On discretely entropy conservative discontinuous Galerkin methods

Jesse Chan

Abstract

High order methods based on diagonal-norm summation by parts operators can be shown to semi-discretely conserve entropy for nonlinear systems of hyperbolic PDEs [1, 2]. These methods can be interpreted as mass lumped discontinuous Galerkin spectral element methods (DG-SEM) [3, 4, 5, 6]. In this work, we describe how to extend discretely entropy conservative schemes for a more general class of DG methods using quadrature-based L^2 projections, flux differencing, and specific DG differentiation operators. This approach also recovers existing methods for Burgers' equation involving generalized SBP operators without boundary nodes and dense norm SBP operators. Numerical experiments confirm the stability and high order accuracy of the proposed methods for the one-dimensional shallow water and compressible Euler equations.

1 Introduction

Numerical simulations in engineering increasingly require higher accuracy without sacrificing computational efficiency. High order methods are more accurate than low order methods per degree of freedom for sufficiently regular solutions, and provide one avenue towards improving fidelity in numerical simulations while maintaining reasonable computational costs. High order methods which can accomodate unstructured meshes are particularly well-suited for problems with complex geometries. Among such methods, high order discontinuous Galerkin (DG) methods are particularly well-suited to the solution of time-dependent hyperbolic problems on modern computing architectures [7, 8].

The accuracy observed for high order methods can be attributed in part to their low numerical dissipation and dispersion properties [9]. While low numerical dissipation is advantageous for problems in linear wave propagation [10], the high numerical dissipation present in low order schemes adds a stabilizing effect for nonlinear hyperbolic problems [11]. As a result, high order schemes for nonlinear conservation laws tend to require additional stabilization, including (but not limited to) filtering [7], slope limiting [12], artificial viscosity [13], and polynomial de-aliasing [14, 15]. Moreover, stabilized numerical methods can still fail, requiring user intervention or heuristic modifications to achieve non-divergent solutions.

For linear wave propagation problems, numerical methods can be shown to be semi-discretely energy stable, even in the presence of curvilinear coordinates or variable coefficients [16, 17, 18, 19]. This semi-discrete stability can be used to show that, under a stable timestep restriction (CFL), discrete solutions obtained by explicit time-stepping do not suffer from un-physical growth in time. For nonlinear systems of conservation laws, typical methods for fluid dynamics do not admit formal proofs of stability. However, progress has been made in designing provably discretely stable methods, beginning with the introduction of entropy conservative and entropy stable finite volume schemes by Tadmor [20, 21]. These schemes rely on a specific entropy conservative flux which satisfies a condition involving the entropy variables and entropy potential, and have been extended to low order finite volume methods on unstructured grids [22].

The construction of entropy conservative high order schemes was done much more recently for the compressible Euler and Navier-Stokes equations in [1, 2] based on a spectral collocation approach on tensor product elements, which can be interpreted as a mass-lumped DG spectral element (DG-SEM) scheme. The proof of entropy conservation relies on the presence of a diagonal mass matrix, as well as the summation by parts property [23]. Similar entropy-stable schemes were constructed for the shallow water and MHD equations [4, 5, 24]. More recently, entropy conservative and entropy stable schemes were extended to unstructured triangular meshes in [6] using the split-form methodology of [3] and multi-dimensional SBP operators [25].

To the authors knowledge, the construction of entropy conservative and entropy stable schemes have relied on the use of lumped diagonal mass matrices and nodal collocation (diagonal norm SBP operators). Entropy stable high order finite element and DG methods have been proposed [26], but the proofs are often given at the continuous level, relying on exact integration or the chain rule, neither of which holds at the discrete level. Thus, while DG-SEM based on flux differencing is entropy conservative, this is tied to a specific choice of basis and quadrature. Diagonal-norm SBP operators can be constructed for triangles and tetrahedra [27, 25, 6]; however, the number of nodal points for such operators is greater than the dimension of the natural polynomial approximation space. Furthermore, appropriate point sets have only been constructed for $N \leq 4$ in three dimensions [28].

This work focuses on the construction of entropy conservative high order DG schemes for systems of conservation laws. In order to generalize beyond collocation DG and diagonal norm SBP operators, we will consider dense (non-lumped) mass matrices and more general quadrature rules (e.g. rules without boundary points, rules with more points than the dimension of the corresponding approximation space). We will present proofs of discrete entropy stability in terms of integrals and L^2 projections, which we assume are computed using an appropriate quadrature rule. We stress that, while we utilize a continuous notation involving integrals rather than discrete sums, the proofs rely only on properties of quadrature-based integration and L^2 projection.

The outline of the paper is as follows: Section 2 will briefly review the construction of entropy conservative collocation methods on a single element. Section 3 will describe how to construct analogous entropy conservative methods on single element. Section 4 will discuss extensions to multiple elements, including comparisons of different coupling terms and entropy stable fluxes. We note that, for simplicity and clarity of presentation, the proposed methods and proofs are given in one dimension. However, the extension to multiple dimensions (including simplicial elements) is straightforward and will be described in an upcoming manuscript. Finally, Section 5 presents numerical results which verify the high order accuracy and discrete entropy conservation of the proposed methods.

2 Entropy conservative diagonal norm collocation DG methods

We will begin by reviewing continuous entropy theory, as well as existing high order methods which are provably entropy conservative at the discrete level. These methods were introduced in [1, 2] for tensor product elements, though we adopt methods of proof introduced more recently in [29, 6]. These methods all share a basis in diagonal norm summation by parts operators. Because they can be derived from a nodal discontinuous Galerkin framework through the collocation of nodal points and quadrature points, we refer to them as "diagonal norm collocation DG methods" in this work.

2.1 Entropy stability for systems of hyperbolic PDEs

We consider systems of nonlinear conservation laws in one dimension with n variables

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = 0, \qquad \boldsymbol{u}(x,t) = (u_1(x,t), \ldots, u_n(x,t)). \tag{1}$$

where the fluxes $f(\boldsymbol{u})$ are smooth functions of the vector of conservative variables $\boldsymbol{u}(x,t)$.The Jacobian matrix $\boldsymbol{A}(\boldsymbol{u})$ is defined entrywise as

$$(\boldsymbol{A}(\boldsymbol{u}))_{ij} = \left(\frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial \boldsymbol{u}_j}\right)_i.$$

We are interested in systems for which there exists a convex entropy function $U(\boldsymbol{u})$ such that

$$U''(\boldsymbol{u})\boldsymbol{A}(\boldsymbol{u}) = \left(U''(\boldsymbol{u})\boldsymbol{A}(\boldsymbol{u})\right)^T. \tag{2}$$

For systems with convex entropy functions, one can define entropy variables $\boldsymbol{v} = U'(\boldsymbol{u})$. The convexity of the $U(\boldsymbol{u})$ guarantees that the mapping between conservative and entropy variables is invertible.

