DISCRETELY ENTROPY STABLE WEIGHT-ADJUSTED DISCONTINUOUS GALERKIN METHODS ON CURVILINEAR MESHES

JESSE CHAN, LUCAS C. WILCOX

Abstract. We construct entropy conservative and entropy stable high order accurate discontinuous Galerkin (DG) discretizations for time-dependent nonlinear hyperbolic conservation laws on curvilinear meshes. The resulting schemes preserve a semi-discrete quadrature approximation of a continuous global entropy inequality. The proof requires the satisfaction of a discrete geometric conservation law, which we enforce through an appropriate polynomial approximation. We extend the construction of entropy conservative and entropy stable DG schemes to the case when high order accurate curvilinear mass matrices are approximated using low-storage weight-adjusted approximations, and describe how to retain global conservation properties under such an approximation. The theoretical results are verified through numerical experiments for the compressible Euler equations on triangular and tetrahedral meshes.

1. Introduction

High order discontinuous Galerkin (DG) methods are attractive for the simulation of time-dependent wave propagation due to their low numerical dispersion and dissipation [5, 6] and ability to handle unstructured meshes and complex geometries. These same properties make them attractive for the resolution of transient waves and vortices in compressible flow [7]. However, whereas the construction of stable DG methods for wave propagation is relatively well-established, it is not possible to extend the same formulations directly from linear wave problems to the nonlinear conservation laws which govern compressible fluid flow.

The low numerical dissipation of high order DG methods combined with the lack of inherently stable formulations for nonlinear conservation laws has given high order discretizations the reputation of being non-robust and highly sensitive to instabilities and under-resolved features [7]. This instability is addressed in practice by adding additional stabilization through limiting, filtering, or artificial viscosity [8, 9, 10, 11]. However, these approaches are typically ad-hoc, and do not guarantee the stability of the resulting scheme. Moreover, high order accuracy can be lost if the stabilization is too strong.

The lack of inherently stable formulations was addressed for high order nodal DG methods on quadrilateral and hexahedral meshes in [12, 13]. The resulting schemes ensure that that the numerical solution satisfies a semi-discrete version of an entropy inequality, independently of discrete effects such as under-integration. This results in significantly more robust high order simulations, where the numerical solution does not blow up even in the presence of under-resolved features such as shock discontinuities or turbulence. High order entropy stable schemes have since been extended to staggered grid and non-conforming [14, 15] tensor product elements, as well as to simplicial meshes [16, 1, 17].

Entropy stable methods have largely relied on a finite difference summation-by-parts (SBP) framework. The summation-by-parts property holds for spectral element DG methods (DG-SEM), which assume a polynomial basis which collocates the solution at Gauss–Legendre–Lobatto (GLL) quadrature nodes on tensor product elements. The collocation approach requires the number of quadrature nodes to be identical to the number of basis functions. Because analogous quadrature rules on the triangle and tetrahedron must contain more nodes than the dimension of the underlying polynomial space [18, 19], GLL-like collocation cannot be replicated on simplicial elements. It is still possible to construct entropy stable schemes on simplicial elements within the SBP framework [16, 17]. However, the resulting SBP operators are not "modal", in the sense that the matrices are not associated with an underlying basis or approximation space.

Entropy stable schemes were extended to modal discretizations on affine meshes in [1], allowing for the use of arbitrary basis functions and over-integrated quadrature rules. In this work, we show how to extend the construction of modal high order entropy stable schemes to curved meshes. As noted in [17], this requires the use of a split formulation for the geometric terms involved in differentiation, as well as the satisfaction of

a discrete geometric conservation law (GCL) [20, 2]. We also show how to extend entropy stable schemes to accomodate weight-adjusted mass matrices, which provide low storage approximations of inverse weighted mass matrices appearing for curved meshes [4]. These weight-adjusted approximations can be applied more efficiently than inverse weighted mass matrices on many-core architectures such as Graphics Processing Units (GPUs) [21]. However, additional steps are required to ensure entropy stability and the conservation of mass, momentum, and energy under weight-adjusted mass matrices.

The paper is organized as follows: Section 2 reviews the derivation of entropy inequalities for systems of nonlinear conservation laws, and describes why this does not hold under a high order DG discretization. Section 3 describes the construction of entropy stable DG methods for curved meshes using weighted mass matrices, and Section 4 describes how to extend this to the case of weight-adjusted mass matrices. Section 5 describes how to enforce the geometric conservation law using a modification of the approach described in [2], and Section 6 concludes by presenting two and three-dimensional numerical experiments which verify the accuracy and stability of the presented schemes.

2. Systems of nonlinear conservation laws

This work addresses high order accurate schemes for the following system of n nonlinear conservation laws in d dimensions

$$\frac{\partial \boldsymbol{u}}{\partial t} + \sum_{j=1}^{d} \frac{\partial \boldsymbol{f}_j(\boldsymbol{u})}{\partial x_j} = 0, \qquad \boldsymbol{u} : \mathbb{R}^d \times [0,\infty) \to \mathbb{R}^n, \qquad \boldsymbol{f}_j : \mathbb{R}^n \to \mathbb{R}^n, \tag{1}$$

where $\boldsymbol{u}(\boldsymbol{x},t)$ denotes the *conservative variables* for this system. We are interested in nonlinear conservation laws for which an entropy function $U(\boldsymbol{u})$ exists, where $U(\boldsymbol{u})$ is convex with respect to the conservative variables $\boldsymbol{u}$. If this function exists, then it is possible to define *entropy variables* $\boldsymbol{v}(\boldsymbol{u}) = \frac{\partial U}{\partial \boldsymbol{u}}$. These functions symmetrize the system of nonlinear conservation laws (1) [22].

It can be shown (see, for example, [23]) that symmetrization is equivalent to the existence of entropy flux functions $F_j(\boldsymbol{u})$ and entropy potentials ψ_j such that

$$\boldsymbol{v}^T \frac{\partial \boldsymbol{f}_j}{\partial \boldsymbol{u}} = \frac{\partial F_j(\boldsymbol{u})}{\partial \boldsymbol{u}}^T, \qquad \psi_j(\boldsymbol{v}) = \boldsymbol{v}^T \boldsymbol{f}_j(\boldsymbol{u}(\boldsymbol{v})) - F_j(\boldsymbol{u}(\boldsymbol{v})), \qquad \psi_j'(\boldsymbol{v}) = \boldsymbol{f}_j(\boldsymbol{u}(\boldsymbol{v})).$$

Smooth solutions of (1) can be shown to satisfy a conservation of entropy by multiplying (1) by $\boldsymbol{v}(\boldsymbol{u})$. Using the definition of the entropy variables, entropy flux, and the chain rule yields

$$\boldsymbol{v}^T \frac{\partial \boldsymbol{f}_j(\boldsymbol{u})}{\partial x_j} = \frac{\partial U(\boldsymbol{u})}{\partial \boldsymbol{u}}^T \frac{\partial \boldsymbol{f}_j(\boldsymbol{u})}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{u}}{\partial x_j} = \frac{\partial F_j(\boldsymbol{u})}{\partial x_j}, \qquad \frac{\partial U(\boldsymbol{u})}{\partial t} + \sum_{j=1}^{d} \frac{\partial F_j(\boldsymbol{u})}{\partial x_j} = 0. \tag{2}$$

Let $\Omega \subset \mathbb{R}^d$ be a closed domain with boundary $\partial\Omega$. Integrating over Ω and using Gauss' theorem on the spatial derivative yields

$$\int_\Omega \frac{\partial U(\boldsymbol{u})}{\partial t} \, \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega} \sum_{j=1}^{d} \left(\boldsymbol{v}(\boldsymbol{u})^T \boldsymbol{f}_j(\boldsymbol{u}) - \psi_j\left(\boldsymbol{v}(\boldsymbol{u})\right) \right) n_j \, \mathrm{d}\boldsymbol{x} = 0, \tag{3}$$

where $\boldsymbol{n} = (n_1, \ldots, n_d)^T$ denotes the unit outward normal vector on $\partial\Omega$.

General solutions (including non-smooth solutions such as shocks) satisfy an entropy *inequality*

$$\int_\Omega \frac{\partial U(\boldsymbol{u})}{\partial t} \, \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega} \sum_{j=1}^{d} \left(\boldsymbol{v}(\boldsymbol{u})^T \boldsymbol{f}_j(\boldsymbol{u}) - \psi_j\left(\boldsymbol{v}(\boldsymbol{u})\right) \right) n_j \, \mathrm{d}\boldsymbol{x} \leq 0, \tag{4}$$

which results from considering solutions of an appropriate viscous form of the equations (1) and taking the limit as viscosity vanishes. In this work, schemes which satisfy a discrete form of (4) will be constructed by first enforcing a discrete version of entropy conservation (3), then adding an appropriate numerical dissipation which will enforce the entropy inequality (4).

2.1. Standard DG formulations for nonlinear conservation laws.

We begin by reviewing the construction of standard high order accurate DG formulations for (1).

2.1.1. *Mathematical notation.* Let the domain $\Omega \subset \mathbb{R}^d$ be decomposed into elements (subdomains) D^k, and let $\widehat{D}$ denote a d-dimensional reference element with boundary $\partial\widehat{D}$. Let $\widehat{\boldsymbol{x}} = \{\widehat{x}_1, \ldots, \widehat{x}_d\}$ denote coordinates on $\widehat{D}$, and let $\widehat{n}_i$ denote and the ith component of the unit normal vector on $\partial\widehat{D}$. We assume that $\widehat{n}_i$ is constant; i.e., that the faces of the reference element are planar (this assumption holds for all commonly used reference elements [24]).

We will assume that each physical element D^k is the image of $\widehat{D}$ under some smoothly differentiable mapping $\boldsymbol{\Phi}_k(\widehat{\boldsymbol{x}})$ such that

$$\boldsymbol{x} = \boldsymbol{\Phi}_k(\widehat{\boldsymbol{x}}), \qquad \boldsymbol{x} \in D^k.$$

This also implies that integrals over physical elements can be mapped back to the reference element as follows

$$\int_{D^k} u \,\mathrm{d}\boldsymbol{x} = \int_{\widehat{D}} u J^k \,\mathrm{d}\widehat{\boldsymbol{x}},$$

where J^k denotes the determinant of the Jacobian of $\boldsymbol{\Phi}_k$. Integrals over physical faces of D^k can similarly be mapped back to reference faces.

We define an approximation space using degree N polynomials on the reference element. For example, on a d-dimensional reference simplex, the natural polynomial space are total degree N polynomials

$$P^N\left(\widehat{D}\right) = \left\{ \widehat{x}_1^{i_1} \ldots \widehat{x}_d^{i_d}, \quad \widehat{\boldsymbol{x}} \in \widehat{D}, \quad 0 \leq \sum_{k=1}^{d} i_k \leq N \right\}.$$

Other element types possess different natural polynomial spaces [24], but typically contain the space of total degree N polynomials. This work is directly applicable to other elements and spaces as well. We denote the dimension of the approximation space P^N as $N_p = \dim\left(P^N\left(\widehat{D}\right)\right)$. We also define trace spaces for each face of the reference element. Let $\widehat{f}$ be a face of the reference element $\widehat{D}$. The trace space over $\widehat{f}$ is defined as the space of traces of functions in $P^N\left(\widehat{D}\right)$

$$P_f^N\left(\widehat{f}\right) = \left\{ u|_{\widehat{f}}, \quad u \in P^N\left(\widehat{D}\right) \right\}, \qquad \widehat{f} \in \partial\widehat{D}.$$

We denote the dimension of the trace space as $\dim\left(P_f^N\left(\widehat{f}\right)\right) = N_p^f$.

We next define the L^2 norm and inner products over the reference element $\widehat{D}$ and the surface of the reference element $\partial\widehat{D}$ as

$$(\boldsymbol{u}, \boldsymbol{v})_{\widehat{D}} = \int_{\widehat{D}} \boldsymbol{u} \cdot \boldsymbol{v} \,\mathrm{d}\widehat{\boldsymbol{x}}, \qquad \|\boldsymbol{u}\|_{\widehat{D}}^2 = (\boldsymbol{u}, \boldsymbol{u})_{\widehat{D}}, \qquad \langle \boldsymbol{u}, \boldsymbol{v} \rangle_{\partial\widehat{D}} = \int_{\partial\widehat{D}} \boldsymbol{u} \cdot \boldsymbol{v} \,\mathrm{d}\widehat{\boldsymbol{x}}.$$

We also introduce the continuous L^2 projection operator Π_N and lifting operator L. For $u \in L^2\left(\widehat{D}\right)$, the L^2 projection $\Pi_N u$ is defined through

$$\int_{\widehat{D}} \Pi_N u v \,\mathrm{d}\widehat{\boldsymbol{x}} = \int_{\widehat{D}} u v \,\mathrm{d}\widehat{\boldsymbol{x}}, \qquad \forall v \in P^N\left(\widehat{D}\right). \tag{5}$$

Likewise, for a boundary function $u \in L^2\left(\partial\widehat{D}\right)$, the lifting operator L [25, 26] is defined through

$$(Lu, v)_{\widehat{D}} = \langle u, v \rangle_{\partial\widehat{D}}, \qquad \forall v \in P^N\left(\widehat{D}\right). \tag{6}$$

Finally, we introduce L^2, L^∞ Sobolev norms and spaces, which will be utilized for error estimates. The L^2 space is defined as the space of functions with finite L^2 norm. The Lebesgue L^∞ norm and the associated L^∞ space over a general domain Ω are

$$\|u\|_{L^\infty(\Omega)} = \inf\left\{C \geq 0 : |u(\boldsymbol{x})| \leq C \quad \forall \boldsymbol{x} \in \Omega\right\}, \qquad L^\infty(\Omega) = \left\{u : \Omega \to \mathbb{R}, \quad \|u\|_{L^\infty(\Omega)} < \infty\right\}.$$

The L^2 and L^∞ Sobolev norms of degree s are then defined as

$$\|u\|_{W^{s,2}(\Omega)}^2 = \sum_{|\alpha| \leq s} \|D^\alpha u\|_{L^2(\Omega)}^2, \qquad \|u\|_{W^{s,\infty}(\Omega)} = \max_{|\alpha| \leq s} \|D^\alpha u\|_{L^\infty(\Omega)},$$

respectively. Here $\alpha = \{\alpha_1, \dots, \alpha_d\}$ is a multi-index of order $|\alpha| = \alpha_1 + \cdots + \alpha_d$ such that

$$D^\alpha u = \frac{\partial^{\alpha_1}}{\partial x_1^{\alpha_1}} \cdots \frac{\partial^{\alpha_d}}{\partial x_d^{\alpha_d}} u.$$

The Sobolev spaces $W^{s,2}$ and $W^{s,\infty}$ are defined as the spaces of functions with finite L^2 and L^∞ Sobolev norms of degree s, respectively.

2.1.2. *Discontinuous Galerkin formulations and the L^2 projection.* Discontinuous Galerkin methods have been widely applied to systems of nonlinear conservation laws (1) [27, 28, 29]. The development of new discontinuous Galerkin methods for nonlinear conservation laws has focused heavily on the choice of numerical flux [30] or the development of slope limiters [9, 31] and artificial viscosity strategies [8, 10, 32]. However, the treatment of the underlying volume discretization remains relatively unchanged between each of these approaches.

Ignoring terms involving filters, limiters, or artificial viscosity, a semi-discrete "weak" DG formulation for (1) can be given locally over an element D^k: find $\boldsymbol{u} \in \left(P^N\left(D^k\right) \times [0,\infty)\right)^n$ such that

$$\int_{D^k} \left(\frac{\partial \boldsymbol{u}}{\partial t} \cdot \boldsymbol{v} - \sum_{j=1}^d \boldsymbol{f}_j(\boldsymbol{u}) \cdot \frac{\partial \boldsymbol{v}}{\partial x_i} \right) \mathrm{d}\boldsymbol{x} + \sum_{j=1}^d \int_{\partial D^k} \left(\boldsymbol{f}_j^*\left(\boldsymbol{u}^+, \boldsymbol{u}\right)\right) \cdot \boldsymbol{v} n_j \, \mathrm{d}\boldsymbol{x} = 0, \qquad \forall \boldsymbol{v} \in \left(P^N\left(D^k\right)\right)^n, \tag{7}$$

where the numerical flux $\boldsymbol{f}^*$ is a function of the solution $\boldsymbol{u}$ on both D^k and neighboring elements.

Unfortunately, solutions to (7) do not (in general) obey a discrete version of the entropy inequality (4). Since (4) is a generalized statement of energy stability, the lack of a discrete entropy inequality implies that the discrete solution can blow up in finite time. The reason for this is due to the fact that, in practice, the integrals in (7) are not computed exactly and are instead approximated using polynomially exact quadratures. This is compounded by the fact that the nonlinear flux function $\boldsymbol{f}_j(\boldsymbol{u})$ is often rational and impossible to integrate exactly using polynomial quadratures.

While this paper focuses on curved meshes, the inexactness of quadrature leads to the loss of the chain rule and thus the loss of entropy conservation or entropy dissipation (4) even on affine meshes. We can rewrite (7) in a strong form using a discrete quadrature-based L^2 projection. For polynomial approximation spaces on affine meshes, $\frac{\partial \boldsymbol{v}}{\partial x_i}$ is polynomial. Then, mapping (7) back to the reference element $\widehat{D}$ and using the L^2 projection and (5), we have that

$$\int_{D^k} \boldsymbol{f}_j(\boldsymbol{u}) \cdot \frac{\partial \boldsymbol{v}}{\partial x_i} \, \mathrm{d}\boldsymbol{x} = \int_{\widehat{D}} \Pi_N \boldsymbol{f}_j(\boldsymbol{u}) \cdot \frac{\partial \boldsymbol{v}}{\partial x_i} J^k \, \mathrm{d}\boldsymbol{x}.$$

Thus, integrating by parts (7) recovers a "strong" DG formulation involving the projection operator

$$\begin{aligned} &\int_{D^k} \left(\frac{\partial \boldsymbol{u}}{\partial t} - \sum_{j=1}^d \frac{\partial \Pi_N \boldsymbol{f}_j(\boldsymbol{u})}{\partial x_j} \right) \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} \\ &+ \sum_{j=1}^d \int_{\partial D^k} \left(\boldsymbol{f}_j^*\left(\boldsymbol{u}^+, \boldsymbol{u}\right) - \Pi_N \boldsymbol{f}_j(\boldsymbol{u})\right) \cdot \boldsymbol{v} n_j \, \mathrm{d}\boldsymbol{x} = 0, \qquad \forall \boldsymbol{v} \in \left(P^N\left(D^k\right)\right)^n. \end{aligned} \tag{8}$$

From this, we see that our discrete scheme does not differentiate the nonlinear flux function $\boldsymbol{f}_j(\boldsymbol{u})$ exactly, but instead differentiates the projection of $\Pi_N \boldsymbol{f}_j(\boldsymbol{u})$ onto polynomials of degree N. Because the L^2 projection operator is introduced, the chain rule no longer holds at the discrete level and step (2) of the proof of entropy conservation is no longer valid. Thus, ensuring discrete entropy stability will require a discrete formulation of the system of nonlinear conservation laws (1) from which we can prove a discrete entropy inequality without relying on the chain rule.

3. Discretely entropy stable DG methods on curved meshes

We will first show how to construct discretely entropy stable high order accurate DG methods on curvilinear meshes, but will present this using a matrix formulation as opposed to a continuous formulation. This is to ensure that the effects of discretization, nonlinear, and quadrature are accounted for in the proof of

semi-discrete entropy stability. We first introduce quadrature-based matrices, which we will then use to construct discretely entropy stable DG formulations.