It can be shown (see, for example, [30]) that (2) is equivalent to the existence of an entropy flux functions $F(\boldsymbol{u})$ such that

$$\boldsymbol{v}^T \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} = \frac{\partial F(\boldsymbol{u})}{\partial \boldsymbol{u}}^T.$$

The flux function and entropy flux function are further related through the so-called entropy potential ψ_k

$$\psi(\boldsymbol{v}) = \boldsymbol{v}^T \boldsymbol{f}(\boldsymbol{u}(\boldsymbol{v})) - F(\boldsymbol{u}(\boldsymbol{v})), \qquad \psi'(\boldsymbol{v}) = \boldsymbol{f}(\boldsymbol{u}(\boldsymbol{v})).$$

When $\boldsymbol{u}$ is smooth, multiplying (1) on the left by $\boldsymbol{v}^T = U'(\boldsymbol{u})^T$, applying the definition of the entropy flux and using the chain rule yields the conservation of entropy

$$\frac{\partial U(\boldsymbol{u})}{\partial t} + \frac{\partial F(\boldsymbol{u})}{\partial x} = 0.$$

More generally, it can be shown that physically relevant solutions of (1) (defined as the limiting solution for an appropriately defined vanishing viscosity) satisfy the entropy inequality

$$\frac{\partial U(\boldsymbol{u})}{\partial t} + \frac{\partial F(\boldsymbol{u})}{\partial x} \leq 0.$$

We assume now that the domain is the interval $[-1, 1]$. Integrating the entropy inequality over this interval and using the definition of the entropy potential then yields

$$\int_{-1}^{1} \frac{\partial U(\boldsymbol{u})}{\partial t} + \left(\boldsymbol{v}^T \boldsymbol{f}(\boldsymbol{u}(\boldsymbol{v})) - \psi(\boldsymbol{v})\right)\Big|_{-1}^{1} \leq 0. \tag{3}$$

The focus of this work is the construction of methods which satisfy a discrete analogue of this entropy inequality. We begin by reviewing existing discretely entropy conservative methods based on collocation and diagonal norm summation by parts (SBP) operators.

2.2 Entropy conservative collocation methods based on SBP operators

Let $x_i \in [-1, 1]$ denote the $(N + 1)$ point Gauss-Lobatto-Legendre (GLL) quadrature rule with positive weights $w_1, \ldots, w_{N+1}$. Let $\ell_i(x)$ denote the nodal Lagrange basis defined at GLL points, and define the differentiation matrix $\boldsymbol{D}$ be defined as

$$\boldsymbol{D}_{ij} = \frac{\partial \ell_j(x_i)}{\partial x}.$$

The matrix $\boldsymbol{D}$ maps from nodal values of a polynomial to nodal values of its derivative. We also define the mass matrix $\boldsymbol{M}$

$$\boldsymbol{M}_{ij} = \int_{-1}^{1} \ell_i(x) \ell_j(x) \, \mathrm{d}x \approx \delta_{ij} w_i,$$

where the approximation results from evaluating the integral using GLL quadrature. For this choice of quadrature, the Kronecker property $\ell_j(x_i) = \delta_{ij}$ implies that $\boldsymbol{M}$ is a diagonal matrix whose entries are simply the quadrature weights.

The mass and derivative matrices may be used to define the diagonal norm SBP operator $\boldsymbol{S} = \boldsymbol{M}\boldsymbol{D}$, with the summation-by-parts property

$$\boldsymbol{S} = \boldsymbol{B} - \boldsymbol{S}^T, \qquad \boldsymbol{B}_{ij} = \begin{cases} 1 & i = j = 1 \\ 1 & i = j = N + 1 \\ 0 & \text{otherwise} \end{cases}.$$

This property is simply a restatement of integration by parts, using that the derivative of $\ell_j(x) \in P^{N-1}$ and that GLL quadrature is exact for polynomials of degree $2N-1$.

We seek a polynomial approximation to the solution $\boldsymbol{u}(x)$ of (1). A DG formulation in the "strong form" [7] is given as

$$\int_{-1}^{1} \left(\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} \right) \boldsymbol{w} + \left(\boldsymbol{f}^* - \boldsymbol{f}(\boldsymbol{u}) \right) \boldsymbol{w}|_{-1}^{1} = 0, \qquad \forall \boldsymbol{w} \in V_h.$$

Here, $\boldsymbol{f}^*$ is a numerical flux used for the weak imposition of boundary or continuity conditions.

We derive a DG-SEM approximation by evaluating the above integrals using GLL quadrature. We furthermore assume that the solution is approximated using a nodal basis at GLL points

$$\boldsymbol{u}(x,t) = \sum_{j=1}^{N+1} \boldsymbol{u}_j(t) \ell_j(x).$$

The resulting semi-discrete system yields a nodal collocation scheme for $\boldsymbol{u}_i$

$$\frac{\partial \boldsymbol{u}_i}{\partial t} + \sum_{j=1}^{N+1} \left((\boldsymbol{D} \otimes \boldsymbol{I})_{ij} \, \boldsymbol{f}(\boldsymbol{u}_j) + \left(\boldsymbol{M}^{-1} \boldsymbol{B} \right)_{ij} \left(\boldsymbol{f}_i^* - \boldsymbol{f}(\boldsymbol{u}_i) \right) \right) = 0.$$

This formulation conserves the quantities $\boldsymbol{u}_i$, but (except in specific cases) does not satisfy a discrete analogue of the statement of entropy conservation (3). Entropy conservative schemes may be constructed based on the SBP property and a "flux differencing" approach [3] involving a definition of entropy conservative/stable fluxes given by Tadmor [20, 21].

Definition 1. Let $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ be a bivariate function such that it is symmetric and consistent with the flux function $\boldsymbol{f}(\boldsymbol{u})$

$$\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R) = \boldsymbol{f}_S(\boldsymbol{u}_R, \boldsymbol{u}_L), \qquad \boldsymbol{f}_S(\boldsymbol{u}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u})$$

The numerical flux $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ is entropy conservative if, for entropy variables $\boldsymbol{v}_L = \boldsymbol{v}(\boldsymbol{u}_L), \boldsymbol{v}_R = \boldsymbol{v}(\boldsymbol{u}_R)$

$$\left(\boldsymbol{v}_L - \boldsymbol{v}_R \right)^T \boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R) = (\psi_L - \psi_R), \qquad \psi_L = \psi(\boldsymbol{v}(\boldsymbol{u}_L)), \quad \psi_R = \psi(\boldsymbol{v}(\boldsymbol{u}_R)).$$

Similarly, a flux $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ is referred to as entropy stable if $\left(\boldsymbol{v}_L - \boldsymbol{v}_R \right)^T \boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R) \leq (\psi_L - \psi_R)$.

An entropy conservative nodal collocation method can then be derived by modifying the flux derivative as follows

$$\frac{\partial \boldsymbol{u}_i}{\partial t} + \sum_{j=1}^{N+1} \left((2\boldsymbol{D} \otimes \boldsymbol{I})_{ij} \, \boldsymbol{f}_S(\boldsymbol{u}_i, \boldsymbol{u}_j) + \left(\boldsymbol{M}^{-1} \boldsymbol{B} \right)_{ij} \left(\boldsymbol{f}_i^* - \boldsymbol{f}(\boldsymbol{u}_i) \right) \right) = 0. \tag{4}$$

The replacement of the derivative of the flux function with flux differencing allows one to prove that (4) conserves a discrete entropy [29, 6]. We reproduce these proofs here, as we will refer to them in following sections.

Theorem 1. *The collocation method (4) is discretely entropy conservative in the sense that*

$$\boldsymbol{1}^T \left(\boldsymbol{M} \frac{\partial U(\boldsymbol{u})}{\partial t} + \boldsymbol{B} \left(\boldsymbol{v}^T \boldsymbol{f}^* - \boldsymbol{\psi} \right) \right) = 0,$$

where $\boldsymbol{1}$ *denotes the vector of ones and* $\boldsymbol{\psi}_i = \psi(\boldsymbol{v}_i)$ *is the entropy potential in terms of the nodal value of* $\boldsymbol{u}_i$*.*

Proof. Let $\boldsymbol{v}_i = \boldsymbol{v}(\boldsymbol{u}_i)$ denote the entropy variables, defined in terms of the nodal values $\boldsymbol{u}_i$. We multiply (4) by $(\boldsymbol{M}\boldsymbol{v})^T$. Using the fact that $\boldsymbol{M}$ is diagonal, the time derivative term can be manipulated to

$$\boldsymbol{v}^T \boldsymbol{M} \frac{\partial \boldsymbol{u}}{\partial t} = \sum_{i=1}^{N+1} \boldsymbol{M}_{ii} \boldsymbol{v}_i^T \frac{\partial \boldsymbol{u}_i}{\partial t} = \sum_{i=1}^{N+1} \boldsymbol{M}_{ii} \frac{\partial S_i}{\partial t} = \boldsymbol{1}^T \boldsymbol{M} \frac{\partial S_i}{\partial t},$$

where we have used continuity in time to arrive at the point-wise relation

$$\boldsymbol{v}^T \frac{\partial \boldsymbol{u}}{\partial t} = \frac{\partial U(\boldsymbol{u})}{\partial \boldsymbol{u}}^T \frac{\partial \boldsymbol{u}}{\partial t} = \frac{\partial U(\boldsymbol{u})}{\partial t}.$$

The spatial derivative term can be manipulated using the summation by parts property $\boldsymbol{M}\boldsymbol{D} = \boldsymbol{S} = \boldsymbol{B} - \boldsymbol{S}^T$

$$\begin{aligned}\sum_{i,j=1}^{N+1} \boldsymbol{v}_i^T 2\boldsymbol{M}\boldsymbol{D}_{ij} \boldsymbol{f}_S(\boldsymbol{u}_i, \boldsymbol{u}_j) &= \sum_{i,j=1}^{N+1} \boldsymbol{v}_i^T (\boldsymbol{S} + \boldsymbol{B} - \boldsymbol{S}^T)_{ij} \boldsymbol{f}_S(\boldsymbol{u}_i, \boldsymbol{u}_j) \\ &= \sum_{i,j=1}^{N+1} \boldsymbol{S}_{ij} \boldsymbol{v}_i^T \boldsymbol{f}_S(\boldsymbol{u}_i, \boldsymbol{u}_j) - \boldsymbol{S}_{ji} \boldsymbol{v}_i^T \boldsymbol{f}_S(\boldsymbol{u}_i, \boldsymbol{u}_j) + \boldsymbol{B}_{ij} \boldsymbol{v}_i^T \boldsymbol{f}_S(\boldsymbol{u}_i, \boldsymbol{u}_j). \end{aligned} \tag{5}$$

Rearranging indices and using symmetry of $\boldsymbol{f}_S(\boldsymbol{u}_i, \boldsymbol{u}_j)$ then yields that

$$\begin{aligned}\sum_{i,j=1}^{N+1} \boldsymbol{S}_{ij} \boldsymbol{v}_i^T \boldsymbol{f}_S(\boldsymbol{u}_i, \boldsymbol{u}_j) - \boldsymbol{S}_{ji} \boldsymbol{v}_i^T \boldsymbol{f}_S(\boldsymbol{u}_i, \boldsymbol{u}_j) &= \sum_{i,j=1}^{N+1} \boldsymbol{S}_{ij} \left(\boldsymbol{v}_i - \boldsymbol{v}_j\right)^T \boldsymbol{f}_S(\boldsymbol{u}_i, \boldsymbol{u}_j) \\ &= \sum_{i,j=1}^{N+1} \boldsymbol{S}_{ij} \left(\boldsymbol{\psi}_i - \boldsymbol{\psi}_j\right) = -\boldsymbol{1}^T \boldsymbol{S} \boldsymbol{\psi} = -\boldsymbol{1}^T \boldsymbol{B} \boldsymbol{\psi}, \end{aligned}$$

where we have used $\boldsymbol{S}\boldsymbol{1} = 0$ and the summation by parts property in the last line.

Using that $\boldsymbol{B}$ is diagonal, combining the latter term in (5) with boundary terms involving the numerical flux yields

$$\sum_{i,j=1}^{N+1} \boldsymbol{B}_{ij} \boldsymbol{v}_i^T \boldsymbol{f}_S(\boldsymbol{u}_i, \boldsymbol{u}_j) + \boldsymbol{v}_i^T \boldsymbol{B}_{ij} \left(\boldsymbol{f}_i^* - \boldsymbol{f}(\boldsymbol{u}_i)\right) = \boldsymbol{1}^T \boldsymbol{B} \left(\boldsymbol{v}^T \boldsymbol{f}^*\right),$$

where we have used the consistency of $\boldsymbol{f}_S(\boldsymbol{u}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u})$. Combining everything results in a discrete statement of the conservation of entropy (3). □

We note that, unlike the proof of continuous entropy conservation, the proof of discrete entropy conservation does not require the use of the chain rule, which may not hold at the discrete level. This proof of entropy conservation can also be extended beyond a single element by specifying appropriate simultaneous approximation terms (SBP-SAT). This can be done by taking $\boldsymbol{f}_i^*$ to be the entropy conservative flux $f_S(u_L, u_R)$ at the interface between two elements [2, 29, 6]. Similarly, employing an entropy stable flux at element interfaces results in an entropy stable method which satisfies a global entropy inequality.

It is important to emphasize that conservation of entropy does not, in general, guarantee stability of the numerical scheme unless the solution satisfies additional constraints. For example, the convex entropy associated with the compressible Euler and Navier-Stokes equations is well-defined only if the density and pressure are positive, and steps must be taken in any numerical scheme to guarantee that the discrete solution of such constraints. For DG methods, this is most commonly done using bound and positivity-preserving limiters, such as those introduced in [31, 32].

3 Entropy conservative DG methods on a single element

In this section, we focus on constructing entropy conservative DG schemes for general choices of basis and quadrature. These include (as special cases) dense norm SBP operators and generalized SBP (GSBP) operators which do not contain boundary points, which introduce additional difficulties in constructing entropy conservative DG schemes. These difficulties present themselves primarily in two areas:

1. the generalization of flux differencing to a continuous formulation.

2. the proof of entropy stability assumes a diagonal norm mass matrix, which commutes with point-wise multiplication by the (nodal values of) the entropy variables.

We address both areas of difficulty in generalizing the entropy conservative DG-SEM scheme (4). In particular, we give a continuous DG formulation with flux differencing, and prove that the generalized scheme is entropy conservative (stable) on a single element in the following sections. As an example, we illustrate how to apply the proposed methods to produce energy conservative formulations of Burgers' equation.

3.1 Notation

Fix notation: switch V_h to P^N We assume that the domain $\Omega \in \mathbb{R}^d$ is exactly triangulated by a mesh which consists of the union of non-overlapping elements D^k. We further assume that each element D^k is the image of a reference element $\widehat{D}$ under the local elemental mapping

$$x^k = \boldsymbol{\Phi}^k \widehat{x},$$

where x^k denote physical coordinates on D^k and $\widehat{x}$ denote coordinates on the reference element. Finally, we identify J as the determinant of the Jacobian of $\boldsymbol{\Phi}^k$, and refer to it as the Jacobian for the remainder of this work. For clarity and simplicity of presentation, we assume $\boldsymbol{\Phi}^k$ is affine in this case, such that J is constant over each element D^k.