3.1. **Basis and quadrature rules.** We now introduce quadrature-based matrices for the d-dimensional reference element $\widehat{D}$, which we will use to construct matrix-vector formulations of DG methods. Assuming $u(\widehat{\boldsymbol{x}}) \in P^N\left(\widehat{D}\right)$, it can be represented in terms of the vector of coefficients $\boldsymbol{u}$ using some polynomial basis ϕ_i of degree N and dimension N_p

$$u(\widehat{\boldsymbol{x}}) = \sum_{j=1}^{N_p} \boldsymbol{u}_j \phi_j(\widehat{\boldsymbol{x}}), \qquad P^N\left(\widehat{D}\right) = \operatorname{span}\left\{\phi_i(\widehat{x})\right\}_{i=1}^{N_p}.$$

We construct quadrature-based matrices based on ϕ_i and appropriate volume and surface quadrature rules. The volume and surface quadrature rules are given by points and positive weights $\{(\widehat{\boldsymbol{x}}_i, \widehat{w}_i)\}_{i=1}^{N_q}$ and $\left\{(\widehat{\boldsymbol{x}}_i^f, \widehat{w}_i^f)\right\}_{i=1}^{N_q^f}$, respectively. We make the following assumptions on the strength of these quadratures:

Assumption 1 (Integration by parts under quadrature)**.** *The volume quadrature rule $\{(\widehat{\boldsymbol{x}}_i, \widehat{w}_i)\}_{i=1}^{N_q}$ is exact for polynomials of degree $2N-1$. Additionally, for any $u, v \in P^N\left(\widehat{D}\right)$, integration by parts*

$$\left(\frac{\partial u}{\partial \widehat{x}_i}, v\right)_{\widehat{D}} = \langle u, v\widehat{n}_i\rangle_{\partial\widehat{D}} - \left(u, \frac{\partial v}{\partial \widehat{x}_i}\right)_{\widehat{D}}$$

holds when the volume and surface integrals are approximated using quadrature.

Assumption 1 holds, for example, for any surface quadrature rule which is exact for degree $2N$ polynomials on the boundary of the reference element $\partial\widehat{D}$.

3.2. **Reference element matrices.** Let $\boldsymbol{W}, \boldsymbol{W}_f$ denote diagonal matrices whose entries are volume and surface quadrature weights, respectively. The surface quadrature weights are given by quadrature weights on reference faces, which are mapped to faces of the reference element. We define the volume and surface quadrature interpolation matrices $\boldsymbol{V}_q$ and $\boldsymbol{V}_f$ such that

$$\begin{aligned} (\boldsymbol{V}_q)_{ij} &= \phi_j(\widehat{\boldsymbol{x}}_i), \qquad 1 \le j \le N_p, \qquad 1 \le i \le N_q, \\ (\boldsymbol{V}_f)_{ij} &= \phi_j(\widehat{\boldsymbol{x}}_i^f), \qquad 1 \le j \le N_p, \qquad 1 \le i \le N_q^f, \end{aligned} \tag{9}$$

which map coefficients $\boldsymbol{u}$ to evaluations of u at volume and surface quadrature points.

Next, let $\boldsymbol{D}_i$ denote the differentiation matrix with respect to the ith coordinate, defined implicitly through the relations

$$u(\widehat{\boldsymbol{x}}) = \sum_{j=1}^{N_p} \boldsymbol{u}_j \phi_j(\widehat{\boldsymbol{x}}), \qquad \frac{\partial u}{\partial \widehat{\boldsymbol{x}}_i} = \sum_{j=1}^{N_p} \left(\boldsymbol{D}_i \boldsymbol{u}\right)_j \phi_j(\widehat{\boldsymbol{x}}).$$

The matrix $\boldsymbol{D}_i$ maps basis coefficients of some polynomial $u \in P^N\left(\widehat{D}\right)$ to coefficients of its ith derivative with respect to the reference coordinate $\widehat{\boldsymbol{x}}$, and is sometimes referred to as a "modal" differentiation matrix (with respect to a general non-nodal "modal" basis [19]).

Using the volume quadrature interpolation matrix $\boldsymbol{V}_q$, we can compute a quadrature-based mass matrix $\boldsymbol{M}$ by evaluating L^2 inner products of different basis functions using quadrature

$$\boldsymbol{M} = \boldsymbol{V}_q^T \boldsymbol{W} \boldsymbol{V}_q, \qquad \boldsymbol{M}_{ij} = \sum_{k=1}^{N_q} \widehat{w}_k \phi_j(\widehat{\boldsymbol{x}}_k)\phi_i(\widehat{\boldsymbol{x}}_k) \approx \int_{\widehat{D}} \phi_j \phi_i \, \mathrm{d}\widehat{\boldsymbol{x}} = (\phi_j, \phi_i)_{\widehat{D}}.$$

The approximation in the formula for the mass matrix becomes an equality if the volume quadrature rule is exact for polynomials of degree $2N$. The mass matrix is symmetric and positive definite under Assumption 1; however, we do not make any distinctions between diagonal and dense (lumped) mass matrices in this work.

The mass matrix appears in the discretization of L^2 projection (5) and lift operators (6) using quadrature. The result are quadrature-based L^2 projection and lift operators $\boldsymbol{P}_q, \boldsymbol{L}_q$,

$$\boldsymbol{P}_q = \boldsymbol{M}^{-1} \boldsymbol{V}_q^T \boldsymbol{W}, \qquad \boldsymbol{L}_q = \boldsymbol{M}^{-1} \boldsymbol{V}_f^T \boldsymbol{W}_f, \tag{10}$$

which are discretizations of the continuous L^2 projection operator Π_N and continuous lift operator L. The matrix $\boldsymbol{P}_q$ maps a function (in terms of its evaluation at quadrature points) to coefficients of the L^2 projection in the basis $\phi_j(x)$, while the matrix $\boldsymbol{L}_q$ "lifts" a function (evaluated at surface quadrature points) from the boundary of an element to coefficients of a basis defined in the interior of the element.

Finally, we introduce quadrature-based operators $\boldsymbol{D}_N^i$ which will be used to construct discretizations of our nonlinear conservation laws. This operator was introduced in [1] as a "decoupled summation-by-parts" operator

$$\boldsymbol{D}_N^i = \begin{bmatrix} \boldsymbol{V}_q \boldsymbol{D}_i \boldsymbol{P}_q - \frac{1}{2}\boldsymbol{V}_q \boldsymbol{L}_q \text{diag}\left(\widehat{\boldsymbol{n}}_i \circ \widehat{\boldsymbol{J}}_f\right) \boldsymbol{V}_f \boldsymbol{P}_q & \frac{1}{2}\boldsymbol{V}_q \boldsymbol{L}_q \text{diag}\left(\widehat{\boldsymbol{n}}_i \circ \widehat{\boldsymbol{J}}_f\right) \\ -\frac{1}{2}\text{diag}\left(\widehat{\boldsymbol{n}}_i \circ \widehat{\boldsymbol{J}}_f\right) \boldsymbol{V}_f \boldsymbol{P}_q & \frac{1}{2}\text{diag}\left(\widehat{\boldsymbol{n}}_i \circ \widehat{\boldsymbol{J}}_f\right) \end{bmatrix} \tag{11}$$

where $\widehat{\boldsymbol{n}}_i$ is the vector containing values of the ith component of the unit normal on the surface of the reference element $\widehat{D}$, and $\widehat{\boldsymbol{J}}_f$ is the vector containing values of the face Jacobian factor $\widehat{J}_f$ which result from mapping a face of $\widehat{D}$ to a reference face. Here $\widehat{\boldsymbol{n}}_i \circ \widehat{\boldsymbol{J}}_f$ is the Hadamard product (i.e., the entrywise product) of the vectors $\widehat{\boldsymbol{n}}_i$ and $\widehat{\boldsymbol{J}}_f$. When combined with projection and lifting matrices, $\boldsymbol{D}_N^i$ produces a high order approximation of non-conservative products. Let $\boldsymbol{f}, \boldsymbol{g}$ denote vectors containing the evaluation of functions $f(\boldsymbol{x}), g(\boldsymbol{x})$ at both volume and surface quadrature points

$$\begin{bmatrix} \boldsymbol{P}_q & \boldsymbol{L}_q \end{bmatrix} \text{diag}\left(\boldsymbol{f}\right) \boldsymbol{D}_N^i \boldsymbol{g} \approx f \frac{\partial g}{\partial \widehat{x}_i}.$$

It was shown in [1] that the matrix $\boldsymbol{D}_N^i$ satisfies several key properties. First, it can be observed that $\boldsymbol{D}_N^i \mathbf{1} = 0$, where $\mathbf{1}$ is the vector of all ones. Second, $\boldsymbol{D}_N^i$ satisfies a summation-by-parts property. Let $\boldsymbol{Q}_N^i$ be the scaling of $\boldsymbol{D}_N^i$ by the diagonal matrix of volume and surface quadrature weights

$$\boldsymbol{Q}_N^i = \boldsymbol{W}_N \boldsymbol{D}_N^i, \qquad \boldsymbol{W}_N = \begin{pmatrix} \boldsymbol{W} & \\ & \boldsymbol{W}_f \end{pmatrix}.$$

Then, $\boldsymbol{Q}_N^i$ satisfies the following discrete analogue of integration by parts

$$\boldsymbol{Q}_N^i + \left(\boldsymbol{Q}_N^i\right)^T = \boldsymbol{B}_N^i, \qquad \boldsymbol{B}_N = \begin{pmatrix} \mathbf{0} & \\ & \boldsymbol{W}_f \text{diag}\left(\widehat{\boldsymbol{n}}_i \circ \widehat{\boldsymbol{J}}_f\right) \end{pmatrix}. \tag{12}$$

The matrix $\boldsymbol{D}_N^i$ reduces to polynomial differentiation when applied to polynomials, in the sense that

$$\boldsymbol{D}_N^i \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_q \boldsymbol{D}_i \\ \mathbf{0} \end{bmatrix}. \tag{13}$$

3.3. **Matrices on curved physical elements.** The key difference between curvilinear and affine meshes is that geometric terms now vary spatially over each element. In practice, derivatives are computed over the reference element and mapped to the physical element D^k through a change of variables formula

$$J^k \frac{\partial u}{\partial x_i} = \sum_{j=1}^{d} G_{ij}^k \frac{\partial u}{\partial \widehat{x}_j}, \qquad G_{ij}^k = J^k \frac{\partial \widehat{x}_j}{\partial x_i},$$

where we have defined the elements of the matrix $\boldsymbol{G}^k$ as the derivatives of the reference coordinates $\widehat{x}_j$ with respect to the physical coordinates x_i on D^k times the Jacobian of the transformation from reference to physical coordinates J^k. We denote evaluations of G_{ij}^k at both volume and surface quadrature points as the vector $\boldsymbol{G}_{ij}^k$.

We assume in this work that the mesh is stationary. It can be shown at the continuous level that, for any differentiable and invertible mapping, the quantity $\boldsymbol{G}^k$ satisfies a geometric conservation law (GCL) [2, 20]

$$\sum_{j=1}^{d} \frac{\partial}{\partial \widehat{x}_j} G_{ij}^k = 0, \tag{14}$$

or that $\widehat{\nabla} \cdot \boldsymbol{G}^k = 0$. Using (14), the scaled physical derivative $J^k \frac{\partial u}{\partial x_i}$ can be computed via

$$J^k \frac{\partial u}{\partial x_i} = \frac{1}{2} \sum_{j=1}^{d} \left(G_{ij}^k \frac{\partial u}{\partial \widehat{x}_j} + \frac{\partial \left(G_{ij}^k u \right)}{\partial \widehat{x}_j} \right). \tag{15}$$

We will require the following assumptions on the mesh, as well as the geometric terms and outward normal vectors:

Assumption 2 (Mesh assumptions)**.** *We assume that the mesh is quasi-uniform. The mesh is also assumed to be watertight, such that normals are consistent across neighboring elements as follows: for a shared face f between D^k and $D^{k,+}$, the scaled outward normal vectors for each element are equal and opposite at all points such that*

$$\boldsymbol{n} J_f^k = -\boldsymbol{n}^+ J_f^{k,+}. \tag{16}$$

We also assume that the scaled matrix of geometric terms transforms scaled reference normal vectors to scaled physical normals, such that

$$\sum_{j=1}^{d} G_{ij}^k \widehat{n}_j \widehat{J}_f = n_i J_f^k, \qquad \left(\sum_{j=1}^{d} \left(\boldsymbol{G}_{ij}^k \right)_f \circ \left(\widehat{\boldsymbol{n}}_j \circ \widehat{\boldsymbol{J}}_f \right) \right) = \left(\boldsymbol{n}_i \circ \boldsymbol{J}_f^k \right), \tag{17}$$

where $\boldsymbol{n}_j$ and $\boldsymbol{J}_f^k$ are vectors containing evaluations of the physical unit normals and face Jacobian factors for D^k at surface quadrature points, respectively. Likewise, $\left(\boldsymbol{G}_{ij}^k \right)_f$ is a vector containing evaluations of G_{ij}^k at the surface quadrature points.

The properties (16) and (17) hold at the continuous level for a watertight mesh [33], and thus at all points where the geometric terms are computed exactly. However, we will also consider cases where the geometric terms G_{ij}^k are modified to enforce a discrete form of (14); in these situations, it will be important to ensure that (17) holds after such modifications.

Similar to what is done to stabilize finite difference discretizations [34, 35], we define physical differentiation matrices based on the approximation of (15). Define $\boldsymbol{D}_k^i$ as

$$\boldsymbol{D}_k^i = \frac{1}{2} \sum_{j=1}^{d} \left(\text{diag} \left(\boldsymbol{G}_{ij}^k \right) \boldsymbol{D}_N^j + \boldsymbol{D}_N^j \text{diag} \left(\boldsymbol{G}_{ij}^k \right) \right).$$

Using properties of the Hadamard product [36], we can rewrite $\boldsymbol{D}_k^i$ as

$$\boldsymbol{D}_k^i = \sum_{j=1}^{d} \left(\boldsymbol{D}_N^j \circ \left\{ \left\{ \boldsymbol{G}_{ij}^k \right\} \right\} \right), \qquad \left\{ \left\{ \boldsymbol{G}_{ij}^k \right\} \right\}_{mn} = \frac{1}{2} \left(\left(\boldsymbol{G}_{ij}^k \right)_m + \left(\boldsymbol{G}_{ij}^k \right)_n \right), \tag{18}$$

where $\left\{ \left\{ \boldsymbol{G}_{ij}^k \right\} \right\}$ denotes the matrix of averages between each of the entries of $\boldsymbol{G}_{ij}^k$. From Assumption 2 and (18), it is straightforward to show that (because $\left\{ \left\{ \boldsymbol{G}_{ij}^k \right\} \right\}$ is symmetric) $\boldsymbol{Q}_k^i = \boldsymbol{W}_N \boldsymbol{D}_k^i$ also satisfies a summation-by-parts property

$$\boldsymbol{Q}_k^i + \left(\boldsymbol{Q}_k^i \right)^T = \boldsymbol{B}_k^i, \qquad \boldsymbol{B}_k^i = \begin{pmatrix} \boldsymbol{0} & \\ & \boldsymbol{W}_f \text{diag} \left(\boldsymbol{n}_i \circ \boldsymbol{J}_f^k \right) \end{pmatrix}. \tag{19}$$

Curvilinear mappings also imply that integrals over each physical element D^k are no longer simple scalings of integrals over $\widehat{D}$. The L^2 projection of $u \in L^2 \left(D^k \right)$ over a curvilinear element D^k is defined through

$$\left(\Pi_N^k u, v \right)_{D^k} = \left(u, v \right)_{D^k}, \qquad \forall v \in P^N \left(\widehat{D} \right). \tag{20}$$

Mapping integrals to the reference element $\widehat{D}$ yields

$$\left(\Pi_N^k u, v J^k \right)_{\widehat{D}} = \left(u, v J^k \right)_{\widehat{D}}, \qquad \forall v \in P^N \left(\widehat{D} \right). \tag{21}$$

For affine elements, J^k is constant and can be cancelled. Thus, the L^2 projection over affine elements is equivalent to simply taking the L^2 projection of a function over the reference element. However, for curved elements, J acts as a spatially varying weight within the L^2 inner product.

Discretizing (21) requires a weighted mass matrix. We define a curved mass matrix over an element D^k by weighting the discrete L^2 norm with values of J at quadrature points

$$\boldsymbol{M}^k = \boldsymbol{V}_q^T \boldsymbol{W} \text{diag}\left(\boldsymbol{J}^k\right) \boldsymbol{V}_q, \tag{22}$$

where $\boldsymbol{J}^k$ is a vector containing evaluation of the physical Jacobian factors for D^k at volume quadrature points. Then, curvilinear L^2 projection and lift matrices can be defined in a manner analogous to (10)

$$\boldsymbol{P}_q^k = \left(\boldsymbol{M}^k\right)^{-1} \boldsymbol{V}_q^T \boldsymbol{W} \text{diag}\left(\boldsymbol{J}^k\right), \qquad \boldsymbol{L}_q^k = \left(\boldsymbol{M}^k\right)^{-1} \boldsymbol{V}_f^T \boldsymbol{W}_f \text{diag}\left(\boldsymbol{J}_f^k\right). \tag{23}$$

These matrices are distinct from element to element, reflecting the fact that problem (21) is distinct from element to element.

3.4. **A discretely entropy stable DG formulation on curved meshes.** Given the matrices in Section 3.3, we can now define a local entropy stable DG formulation on an element D^k. Here, we seek an approximation solution $\boldsymbol{u}_N(\boldsymbol{x}, t)$ to (1), which is represented using vector-valued coefficients $\boldsymbol{u}_h(t)$ such that

$$\boldsymbol{u}_N(\boldsymbol{x}, t) = \sum_{j=1}^{N_p} \left(\boldsymbol{u}_h(t)\right)_j \phi_j(\boldsymbol{x}), \qquad \left(\boldsymbol{u}_h(t)\right)_j \in \mathbb{R}^n.$$

Since the coefficients are vector valued, we assume that all matrices act component-wise on $\boldsymbol{u}_h$ in the Kronecker product sense.

We first define the numerical fluxes $\boldsymbol{f}_{i,S}\left(\boldsymbol{u}_L, \boldsymbol{u}_R\right)$ as the bivariate function of "left" and "right" conservative variable states $\boldsymbol{u}_L, \boldsymbol{u}_R$.

Definition 1. The numerical flux $\boldsymbol{f}_{i,S}\left(\boldsymbol{u}_L, \boldsymbol{u}_R\right)$ is entropy conservative (or entropy stable) if it satisfies the following conditions:

(1) $\boldsymbol{f}_{i,S}\left(\boldsymbol{u}_L, \boldsymbol{u}_R\right) = \boldsymbol{f}_{i,S}\left(\boldsymbol{u}_R, \boldsymbol{u}_L\right)$ (symmetry).
(2) $\boldsymbol{f}_{i,S}\left(\boldsymbol{u}, \boldsymbol{u}\right) = \boldsymbol{f}_i\left(\boldsymbol{u}\right)$ (consistency).
(3) $\boldsymbol{f}_{i,S}$ is referred to as entropy conservative if it satisfies conditions 1, 2, and

$$\left(\boldsymbol{v}_L - \boldsymbol{v}_R\right)^T \boldsymbol{f}_{i,S}\left(\boldsymbol{u}_L, \boldsymbol{u}_R\right) = \psi_i\left(\boldsymbol{u}_L\right) - \psi_i\left(\boldsymbol{u}_R\right).$$

We now introduce the L^2 projection of the entropy variables $\boldsymbol{v}_h$ and the entropy-projected conservative variables $\widetilde{\boldsymbol{u}}$

$$\boldsymbol{u}_q = \boldsymbol{V}_q \boldsymbol{u}_h, \qquad \boldsymbol{v}_h = \boldsymbol{P}_q^k \boldsymbol{v}\left(\boldsymbol{u}_q\right), \qquad \widetilde{\boldsymbol{v}} = \begin{bmatrix} \widetilde{\boldsymbol{v}}_q \\ \widetilde{\boldsymbol{v}}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \boldsymbol{v}_h, \qquad \widetilde{\boldsymbol{u}} = \begin{bmatrix} \widetilde{\boldsymbol{u}}_q \\ \widetilde{\boldsymbol{u}}_f \end{bmatrix} = \boldsymbol{u}\left(\widetilde{\boldsymbol{v}}\right). \tag{24}$$

In (24), the entropy-projected conservative variables $\widetilde{\boldsymbol{u}}$ denote the evaluation of the conservative variables in terms of the projected entropy variables at volume and face quadrature points. We note that, under an appropriate choice of quadrature on quadrilaterals and hexahedra, this approach is equivalent to the approach taken in [14], where the entropy variables are evaluated at Gauss nodes, then interpolated to a different set of nodes and used to compute the nonlinear fluxes.