We will approximate solution components over each element D^k from an approximation space $V_h\left(D^k\right)$, which we define as the composition of the mapping $\boldsymbol{\Phi}^k$ and a reference approximation space $V_h\left(\widehat{D}\right)$

$$V_h\left(D^k\right) = \boldsymbol{\Phi}^k \circ V_h\left(\widehat{D}\right).$$

The global approximation space $V_h\left(\Omega\right)$ is then defined as the direct sum of local approximation spaces

$$V_h\left(\Omega\right) = \bigoplus_{D^k} V_h\left(D^k\right).$$

We assume that $V_h(D^k) \subset H^1(D^k)$; in other words, that all elements of each local approximation space are square integrable and admit well-defined traces.

We define the L^2 norm and inner products over the element D^k and the surface of the element ∂D^k[1]

$$\left(u, v\right)_{D^k} = \int_{D^k} uv \, \mathrm{d}x, \qquad \|u\|^2_{D^k} = \left(u, u\right)_{D^k}, \qquad \left\langle u, v\right\rangle_{\partial D^k} = \int_{\partial D^k} uv \, \mathrm{d}x.$$

We also define the local L^2 projection operator $\Pi_N : L^2(D^k) \rightarrow V_h(D^k)$ such that

$$\left(\Pi_N u, v\right)_{D^k} = \left(u, v\right)_{D^k}, \qquad \forall v \in V_h\left(D^k\right).$$

Throughout this work, we will use that this definition of the L^2 projection holds even when integrals are approximated using quadrature. We will make no assumptions on quadrature except that specify assumptions - integrate polynomials of degree N on the reference element exactly, such that integration by parts holds. Add footnote that we can also generalize to the under-integrated case.

Finally, we define the jump and average of discontinuous functions across element interfaces. Let $D^{k,+}$ denote the neighboring element of across a face f of D^k, and let u^+ denote the values of u on $D^{k,+}$. The jump of u across f on D^k is then defined as

$$[\![u]\!] = u^+ - u, \qquad \{\{u\}\} = \frac{u^+ + u}{2}.$$

[1] In one dimension, D^k is some interval $[x_L, x_R]$ and the L^2 inner product over the surface reduces to

$$\langle u, v\rangle_{\partial D^k} = u(x_L)v(x_L) + u(x_R)v(x_R).$$

However, the presented proofs in this work will not utilize this simplification, and are thus generalizable to higher dimensions in a straightforward manner.

The jump and average of vector fields are defined component-wise using the jumps and averages of components. When the face f lies on the boundary, we define $u^+ = u$, such that $[\![u]\!] = 0$ and $\{\{u\}\} = u$ on $\partial\Omega$.

3.2 A continuous interpretation of flux differencing

In this section, we offer a continuous interpretation of flux differencing [1, 29, 6], which also encompasses the split form methodology proposed in [3]. This interpretation will guide the construction of entropy conservative DG schemes.

Assume that $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ is the symmetric and consistent two-point flux defined in (1). Consider the bilinear function

$$\boldsymbol{f}_S\left(\boldsymbol{u}(x), \boldsymbol{u}(y)\right), \qquad x, y \in \mathbb{R}.$$

Flux differencing can be interpreted as differentiating $\boldsymbol{f}_S\left(\boldsymbol{u}(x), \boldsymbol{u}(y)\right)$ along one coordinate x, then evaluating the result with $y = x$

$$\frac{\partial \boldsymbol{f}(\boldsymbol{u}(x))}{\partial x} \Longrightarrow 2\left.\frac{\partial \boldsymbol{f}_S\left(\boldsymbol{u}(x), \boldsymbol{u}(y)\right)}{\partial x}\right|_{y=x}. \tag{6}$$

It was shown in [6] that flux differencing is a consistent approximation of the flux derivative.

It is helpful to first understand flux differencing as a method for recovering split formulations. We illustrate this using the one-dimensional Burgers equation on domain $[-1, 1]$, which is given in conservative form as

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \qquad f(u) = \frac{1}{2}u^2.$$

Assuming smooth u, Burgers' equation conserves a square entropy $U(\boldsymbol{u}) = u^2/2$. This can be shown by multiplying by u, integrating over $[-1, 1]$, and using $\frac{\partial u}{\partial t}u = \frac{1}{2}\frac{\partial u^2}{\partial t}$

$$\frac{1}{2}\frac{\partial}{\partial t}\int_{-1}^{1} u^2 = -\left.\frac{u^3}{3}\right|_{-1}^{1}. \tag{7}$$

It is well known that DG formulations based on a split formulation can be shown to conserve the square entropy u^2 [23, 33]. This form may be also be recovered using flux differencing under the two-point flux

$$f_S(u_L, u_R) = \frac{1}{6}(u_L^2 + u_L u_R + u_R^2).$$

Applying (6) yields the split form of Burgers' equation

$$2\left.\frac{\partial f_S(u(x), u(y))}{\partial x}\right|_{y=x} = \frac{1}{3}\left.\frac{\partial\left(u(x)^2 + u(x)u(y) + u(y)^2\right)}{\partial x}\right|_{y=x} = \frac{1}{3}\left(\frac{\partial u^2}{\partial x} + u\frac{\partial u}{\partial x}\right).$$

We note that recovering the split-form of Burgers' equation relies on the property that

$$\frac{\partial\left(u(y)^2\right)}{\partial x} = u(y)^2\frac{\partial\left(1\right)}{\partial x} = 0.$$

Thus, recovery of the split form can be achieved by using flux differencing in combination with any differential operator D such that $D1 = 0$.

3.3 Discrete differential operators and weighted integration by parts

Both the conservative and split forms of Burgers' equation are equivalent at the continuous level (assuming continuity of u and exact differentiation); however, this is no longer true at the discrete level. Suppose now that the solution is approximated by $u \in V_h(\Omega)$ on a single element. Since $f(u) = u^2/2 \notin V_h(\Omega)$ for general choices of u, V_h, it is not possible to differentiate $f(u)$ exactly without prior knowledge of the form of the nonlinearity. Indeed, it is more common to first project $f(u)$ to V_h, then differentiate, such that the discrete scheme is equivalent to solving for $u \in V_h(\Omega)$ such that

$$\int_{-1}^{1} \left(\frac{\partial u}{\partial t} + \frac{\partial \Pi_N f(u)}{\partial x} \right) w \, \mathrm{d}x = 0, \qquad \forall w \in V_h(\Omega).$$

This scheme does not, in general, conserve energy.