We now introduce a semi-discrete DG formulation for $\boldsymbol{u}_h$

$$\begin{aligned} &\frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{dt}} + \begin{bmatrix} \boldsymbol{P}_q^k & \boldsymbol{L}_q^k \end{bmatrix} \sum_{j=1}^{d} \left(2\boldsymbol{D}_k^j \circ \boldsymbol{F}_{j,S}\right) \boldsymbol{1} + \sum_{j=1}^{d} \boldsymbol{L}_q^k \text{diag}\left(\boldsymbol{n}_j\right) \left(\boldsymbol{f}_j^* - \boldsymbol{f}_j(\widetilde{\boldsymbol{u}}_f)\right) = 0, \\ &\left(\boldsymbol{F}_{j,S}\right)_{mn} = \boldsymbol{f}_{j,S}\left(\left(\widetilde{\boldsymbol{u}}\right)_m, \left(\widetilde{\boldsymbol{u}}\right)_n\right), \qquad 1 \le m, n \le N_q + N_q^f, \\ &\boldsymbol{f}_j^* = \boldsymbol{f}_{j,S}(\widetilde{\boldsymbol{u}}_f^+, \widetilde{\boldsymbol{u}}_f) \text{ on interior faces}, \end{aligned} \tag{25}$$

where $\widetilde{\boldsymbol{u}}^+$ denotes the values of the entropy-projected conservative variables on the neighboring element across each face of D^k, and $\boldsymbol{f}_j^*$ on the boundary denotes the jth component of some numerical flux through which boundary conditions are imposed. Note that the face/surface Jacobian factors $\boldsymbol{J}_f^k$ are incorporated into the definition of $\boldsymbol{L}_q^k$.

Define the diagonal boundary quadrature matrix $\boldsymbol{W}_{\partial\Omega}$ such that

$$\left(\boldsymbol{W}_{\partial\Omega}\right)_{ii} = \begin{cases} \boldsymbol{W}_f, & \text{if } \widehat{\boldsymbol{x}}_i^f \text{ is on the } \partial\Omega \\ 0, & \text{otherwise.} \end{cases}$$

We have the following semi-discrete statement of entropy conservation:

Theorem 1. *Let $\boldsymbol{f}_{i,S}$ be an entropy conservative flux from Definition 1 and assume that $\boldsymbol{Q}_k^j \mathbf{1} = 0$ for $j = 1, \dots, d$ over each element D^k. Then, (25) is entropy conservative in the sense that*

$$\sum_k \mathbf{1}^T \text{diag}\left(\boldsymbol{J}^k\right) \boldsymbol{W} \frac{\mathrm{d}U(\boldsymbol{u}_q)}{\mathrm{dt}} = \sum_k \sum_{j=1}^d \mathbf{1}^T \text{diag}\left(\boldsymbol{n}_j \circ \boldsymbol{J}_f^k\right) \boldsymbol{W}_{\partial\Omega} \left(\psi_j\left(\widetilde{\boldsymbol{u}}_f\right) - \widetilde{\boldsymbol{v}}_f^T \boldsymbol{f}_j^*\right).$$

Proof. Under the assumption that $\boldsymbol{Q}_k^j \mathbf{1} = 0$ for $i = 1, \dots, d$ over each element and (19), the proof of entropy conservation is identical to that of [1]. □

An entropy stable scheme can be constructed by adding an entropy-dissipating penalty term, such as a Lax-Friedrichs penalization or the matrix dissipation terms introduced in [37, 38]. For example, Lax-Friedrichs penalization can be incorporated by replacing the flux term with

$$\boldsymbol{L}_q^k \text{diag}\left(\boldsymbol{n}_j\right)\left(\boldsymbol{f}_j^* - \boldsymbol{f}(\boldsymbol{u})\right) \Longrightarrow \boldsymbol{L}_q^k \left(\text{diag}\left(\boldsymbol{n}_j\right)\left(\boldsymbol{f}_j^* - \boldsymbol{f}(\boldsymbol{u})\right) - \frac{\lambda}{2}[\![\widetilde{\boldsymbol{u}}_f]\!]\right),$$

where λ is an estimate of the maximum eigenvalue of $\frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial \boldsymbol{u}}$ [16, 1].

4. Discretely stable and low storage DG methods on curved meshes

A disadvantage of the formulation (25) is high storage costs, especially at high orders of approximation. While the matrices $\boldsymbol{Q}_k^i$ can be applied to a vector without needing to explicitly store the matrix, the projection and lifting matrices (23) differ from element to element, necessitating either explicit pre-computation and storage or the assembly and inversion of a weighted mass matrix for each right hand side evaluation. The latter option is computationally expensive, while the former option increases storage costs. This increase in storage can result in suboptimal performance on modern computational architectures [21], due to the increasing cost of memory operations and data movement compared to arithmetic operations.

In this section, we present a discretely entropy stable scheme which avoids this high storage cost through the use of a low-storage weight-adjusted approximation to the inverse of a weighted mass matrix. To ensure a discrete entropy conservation or a discrete entropy inequality, we also modify the formulation (25) to take into account the use of a weight-adjusted mass matrix.

4.1. A weight-adjusted approximation to the curvilinear mass matrix.

The presence of the weighted L^2 inner product $\left(J^k \Pi_N^k u, v\right)_{\widehat{D}}$ in (21) results in the presence of a weighted mass matrix. Because the weight J^k varies spatially over each element, the inverse of a weighted mass matrix is no longer a scaling of the inverse reference mass matrix. The motivation for the weight-adjusted mass matrix is to replace the inversion of weighted mass matrices over each element with the application of inverse reference mass matrices and quadrature-based operations involving the spatially varying weights J^k [3, 4].

To define a weight-adjusted approximation to the curvilinear L^2 inner product, we first define the operator $T_w^{-1} : L^2 \to P^N$ as follows

$$\left(w T_w^{-1} u, v\right)_{\widehat{D}} = (u, v)_{\widehat{D}}, \qquad \forall v \in P^N\left(\widehat{D}\right). \tag{26}$$

Roughly speaking, $T_w u$ approximates u/w. Thus, taking $w = 1/J^k$ provides an approximation of the curvilinear L^2 inner product

$$\left(J^k u, v\right)_{\widehat{D}} \approx \left(T_{1/J^k}^{-1} u, v\right)_{\widehat{D}}.$$

Computing $T_{1/J^k}^{-1} u$ requires solving (26). Let $u \in P^N\left(\widehat{D}\right)$, and let $\boldsymbol{u}_J$ denote coefficients for the polynomial $T_{1/J^k}^{-1} u$. This results in the following matrix system

$$\boldsymbol{M}_{1/J^k} \boldsymbol{u}_J = \boldsymbol{M} \boldsymbol{u}, \qquad \boldsymbol{M}_{1/J^k} = \boldsymbol{V}_q^T \boldsymbol{W} \text{diag}\left(1/\boldsymbol{J}^k\right) \boldsymbol{V}_q,$$

which implies that, when restricted to polynomials, the matrix form of T_{1/J^k}^{-1} is $\boldsymbol{M}_{1/J^k}^{-1}\boldsymbol{M}$. Then, the weight-adjusted mass matrix is the Gram matrix with respect to the weight-adjusted inner product $\left(T_{1/J^k}^{-1}u,v\right)_{\widehat{D}}$, such that

$$\boldsymbol{M}^k \approx \boldsymbol{M}\boldsymbol{M}_{1/J^k}^{-1}\boldsymbol{M}, \qquad \left(\boldsymbol{M}^k\right)^{-1} \approx \boldsymbol{M}^{-1}\boldsymbol{M}_{1/J^k}\boldsymbol{M}^{-1}.$$

The inverse of the weight-adjusted mass matrix can be applied in a matrix-free fashion by using quadrature to form $\boldsymbol{M}_{1/J^k}$. This requires storage of the inverse reference mass matrix and the values of $\boldsymbol{J}^k$ at quadrature points. Assuming that the number of quadrature points scales as $O(N^d)$ in d dimensions, this yields a storage cost of $O(N^d)$ per-element compared to an $O(N^{2d})$ per element storage cost required for the storage of inverse weighted mass matrices $\left(\boldsymbol{M}^k\right)^{-1}$. This application of the weight-adjusted mass matrix is typically applied using the L^2 projection matrix $\boldsymbol{P}_q$ as follows

$$\boldsymbol{M}^{-1}\boldsymbol{M}_{1/J^k}\boldsymbol{M}^{-1} = \boldsymbol{P}_q \text{diag}\left(1/\boldsymbol{J}^k\right)\boldsymbol{V}_q\boldsymbol{M}^{-1}.$$

When evaluating the right hand side of a semi-discrete formulation such as (25), the inverse mass matrix is typically merged into operations on the right hand side, such that the main work in applying the weight-adjusted mass matrix consists of applying the interpolation matrix $\boldsymbol{V}_q$, scaling by pointwise values of $1/\boldsymbol{J}^k$ at quadrature points, and multiplying by the L^2 projection matrix $\boldsymbol{P}_q$.

4.2. **A discretely entropy stable low storage DG formulation on curved meshes.** Given the weight-adjusted inverse mass matrix, we can also define a weight-adjusted version of the L^2 projection over a curved element D^k. We refer to this operator as $\widetilde{\Pi}_N^k : L^2 \to P^N$, which satisfies

$$\left(T_{1/J^k}^{-1}\widetilde{\Pi}_N^k u, v\right)_{\widehat{D}} = \left(uJ^k, v\right)_{\widehat{D}}, \qquad \forall v \in P^N\left(\widehat{D}\right).$$

It was shown in [4] that $\widetilde{\Pi}_N^k$ is given explicitly by

$$\widetilde{\Pi}_N^k u = \Pi_N\left(\frac{1}{J^k}\Pi_N\left(uJ^k\right)\right), \tag{27}$$

where Π_N is the L^2 projection operator on the reference element $\widehat{D}$. We can discretize $\widetilde{\Pi}_N^k$ using quadrature to yield a weight-adjusted projection matrix $\widetilde{\boldsymbol{P}}_q^k$

$$\begin{aligned}\widetilde{\boldsymbol{P}}_q^k &= \boldsymbol{M}^{-1}\boldsymbol{M}_{1/\boldsymbol{J}^k}\boldsymbol{M}^{-1}\boldsymbol{V}_q^T\boldsymbol{W}\text{diag}\left(\boldsymbol{J}^k\right) = \boldsymbol{M}^{-1}\boldsymbol{V}_q^T\boldsymbol{W}\text{diag}\left(1/\boldsymbol{J}^k\right)\boldsymbol{V}_q\boldsymbol{P}_q\text{diag}\left(\boldsymbol{J}^k\right)\\ &= \boldsymbol{P}_q\text{diag}\left(1/\boldsymbol{J}^k\right)\boldsymbol{V}_q\boldsymbol{P}_q\text{diag}\left(\boldsymbol{J}^k\right).\end{aligned} \tag{28}$$

We can similarly define a weight-adjusted lifting matrix $\widetilde{\boldsymbol{L}}_q$ by replacing the weighted mass matrix in (23) with the weight-adjusted mass matrix

$$\begin{aligned}\widetilde{\boldsymbol{L}}_q^k &= \boldsymbol{M}^{-1}\boldsymbol{M}_{1/\boldsymbol{J}^k}\boldsymbol{M}^{-1}\boldsymbol{V}_f^T\boldsymbol{W}_f\text{diag}\left(\boldsymbol{J}_f^k\right) = \boldsymbol{M}^{-1}\boldsymbol{V}_q^T\boldsymbol{W}\text{diag}\left(1/\boldsymbol{J}^k\right)\boldsymbol{V}_q\boldsymbol{L}_q\text{diag}\left(\boldsymbol{J}_f^k\right)\\ &= \boldsymbol{P}_q\text{diag}\left(1/\boldsymbol{J}^k\right)\boldsymbol{V}_q\boldsymbol{L}_q\text{diag}\left(\boldsymbol{J}_f^k\right).\end{aligned} \tag{29}$$

We can now introduce the weight-adjusted projection of the entropy variables $\boldsymbol{v}_h$ and the corresponding entropy-projected conservative variables $\widetilde{\boldsymbol{u}}$

$$\boldsymbol{u}_q = \boldsymbol{V}_q\boldsymbol{u}_h, \qquad \boldsymbol{v}_h = \widetilde{\boldsymbol{P}}_q^k\boldsymbol{v}\left(\boldsymbol{u}_q\right), \qquad \widetilde{\boldsymbol{v}} = \begin{bmatrix}\boldsymbol{V}_q\\ \boldsymbol{V}_f\end{bmatrix}\boldsymbol{v}_h, \qquad \widetilde{\boldsymbol{u}} = \begin{bmatrix}\widetilde{\boldsymbol{u}}_q\\ \widetilde{\boldsymbol{u}}_f\end{bmatrix} = \boldsymbol{u}\left(\widetilde{\boldsymbol{v}}\right). \tag{30}$$

A semi-discrete DG formulation for $\boldsymbol{u}_h$ can be constructed using the variables defined in (30)

$$\begin{aligned}&\frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{dt}} + \begin{bmatrix}\widetilde{\boldsymbol{P}}_q^k & \widetilde{\boldsymbol{L}}_q^k\end{bmatrix}\sum_{j=1}^d\left(2\boldsymbol{D}_k^j \circ \boldsymbol{F}_{j,S}\right)\boldsymbol{1} + \sum_{j=1}^d \widetilde{\boldsymbol{L}}_q^k\text{diag}\left(\boldsymbol{n}_j\right)\left(\boldsymbol{f}_j^* - \boldsymbol{f}_j(\widetilde{\boldsymbol{u}}_f)\right) = 0,\\ &\left(\boldsymbol{F}_{j,S}\right)_{mn} = \boldsymbol{f}_{j,S}\left(\left(\widetilde{\boldsymbol{u}}\right)_m, \left(\widetilde{\boldsymbol{u}}\right)_n\right), \qquad 1 \le m,n \le N_q + N_q^f,\\ &\boldsymbol{f}_j^* = \boldsymbol{f}_{j,S}(\widetilde{\boldsymbol{u}}_f^+, \widetilde{\boldsymbol{u}}_f) \text{ on interior faces.}\end{aligned} \tag{31}$$

Since the weight-adjusted mass matrix inverse is low-storage, and since the matrices $\boldsymbol{D}_k^j$ in (18) can be assembled from reference matrices $\boldsymbol{D}_N^i$ and the values of geometric terms at quadrature points, the overall

scheme requires only $O(N^d)$ storage per element. We can additionally show that formulation (31) is entropy conservative in the same sense as Theorem 1:

Theorem 2. *Let $\boldsymbol{f}_{i,S}$ be an entropy conservative flux from Definition 1 and assume that $\boldsymbol{Q}_k^j \mathbf{1} = 0$ for $j = 1, \ldots, d$ over each element D^k. Then, (31) is entropy conservative in the sense that*

$$\sum_k \mathbf{1}^T \mathrm{diag}\left(\boldsymbol{J}^k\right) \boldsymbol{W} \frac{\mathrm{d}U(\boldsymbol{u}_q)}{\mathrm{dt}} = \sum_k \sum_{j=1}^d \mathbf{1}^T \mathrm{diag}\left(\boldsymbol{n}_j \circ \boldsymbol{J}_f^k\right) \boldsymbol{W}_{\partial\Omega} \left(\psi_j\left(\widetilde{\boldsymbol{u}}_f\right) - \widetilde{\boldsymbol{v}}_f^T \boldsymbol{f}_j^*\right).$$

Proof. The proof is similar to that of Theorem 1. First, recall from the definitions of the weight-adjusted projection matrix (28) and weight-adjusted lift matrix (29) that

$$\widetilde{\boldsymbol{P}}_q^k = \boldsymbol{M}^{-1} \boldsymbol{M}_{1/J^k} \boldsymbol{M}^{-1} \boldsymbol{V}_q^T \boldsymbol{W} \mathrm{diag}\left(\boldsymbol{J}^k\right), \tag{32}$$

$$\widetilde{\boldsymbol{L}}_q^k = \boldsymbol{M}^{-1} \boldsymbol{M}_{1/J^k} \boldsymbol{M}^{-1} \boldsymbol{V}_f^T \boldsymbol{W}_f \mathrm{diag}\left(\boldsymbol{J}_f^k\right). \tag{33}$$

Then, multiplying by the weight-adjusted mass matrix $\boldsymbol{M}\boldsymbol{M}_{1/J^k}^{-1}\boldsymbol{M}$ on both sides of (31) and using (32), (33) yields the weak form of (31)

$$\boldsymbol{M}\boldsymbol{M}_{1/J^k}^{-1}\boldsymbol{M}\frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{dt}} + \begin{bmatrix} \boldsymbol{V}_q^T & \boldsymbol{V}_f^T \end{bmatrix} \sum_{j=1}^d \left(2\boldsymbol{Q}_k^j \circ \boldsymbol{F}_{j,S}\right) \mathbf{1} + \sum_{j=1}^d \boldsymbol{V}_f^T \boldsymbol{W}_f \mathrm{diag}\left(\boldsymbol{n}_j\right) \left(\boldsymbol{f}_j^* - \boldsymbol{f}_j(\widetilde{\boldsymbol{u}}_f)\right) = 0. \tag{34}$$

Testing with the weight-adjusted projection of the entropy variables $\boldsymbol{v}_h = \widetilde{\boldsymbol{P}}_q^k \boldsymbol{v}\left(\boldsymbol{u}_q\right)$ and using (32) then yields for the time term

$$\begin{aligned}
\left(\widetilde{\boldsymbol{P}}_q^k \boldsymbol{v}\left(\boldsymbol{u}_q\right)\right)^T \boldsymbol{M}\boldsymbol{M}_{1/J^k}^{-1}\boldsymbol{M}\frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{dt}} &= \boldsymbol{v}\left(\boldsymbol{u}_q\right)^T \boldsymbol{W} \mathrm{diag}\left(\boldsymbol{J}^k\right) \boldsymbol{V}_q \boldsymbol{M}^{-1} \boldsymbol{M}_{1/J^k} \boldsymbol{M}^{-1} \boldsymbol{M}\boldsymbol{M}_{1/J^k}^{-1}\boldsymbol{M}\frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{dt}} \\
&= \boldsymbol{v}\left(\boldsymbol{u}_q\right)^T \boldsymbol{W} \mathrm{diag}\left(\boldsymbol{J}^k\right) \frac{\mathrm{d}\boldsymbol{V}_q \boldsymbol{u}_h}{\mathrm{dt}} = \mathbf{1}^T \boldsymbol{W} \mathrm{diag}\left(\boldsymbol{J}^k\right) \left(\mathrm{diag}\left(\boldsymbol{v}\left(\boldsymbol{u}_q\right)\right) \frac{\mathrm{d}\boldsymbol{u}_q}{\mathrm{dt}}\right) \\
&= \mathbf{1}^T \boldsymbol{W} \mathrm{diag}\left(\boldsymbol{J}^k\right) \frac{\mathrm{d}U(\boldsymbol{u}_q)}{\mathrm{dt}}.
\end{aligned}$$

The remainder of the proof is identical to that of Theorem 1 and [1]. □

4.3. **Analysis of weight-adjusted projection.** The construction of the discretely entropy stable weight-adjusted DG formulation (31) replaces the L^2 projection operator Π_N^k with the weight-adjusted projection operator $\widetilde{\Pi}_N^k$. While this preserves entropy stability, it is unclear whether $\widetilde{\Pi}_N^k$ is high order accurate. In this section, we prove that the weight-adjusted projection is high order accurate due to the fact that, for a fixed geometric mapping and sufficiently regular u, the difference between the L^2 and weight-adjusted projection is $\left\|\Pi_N^k u - \widetilde{\Pi}_N^k u\right\|_{L^2(\Omega)} = O(h^{N+2})$. Because the approximation error for the L^2 projection is $O(h^{N+1})$ for sufficiently regular u, the difference between the L^2 and weight-adjusted projection converges faster than the L^2 best approximation error. Consequentially, solutions computed using the L^2 and weight-adjusted projection are typically indistinguishable for a fixed geometric mapping [39].