We can derive a semi-discrete Galerkin scheme based which conserves energy discretely using flux differencing and a different choice of discrete differentiation operator D_N, which was motivated by the SBP operators introduced in [6]. Let $w(x)$ be some bounded real-valued function; then, we can define the discrete differential operator $D_N : H^1 \to V_h$ such that, for $u \in H^1$,

$$(D_N u, vw)_{\widehat{D}} = \left(\frac{\partial \Pi_N u}{\partial x}, vw \right)_{\widehat{D}} + \frac{1}{2} \langle u - \Pi_N u, vw \boldsymbol{n}_x \rangle_{\partial \widehat{D}} + \frac{1}{2} \left\langle u - \Pi_N u, \Pi_N \left(vw \right) \boldsymbol{n}_x \right\rangle_{\widehat{D}}, \qquad \forall v \in V_h. \tag{8}$$

This operator satisfies a form of integration by parts, which is summarized by the following lemma:

Lemma 1. *The linear operator D_N satisfies a "weighted" integration by parts property with respect the L^2 inner product and any bounded real-valued function $w(x)$*

$$(D_N u, wv)_{\widehat{D}} = \langle u, wv \boldsymbol{n}_x \rangle_{\partial \widehat{D}} - (u, D_N(wv))_{\widehat{D}}.$$

Proof. Integrating the left hand side by parts gives

$$-\left(\frac{\partial \Pi_N (vw)}{\partial x}, u \right)_{\widehat{D}} + \frac{1}{2} \langle u - \Pi_N u, vw \boldsymbol{n}_x \rangle_{\partial \widehat{D}} + \frac{1}{2} \left\langle u + \Pi_N u, \Pi_N \left(vw \right) \boldsymbol{n}_x \right\rangle_{\partial \widehat{D}}, \qquad \forall v \in V_h.$$

Rearranging boundary terms yields

$$-\left(\frac{\partial \Pi_N (vw)}{\partial x}, u \right)_{\widehat{D}} + \frac{1}{2} \left\langle vw + \Pi_N(vw), u \boldsymbol{n}_x \right\rangle_{\partial \widehat{D}} - \frac{1}{2} \left\langle vw - \Pi_N(vw), \Pi_N u \boldsymbol{n}_x \right\rangle_{\partial \widehat{D}}.$$

Adding and subtracting $\langle vw, u \boldsymbol{n}_x \rangle_{\partial \widehat{D}}$ completes the proof. □

Direct computations also show that $(D_N 1, vw)_{\widehat{D}} = 0$ for all bounded weights $w(x)$ and $\forall v \in V_h$. Thus, we can apply (6) with D_N to construct a discrete formulation for Burgers' equation

$$\left(\frac{\partial u}{\partial t}, v \right) + \frac{1}{3} \left(D_N u^2 + u D_N u, v \right) = \left(\frac{\partial u}{\partial t}, v \right)_{\widehat{D}} + \frac{1}{3} \left(D_N u^2, v \right)_{\widehat{D}} + \frac{1}{3} \left(D_N u, uv \right)_{\widehat{D}}, \qquad \forall v \in V_h. \tag{9}$$

To show this formulation is discretely energy conservative, we take $v = u$ and use Lemma 1 to cancel volume terms, yielding

$$\frac{1}{2} \frac{\partial}{\partial t} (u, u)_{\widehat{D}} = -\frac{1}{3} \left\langle u^2, u \right\rangle_{\partial \widehat{D}},$$

which is a discrete analogue of the conservation condition (7). This is equivalent to the skew-symmetric split formulation described in [23]; however, as noted in [6], it can be derived from flux differencing without explicit knowledge of the skew form. We will show later in Section 4.2.2 that a natural extension of this formulation to multiple elements recovers the weak form of the scheme introduced in [33].

3.4 Conservation of entropy on one element

While flux differencing can be used to recover split formulations, it can also be used to construct entropy stable formulations which do not correspond to skew-symmetric splittings. In this section we will show that, for any sufficiently accurate quadrature rule, it is possible to construct discretizations which satisfy a discrete equivalent of conservation of entropy (3).

For the proof, we will need the following assumption:

Assumption 1. *The numerical flux function $\boldsymbol{f}_S$ admits an absolutely convergent expansion of the form*

$$\boldsymbol{f}_S(x,y) = \sum_{i=1}^{\infty} \boldsymbol{f}_i(x)\boldsymbol{g}_i(y),$$

where $\boldsymbol{f}_i, \boldsymbol{g}_i$ are real and bounded over each element D^k.

This assumption is satisfied, for example, by $\boldsymbol{f}_S$ which are real analytic and thus admit a Taylor expansion in x and y. In Appendix A, we show that this assumption also holds for the choices of flux functions $\boldsymbol{f}_S$ considered in this work, assuming the solution satisfies physical bound constraints. However, we note that this assumption is not required in the computation implementations of the proposed methods, and may be an overly restrictive condition on $\boldsymbol{f}_S$.

We now introduce a discretization for the nonlinear system of conservation laws (1) on $[-1,1]$: find $\boldsymbol{u} \in V_h$ such that

$$\left(\frac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{w}\right)_{\widehat{D}} + \left(\left.\left(2D_N \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\right)\right|_{y=x}, \boldsymbol{w}\right)_{\widehat{D}} = \left\langle \boldsymbol{f}^* - \boldsymbol{f}(\boldsymbol{u}_{\boldsymbol{v}}), \boldsymbol{w}\right\rangle_{\partial \widehat{D}}, \qquad \boldsymbol{w} \in V_h, \tag{10}$$

where $\boldsymbol{f}^*$ is a numerical flux used to impose boundary conditions weakly, and $\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y)$ are evaluations of $\boldsymbol{u}$ in terms of the projections of entropy variables $\boldsymbol{v}(\boldsymbol{u}(x))$

$$\boldsymbol{u}_{\boldsymbol{v}}(x) = \boldsymbol{u}\left((\Pi_N \boldsymbol{v})(x)\right). \tag{11}$$

Theorem 2. *The formulation (10) is semi-discretely entropy conservative in the sense that*

$$\left(\frac{\partial U(\boldsymbol{u})}{\partial t}, 1\right)_{\widehat{D}} = \langle \psi, 1\rangle_{\partial \widehat{D}} - \langle \boldsymbol{f}^*, \Pi_N \boldsymbol{v}\rangle_{\partial \widehat{D}}.$$

Proof. We take the test function $\boldsymbol{w}$ to be the projection of the entropy variables. On the left hand side, we may use the fact that, under a method of lines discretization,

$$\boldsymbol{u}(x,t) = \sum_{j=1}^{N_p} \boldsymbol{u}_j(t)\phi_j(x), \qquad \frac{\partial \boldsymbol{u}(x,t)}{\partial t} \in V_h.$$

Thus, testing with $\Pi_N \boldsymbol{v}$ and assuming continuity in time yields

$$\left(\frac{\partial \boldsymbol{u}}{\partial t}, \Pi_N \boldsymbol{v}(\boldsymbol{u})\right)_{\widehat{D}} = \left(\frac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{v}(\boldsymbol{u})\right)_{\widehat{D}} = \left(\frac{\partial U(\boldsymbol{u})}{\partial t}, 1\right)_{\widehat{D}}, \qquad \forall \boldsymbol{u} \in V_h.$$