We first note that $\widetilde{\Pi}_N^k = \Pi_N\left(\frac{1}{J^k}\Pi_N\left(uJ^k\right)\right)$ is self-adjoint with respect to the J-weighted L^2 inner product

$$\left(J^k \widetilde{\Pi}_N^k u, v\right)_{\widehat{D}} = \left(\Pi_N\left(\frac{1}{J^k}\Pi_N\left(uJ^k\right)\right), vJ^k\right)_{\widehat{D}} = \left(uJ^k, \Pi_N\left(\frac{1}{J^k}\Pi_N\left(vJ^k\right)\right)\right)_{\widehat{D}} = \left(uJ^k, \widetilde{\Pi}_N^k v\right)_{\widehat{D}}. \tag{35}$$

Furthermore, using that the operator T_{1/J^k}^{-1} is self-adjoint for $v \in P^N\left(\widehat{D}\right)$ with respect to the L^2 inner product [3], we find that a projection-like property holds for the weight-adjusted L^2 inner product

$$\left(T_{1/J^k}^{-1} \widetilde{\Pi}_N^k u, v\right)_{\widehat{D}} = \left(\frac{1}{J^k}\Pi_N(uJ^k), T_{1/J^k}^{-1} v\right)_{\widehat{D}} = \left(\Pi_N(uJ^k), v\right)_{\widehat{D}} = \left(uJ^k, v\right)_{\widehat{D}}, \qquad \forall v \in P^N \widehat{D}. \tag{36}$$

To prove $\left\|\Pi_N^k u - \widetilde{\Pi}_N^k u\right\|_{L^2\left(\widehat{D}\right)} = O(h^{N+2})$, we use a generalized inverse inequality and results from [3, 4]. We first introduce a modification of Theorem 3.1 in [40, 3]

Theorem 3. *Let D^k be a quasi-regular element with representative size $h = \operatorname{diam}\left(D^k\right)$, and let $\widehat{D}$ be the reference element. For $N \geq 0$, $w \in W^{N+1,\infty}\left(D^k\right)$, and $u \in W^{r,2}\left(D^k\right)$,*

$$\left\| u - \frac{1}{w}\Pi_N\left(uw\right)\right\|_{L^2\left(\widehat{D}\right)} \leq Ch^{\min(r,N+1)} \left\|\frac{1}{\sqrt{J^k}}\right\|_{L^\infty(D^k)} \left\|\frac{1}{w}\right\|_{L^\infty(D^k)} \|w\|_{W^{N+1,\infty}(D^k)} \|u\|_{W^{r,2}(D^k)}.$$

The proof is a straightforward modification of the proofs presented in [40, 3] accounting for reduced regularity of u when $r < (N+1)$. The next result we need is a generalized inverse inequality.

Lemma 1. *Let $v \in P^N\left(\widehat{D}\right)$, and let $h = \operatorname{diam}\left(D^k\right)$. Then,*

$$\|v\|_{W^{N+1,2}(D^k)} \leq C_N h^{-N} \left\|\sqrt{J^k}\right\|_{L^\infty} \left\|\frac{1}{\sqrt{J^k}}\right\|_{L^\infty} \left\|\frac{1}{J^k}\boldsymbol{G}^k\right\|_{W^{N,\infty}(D^k)} \|v\|_{L^2(D^k)}.$$

where C_N depends on N, but is independent of h.

Proof. By applying Faà di Bruno's formula, we can express the degree $(N+1)$ Sobolev norm of v on D^k in terms of derivatives of v on the reference element $\widehat{D}$. Noting that all $(N+1)$ derivatives of v disappear for $v \in P^N\left(\widehat{D}\right)$ allows us to bound the degree $(N+1)$ Sobolev norm of v by its degree N Sobolev norm

$$\|v\|_{W^{N+1,2}(D^k)} \leq C_N \left\|\frac{1}{J^k}\boldsymbol{G}^k\right\|_{W^{N,\infty}(D^k)} \|v\|_{W^{N,2}(D^k)},$$

where $\boldsymbol{G}^k$ is the matrix of scaled geometric terms for D^k. Then, a scaling argument [33, 41] yields

$$\|v\|_{W^{N,2}(D^k)} \leq C_1 h^{-N} \left\|\sqrt{J^k}\right\|_{L^\infty} \|v\|_{W^{N,2}\left(\widehat{D}\right)}. \tag{37}$$

The quantity $\|v\|_{W^{N,2}\left(\widehat{D}\right)}$ can be bounded by noting that $v \in P^N\left(\widehat{D}\right)$. Since $P^N\left(\widehat{D}\right)$ is finite-dimensional, the Sobolev norm can be bounded from above by the L^2 norm of $\widehat{D}$ with a constant C_2 depending on N, d. By another scaling argument, we have

$$\|v\|_{W^{N+1,2}\left(\widehat{D}\right)} \leq C_2 \|v\|_{L^2\left(\widehat{D}\right)} \leq \left\|\frac{1}{\sqrt{J^k}}\right\|_{L^\infty} C_2 \|v\|_{L^2(D^k)}. \tag{38}$$

□

We can now prove that $\widetilde{\Pi}_N^k u$ is superconvergent to the curvilinear L^2 projection $\Pi_N^k u$:

Theorem 4. *Let $u \in W^{r,2}\left(D^k\right)$. The difference between the L^2 projection $\Pi_N^k u$ and the weight-adjusted projection $\widetilde{\Pi}_N^k u$ is*

$$\left\|\Pi_N^k u - \widetilde{\Pi}_N^k u\right\|_{L^2(D^k)} \leq C_N C_J h^{\min(r,N+1)+1} \|u\|_{W^{N+1,2}(D^k)},$$

where C_N is a mesh-independent constant which depends on N, d and C_J is

$$C_J = \left\|J^k\right\|_{L^\infty(D^k)}^{1.5} \left\|\frac{1}{J^k}\right\|_{L^\infty(D^k)}^{1.5} \left\|\frac{1}{J^k}\right\|_{W^{N+1,\infty}(D^k)} \left\|J^k\right\|_{W^{N+1,\infty}(D^k)} \left\|\frac{1}{J^k}\boldsymbol{G}^k\right\|_{W^{N,\infty}(D^k)}.$$

Proof. We can rewrite the norm of the difference between the weight-adjusted and L^2 projections

$$\left\|\Pi_N^k u - \widetilde{\Pi}_N^k u\right\|_{L^2(D^k)}^2 = \left(\Pi_N^k u - \widetilde{\Pi}_N^k u, vJ^k\right)_{\widehat{D}}, \qquad v = \Pi_N^k u - \widetilde{\Pi}_N^k u.$$

Because $v \in P^N\left(\widehat{D}\right)$, we can also evaluate the squared error as

$$\begin{aligned}\left\|\Pi_N^k u - \widetilde{\Pi}_N^k u\right\|_{L^2(D^k)}^2 &= \left|\left(\Pi_N^k u, vJ^k\right)_{\widehat{D}} - \left(\widetilde{\Pi}_N^k u, vJ^k\right)_{\widehat{D}}\right| = \left|\left(u, vJ^k\right)_{\widehat{D}} - \left(\widetilde{\Pi}_N^k u, vJ^k\right)_{\widehat{D}}\right| \\ &\leq \left\|J^k\right\|_{L^\infty(D^k)} \left|\left(u - \widetilde{\Pi}_N^k u, v\right)_{\widehat{D}}\right|.\end{aligned}$$

We can then note that $\widetilde{\Pi}_N^k u = \Pi_N\left(\frac{1}{J^k}\Pi_N\left(uJ^k\right)\right)$ to show that

$$\begin{aligned}\left(u-\widetilde{\Pi}_N^k u, v\right)_{\widehat{D}} &= \left(uJ^k, \frac{v}{J^k}\right)_{\widehat{D}} - \left(\Pi_N\left(\frac{1}{J^k}\Pi_N\left(uJ^k\right)\right), v\right)_{\widehat{D}} \\ &= \left(uJ^k, \frac{v}{J^k}\right)_{\widehat{D}} - \left(\Pi_N\left(uJ^k\right), \frac{v}{J^k}\right)_{\widehat{D}}.\end{aligned}$$

Adding and subtracting $\left(\Pi_N\left(uJ^k\right), \Pi_N\left(\frac{v}{J^k}\right)\right)_{\widehat{D}}$ and using Theorem 3 (noting that $v \in W^{N+1,2}\left(D^k\right)$ since it is polynomial) gives

$$\begin{aligned}\left\|\Pi_N^k u - \widetilde{\Pi}_N^k u\right\|_{L^2(D^k)}^2 &\leq \left\|J^k\right\|_{L^\infty(D^k)} \left|\left(u-\widetilde{\Pi}_N^k u, v\right)_{\widehat{D}}\right| \\ &= \left\|J^k\right\|_{L^\infty(D^k)} \left|\left(uJ^k - \Pi_N\left(uJ^k\right), \frac{v}{J^k} - \Pi_N\left(\frac{v}{J^k}\right)\right)_{\widehat{D}}\right| \\ &\leq \left\|J^k\right\|_{L^\infty(D^k)} \left\|uJ^k - \Pi_N\left(uJ^k\right)\right\|_{\widehat{D}} \left\|\frac{v}{J^k} - \Pi_N\left(\frac{v}{J^k}\right)\right\|_{\widehat{D}} \\ &\leq C h^{\min(r,N+1)+N+1} \widetilde{C}_J \left\|u\right\|_{W^{r,2}(D^k)} \left\|v\right\|_{W^{N+1,2}(D^k)},\end{aligned}$$

where

$$\widetilde{C}_J = \left\|J^k\right\|_{L^\infty(D^k)} \left\|\frac{1}{J^k}\right\|_{L^\infty(D^k)} \left\|J^k\right\|_{W^{N+1,\infty}(D^k)} \left\|\frac{1}{J^k}\right\|_{W^{N+1,\infty}(D^k)}.$$

Applying Lemma 1 then yields

$$\left\|\Pi_N^k u - \widetilde{\Pi}_N^k u\right\|_{L^2(D^k)}^2 \leq C_N h^{\min(r,N+1)+1} C_J \left\|u\right\|_{W^{N+1,2}(D^k)} \left\|v\right\|_{L^2(D^k)}.$$

Dividing through by $\left\|v\right\|_{L^2(D^k)} = \left\|\Pi_N^k u - \widetilde{\Pi}_N^k u\right\|_{L^2(D^k)}$ gives the desired result. □

Theorem 3 can be used to show that the L^2 error between $\Pi_N^k u, \widetilde{\Pi}_N^k u$ and $u \in W^{r,2}\left(D^k\right)$ is $O(h^r)$. Theorem 4 demonstrates that the L^2 difference between $\Pi_N^k u, \widetilde{\Pi}_N^k u$ is $O(h^{r+1})$, or at least one order higher than the approximation error. We note that optimal convergence of the weight-adjusted projection requires that the geometric mapping $\boldsymbol{\Phi}_k$ is asymptotically affine (i.e., the Sobolev norm of $J^k, \boldsymbol{G}^k$ does not grow under mesh refinement), which is ensured under nested mesh refinement and appropriate curvilinear blending strategies [42, 40, 4].

4.3.1. *Local and global conservation.* We next address local conservation of the weight-adjusted scheme (31), which is also referred to as primary conservation [43, 12, 13, 15]. We begin by noting that (31) is locally conservative with respect to the weight-adjusted inner product. Testing (34) with $\mathbf{1}$ yields

$$\mathbf{1}^T \boldsymbol{M} \boldsymbol{M}_{1/J^k}^{-1} \boldsymbol{M} \frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{dt}} + \sum_{j=1}^{d} \mathbf{1}^T \left(2\boldsymbol{Q}_k^j \circ \boldsymbol{F}_{j,S}\right) \mathbf{1} + \sum_{j=1}^{d} \mathbf{1}^T \boldsymbol{W}_f \mathrm{diag}\left(\boldsymbol{n}_j\right) \left(\boldsymbol{f}_j^* - \boldsymbol{f}_j(\widetilde{\boldsymbol{u}}_f)\right) = 0.$$

Local conservation can be shown by applying the SBP property (19) and noting that $\left(\boldsymbol{B}_k^i \circ \boldsymbol{F}_{j,S}\right)\mathbf{1} = \boldsymbol{W}_f \mathrm{diag}\left(\boldsymbol{n}_j \circ \boldsymbol{J}_f^k\right) \boldsymbol{f}_j(\boldsymbol{u})$ (using the consistency of $\boldsymbol{f}_S$ and diagonal nature of $\boldsymbol{B}_k^i$) yields

$$\mathbf{1}^T \boldsymbol{M} \boldsymbol{M}_{1/J^k}^{-1} \boldsymbol{M} \frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{dt}} + \sum_{j=1}^{d} \mathbf{1}^T \left(\left(\boldsymbol{Q}_k^j - \left(\boldsymbol{Q}_k^j\right)^T\right) \circ \boldsymbol{F}_{j,S}\right) \mathbf{1} + \sum_{j=1}^{d} \mathbf{1}^T \boldsymbol{W}_f \mathrm{diag}\left(\boldsymbol{n}_j\right) \left(\boldsymbol{f}_j^*\right) = 0.$$

Since $\left(\boldsymbol{Q}_k^j - \left(\boldsymbol{Q}_k^j\right)^T\right)$ is skew-symmetric and $\boldsymbol{F}_{j,S}$ is symmetric, the Hadamard product of these two matrices is skew-symmetric. As a result, $\mathbf{1}^T \left(\left(\boldsymbol{Q}_k^j - \left(\boldsymbol{Q}_k^j\right)^T\right) \circ \boldsymbol{F}_{j,S}\right) \mathbf{1} = 0$ and

$$\mathbf{1}^T \boldsymbol{M} \boldsymbol{M}_{1/J^k}^{-1} \boldsymbol{M} \frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{dt}} + \sum_{j=1}^{d} \mathbf{1}^T \boldsymbol{W}_f \mathrm{diag}\left(\boldsymbol{n}_j\right) \left(\boldsymbol{f}_j^*\right) = 0. \tag{39}$$

Global conservation is shown by summing (39) over all elements D^k. Because the flux $\boldsymbol{f}_j^*$ is single-valued on each face and the outward normal $\boldsymbol{n}_j$ changes sign on adjacent elements, the contributions $\mathbf{1}^T \boldsymbol{W}_f \mathrm{diag}\left(\boldsymbol{n}_j\right)\left(\boldsymbol{f}_j^*\right)$ cancel on all interior interfaces.[1] On periodic meshes, this yields conservation with respect to the weight-adjusted mass matrix

$$\sum_k \mathbf{1}^T \boldsymbol{M} \boldsymbol{M}_{1/J^k}^{-1} \boldsymbol{M} \frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{dt}} = 0.$$

However, while the weight-adjusted approximation of the mass matrix is high order accurate and efficient, it does not preserve the average over a physical element, which is equivalent to the J-weighted average over the reference element. This is due to the fact that, in general,

$$\int_{\widehat{D}} u J^k \,\mathrm{d}\widehat{\boldsymbol{x}} - \int_{\widehat{D}} T_{1/J^k}^{-1} u \,\mathrm{d}\widehat{\boldsymbol{x}} \approx \mathbf{1}^T \boldsymbol{M}_{J^k} \boldsymbol{u} - \mathbf{1}^T \boldsymbol{M} \boldsymbol{M}_{1/J^k}^{-1} \boldsymbol{M} \boldsymbol{u} \neq 0. \tag{40}$$

Results in [3] show that the difference between the true mean and weight-adjusted mean in (40) converges extremely fast at a rate of $O(h^{2N+2})$. However, for systems of conservation laws, it is often desired that the local element average is preserved exactly up to machine precision. We present two simpler approaches to ensuring local conservation in this section.

The simplest way to ensure local conservation is to approximate J using a degree N polynomial and utilize a sufficiently accurate quadrature. Then, we have the following lemma:

Lemma 2. *Let $J^k \in P^N$, and let integrals be computed using quadrature which is exact for degree $2N$ polynomials. Then,*

$$\mathbf{1}^T \boldsymbol{M} \boldsymbol{M}_{1/J^k}^{-1} \boldsymbol{M} \boldsymbol{u} = \int_{\widehat{D}} T_{1/J^k}^{-1} u \,\mathrm{d}\widehat{\boldsymbol{x}} = \int_{\widehat{D}} u J^k \,\mathrm{d}\widehat{\boldsymbol{x}}.$$

Proof. The proof relies on (26), which states that $\left(\frac{1}{J^k} T_{1/J^k}^{-1} u, v\right)_{\widehat{D}} = (u, v)_{\widehat{D}}$ for all $v \in P^N$. If J^k is a polynomial of degree N, then taking $v = 1$ yields

$$\left(T_{1/J^k}^{-1} u, 1\right)_{\widehat{D}} = \left(\frac{1}{J^k} T_{1/J^k}^{-1} u, J^k\right)_{\widehat{D}} = \left(u, J^k\right)_{\widehat{D}} = \int_{\widehat{D}} u J^k \,\mathrm{d}\widehat{\boldsymbol{x}}.$$

Additionally, the proof still holds if integrals are approximated using a quadrature rule which exactly integrates $u J^k \in P^{2N}$. □

We note that, for isoparametric curved elements, $J^k \notin P^N$ in general (in 2D, $J^k \in P^{2N-2}$, while in 3D, $J^k \in P^{3N-3}$ [44]). Thus, to ensure local conservation, we will approximate J^k using a degree N polynomial (for example, the interpolant or L^2 projection onto P^N). We note that this approximation is required only in the weight-adjusted mass matrix, and does not modify the scaled geometric terms G_{ij}^k.

The second approach relies on a simple correction which restores exact conservation of the true mean. In [4], it was shown that a rank one correction of the weight-adjusted mass matrix inverse preserves the mean exactly. However, this requires the use of the Sherman–Morrison formula to compute the inverse of a rank one matrix update, which can be cumbersome to incorporate. We present a simpler explicit correction formula which does not involve matrices. Let u_J and u_{WADG} be defined through

$$\begin{aligned} \left(u_J J, v\right)_{\widehat{D}} &= (f, v)_{\widehat{D}}, \qquad \forall v \in P^N\left(\widehat{D}\right), \\ \left(T_{1/J}^{-1} u_{\mathrm{WADG}}, v\right)_{\widehat{D}} &= (f, v)_{\widehat{D}}, \qquad \forall v \in P^N\left(\widehat{D}\right), \end{aligned}$$

where u_J corresponds to the inversion of the weighted mass matrix and u_{WADG} corresponds to the inversion of a weight-adjusted mass matrix. For example, if $f = uJ$ for some function $u(\boldsymbol{x})$, then $u_J = \Pi_N^k u$ and $u_{\mathrm{WADG}} = \widetilde{\Pi}_N^k u$. To ensure both local and global conservation, we require that the weighted average of

[1] Conservation still holds when incorporating entropy dissipation through penalty terms involving jumps. This is because the definition of the jump changes sign on adjacent elements, such that jump contributions cancel when summing over all elements.

u_{WADG} is the same as the weighted average of u_J. Let the conservative weight-adjusted $\widetilde{u}_{\text{WADG}}$ be defined as

$$\widetilde{u}_{\text{WADG}} = u_{\text{WADG}} + \frac{\int_{\widehat{D}} (f - J u_{\text{WADG}}) \, \mathrm{d}\widehat{\boldsymbol{x}}}{\int_{\widehat{D}} J \, \mathrm{d}\widehat{\boldsymbol{x}}}. \tag{41}$$

Taking the weighted integral of $\widetilde{u}_{\text{WADG}}$ yields

$$\int_{\widehat{D}} \widetilde{u}_{\text{WADG}} J \, \mathrm{d}\widehat{\boldsymbol{x}} = \int_{\widehat{D}} f \, \mathrm{d}\widehat{\boldsymbol{x}}.$$

In other words, (41) ensures local conservation by correcting the weighted average of u_{WADG} to match that of u_J. Applying this correction to the right hand side of (31) then yields a scheme which locally and globally conserves mean values of the conservative variables.