For the spatial derivative, we expand $\boldsymbol{f}_S$ out using Assumption 1 and use Lemma 1

$$\begin{aligned}
&\left(\left.\left(D_N \sum_{i=1}^{\infty} \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}(x))\boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}}(y))\right)\right|_{y=x}, \Pi_N \boldsymbol{v}(x)\right)_{\widehat{D}} = \sum_{i=1}^{\infty} \left(D_N \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}), \boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}})\Pi_N \boldsymbol{v}\right)_{\widehat{D}} \\
&= \sum_{i=1}^{\infty} \langle \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}})\boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}}), \Pi_N \boldsymbol{v}\rangle_{\partial \widehat{D}} - \left(\boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}), D_N\left(\boldsymbol{g}_i \Pi_N \boldsymbol{v}\right)\right)_{\widehat{D}} \\
&= \langle \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}, \boldsymbol{u}_{\boldsymbol{v}}), \Pi_N \boldsymbol{v}(x)\rangle_{\partial \widehat{D}} - \left(\left.\left(D_N\left(\boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\Pi_N \boldsymbol{v}(x)\right)\right)\right|_{y=x}, 1\right)_{\widehat{D}} \\
&= \langle \boldsymbol{f}(\boldsymbol{u}_{\boldsymbol{v}}), \Pi_N \boldsymbol{v}(x)\rangle_{\partial \widehat{D}} - \left(\left.\left(D_N\left(\boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\Pi_N \boldsymbol{v}(x)\right)\right)\right|_{y=x}, 1\right)_{\widehat{D}},
\end{aligned}$$

where we have used the consistency of $\boldsymbol{f}_S$ in the last step. Noting that

$$\left.\left(D_N \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\right)\right|_{y=x} \boldsymbol{v}(x) = \left.\left(D_N\left(\boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\boldsymbol{v}(y)\right)\right)\right|_{y=x},$$

the volume terms can be manipulated to yield

$$
\begin{aligned}
&\left(\left.\left(D_N \left(\boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y)) \left(\Pi_N \boldsymbol{v}(y) - \Pi_N \boldsymbol{v}(x)\right)\right)\right)\right|_{y=x}, 1 \right)_{\widehat{D}} \\
&= \left(\left.\left(D_N \left(\psi(y) - \psi(x)\right)\right)\right|_{y=x}, 1 \right)_{\widehat{D}} = \left(-D_N \psi(x), 1\right)_{\widehat{D}} = \left\langle -\psi, 1 \right\rangle_{\partial \widehat{D}},
\end{aligned}
$$

where we have used Definition 1 in going from the first to second integral and Lemma 1 in the final step. □

We see that, in the proof of Theorem 2, recovering the time derivative of entropy on the left hand side requires testing with the projection of the entropy variables. For collocation-type DG methods, L^2 projection and interpolation are equivalent, which implies that evaluating the flux $\boldsymbol{f}_S$ using point-wise nodal values of the solution satisfies the entropy conservation condition given in Definition 1. For diagonal norm collocation-type DG methods, the method can also be simplified by noting that the projection operator reduces to a nodal interpolation operator.

In general, the projection of the entropy variables does not agree with their point-wise evaluation. However, the entropy conservation condition given in Definition 1 holds so long as the flux is evaluated in a manner consistent with the entropy variables. This motivates (11) as a non-standard way of evaluating the flux, which is the primary difference between the construction of entropy conservative schemes for diagonal norm collocation-type DG methods and projection-based DG methods.

Remark. *In the proof of Theorem 2, Assumption 1 is used only to formally apply Lemma 1. The use of this assumption can be avoided by proving instead a discrete conservation of entropy in terms of a fixed quadrature rule and the quadrature-based matrices introduced in Section 4.3.*

4 Entropy conservative DG methods on multiple elements

In this section, we discuss the extension of entropy conservative DG schemes to multiple elements. The primary difficulty in this case is the construction of appropriate interface (SBP-SAT) terms. Entropy stable formulations based on dense norm or GSBP operators require SBP-SAT terms which are either non-trivial to derive or have a wide stencil [34, 33]. We present an approach which instead utilizes a global approach to flux differencing to derive entropy conservative interface terms, and show that it recovers existing energy stable SBP-SAT terms for Burgers' equation.

4.1 Global DG differentiation operators

The construction of energy and entropy conservative DG-SBP methods requires the specification of appropriate SBP-SAT terms, which are often derived by reverse-engineering boundary terms such that interface contributions cancel [34, 35, 33]. We present an alternative systematic approach to the derivation of interface terms for DG formulations by combining flux differencing with global discrete DG differential operators, which have been used to simplify the notation and analysis of DG formulations for elliptic equations [36, 37]. This approach does not require the specification of appropriate SBP-SAT terms in order to prove entropy conservation; rather, explicit expressions for SBP-SAT terms may be derived in a systematic manner using flux differencing and the definitions of global DG operators.

We adapt the construction of local differentiation operators introduced previously to construct a global DG differentiation operator

$$
\left(D_h^x u, vw\right)_{\Omega} = \sum_k \left(\frac{\partial \Pi_N u}{\partial x}, vw\right)_{D^k} + \frac{1}{2}\left\langle u^+ - \Pi_N u, vw\boldsymbol{n}_x\right\rangle_{\partial D^k} + \frac{1}{2}\left\langle u - \Pi_N u, \Pi_N(vw)\boldsymbol{n}_x\right\rangle_{\partial D^k}, \tag{12}
$$

where $w(x)$ is a bounded smooth function over each element and $\left(D_h^x u, v\right)_{\Omega} = \sum_k \left(D_h^x u, v\right)_{D^k}$. Entropy conservative DG schemes can be constructed using the fact that D_h^x satisfies a *global* analogue of integration by parts:

Lemma 2. *Let D_h^x be defined as in (12), and define $v^+ = v$ on $\partial\Omega$. Then,*

$$\left(D_h^x u, vw\right)_\Omega = \left\langle u, vw\boldsymbol{n}_x\right\rangle_{\partial\Omega} - \left(u, D_h^x(vw)\right)_\Omega .$$

for $u \in H^1(\Omega_h)$ and $\forall v \in V_h(\Omega_h)$.

Proof. We may rewrite $\left(D_h^x u, vw\right)_\Omega$ using the definition of D_N in (8)

$$\left(D_h^x u, vw\right)_\Omega = \sum_k \left(D_N u, vw\right)_{D^k} + \frac{1}{2}\left\langle [\![u]\!], vw\boldsymbol{n}_x\right\rangle_{\partial D^k \setminus \partial\Omega} .$$

Applying Lemma 1 over each element yields

$$\begin{aligned}
\left(D_h^x u, vw\right)_\Omega &= \sum_k \left(-u, D_N(vw)\right)_{D^k} + \left\langle u + \frac{1}{2}[\![u]\!], vw\boldsymbol{n}_x\right\rangle_{\partial D^k} \\
&= \sum_k \left(-u, D_N(vw)\right)_{D^k} + \left\langle \{\{u\}\}, vw\boldsymbol{n}_x\right\rangle_{\partial D^k} \\
&= \sum_k \left(-u, D_N(vw)\right)_{D^k} - \left\langle [\![vw]\!], u\boldsymbol{n}_x\right\rangle_{\partial D^k \setminus \partial\Omega} + \left\langle vw, u\boldsymbol{n}_x\right\rangle_{\partial\Omega} \\
&= \left\langle vw, u\boldsymbol{n}_x\right\rangle_{\partial\Omega} - \left(u, D_h^x(vw)\right)_\Omega .
\end{aligned}$$

□

Remark. *It is possible to define a simpler global DG differentiation operator based on the composition of a discrete differentiation operator and the L^2 projection. Let $D_h^x : V_h \to V_h$ be defined as*

$$\left(D_h^x u, v\right)_\Omega = \sum_k \left(\frac{\partial u}{\partial x}, v\right)_{D^k} + \frac{1}{2}\left\langle [\![u]\!], v\boldsymbol{n}_x\right\rangle_{\partial D^k}, \qquad \forall v \in V_h.$$

Then, using that the range of D_h^x is V_h, it is straightforward to show that $D_h^x\Pi_N$ satisfies an analogue of integration by parts, such that

$$\left(D_h^x \Pi_N u, v\right)_\Omega = \left\langle \Pi_N u, v\right\rangle_{\partial\Omega} - \left(D_h^x \Pi_N v, u\right)_\Omega, \qquad \forall u, v \in L^2(\Omega) .$$

Unfortunately, while (12) requires only traces of neighboring elements, this construction results in a wide stencil involving all degrees of freedom on neighboring elements due to the presence of $[\![u]\!]$ and the L^2 projection. This choice of global DG differentiation operator is analogous to the choice of SBP-SAT terms specified in [34] for SBP operators on simplicial elements without boundary nodes.