This approach is applicable to an arbitrary weight, and can be generalized to matrix-valued weights as well [21]. Moreover, using Theorem 6 in [3], one can show that the L^2 norm of the difference $\widetilde{u}_{\text{WADG}} - u_{\text{WADG}}$ is $O(h^{2N+1})$, and does not affect high order accuracy.

5. Enforcing the discrete geometric conservation law

An important aspect of Theorem 1 is the assumption that $\boldsymbol{Q}_k^j \mathbf{1} = 0$ for $j = 1, \ldots, d$. However, this is not always guaranteed to hold for $\boldsymbol{Q}_k^j$ as defined through (18). In this section, we discuss methods of constructing the geometric terms G_{ij}^k for curvilinear meshes in a way that ensures $\boldsymbol{Q}_k^j \mathbf{1} = 0$.

From (18), the condition $\boldsymbol{Q}_k^j \mathbf{1} = 0$ is equivalent to

$$\begin{aligned}\boldsymbol{Q}_k^j \mathbf{1} = \boldsymbol{W}_N \sum_{j=1}^{d} \boldsymbol{D}_N^j \circ \left\{\{\boldsymbol{G}_{ij}^k\}\right\} \mathbf{1} &= \frac{1}{2} \boldsymbol{W}_N \sum_{j=1}^{d} \left(\operatorname{diag}\left(\boldsymbol{G}_{ij}^k\right) \boldsymbol{D}_N^j \mathbf{1} + \boldsymbol{D}_N^j \operatorname{diag}\left(\boldsymbol{G}_{ij}^k\right) \mathbf{1} \right) \\ &= \frac{1}{2} \boldsymbol{W}_N \sum_{j=1}^{d} \boldsymbol{D}_N^j \boldsymbol{G}_{ij}^k = 0,\end{aligned}$$

where we have used that $\boldsymbol{D}_N^j \mathbf{1} = 0$ to eliminate the first term. Since $\boldsymbol{W}_N$ is a diagonal matrix with positive entries, $\boldsymbol{Q}_k^j \mathbf{1} = 0$ is equivalent to ensuring that a discrete version of the GCL (14) holds

$$\sum_{j=1}^{d} \boldsymbol{D}_N^j \boldsymbol{G}_{ij}^k = 0. \tag{42}$$

This condition is required to ensure that free-stream preservation holds at the discrete level. In other words, we wish to ensure that the semi-discrete scheme preserves (for $\boldsymbol{u}$ constant)

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{f}(\boldsymbol{u}) = \frac{\partial \boldsymbol{u}}{\partial t} = 0.$$

For isoparametric geometric mappings (where the degree of the mapping matches the degree of the polynomial approximation) in two dimensions, the GCL is naturally enforced by the "cross-product" form, noting that the scaled metric terms G_{ij}^k are exactly polynomials of degree N. As a result, computing the metric terms exactly automatically enforces that both the continuous GCL (14) and the discrete GCL (42) are satisfied. However, the discrete GCL is not always maintained at the discrete level in 3D.

In three dimensions, geometric terms are typically computed in "cross-product" form

$$\begin{bmatrix} G_{1i}^k \\ G_{2i}^k \\ G_{3i}^k \end{bmatrix} = \frac{\partial \boldsymbol{x}}{\partial \widehat{x}_j} \times \frac{\partial \boldsymbol{x}}{\partial \widehat{x}_k}, \qquad (i, j, k) = (1, 2, 3), \text{ cyclic.} \tag{43}$$

Note the abuse of notation here and in the sequel, the superscript k refers to the element number and the subscript k to the cyclic index. This formula can be used to compute the geometric terms exactly at volume and surface quadrature points. However, because $\frac{\partial \boldsymbol{x}}{\partial \widehat{x}_j}, \frac{\partial \boldsymbol{x}}{\partial \widehat{x}_k} \in P^{N-1}$, the geometric terms G_{ij}^k are polynomials of degree P^{2N-2}. The discrete GCL condition holds only if $G_{ij}^k \in P^{2N-2}$ are differentiated exactly; however,

because applying $\boldsymbol{D}_N^j$ involves the L^2 projection, and because G_{ij}^k and its L^2 projection onto degree N polynomials can differ, the discrete GCL (42) does not hold in general [2].

This can be remedied by using an alternative form of the geometric terms, which ensures that (42) is satisfied a-priori [20, 45, 2]. The geometric terms G_{ij}^k can also be computed using a "conservative curl" form, where G_{ij}^k are the image of the curl applied to some quantity as follows:

$$\begin{bmatrix} G_{1i}^k \\ G_{2i}^k \\ G_{3i}^k \end{bmatrix} = \begin{bmatrix} \left(-\widehat{\nabla} \times \left(x_3 \widehat{\nabla} x_2\right)\right)_i \\ \left(\widehat{\nabla} \times \left(x_3 \widehat{\nabla} x_1\right)\right)_i \\ \left(\widehat{\nabla} \times \left(x_1 \widehat{\nabla} x_2\right)\right)_i \end{bmatrix}, \tag{44}$$

where the subscript i denotes the ith component of the vector quantity. From (44), it can be observed that, because the divergence of a curl vanishes, the continuous GCL condition (14) holds. The central idea of [45, 2] is to use (44), but to interpolate before applying the curl

$$\begin{bmatrix} G_{1i}^k \\ G_{2i}^k \\ G_{3i}^k \end{bmatrix} = \begin{bmatrix} \left(-\widehat{\nabla} \times I_N\left(x_3 \widehat{\nabla} x_2\right)\right)_i \\ \left(\widehat{\nabla} \times I_N\left(x_3 \widehat{\nabla} x_1\right)\right)_i \\ \left(\widehat{\nabla} \times I_N\left(x_1 \widehat{\nabla} x_2\right)\right)_i \end{bmatrix}, \tag{45}$$

where I_N denotes the degree N polynomial interpolation operator. Since the geometric terms are still computed by applying a curl, the continuous GCL condition (14) is still satisfied. We shall also show that this approximation also satisfies the discrete GCL condition.

We adopt a slight modification of (45) in this work which is tailored towards triangular and tetrahedral elements. Because the geometric terms are computed by applying the curl, the geometric terms are approximated as degree $(N-1)$ polynomials rather than degree N polynomials, which can reduce accuracy. Instead, we approximate geometric terms by using the interpolation operator I_{N+1} onto degree $(N+1)$ polynomials, then interpolating back to degree N polynomials

$$\begin{bmatrix} G_{1i}^k \\ G_{2i}^k \\ G_{3i}^k \end{bmatrix} = \begin{bmatrix} I_N\left(-\widehat{\nabla} \times I_{N+1}\left(x_3 \widehat{\nabla} x_2\right)\right)_i \\ I_N\left(\widehat{\nabla} \times I_{N+1}\left(x_3 \widehat{\nabla} x_1\right)\right)_i \\ I_N\left(\widehat{\nabla} \times I_{N+1}\left(x_1 \widehat{\nabla} x_2\right)\right)_i \end{bmatrix}. \tag{46}$$

For any $u \in P^N\left(\widehat{D}\right)$, $\frac{\partial u}{\partial \widehat{x}_i} \in P^{N-1}\left(\widehat{D}\right)$ for $i = 1, 2, 3$. Thus, the interpolation to degree N polynomials is exact, since the derivatives of a degree $(N+1)$ polynomial are degree N on triangles and tetrahedra.

Remark. *The accuracy of (46) depends on the choice of interpolation points. It is well known that interpolating at equispaced points can result in inaccurate polynomial approximations. One can determine good interpolation point sets by optimizing over some measure of interpolation quality (such as the Lebesgue constant), and in practice, sets of interpolation points are pre-computed for some polynomial degrees $N = 1, \ldots, N_{\max}$ on the reference element and stored [46, 47], or explicitly computed as the image of equispaced points under an appropriately defined mapping [48, 49, 50].*

To prove that the construction (46) satisfies Assumption 2, we must assume that the interpolation points for a degree N element include an appropriate number of points on each face. We note that these assumptions exclude interpolation points which lie purely in the interior of an element, such as those introduced in [51, 52]. We can now show that the geometric terms satisfy all conditions necessary to guarantee entropy stability:

Theorem 5. *Let the mesh consist of triangles or tetrahedral elements which satisfy Assumption 2, and let the interpolation points which define the degree N interpolation operator be distributed such that N_p^f points lie on each face. Then, the approximate geometric terms $\widetilde{J^k \boldsymbol{G}_{ij}^k}$ and approximate scaled normals $\widetilde{n_i J_f^k}$ computed using (46) and (17)) satisfy both the discrete GCL condition (42) and Assumption 2. Additionally, the error*

in the approximation satisfies

$$\left\| G_{ij}^k - \widetilde{G_{ij}^k} \right\|_{L^2(\Omega)}^2 \leq C_N \left| \Omega \right| h^{N+2} \sqrt{\sum_{i=1}^{d} \sum_{k} \left\| \boldsymbol{r}_i \right\|_{W^{N+2,2}(D^k)}^2}.$$

Proof. The satisfaction of (42) relies on the fact that $\widetilde{G_{ij}^k}$ is a degree N polynomial and is equal to its own L^2 projection. Let $\widetilde{\boldsymbol{G}_{ij}^k}$ denote the polynomial coefficients of $\widetilde{G_{ij}^k}$. Then, applying $\boldsymbol{D}_N^j$ to evaluations of $\widetilde{G_{ij}^k}$ at volume and surface quadrature points and using (13), we have

$$\sum_{j=1}^{d} \boldsymbol{D}_N^j \begin{bmatrix} \boldsymbol{V}_q \\ \boldsymbol{V}_f \end{bmatrix} \widetilde{\boldsymbol{G}_{ij}^k} = \sum_{j=1}^{d} \begin{bmatrix} \boldsymbol{V}_q \boldsymbol{D}_i \widetilde{\boldsymbol{G}_{ij}^k} \\ 0 \end{bmatrix}.$$

The entries of $\boldsymbol{V}_q \boldsymbol{D}_i \widetilde{\boldsymbol{G}_{ij}^k}$ correspond to values of the derivatives of $\widetilde{G_{ij}^k}$ evaluated at quadrature points. Since

$$\sum_{j=1}^{d} \frac{\partial}{\partial \widehat{x}_j} \widetilde{G_{ij}^k} = 0$$

by construction using (46), $\sum_{j=1,\ldots,d} \boldsymbol{V}_q \boldsymbol{D}_i \widetilde{\boldsymbol{G}_{ij}^k} = \boldsymbol{0}$ as well.

Equation (17) of Assumption 2 is satisfied by directly constructing the scaled normals $\widetilde{\boldsymbol{n}} J_f^k$ using values of $\widetilde{G_{ij}^k}$ at quadrature points. We must now prove that the construction of $\widetilde{\boldsymbol{n}} J_f^k$ implies that Equation (16) holds. This is not immediately clear; since the normals are constructed from the approximate geometric terms and the formula (17), it is not guaranteed that $\widetilde{\boldsymbol{n}}^+ J_f^{k,+} = -\widetilde{\boldsymbol{n}} J_f^k$ will hold across a shared face. However, the scaled normal vectors involve only nodal values on the shared face because the normals are computed in terms of the tangential reference derivatives [2]. Thus, assuming a watertight mesh, the interpolation nodes on two neighboring elements will coincide for a shared face f, such that the trace of $\widetilde{G_{ij}^k}$ from either neighboring element will be the same lower-dimensional polynomial on f. This is sufficient to ensure that the scaled normal vectors $\widetilde{\boldsymbol{n}}^+ J_f^{k,+}, \widetilde{\boldsymbol{n}} J_f^k$ will be equal and opposite.

The local L^2 error $\left\| G_{ij}^k - \widetilde{G_{ij}^k} \right\|_{L^2(D^k)}$ can be bounded by noting that, since the error $G_{ij}^k - \widetilde{G_{ij}^k}$ consists of linear combinations of derivatives of the interpolation error $\boldsymbol{r}_i - I_{N+1} \boldsymbol{r}_i$, it can be bounded by the H^1-seminorm of the latter quantity

$$\begin{aligned} \left\| G_{ij}^k - \widetilde{G_{ij}^k} \right\|_{L^2(D^k)} &\leq \sum_{i=1}^{d} C_1 \left\| \sqrt{J^k} \right\|_{L^2(\widehat{D})} \left| (\boldsymbol{r}_i - I_{N+1} \boldsymbol{r}_i) \right|_{H^1(\widehat{D})} \\ &\leq \sum_{i=1}^{d} \widetilde{C}_N \left\| \sqrt{J^k} \right\|_{L^2(\widehat{D})} \left| \boldsymbol{r}_i \right|_{W^{N+2,2}(\widehat{D})}, \end{aligned}$$

where we have used the Bramble-Hilbert lemma [41] on the reference element in the last step. Since it is applied on the reference element $\widehat{D}$ rather than the physical element D^k, the constant $\widetilde{C}_N$ depends on the reference element and order of approximation, but not the mesh size h. A scaling argument for quasi-uniform meshes then yields that

$$\left| \boldsymbol{r}_i \right|_{W^{N+2,2}(\widehat{D})} \leq C_2 h^{N+2} \left\| \boldsymbol{r}_i \right\|_{W^{N+2,2}(D^k)}. \tag{47}$$

The global estimate results from squaring (47), summing over all elements and using $\left\| \sqrt{J^k} \right\|_{L^2(\widehat{D})}^2 = \left| D^k \right|$. □

Remark. *It should be pointed out that this approach does not work on hexahedral elements. This is due to the fact that the natural polynomial space on hexahedral elements is the tensor product space* $Q^N\left(\widehat{D}\right)$

$$Q^N\left(\widehat{D}\right) = \left\{ \widehat{x}_1^{i_1} \ldots \widehat{x}_d^{i_d}, \quad \widehat{\boldsymbol{x}} \in \widehat{D}, \quad 0 \leq i_k \leq N \right\}.$$

For $u \in Q^{N+1}\left(\widehat{D}\right)$, $\frac{\partial u}{\partial \widehat{x}_i}$ is at most degree N in the coordinate $\widehat{x}_i$, but can remain a polynomial of degree $N+1$ in all other coordinates. Thus, interpolating from degree $N+1$ to degree N polynomials in (46) introduces aliasing errors and is no longer exact. The result of (46) is no longer the image of a curl, and thus does not satisfy the discrete GCL by construction.

We briefly outline how to compute $\widetilde{G_{ij}^k}$ in three dimensions. Let $\left\{\widehat{\boldsymbol{x}}_i^N\right\}_{i=1}^{N_p}$ denote the set of degree N interpolation points, and let $\ell_i^N(\widehat{\boldsymbol{x}})$ denote the ith degree N Lagrange basis function on the reference element. We define interpolation matrices $\boldsymbol{V}_N^{N+1}$ and $\boldsymbol{V}_{N+1}^N$ between degree N and $N+1$ polynomials such that

$$\begin{aligned} \left(\boldsymbol{V}_N^{N+1}\right)_{ij} &= \ell_j^N(\widehat{\boldsymbol{x}}_i^{N+1}), \qquad 1 \le i \le N_p, \qquad 1 \le i \le (N+1)_p \\ \left(\boldsymbol{V}_{N+1}^{N}\right)_{ij} &= \ell_j^{N+1}(\widehat{\boldsymbol{x}}_i^{N}), \qquad 1 \le i \le (N+1)_p, \qquad 1 \le i \le N_p, \end{aligned} \tag{48}$$

where $N_p, (N+1)_p$ denotes the number of interpolation points for degree N and $N+1$ polynomials, respectively. Next, let $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ denote vectors containing x_1, x_2, x_3 coordinates of degree N interpolation points on a curved physical element D^k, and let $\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{x}}_2, \widetilde{\boldsymbol{x}}_3$ denote their evaluation at degree $(N+1)$ interpolation points

$$\widetilde{\boldsymbol{x}}_1 = \boldsymbol{V}_N^{N+1}\boldsymbol{x}_1, \qquad \widetilde{\boldsymbol{x}}_2 = \boldsymbol{V}_N^{N+1}\boldsymbol{x}_2, \qquad \widetilde{\boldsymbol{x}}_3 = \boldsymbol{V}_N^{N+1}\boldsymbol{x}_3. \tag{49}$$

Let $\widetilde{\boldsymbol{D}}_i^{N+1}$ denote the nodal differentiation matrix of degree $N+1$ with respect to the ith coordinate direction. The geometric factors are computed as follows:

$$\begin{aligned}
\boldsymbol{G}_{11}^{\boldsymbol{k}} &= \boldsymbol{V}_{N+1}^N\left(\widetilde{\boldsymbol{D}}_3^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_2^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_3\right) - \widetilde{\boldsymbol{D}}_2^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_3^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_3\right)\right) \\
\boldsymbol{G}_{12}^{\boldsymbol{k}} &= \boldsymbol{V}_{N+1}^N\left(\widetilde{\boldsymbol{D}}_1^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_3^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_3\right) - \widetilde{\boldsymbol{D}}_3^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_1^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_3\right)\right) \\
\boldsymbol{G}_{13}^{\boldsymbol{k}} &= \boldsymbol{V}_{N+1}^N\left(\widetilde{\boldsymbol{D}}_2^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_1^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_3\right) - \widetilde{\boldsymbol{D}}_1^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_2^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_3\right)\right) \\
\boldsymbol{G}_{21}^{\boldsymbol{k}} &= -\boldsymbol{V}_{N+1}^N\left(\widetilde{\boldsymbol{D}}_3^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_2^{N+1}\widetilde{\boldsymbol{x}}_1\right)\circ\widetilde{\boldsymbol{x}}_3\right) - \widetilde{\boldsymbol{D}}_2^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_3^{N+1}\widetilde{\boldsymbol{x}}_1\right)\circ\widetilde{\boldsymbol{x}}_3\right)\right) \\
\boldsymbol{G}_{22}^{\boldsymbol{k}} &= -\boldsymbol{V}_{N+1}^N\left(\widetilde{\boldsymbol{D}}_1^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_3^{N+1}\widetilde{\boldsymbol{x}}_1\right)\circ\widetilde{\boldsymbol{x}}_3\right) - \widetilde{\boldsymbol{D}}_3^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_1^{N+1}\widetilde{\boldsymbol{x}}_1\right)\circ\widetilde{\boldsymbol{x}}_3\right)\right) \\
\boldsymbol{G}_{23}^{\boldsymbol{k}} &= -\boldsymbol{V}_{N+1}^N\left(\widetilde{\boldsymbol{D}}_2^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_1^{N+1}\widetilde{\boldsymbol{x}}_1\right)\circ\widetilde{\boldsymbol{x}}_3\right) - \widetilde{\boldsymbol{D}}_1^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_2^{N+1}\widetilde{\boldsymbol{x}}_1\right)\circ\widetilde{\boldsymbol{x}}_3\right)\right) \\
\boldsymbol{G}_{31}^{\boldsymbol{k}} &= -\boldsymbol{V}_{N+1}^N\left(\widetilde{\boldsymbol{D}}_3^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_2^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_1\right) - \widetilde{\boldsymbol{D}}_2^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_3^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_1\right)\right) \\
\boldsymbol{G}_{32}^{\boldsymbol{k}} &= -\boldsymbol{V}_{N+1}^N\left(\widetilde{\boldsymbol{D}}_1^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_3^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_1\right) - \widetilde{\boldsymbol{D}}_3^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_1^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_1\right)\right) \\
\boldsymbol{G}_{33}^{\boldsymbol{k}} &= -\boldsymbol{V}_{N+1}^N\left(\widetilde{\boldsymbol{D}}_2^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_1^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_1\right) - \widetilde{\boldsymbol{D}}_1^{N+1}\left(\left(\widetilde{\boldsymbol{D}}_2^{N+1}\widetilde{\boldsymbol{x}}_2\right)\circ\widetilde{\boldsymbol{x}}_1\right)\right).
\end{aligned} \tag{50}$$

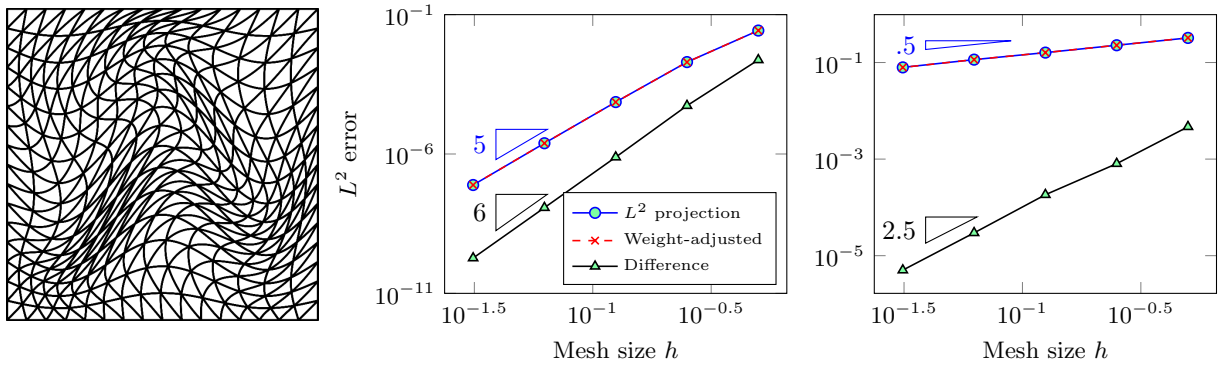
Remark. *We note that the discrete GCL (42) can also be enforced directly through a local constrained minimization problem [53, 17], which yields a solution in terms of a pseudo-inverse. However, we have not found a straightforward way to simultaneously enforce both the discrete GCL condition (42) and Assumption 2 using this approach.*

6. Numerical experiments

In this section, we present numerical results which verify the theoretical results in this work. We first verify that the weight-adjusted L^2 projection $\widetilde{\Pi}_N^k$ and the GCL-satisfying geometric factors G_{ij}^k obey the error estimates in Theorem 4 and Theorem 5. Next, we verify the semi-discrete entropy conservation, primary conservation, and accuracy of the proposed high order accurate methods for the compressible Euler equations on curved meshes in two and three dimensions. For all curved meshes, we utilize the low storage weight-adjusted formulation (31).

In choosing the timestep dt, we follow [24] and set

$$dt = \min_k \mathrm{CFL}\frac{\|J\|_{L^\infty(D^k)}}{C_N\|J^f\|_{L^\infty(\partial D^k)}}, \tag{51}$$



(A) Warped curvilinear mesh

(B) Smooth function

(C) Discontinuous function

FIGURE 1. L^2 errors in approximating both smooth and discontinuous functions using L^2 and weight-adjusted projections on a curved mesh. The approximation order is $N = 4$, and a degree $2N$ quadrature rule is used to compute integrals over the reference triangle.

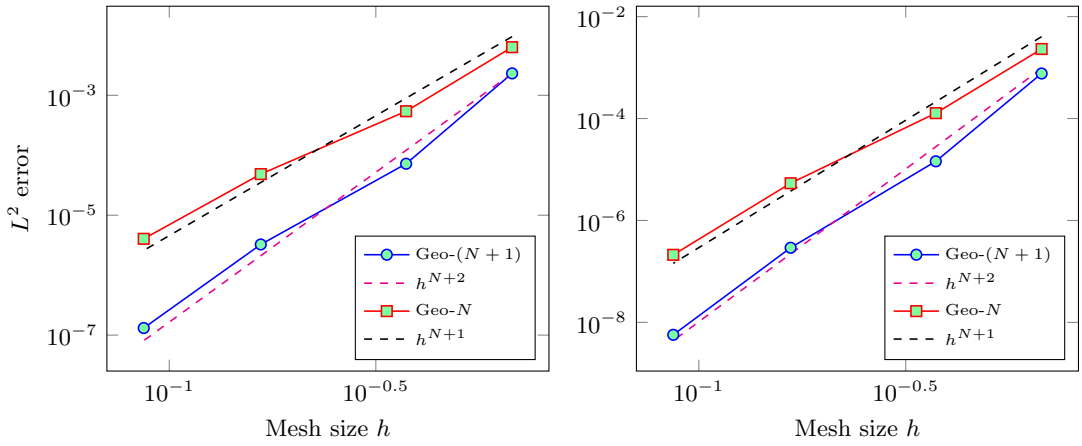
where C_N is the $O(N^2)$ order-dependent constant in the surface trace inequality [24] and CFL is a user-chosen constant. All numerical experiments in this work utilize the five-stage fourth order low storage Runge–Kutta (LSRK-45) time-stepper [54].

6.1. **Accuracy of weight-adjusted projection and geometric terms.** In this section, we verify Theorems 4 and 5 concerning the accuracy of the weight-adjusted projection and modified construction of geometric terms satisfying the discrete geometric conservation law. Figure 1 shows L^2 errors for both the standard L^2 projection (20) and the weight-adjusted projection (28) for a series of warped meshes of degree $N = 4$. Errors are estimated using degree $2N+1$ quadratures for triangles and tetrahedra [55]. We compute L^2 errors for both smooth and discontinuous functions

$$f(\boldsymbol{x}) = e^{x_1+x_2}\sin(\pi x_1)\sin(\pi x_2), \qquad g(\boldsymbol{x}) = f(\boldsymbol{x}) + H(x_1 + x_2 - \sin(\pi x_1)),$$

where H is the Heaviside function. For the smooth function $f(\boldsymbol{x})$, we observe that the difference between the L^2 and weight-adjusted projections indeed converges at a rate of $O\left(h^{N+2}\right)$ as predicted by Theorem 1, such that the L^2 errors for each projection appear identical. The L^2 errors for the L^2 and weight-adjusted projections of the discontinuous function $g(\boldsymbol{x})$ are also virtually identical. However, the difference beteween the L^2 and weight-adjusted projection converges faster than estimated by Theorem 1, with the L^2 error converging as $O(h^{1/2})$ and the difference converging as $O(h^{2+1/2})$.

We next compare the approximation of the geometric factors on a curved three-dimensional mesh. We generate a sequence of quasi-uniform unstructured tetrahedral meshes using GMSH [56] and construct a curvilinear mesh from the distorted coordinates $\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \frac{1}{8}\cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right)\cos\left(\frac{\pi}{2}x_3\right)$. We then compute the L^2 error in approximating geometric terms for each element D^k by computing $G^k_{ij} - \widetilde{G^k_{ij}}$ at quadrature points. We estimate the mesh size as $h = \max_k \left\| J^k/J^k_f \right\|_{L^\infty}$, since $J^k = O(h^d)$ and $J^k_f = O(h^{d-1})$ in d dimensions [24]. Figure 2 shows errors for an $N = 3$ and $N = 4$ mesh. We refer to the construction of approximate geometric terms $\widetilde{G^k_{ij}}$ introduced in [2, 57] as "Geo-N", since the interpolation is performed using a degree N interpolation operator. We refer to the construction of $\widetilde{G^k_{ij}}$ in (46) and Theorem 5 as "Geo-$(N+1)$", since the main interpolation step is performed on degree $(N+1)$ polynomials instead. It can be observed that the Geo-N scheme converges at a rate of $O(h^{N+1})$, while the Geo-(N+1) scheme converges at a rate of $O(h^{N+2})$. We note that the error in both the Geo-N and Geo-$(N+1)$ approximations of $\widetilde{G^k_{ij}}$ converge at the same rate or faster than the best approximation error.



(A) $N = 3$ (B) $N = 4$

FIGURE 2. L^2 errors in the approximation of metric terms G_{ij}^k with metric terms $\widetilde{G_{ij}^k}$ satisfying the discrete GCL condition (42) and Assumption 2.

6.2. **Two-dimensional compressible Euler equations.** The compressible Euler equations in two dimensions are given as follows:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x_1} + \frac{\partial (\rho v)}{\partial x_2} &= 0, \\
\frac{\partial \rho u}{\partial t} + \frac{\partial \left(\rho u^2 + p\right)}{\partial x_1} + \frac{\partial (\rho u v)}{\partial x_2} &= 0, \\
\frac{\partial \rho v}{\partial t} + \frac{\partial (\rho u v)}{\partial x_1} + \frac{\partial \left(\rho v^2 + p\right)}{\partial x_2} &= 0, \\
\frac{\partial E}{\partial t} + \frac{\partial (u(E+p))}{\partial x_1} + \frac{\partial (v(E+p))}{\partial x_2} &= 0.
\end{aligned} \tag{52}$$

In two dimensions, the pressure is $p = (\gamma - 1)\left(E - \frac{1}{2}\rho(u^2 + v^2)\right)$, and the specific internal energy is $\rho e = E - \frac{1}{2}\rho(u^2 + v^2)$.

The choice of convex entropy for the Euler equations is non-unique [58]. However, a unique entropy can be chosen by restricting to choices of entropy variables which symmetrize the viscous heat conduction term in the compressible Navier-Stokes equations [22]. This leads to $U(\boldsymbol{u})$ of the form

$$U(\boldsymbol{u}) = -\frac{\rho s}{\gamma - 1}, \tag{53}$$

where $s = \log\left(\frac{p}{\rho^\gamma}\right)$ is the physical specific entropy. The entropy variables in two dimensions are

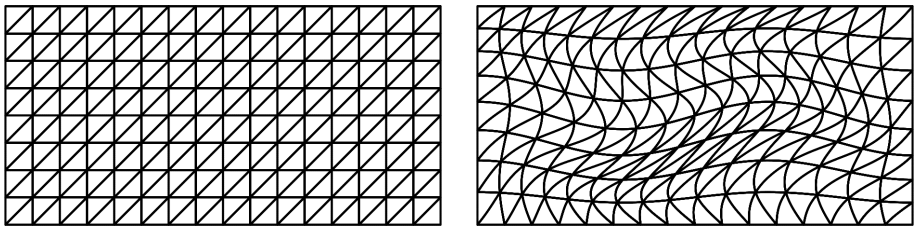
$$v_1 = \frac{\rho e(\gamma + 1 - s) - E}{\rho e}, \qquad v_2 = \frac{\rho u}{\rho e}, \qquad v_3 = \frac{\rho v}{\rho e}, \qquad v_4 = -\frac{\rho}{\rho e}. \tag{54}$$

The conservation variables in terms of the entropy variables are given by

$$\rho = -(\rho e)v_4, \qquad \rho u = (\rho e)v_2, \qquad \rho v = (\rho e)v_3, \qquad E = (\rho e)\left(1 - \frac{v_2^2 + v_3^2}{2v_4}\right), \tag{55}$$

where ρe and s in terms of the entropy variables are

$$\rho e = \left(\frac{(\gamma - 1)}{(-v_4)^\gamma}\right)^{1/(\gamma-1)} e^{\frac{-s}{\gamma-1}}, \qquad s = \gamma - v_1 + \frac{v_2^2 + v_3^2}{2v_4}. \tag{56}$$



(A) Uniform mesh (B) Warped mesh

FIGURE 3. 2D curved meshes used for testing entropy conservation and primary conservation.

The entropy conservative numerical fluxes for the two-dimensional compressible Euler equations are given by Chandrashekar [37]

$$
\begin{aligned}
f^1_{1,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= \{\{\rho\}\}^{\log}\{\{u\}\}, & f^1_{2,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= \{\{\rho\}\}^{\log}\{\{v\}\},\\
f^2_{1,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= f^1_{1,S}\{\{u\}\} + p_{\text{avg}}, & f^2_{2,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= f^1_{2,S}\{\{u\}\},\\
f^3_{1,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= f^2_{2,S}, & f^3_{2,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= f^1_{2,S}\{\{v\}\} + p_{\text{avg}},\\
f^4_{1,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= (E_{\text{avg}} + p_{\text{avg}})\{\{u\}\}, & f^4_{2,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= (E_{\text{avg}} + p_{\text{avg}})\{\{v\}\},
\end{aligned}
\tag{57}
$$

where we have defined the auxiliary quantities

$$
\begin{aligned}
p_{\text{avg}} &= \frac{\{\{\rho\}\}}{2\{\{\beta\}\}}, \qquad E_{\text{avg}} = \frac{\{\{\rho\}\}^{\log}}{2\{\{\beta\}\}^{\log}(\gamma-1)} + \frac{\|\boldsymbol{u}\|^2_{\text{avg}}}{2},\\
\|\boldsymbol{u}\|^2_{\text{avg}} &= 2(\{\{u\}\}^2 + \{\{v\}\}^2) - \left(\{\{u^2\}\} + \{\{v^2\}\}\right).
\end{aligned}
\tag{58}
$$

6.2.1. *Entropy conservation.* We begin by testing the propagation of a shock on a two-dimensional curved mesh using a discontinuous profile on the domain $\Omega = [0,20]\times[-5,5]$. We set the initial velocities to be zero, and initialize the density and pressure as a discontinuous square pulse as in [1]

$$
\rho(\boldsymbol{x},t) = \begin{cases} 3 & |x_1| < 1/2 \text{ and } |x_2| < 1/2 \\ 2 & \text{otherwise,} \end{cases} \qquad u(\boldsymbol{x},t) = v(\boldsymbol{x},t) = 0, \qquad p(\boldsymbol{x},t) = \rho^\gamma.
\tag{59}
$$

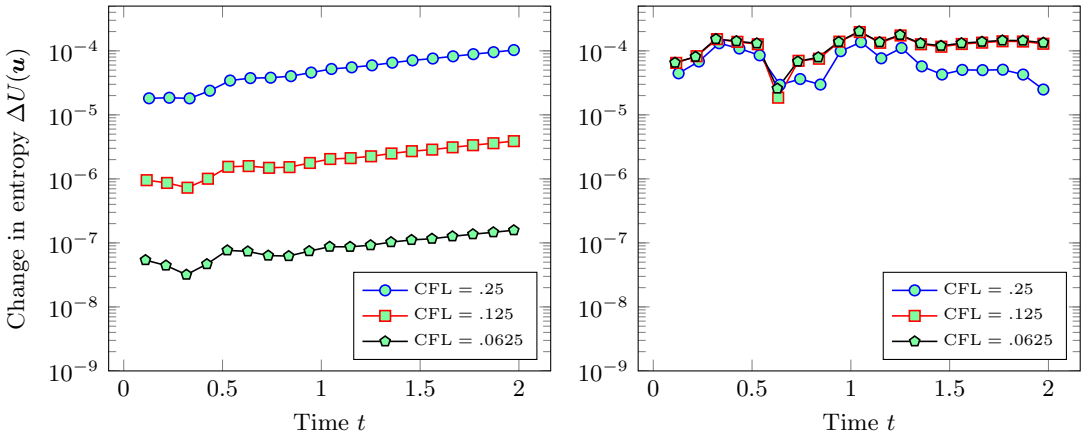
To test the scheme (31), we utilize an entropy conservative flux and run it on a uniform triangular mesh with a curvilinear warping shown in Figure 3. Theorem 2 ensures that, under an entropy conservative flux, (31) is semi-discretely entropy conservative. This does not hold at the fully discrete level; however, it is possible to verify that (31) is entropy stable using other approaches.

First, can examine the entropy RHS, which we define as the right hand side of (31) tested with $\boldsymbol{v}_h$

$$
\text{entropy RHS} = -\sum_{j=1}^{d}\left(\widetilde{\boldsymbol{v}}^T\left(2\boldsymbol{Q}^j_k \circ \boldsymbol{F}_{j,S}\right)\boldsymbol{1} + \widetilde{\boldsymbol{v}}_f^T\boldsymbol{W}_f^k \text{diag}\left(\boldsymbol{n}_j\right)\left(\boldsymbol{f}_j^* - \boldsymbol{f}_j(\widetilde{\boldsymbol{u}}_f)\right)\right).
\tag{60}
$$

For positive density and pressure, (60) should be zero to machine precision. We can also track the change in entropy $\Delta U = |U(\boldsymbol{u}(\boldsymbol{x},t)) - U(\boldsymbol{u}(\boldsymbol{x},0))|$, which should converge to 0 as the timestep dt approaches zero. Furthermore, the rate of convergence should match the order of the time-stepper used [59, 1].

Figure 4 shows the evolution of entropy $U(\boldsymbol{u})$ over time $[0,T]$, where the final time $T = 2$. We compare when the entropy-projected conservative variables $\widetilde{\boldsymbol{u}}$ are computed using the standard L^2 projection on the reference element, and when $\widetilde{\boldsymbol{u}}$ are computed using the weight-adjusted projection. For the standard L^2 projection, the change in entropy does not decrease as dt decreases (for a fixed mesh and order N). When $\widetilde{\boldsymbol{u}}$ is defined using the weight-adjusted projection, the entropy decreases as dt decreases. Moreover, the rate of convergence is approximately $O(dt^{4.675})$, which is slightly higher than the expected rate of $O(dt^4)$ when using LSRK-45. Regardless of whether the standard and weight-adjusted projection was used, the entropy RHS (60) is $O(10^{-14})$, indicating that the proposed scheme is implemented correctly.



(A) With weight-adjusted projection

(B) Without weight-adjusted projection

FIGURE 4. Change in entropy under an entropy conservative formulation with $N = 4$. In both cases, the magnitude of the entropy RHS (60) is $O\left(10^{-14}\right)$.

These results suggest that the weight-adjusted projection is necessary to produce an entropy conservative scheme on curvilinear meshes. However, entropy conservative schemes result in spurious oscillations and lower convergence rates [1]. In practice, dissipative interface terms are added to produce entropy stable schemes. In the presence of interface dissipation, the evolution of entropy over time, L^2 errors for smooth solutions, and the qualitative behavior of the solution are very similar with or without the weight-adjusted projection. This may reflect the fact that aliasing-driven instabilities arise from the spatial discretization, and the entropy RHS (60) is machine precision zero with or without the weight-adjusted projection. In contrast, the presence of the weight-adjusted projection affects only the time-derivative on the left-hand side, and may not play as large a role in suppressing aliasing instabilities.