4.2 Conservation of entropy and entropy stability on multiple elements

Extending the entropy stable DG formulation (10) is done by combining the flux differencing with the global DG differential operator D_h^x

$$\sum_k \left(\frac{\partial \boldsymbol{u}}{\partial t}, \boldsymbol{w}\right)_{D^k} + \left(\left(2D_h^x \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\right)\big|_{y=x}, \boldsymbol{w}\right)_{D^k} = \left\langle \boldsymbol{f}(\boldsymbol{u}_{\boldsymbol{v}}) - \boldsymbol{f}^*, \boldsymbol{w}\right\rangle_{\partial\Omega}, \tag{13}$$

where $\boldsymbol{f}^*$ is some numerical flux used in the imposition of boundary conditions, and the global operator D_h^x is understood to be applied to each component of $\boldsymbol{f}_S(\boldsymbol{u}(x), \boldsymbol{u}(y))$.

This formulation yields a theorem analogous to Theorem 2:

Theorem 3. *The formulation (13) is globally entropy conservative in the sense that*

$$\left(\frac{\partial U(\boldsymbol{u})}{\partial t}, 1\right)_\Omega = \left\langle \psi, 1\right\rangle_{\partial\Omega} - \left\langle \boldsymbol{f}^*, \Pi_N \boldsymbol{v}\right\rangle_{\partial\Omega} .$$

Proof. The proof is identical to the proof of Theorem 2, except that Lemma 2 is used in place of Lemma 1. □

Remark. *The presented analysis focuses on discrete entropy conservation in one dimension. However, the extension to two and three dimensions is straightforward (as shown in [6]), based on analogous definitions of global DG operators in higher dimensions. The construction of higher dimensional DG operators differ from the one-dimensional case in that the evaluation of surface integrals also requires appropriately accurate quadrature rules over the reference element, such that integration by parts holds for elements of the local approximation space.*

Finally, we note that the analysis in previous sections has focused on the construction of entropy conservative schemes. However, entropy is only conserved for smooth solutions and should be dissipated away in the presence of discontinuities and shocks. To this end, one can construct discretely entropy stable schemes by adding additional dissipative terms (for example, by adding HLL or Lax-Friedrichs penalization [6] or matrix dissipation terms [38, 24]). For the numerical experiments presented here, we apply a Lax-Friedrichs penalization, augmenting the flux function at element interfaces with the additional term

$$\boldsymbol{f}_S\left(\boldsymbol{u}_L, \boldsymbol{u}_R\right) \Rightarrow \boldsymbol{f}_S\left(\boldsymbol{u}_L, \boldsymbol{u}_R\right) - \frac{\lambda}{2}[\![\boldsymbol{u}_{\boldsymbol{v}}]\!],$$

where λ is an estimate of the maximum eigenvalue of the flux Jacobian. For an appropriate approximation of λ, Lax-Friedrichs penalization dissipates entropy [6]. We note that, since conservation of entropy requires testing with the projection of the entropy variables, the jump term should involve the evaluations of the conservation variables in terms of the projected entropy variables $\boldsymbol{u}_{\boldsymbol{v}} = \boldsymbol{u}\left(\Pi_N \boldsymbol{v}\right)$ in order to guarantee entropy dissipation.

4.2.1 Local conservation

It is also straightforward to show that the formulation (13) is locally conservative. We follow [6] and take $\boldsymbol{w} = \mathbb{1}_{D^k}$ in the formulation (13)

$$\sum_k \left(\frac{\partial \boldsymbol{u}}{\partial t}, 1\right)_{D^k} + \left(\left(2D_h^x \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\right)\big|_{y=x}, 1\right)_{D^k} = \left\langle \boldsymbol{f}(\boldsymbol{u}_{\boldsymbol{v}}) - \boldsymbol{f}^*, 1\right\rangle_{\partial\Omega \cap D^k}.$$

Using Assumption 1, Lemmas 1 and 2, and the symmetry of $\boldsymbol{f}_S$, one can show that

$$\begin{aligned}
\left(\left(2D_h^x \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\right)\big|_{y=x}, 1\right)_{D^k} &= \sum_{i=1}^{\infty} \left(2D_N \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}), \boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}})\right)_{D^k} + \left\langle [\![\boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}})]\!], \boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}})\right\rangle_{\partial D^k} \\
&= \sum_{i=1}^{\infty} \left(D_N \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}), \boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}})\right)_{D^k} - \left(D_N \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}), \boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}})\right)_{D^k} \\
&\quad + \left\langle \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}), \boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}})\right\rangle_{\partial D^k} + \left\langle \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}^+) - \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}), \boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}})\right\rangle_{\partial D^k} \\
&= \left(D_N \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y)), 1\right)_{D^k} - \left(D_N \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y)), 1\right)_{D^k} \\
&\quad + \left\langle \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}^+, \boldsymbol{u}_{\boldsymbol{v}}), 1\right\rangle_{\partial D^k}.
\end{aligned}$$

The volume terms cancel, leaving

$$\sum_k \left(\frac{\partial \boldsymbol{u}}{\partial t}, 1\right)_{D^k} + \left\langle \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}^+, \boldsymbol{u}_{\boldsymbol{v}}), 1\right\rangle_{\partial D^k} = \left\langle \boldsymbol{f}(\boldsymbol{u}_{\boldsymbol{v}}) - \boldsymbol{f}^*, 1\right\rangle_{\partial\Omega \cap D^k}.$$

On the domain boundary $\partial\Omega$, $\boldsymbol{u}_{\boldsymbol{v}}^+ = \boldsymbol{u}_{\boldsymbol{v}}$ and $\boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}^+, \boldsymbol{u}_{\boldsymbol{v}}) = \boldsymbol{f}(\boldsymbol{u}_{\boldsymbol{v}})$. This implies that the statement of local conservation replaces $\boldsymbol{f}_S$ with the boundary condition numerical flux $\boldsymbol{f}^*$ on element boundaries which intersect with the domain boundary.

We note that this definition of local conservation is given in terms of the conservation variables evaluated as a function of the projected entropy variables

$$\boldsymbol{u}_{\boldsymbol{v}}(x)(x) = \boldsymbol{u}\left(\left(\Pi_N \boldsymbol{v}\right)(x)\right).$$

It is not immediately clear whether this statement of local conservation satisfies the conditions of the classic Lax-Wendroff theorem. However, this form of local conservation does satisfy the generalized definition of local conservation given in [39], which is sufficient to guarantee convergence to the weak solution.