6.2.2. *Local and global conservation.* Next, we check that primary conservation is maintained numerically on curved meshes. We follow [15] and examine the semi-discrete evolution of the average of $\boldsymbol{u}$ over the domain Ω

$$\frac{\partial}{\partial t}\int_\Omega \boldsymbol{u}\,\mathrm{d}\boldsymbol{x} = \sum_k \int_{D^k} \frac{\partial \boldsymbol{u}}{\partial t}\,\mathrm{d}\boldsymbol{x} = \sum_k \int_{\widehat{D}} \frac{\partial \boldsymbol{u}}{\partial t} J^k\,\mathrm{d}\widehat{\boldsymbol{x}} = \sum_k \mathbf{1}^T \boldsymbol{W}\,\mathrm{diag}\left(\boldsymbol{J}^k\right)\boldsymbol{V}_q \frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{dt}}. \tag{61}$$

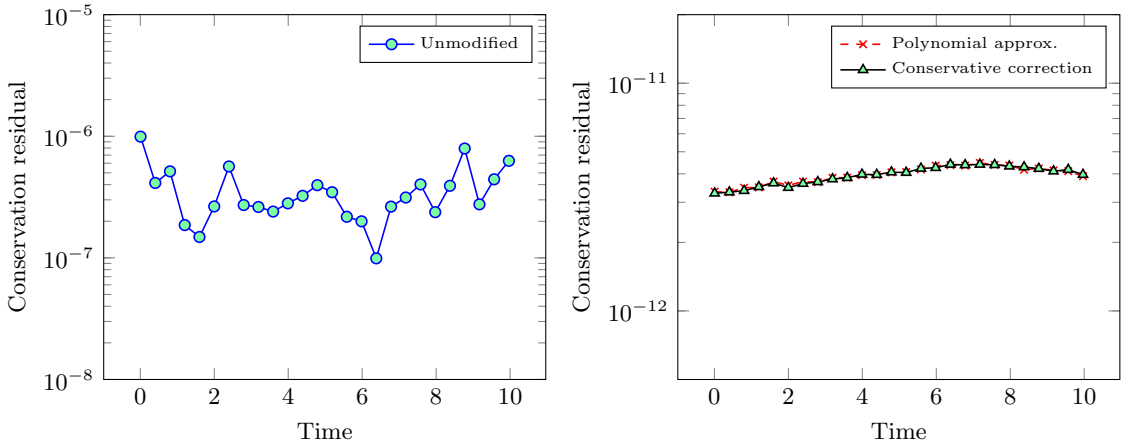
The quantity (61) is computed using quadrature by multiplying the semi-discrete system (31) by $\mathbf{1}^T\boldsymbol{W}\,\mathrm{diag}\left(\boldsymbol{J}^k\right)\boldsymbol{V}_q$

$$\begin{aligned}&\sum_k \mathbf{1}^T \boldsymbol{W}\,\mathrm{diag}\left(\boldsymbol{J}^k\right)\boldsymbol{V}_q \frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{dt}} = \\ &\underbrace{\mathbf{1}^T \boldsymbol{W}\,\mathrm{diag}\left(\boldsymbol{J}^k\right)\boldsymbol{V}_q \left(\begin{bmatrix} \widetilde{\boldsymbol{P}}_q^k & \widetilde{\boldsymbol{L}}_q^k \end{bmatrix} \sum_{j=1}^d \left(2\boldsymbol{D}_k^j \circ \boldsymbol{F}_{j,S}\right)\mathbf{1} + \sum_{j=1}^d \widetilde{\boldsymbol{L}}_q^k \mathrm{diag}\left(\boldsymbol{n}_j\right)\left(\boldsymbol{f}_j^* - \boldsymbol{f}_j(\widetilde{\boldsymbol{u}}_f)\right)\right)}_{\text{conservation residual}}.\end{aligned} \tag{62}$$

We consider three different cases:

(1) **Unmodified:** the weight-adjusted DG scheme (31) is used without any special modifications,
(2) **Polynomial approximation:** the weight-adjusted DG scheme (31) utilizes the L^2 projection of J^k onto $P^N(\widehat{D})$ in the weight-adjusted mass matrix (such that Lemma 2 holds),
(3) **Conservative correction:** the conservation correction (41) is applied to the right hand side of the scheme (31).

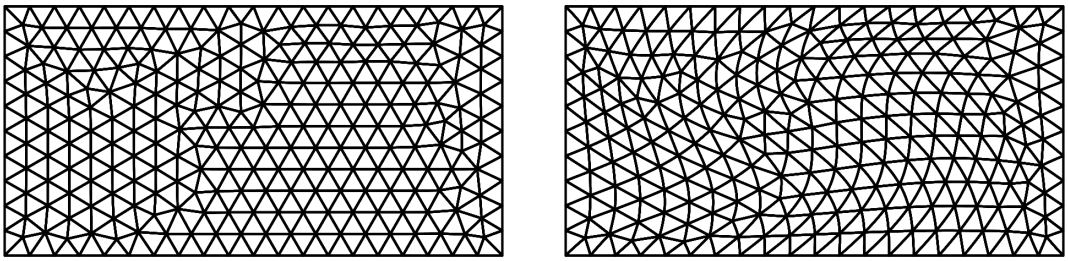
All results use a Lax-Friedrichs flux, a CFL of $1/2$, $N = 4$, and the curved mesh warping shown in Figure 3. We use volume quadrature rules from [55] which are exact for degree $2N+1$ polynomials, and use $(N+1)$-node 1D Gauss–Legendre quadrature rules on the faces.



(A) Conservation residual over time (unmodified)

(B) Conservation residual over time (modified)

FIGURE 5. Conservation residuals for a discontinuous initial condition using the unmodified WADG formulation (31) and two different techniques for restoring local conservation. Both experiments use a Lax-Friedrichs flux, a CFL of $1/2$, $N = 4$, and the curved mesh warping shown in Figure 3.



(A) Affine mesh

(B) Curved mesh

FIGURE 6. Example of affine and warped meshes ($h = 1$ shown) used in convergence studies.

Figure 5a shows computed conservation residuals from (62) on the curved mesh in Figure 3 for the discontinuous pulse initial condition (59). The conservation residual oscillates around $O(10^{-6})$ for the unmodified scheme (31). Modifying the scheme using either the "Polynomial approximation" or "Conservative correction" approaches reduces the conservation error to between $O(10^{-11})$ and $O(10^{-12})$. We have also studied the accuracy of each case in by comparing L^2 errors on a curved mesh for a smooth vortex solution at time $T = 5$. We observe that in all cases, the L^2 errors are virtually identical. For $N = 4$ on an 8×8 mesh, the L^2 errors for each case differ only in the 5th digit, while for the 16×16 and 32×32 meshes, they differ only in the 8th digit. This confirms that both approaches outlined in Section 4.3.1 restore primary conservation with no perceivable effect on accuracy.

6.2.3. *Accuracy and convergence.* Finally, we test the accuracy of the proposed schemes for smooth solutions on curved meshes in two dimensions. We use the isentropic vortex problem [60], which has an analytical

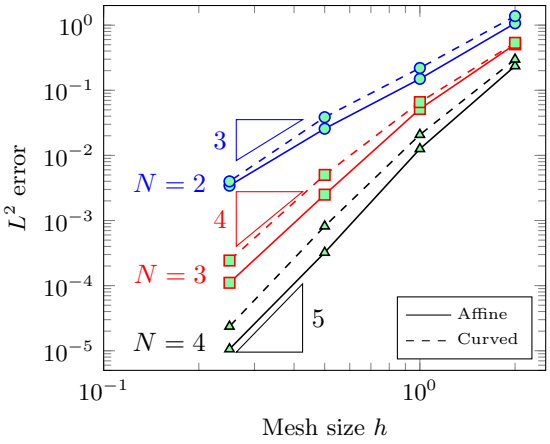


FIGURE 7. Convergence of L^2 errors for the 2D isentropic vortex problem on affine and curved meshes.

solution

$$\rho(\boldsymbol{x},t) = \left(1 - \frac{\frac{1}{2}(\gamma-1)(\beta e^{1-r(\boldsymbol{x},t)^2})^2}{8\gamma\pi^2}\right)^{\frac{1}{\gamma-1}}, \qquad p = \rho^\gamma, \tag{63}$$

$$u(\boldsymbol{x},t) = 1 - \frac{\beta}{2\pi} e^{1-r(\boldsymbol{x},t)^2}(x_2 - c_2), \qquad v(\boldsymbol{x},t) = \frac{\beta}{2\pi} e^{1-r(\boldsymbol{x},t)^2}(x_2 - c_2),$$

where u, v are the x_1 and x_2 velocity and $r(\boldsymbol{x},t) = \sqrt{(x_1 - c_1 - t)^2 + (x_2 - c_2)^2}$. Here, we take $c_1 = 5, c_2 = 0$ and $\beta = 5$. The solution is computed on a periodic rectangular domain $[0, 20] \times [-5, 5]$ at final time $T = 5$. Quasi-uniform triangular meshes are generated using GMSH [56], and a curvilinear warping is applied to the mesh to test the effect of non-affine mappings. This warping is shown in Figure 6 and is defined by mapping nodal positions on each triangle to warped nodal positions $(\widetilde{x_1}, \widetilde{x_2})$ via

$$\widetilde{x_1} = x_1 + \sin(\pi x_1/20)\sin(2\pi(x_2+5)/10)$$
$$\widetilde{x_2} = x_2 - \frac{1}{2}\sin(2\pi x_1/20)\sin(\pi(x_2+5)/10).$$

To ensure primary conservation, we use the "Polynomial Approximation" strategy described in Section 4.3.1 and compute the inverse of the weight-adjusted mass matrix using the degree N L^2 projection $\widehat{\Pi}_N J^k$ instead of J^k. The computed L^2 errors are shown in Figure 7. We observe optimal $O(h^{N+1})$ rates of convergence for $N = 2, N = 4$. For degree $N = 3$, the rate of convergence is slightly higher than $O(h^4)$ and may indicate that the mesh is not yet sufficiently fine for the L^2 error to show the asymptotic convergence rate.

6.3. **Three dimensional compressible Euler equations.** In three dimensions, the compressible Euler equations are given by

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{pmatrix} + \frac{\partial}{\partial x_1}\begin{pmatrix}\rho u\\ \rho u^2 + p\\ \rho uv\\ \rho uw\\ u(E+p)\end{pmatrix} + \frac{\partial}{\partial x_2}\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2 + p\\ \rho vw\\ v(E+p)\end{pmatrix} + \frac{\partial}{\partial x_3}\begin{pmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2 + p\\ w(E+p)\end{pmatrix} = 0, \tag{64}$$

where the pressure p and specific internal energy ρe are defined

$$p = (\gamma-1)\left(E - \frac{1}{2}\rho(u^2+v^2+w^2)\right), \qquad \rho e = E - \frac{1}{2}\rho(u^2+v^2+w^2). \tag{65}$$

The formula for the entropy $U(\boldsymbol{u})$ in three dimensions is the same as the two-dimensional formula (53). The entropy variables in three dimensions are

$$v_1 = \frac{\rho e(\gamma + 1 - s) - E}{\rho e}, \qquad v_2 = \frac{\rho u}{\rho e}, \qquad v_3 = \frac{\rho v}{\rho e}, \qquad v_4 = \frac{\rho w}{\rho e}, \qquad v_5 = -\frac{\rho}{\rho e}. \tag{66}$$

The conservation variables in terms of the entropy variables are given by

$$\rho = -(\rho e)v_5, \qquad \rho u = (\rho e)v_2, \qquad \rho v = (\rho e)v_3, \qquad \rho w = (\rho e)v_4, \qquad E = (\rho e)\left(1 - \frac{v_2^2 + v_3^2 + v_4^2}{2v_5}\right), \tag{67}$$

where ρe and s in terms of the entropy variables are

$$\rho e = \left(\frac{(\gamma - 1)}{(-v_5)^\gamma}\right)^{1/(\gamma-1)} e^{\frac{-s}{\gamma-1}}, \qquad s = \gamma - v_1 + \frac{v_2^2 + v_3^2 + v_4^2}{2v_5}. \tag{68}$$

A set of entropy conservative numerical fluxes for the three-dimensional compressible Euler equations can be written as

$$\boldsymbol{f}_{1,S} = \begin{pmatrix} \{\{\rho\}\}^{\log} \{\{u\}\} \\ \{\{\rho\}\}^{\log} \{\{u\}\}^2 + p_{\text{avg}} \\ \{\{\rho\}\}^{\log} \{\{u\}\} \{\{v\}\} \\ \{\{\rho\}\}^{\log} \{\{u\}\} \{\{w\}\} \\ (E_{\text{avg}} + p_{\text{avg}}) \{\{u\}\} \end{pmatrix}, \qquad \boldsymbol{f}_{2,S} = \begin{pmatrix} \{\{\rho\}\}^{\log} \{\{v\}\} \\ \{\{\rho\}\}^{\log} \{\{u\}\} \{\{v\}\} \\ \{\{\rho\}\}^{\log} \{\{v\}\}^2 + p_{\text{avg}} \\ \{\{\rho\}\}^{\log} \{\{v\}\} \{\{w\}\} \\ (E_{\text{avg}} + p_{\text{avg}}) \{\{v\}\} \end{pmatrix}, \tag{69}$$

$$\boldsymbol{f}_{3,S} = \begin{pmatrix} \{\{\rho\}\}^{\log} \{\{w\}\} \\ \{\{\rho\}\}^{\log} \{\{u\}\} \{\{w\}\} \\ \{\{\rho\}\}^{\log} \{\{v\}\} \{\{w\}\} \\ \{\{\rho\}\}^{\log} \{\{w\}\}^2 + p_{\text{avg}} \\ (E_{\text{avg}} + p_{\text{avg}}) \{\{w\}\} \end{pmatrix}.$$

where we have defined the auxiliary quantities

$$p_{\text{avg}} = \frac{\{\{\rho\}\}}{2\{\{\beta\}\}}, \qquad E_{\text{avg}} = \frac{\{\{\rho\}\}^{\log}}{2(\gamma - 1)\{\{\beta\}\}^{\log}} + \frac{1}{2}\{\{\rho\}\}^{\log} \|\boldsymbol{u}\|_{\text{avg}}^2 \tag{70}$$

$$\|\boldsymbol{u}\|_{\text{avg}}^2 = 2(\{\{u\}\}^2 + \{\{v\}\}^2 + \{\{w\}\}^2) - \left(\{\{u^2\}\} + \{\{v^2\}\} + \{\{w^2\}\}\right).$$

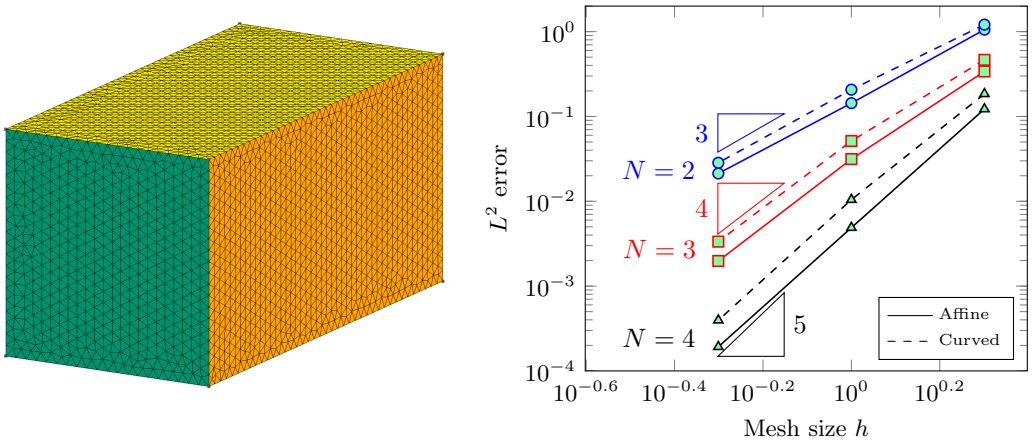
6.3.1. *Accuracy and convergence.* As before, we test the accuracy of the proposed scheme using an isentropic vortex solution adapted to three dimensions. We take the solution to be the extruded 2D vortex propagating in the x_2 direction, whose analytic expression is derived from [61]

$$\begin{aligned} \rho(\boldsymbol{x}, t) &= \left(1 - \frac{(\gamma - 1)}{2}\Pi^2\right)^{\frac{1}{\gamma-1}} \\ \boldsymbol{u}(\boldsymbol{x}, t) &= \Pi \boldsymbol{r}, \\ E(\boldsymbol{x}, t) &= \frac{p_0}{\gamma - 1}\left(1 - \frac{\gamma - 1}{2}\Pi^2\right)^{\frac{\gamma}{\gamma-1}} + \frac{\rho}{2}|\boldsymbol{u}|. \end{aligned}$$

where $\boldsymbol{u} = (u, v, w)^T$ is the velocity vector and

$$\Pi = \Pi_{\max} e^{\frac{1 - \boldsymbol{r}^T \boldsymbol{r}}{2}}, \qquad \boldsymbol{r} = \begin{pmatrix} -(x_2 - c_2 - t) \\ x_1 - c_1 \\ 0 \end{pmatrix}.$$

In this problem, we take $c_1 = c_2 = 5$, $p_0 = 1/\gamma$, and $\Pi_{\max} = 0.4$. The problem is solved on the domain $[0, 10] \times [0, 20] \times [0, 10]$. We use GMSH to construct three unstructured meshes consisting of 1354, 9543,



(A) Mesh for $h = 1/2$

(B) L^2 errors

FIGURE 8. Convergence of L^2 errors for the 3D isentropic vortex problem on unstructured affine and curved meshes, with optimal $O(h^{N+1})$ rates of convergence shown for reference.

and 72923 affine tetrahedra, corresponding to $h = 2$, $h = 1$, and $h = 1/2$ (shown in Figure 8a). We apply a curvilinear warping by mapping nodal positions each tetrahedron to warped nodal positions $(\widetilde{x_1}, \widetilde{x_2}, \widetilde{x_3})$ via

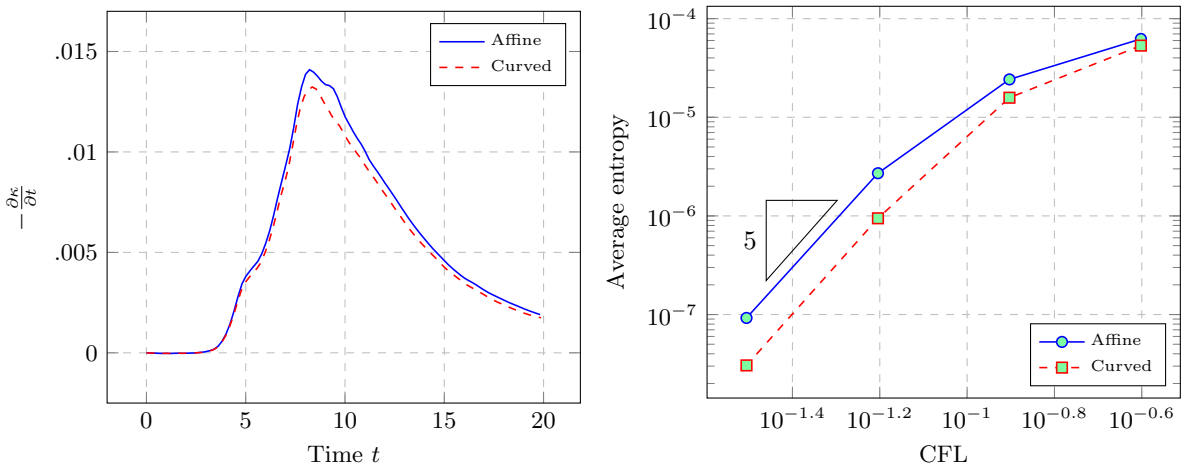
$$\begin{aligned}
\widetilde{x_1} &= x_1 + \frac{1}{2}\sin\left(\pi\frac{x_1}{10}\right)\sin\left(2\pi\frac{x_2}{20}\right)\sin\left(\pi\frac{x_3}{10}\right), \\
\widetilde{x_2} &= x_2 - \sin\left(2\pi\frac{x_1}{10}\right)\sin\left(\pi\frac{x_2}{10}\right)\sin\left(2\pi\frac{x_3}{10}\right), \\
\widetilde{x_3} &= x_3 + \frac{1}{2}\sin\left(\pi\frac{x_1}{10}\right)\sin\left(2\pi\frac{x_2}{10}\right)\sin\left(\pi\frac{x_3}{10}\right).
\end{aligned}$$

The geometric terms are approximated using (46) to ensure the satisfaction of the discrete GCL. Figure 8 shows L^2 errors at final time $T = 5$ for $N = 2, 3, 4$ on both affine and curved meshes. In both cases, we observe optimal $O(h^{N+1})$ asymptotic rates of convergence for $N = 2, 3$, while for $N = 4$ we observe a rate which is slightly higher than $O(h^{N+1/2})$ but not quite $O(h^{N+1})$. This may be due to the fact that the time-stepper is 4th order while the spatial discretization is 5th order.

We also compared approximations of geometric factors using degree N polynomials via (46) and degree $N-1$ polynomials via (45). For the isentropic vortex problem on warped meshes, the approximation of the geometric factors does not impact accuracy significantly. Utilizing (45) and approximating geometric factors with degree $N-1$ polynomials only changed the error in the 5th significant digit, and resulted in the same asymptotic rates of convergence on curved 3D meshes. Future work will explore the effect of a more accurate geometric approximation on curved boundaries where a solid wall boundary condition is applied, where the approximation of geometry has been shown to have a more significant effect [62].

6.3.2. *Inviscid Taylor–Green vortex.* Our last numerical experiment investigates the behavior of entropy stable DG schemes for the inviscid Taylor–Green vortex [63, 35, 17]. The domain is the periodic box $[-\pi, \pi]^3$, and the initial conditions are given as

$$\begin{aligned}
\rho &= 1 \\
u &= \sin(x_1)\cos(x_2)\cos(x_3), \\
v &= -\cos(x_1)\sin(x_2)\cos(x_3), \\
w &= 0, \\
p &= \frac{100}{\gamma} + \frac{1}{16}\left(\cos(2x_1) + \cos(2x_2)\right)\left(2 + \cos(2x_3)\right).
\end{aligned}$$



(A) KE dissipation rate for $N = 3$, $h = \pi/8$

(B) Entropy conservative convergence of $\int_\Omega U(\boldsymbol{u})$

FIGURE 9. Evolution of the kinetic energy dissipation rate over time on affine and curvilinear meshes, as well as dependence of average entropy over the domain $\int_\Omega U(\boldsymbol{u})$ at time $T = 20$ for an entropy conservative formulation.

The Taylor–Green vortex is used to study the transition and decay of turbulence [64]. In the absence of viscosity, the Taylor–Green vortex develops smaller and smaller scales, implying that for sufficiently large times, the solution contains under-resolved features. We study the evolution of the kinetic energy $\kappa(t)$

$$\kappa(t) = \frac{1}{|\Omega|} \int_\Omega \rho \boldsymbol{u} \cdot \boldsymbol{u} \, \mathrm{d}\boldsymbol{x},$$

as well as the kinetic energy dissipation rate $-\frac{\partial \kappa}{\partial t}$, which we approximate by differencing $\kappa(t)$. Figure 9 shows the evolution of $\kappa(t)$ over time for both affine and curvilinear meshes with $h = \pi/8$ and $N = 3$. We also use a curved coarse mesh with element size $h = \pi$ test the convergence of the average entropy for a non-dissipative entropy conservative formulation as the timestep decreases. In both cases, the mesh is defined by constructing nodal positions $\widetilde{\boldsymbol{u}}$ from warpings of affine nodal positions $\boldsymbol{x}$ as follows

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} + .125 \sin(x_1) \sin(x_2) \sin(x_3).$$

The kinetic energy is plotted at 100 equally spaced times between $[0, 20]$. The simulation is stable and does not blow up despite the lack of filtering, limiting, or artificial viscosity. The results are qualitatively similar to those reported in the literature, with the kinetic energy dissipation rate increasing around $t = 4$ and peaking with a value of roughly 0.014 before $t = 9$ [64, 35]. We also observe that, for an entropy conservative formulation, the average entropy appears to converge at a rate of $O(dt^5)$ for a 4th order time-stepping method. The same phenomena was also observed in [1, 17].

7. Conclusions

This work describes how to extend entropy conservative and entropy stable DG "modal" discretizations to curvilinear meshes using both weighted and weight-adjusted mass matrices. Assuming that the geometric terms satisfy a discrete geometric conservation law, the presented schemes allow for the use of over-integration while satisfying a semi-discrete entropy equality or inequality on curved meshes. Numerical results show the presented schemes achieve optimal rates of convergence for smooth solutions on both two and three dimensional affine and curvilinear meshes while remaining robust in the presence of under-resolved solutions such as shocks and turbulence.

Several outstanding computational questions remain to be answered. First, the use of weight-adjusted mass matrices is motivated by the low storage requirements and their efficient application on GPUs. However, to guarantee discrete entropy stability, the computation of a weight-adjusted projection is necessary, which

adds additional computational cost compared to the affine case. Furthermore, numerical results suggest that, while computing the entropy-projected conservative variables $\widetilde{\boldsymbol{u}}$ using the weight-adjusted projection is necessary to ensure a semi-discrete conservation of entropy, the difference between using a weight-adjusted projection and regular projection may be negligible in practice. The necessity of the weight-adjusted projection in computing $\widetilde{\boldsymbol{u}}$ should be examined further. Secondly, on triangular and tetrahedral meshes, the proposed schemes involve more computational work than under-integrated collocation-style SBP schemes [19, 16, 17]. A careful computational comparison of the presented schemes with existing methods should be done to weigh the benefits of improved accuracy with additional computational costs.

Finally, we note that while this work has focused on triangular and tetrahedral elements, the approaches outlined here can be extended to more arbitrary pairings of (possibly non-polynomial) approximation spaces and quadratures. For example, similar techniques can be used to construct entropy stable B-spline or Galerkin difference discretizations on curved meshes [65, 39].

8. Acknowledgements

The authors thank Mark H. Carpenter and David C. Del Rey Fernandez for informative discussions. Jesse Chan is supported by NSF DMS-1719818 and DMS-1712639.

References

[1] Jesse Chan. On discretely entropy conservative and entropy stable discontinuous Galerkin methods. *Journal of Computational Physics*, 362:346–374, June 2018. doi:10.1016/j.jcp.2018.02.033.

[2] David A Kopriva. Metric identities and the discontinuous spectral element method on curvilinear meshes. *Journal of Scientific Computing*, 26(3):301–327, March 2006. doi:10.1007/s10915-005-9070-8.

[3] Jesse Chan, Russell J Hewett, and T Warburton. Weight-adjusted discontinuous Galerkin methods: Wave propagation in heterogeneous media. *SIAM Journal on Scientific Computing*, 39(6):A2935–A2961, 2017. doi:10.1137/16m1089186.

[4] Jesse Chan, Russell J Hewett, and T Warburton. Weight-adjusted discontinuous Galerkin methods: Curvilinear meshes. *SIAM Journal on Scientific Computing*, 39(6):A2395–A2421, 2017. doi:10.1137/16m1089198.

[5] Fang Q Hu, MY Hussaini, and Patrick Rasetarinera. An analysis of the discontinuous Galerkin method for wave propagation problems. *Journal of Computational Physics*, 151(2):921–946, May 1999. doi:10.1006/jcph.1999.6227.

[6] Mark Ainsworth. Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. *Journal of Computational Physics*, 198(1):106–130, July 2004. doi:10.1016/j.jcp.2004.01.004.

[7] Zhijian J. Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, Hung T Huynh, Norbert Kroll, Georg May, Per-Olof Persson, Bram van Leer, and Miguel Visbal. High-order CFD methods: Current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, January 2013. doi:10.1002/fld.3767.

[8] Per-Olof Persson and Jaime Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. In *44th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, January 2006. doi:10.2514/6.2006-112.

[9] Lilia Krivodonova. Limiters for high-order discontinuous Galerkin methods. *Journal of Computational Physics*, 226(1):879–896, September 2007. doi:10.1016/j.jcp.2007.05.011.

[10] Garrett E Barter and David L Darmofal. Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation. *Journal of Computational Physics*, 229(5):1810–1827, March 2010. doi:10.1016/j.jcp.2009.11.010.

[11] Jean-Luc Guermond, Richard Pasquetti, and Bojan Popov. Entropy viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 230(11):4248–4267, May 2011. doi:10.1016/j.jcp.2010.11.043.

[12] Travis C Fisher and Mark H Carpenter. High-order entropy stable finite difference schemes for nonlinear conservation laws: Finite domains. *Journal of Computational Physics*, 252:518–557, November 2013. doi:10.1016/j.jcp.2013.06.014.

[13] Mark H Carpenter, Travis C Fisher, Eric J Nielsen, and Steven H Frankel. Entropy stable spectral collocation schemes for the Navier–Stokes equations: Discontinuous interfaces. *SIAM Journal on Scientific Computing*, 36(5):B835–B867, 2014. doi:10.1137/130932193.

[14] Matteo Parsani, Mark H Carpenter, Travis C Fisher, and Eric J Nielsen. Entropy stable staggered grid discontinuous spectral collocation methods of any order for the compressible Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 38(5):A3129–A3162, 2016. doi:10.1137/15m1043510.

[15] Lucas Friedrich, Andrew R Winters, David C Del Rey Fernández, Gregor J Gassner, Matteo Parsani, and Mark H Carpenter. An entropy stable h/p non-conforming discontinuous Galerkin method with the summation-by-parts property. *arXiv preprint arXiv:1712.10234*, 2017. URL: https://arxiv.org/abs/1712.10234.

[16] Tianheng Chen and Chi-Wang Shu. Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws. *Journal of Computational Physics*, 345:427–461, September 2017. doi:10.1016/j.jcp.2017.05.025.

[17] Jared Crean, Jason E Hicken, David C Del Rey Fernández, David W Zingg, and Mark H Carpenter. Entropy-stable summation-by-parts discretization of the Euler equations on general curved elements. *Journal of Computational Physics*, 356:410–438, March 2018. doi:10.1016/j.jcp.2017.12.015.

[18] Brian T Helenbrook. On the existence of explicit *hp*-finite element methods using Gauss–Lobatto integration on the triangle. *SIAM Journal on Numerical Analysis*, 47(2):1304–1318, 2009. doi:10.1137/070685439.

[19] Jason E Hicken, David C Del Rey Fernández, and David W Zingg. Multidimensional summation-by-parts operators: General theory and application to simplex elements. *SIAM Journal on Scientific Computing*, 38(4):A1935–A1958, 2016. doi:10.1137/15m1038360.

[20] PD Thomas and CK Lombard. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal*, 17(10):1030–1037, October 1979. doi:10.2514/3.61273.

[21] Jesse Chan. Weight-adjusted discontinuous Galerkin methods: Matrix-valued weights and elastic wave propagation in heterogeneous media. *International Journal for Numerical Methods in Engineering*, 113(12):1779–1809, March 2018. doi:10.1002/nme.5720.

[22] Thomas JR Hughes, LP Franca, and M Mallet. A new finite element formulation for computational fluid dynamics: I. symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Computer Methods in Applied Mechanics and Engineering*, 54(2):223–234, February 1986. doi:10.1016/0045-7825(86)90127-1.

[23] Michael S Mock. Systems of conservation laws of mixed type. *Journal of Differential Equations*, 37(1):70–88, July 1980. doi:10.1016/0022-0396(80)90089-3.

[24] Jesse Chan, Zheng Wang, Axel Modave, Jean-Francois Remacle, and T Warburton. GPU-accelerated discontinuous Galerkin methods on hybrid meshes. *Journal of Computational Physics*, 318:142–168, August 2016. doi:10.1016/j.jcp.2016.04.003.

[25] Jan S Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer New York, 2008. doi:10.1007/978-0-387-72067-8.

[26] Daniele Antonio Di Pietro and Alexandre Ern. *Mathematical Aspects of Discontinuous Galerkin Methods*. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-22980-0.

[27] Bernardo Cockburn and Chi-Wang Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Mathematics of Computation*, 52(186):411–435, April 1989. doi:10.1090/s0025-5718-1989-0983311-4.

[28] Bernardo Cockburn and Chi-Wang Shu. The Runge–Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, April 1998. doi:10.1006/jcph.1998.5892.

[29] Bernardo Cockburn. Devising discontinuous Galerkin methods for non-linear hyperbolic conservation laws. *Journal of Computational and Applied Mathematics*, 128(1-2):187–204, March 2001. doi:10.1016/s0377-0427(00)00512-4.

[30] Jianxian Qiu, Boo Cheong Khoo, and Chi-Wang Shu. A numerical study for the performance of the Runge–Kutta discontinuous Galerkin method based on different numerical fluxes. *Journal of Computational Physics*, 212(2):540–565, March 2006. doi:10.1016/j.jcp.2005.07.011.

[31] Xiangxiong Zhang, Yinhua Xia, and Chi-Wang Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. *Journal of Scientific Computing*, 50(1):29–62, January 2012. doi:10.1007/s10915-011-9472-8.

[32] A Klöckner, T Warburton, and JS Hesthaven. Viscous shock capturing in a time-explicit discontinuous Galerkin method. *Mathematical Modelling of Natural Phenomena*, 6(3):57–83, 2011. doi:10.1051/mmnp/20116303.

[33] Philippe G Ciarlet. *The finite element method for elliptic problems*. North-Holland Publishing Company, 1978.

[34] Jan Nordström. Conservative finite difference formulations, variable coefficients, energy estimates and artificial dissipation. *Journal of Scientific Computing*, 29(3):375–404, December 2006. doi:10.1007/s10915-005-9013-4.

[35] Gregor J Gassner, Andrew R Winters, and David A Kopriva. Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations. *Journal of Computational Physics*, 327:39–66, December 2016. doi:10.1016/j.jcp.2016.09.013.

[36] Roger A Horn and Charles R Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012.

[37] Praveen Chandrashekar. Kinetic energy preserving and entropy stable finite volume schemes for compressible Euler and Navier-Stokes equations. *Communications in Computational Physics*, 14(5):1252–1286, November 2013. doi:10.4208/cicp.170712.010313a.

[38] Andrew R Winters, Dominik Derigs, Gregor J Gassner, and Stefanie Walch. A uniquely defined entropy stable matrix dissipation operator for high Mach number ideal MHD and compressible Euler simulations. *Journal of Computational Physics*, 332:274–289, March 2017. doi:10.1016/j.jcp.2016.12.006.

[39] Jesse Chan and John A Evans. Multi-patch discontinuous Galerkin isogeometric analysis for wave propagation: Explicit time-stepping and efficient mass matrix inversion. *Computer Methods in Applied Mechanics and Engineering*, 333:22–54, May 2018. doi:10.1016/j.cma.2018.01.022.

[40] T Warburton. A low-storage curvilinear discontinuous Galerkin method for wave problems. *SIAM Journal on Scientific Computing*, 35(4):A1987–A2012, 2013. doi:10.1137/120899662.

[41] Susanne C Brenner and L Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. Springer New York, 3rd edition, 2008. doi:10.1007/978-0-387-75934-0.

[42] Mo Lenoir. Optimal isoparametric finite elements and error estimates for domains involving curved boundaries. *SIAM Journal on Numerical Analysis*, 23(3):562–580, 1986. doi:10.1137/0723036.

[43] Travis C Fisher, Mark H Carpenter, Jan Nordström, Nail K Yamaleev, and Charles Swanson. Discretely conservative finite-difference formulations for nonlinear conservation laws in split form: Theory and boundary conditions. *Journal of Computational Physics*, 234:353–375, February 2013. doi:10.1016/j.jcp.2012.09.026.

[44] A Johnen, J-F Remacle, and C Geuzaine. Geometrical validity of curvilinear finite elements. *Journal of Computational Physics*, 233:359–372, January 2013. doi:10.1016/j.jcp.2012.08.051.

[45] Miguel R Visbal and Datta V Gaitonde. On the use of higher-order finite-difference schemes on curvilinear and deforming meshes. *Journal of Computational Physics*, 181(1):155–185, September 2002. doi:10.1006/jcph.2002.7117.

[46] Qi Chen and Ivo Babuška. The optimal symmetrical points for polynomial interpolation of real functions in the tetrahedron. *Computer Methods in Applied Mechanics and Engineering*, 137(1):89–94, October 1996. doi:10.1016/0045-7825(96)01051-1.

[47] JS Hesthaven. From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex. *SIAM Journal on Numerical Analysis*, 35(2):655–676, 1998. doi:10.1137/s003614299630587x.

[48] MG Blyth and C Pozrikidis. A Lobatto interpolation grid over the triangle. *IMA Journal of Applied Mathematics*, 71(1):153–169, February 2006. doi:10.1093/imamat/hxh077.

[49] T Warburton. An explicit construction of interpolation nodes on the simplex. *Journal of Engineering Mathematics*, 56(3):247–262, November 2006. doi:10.1007/s10665-006-9086-6.

[50] Jesse Chan and T Warburton. A comparison of high order interpolation nodes for the pyramid. *SIAM Journal on Scientific Computing*, 37(5):A2151–A2170, 2015. doi:10.1137/141000105.

[51] DM Williams, Lee Shunn, and Antony Jameson. Symmetric quadrature rules for simplexes based on sphere close packed lattice arrangements. *Journal of Computational and Applied Mathematics*, 266:18–38, August 2014. doi:10.1016/j.cam.2014.01.007.

[52] Freddie D Witherden and Peter E Vincent. On the identification of symmetric quadrature rules for finite element methods. *Computers & Mathematics with Applications*, 69(10):1232–1241, May 2015. doi:10.1016/j.camwa.2015.03.017.

[53] David C Del Rey Fernández, Jason E Hicken, and David W Zingg. Simultaneous approximation terms for multi-dimensional summation-by-parts operators. *Journal of Scientific Computing*, 75(1):83–110, April 2018. doi:10.1007/s10915-017-0523-7.

[54] Mark H Carpenter and Christopher A Kennedy. Fourth-order 2N-storage Runge-Kutta schemes. Technical Report NASA-TM-109112, NAS 1.15:109112, NASA Langley Research Center, 1994. doi:2060/19940028444.

[55] Hong Xiao and Zydrunas Gimbutas. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers & Mathematics with Applications*, 59(2):663–676, January 2010. doi:10.1016/j.camwa.2009.10.027.

[56] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, September 2009. doi:10.1002/nme.2579.

[57] Florian Hindenlang, Gregor J Gassner, Christoph Altmann, Andrea Beck, Marc Staudenmaier, and Claus-Dieter Munz. Explicit discontinuous Galerkin methods for unsteady problems. *Computers & Fluids*, 61:86–93, May 2012. doi:10.1016/j.compfluid.2012.03.006.

[58] Amiram Harten. On the symmetric form of systems of conservation laws with entropy. *Journal of Computational Physics*, 49(1):151–164, January 1983. doi:10.1016/0021-9991(83)90118-3.

[59] Gregor J Gassner, Andrew R Winters, and David A Kopriva. A well balanced and entropy conservative discontinuous Galerkin spectral element method for the shallow water equations. *Applied Mathematics and Computation*, 272:291–308, January 2016. doi:10.1016/j.amc.2015.07.014.

[60] Chi-Wang Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, pages 325–432. Springer Berlin Heidelberg, 1998. doi:10.1007/bfb0096355.

[61] David M Williams and Antony Jameson. Nodal points and the nonlinear stability of high-order methods for unsteady flow problems on tetrahedral meshes. In *21st AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2013. doi:10.2514/6.2013-2830.

[62] Thomas Toulorge, Jonathan Lambrechts, and Jean-François Remacle. Optimizing the geometrical accuracy of curvilinear meshes. *Journal of Computational Physics*, 310:361–380, April 2016. doi:10.1016/j.jcp.2016.01.023.

[63] Geoffrey I Taylor and Albert E Green. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 158(895):499–521, February 1937. doi:10.1098/rspa.1937.0036.

[64] James DeBonis. Solutions of the Taylor-Green vortex problem using high-resolution explicit finite difference methods. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, January 2013. doi:10.2514/6.2013-382.

[65] Jeffrey W Banks and Thomas Hagstrom. On Galerkin difference methods. *Journal of Computational Physics*, 313:310–327, 2016.