4.2.2 Recovery of known SBP-SATs for Burgers' equation

To illustrate the application of the global flux differencing approach, we derive explicit expressions for an energy stable formulation for Burgers' equation. We will assume periodic boundary conditions for simplicity and use the energy conserving flux

$$f_S(u_L, u_R) = \frac{1}{6}\left(u_L^2 + u_L u_R + u_R^2\right).$$

We focus on the discretization of the spatial terms. Applying flux differencing to the flux to yield the following contributions

$$\left(2D_h^x f_S(u(x), u(y))\big|_{y=x}, v\right) = \frac{1}{3}\left(D_h^x\left(u(x)^2 + u(x)u(y) + u(y)^2\right)\big|_{y=x}, v\right) = \frac{1}{3}\left(D_h^x\left(u^2\right) + uD_h^x u, v\right).$$

Expanding out the definition of $D_h^x\left(u^2\right)$ gives

$$\begin{aligned}\left(D_h^x\left(u^2\right), v\right)_\Omega &= \sum_k \left(\frac{\partial \Pi_N u^2}{\partial x}, v\right)_{D^k} + \frac{1}{2}\left\langle \left(u^+\right)^2 - \Pi_N u^2, v\boldsymbol{n}_x\right\rangle_{\partial D^k} + \frac{1}{2}\left\langle u^2 - \Pi_N u^2, \Pi_N v\boldsymbol{n}_x\right\rangle_{\partial D^k} \\ &= \sum_k \left(\frac{\partial \Pi_N u^2}{\partial x}, v\right)_{D^k} + \left\langle \frac{1}{2}\left(\left(u^+\right)^2 + u^2\right) - \Pi_N u^2, v\boldsymbol{n}_x\right\rangle_{\partial D^k}.\end{aligned}$$

Similarly, expanding out $\left(uD_h^x u, v\right)_\Omega$ gives

$$\begin{aligned}\left(uD_h^x u, v\right)_\Omega = \left(D_h^x u, uv\right)_\Omega &= \sum_k \left(\frac{\partial u}{\partial x}, uv\right)_{D^k} + \frac{1}{2}\left\langle u^+ - \Pi_N u, uv\boldsymbol{n}_x\right\rangle_{\partial D^k} + \frac{1}{2}\left\langle u - \Pi_N u, \Pi_N (uv)\right\rangle_{\partial D^k} \\ &= \sum_k \left(\frac{\partial u}{\partial x}, uv\right)_{D^k} + \frac{1}{2}\left\langle u^+ u - u^2, v\boldsymbol{n}_x\right\rangle_{\partial D^k}.\end{aligned}$$

Summing these together yields explicit expressions for the DG weak formulation

$$\sum_k \left(\frac{\partial u}{\partial t} + \frac{1}{3}\left(\frac{\partial \Pi_N u^2}{\partial x} + u\frac{\partial u}{\partial x}\right), v\right)_{D^k} + \left\langle f_S(u^+, u) - \frac{1}{3}\Pi_N u^2 - \frac{1}{6}u^2, v\boldsymbol{n}_x\right\rangle_{\partial D^k} = 0.$$

This formulation, which was derived through the use of flux differencing with the global DG operator (12), recovers the weak form of the split formulation and the correction terms introduced in [33] for Burgers' equation.

4.3 On implementation in terms of the flux function

While it is possible to use (13) to derive explicit DG weak formulations, this can be difficult for complicated flux functions. Instead, we follow [3, 6] and seek an implementation of flux differencing and (13) which requires only the specification of $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ and local matrices.

We begin by rewriting the global DG differentiation operator D_h^x using the lifting operator $L : V_h \to V_h$ [7, 36]

$$\left(Lu, v\right)_{D^k} = \left\langle u, v\right\rangle_{\partial D^k}, \qquad u \in V_h, \qquad \forall v \in V_h.$$

The lift operator can be used to rewrite surface terms which involve the volume L^2 projection

$$\left\langle u, \Pi_N v\right\rangle_{\partial D^k} = \left(Lu, \Pi_N v\right)_{D^k} = \left(Lu, v\right)_{D^k}, \qquad u \in V_h, \qquad v \notin V_h.$$

Then, the global DG differentiation operator can be rewritten as

$$\left(D_h^x u, v\right)_\Omega = \sum_k \left(\frac{\partial \Pi_N u}{\partial x} + \frac{1}{2} L\left(u - \Pi_N u\right)\boldsymbol{n}_x, v\right)_{D^k} + \frac{1}{2}\left\langle u^+ - \Pi_N u, v\boldsymbol{n}_x\right\rangle_{\partial D^k}. \tag{14}$$

Applying Assumption 1, we can rewrite

$$\begin{aligned}
\left(\left.\left(D_h^x \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\right)\right|_{y=x}, \boldsymbol{w}\right)_\Omega = &\sum_k \sum_{i=1}^{\infty} \left(\frac{\partial \Pi_N \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}})}{\partial x}, \boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}})\boldsymbol{w}\right)_{D^k} \\
&+ \sum_k \sum_{i=1}^{\infty} \frac{1}{2}\left\langle \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}^+)\boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}}) - \boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}})\Pi_N \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}), \boldsymbol{w}\boldsymbol{n}_x\right\rangle_{\partial D^k} \\
&+ \sum_k \sum_{i=1}^{\infty} \frac{1}{2}\left(\boldsymbol{g}_i(\boldsymbol{u}_{\boldsymbol{v}}) L\left(\boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}}^+) - \Pi_N \boldsymbol{f}_i(\boldsymbol{u}_{\boldsymbol{v}})\right)\boldsymbol{n}_x, \boldsymbol{w}\right)_{D^k},
\end{aligned}$$

where $\boldsymbol{w}\boldsymbol{n}_x$ denotes component-wise multiplication of $\boldsymbol{w}$ by $\boldsymbol{n}_x$, the x-component of the unit outward normal. Moving the sum inside and simplifying then allows us to re-write the resulting expression as

$$\begin{aligned}
\left(\left.\left(D_h^x \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\right)\right|_{y=x}, \boldsymbol{w}\right)_\Omega = &\sum_k \left(\left.\frac{\partial \Pi_N \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))}{\partial x}\right|_{y=x}, \boldsymbol{w}\right)_{D^k} \\
&+ \frac{1}{2}\left\langle \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}^+, \boldsymbol{u}_{\boldsymbol{v}}) - \left.\Pi_N^x \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\right|_{y=x}, \boldsymbol{w}\boldsymbol{n}_x\right\rangle_{\partial D^k} \\
&+ \frac{1}{2}\left(\left.\left(L^x\left(\boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y)) - \Pi_N^x \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\right)\boldsymbol{n}_x\right)\right|_{y=x}, \boldsymbol{w}\right)_{D^k},
\end{aligned} \tag{15}$$

where the superscripts L^x, Π_N^x are used to emphasize that the lift and projection operators are applied in x and not in y. This can be used to practically evaluate $\left.\left(\boldsymbol{D}_h^x \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\right)\right|_{y=x}$ using only reference operators and evaluations of $\boldsymbol{f}_S$. However, unlike previous entropy conservative collocation DG methods, the boundary terms in (15) interact with volume terms through the L^2 projection and the lift operator. This can be simplified for diagonal norm SBP operators with boundary nodes, using equivalences between L^2 projection and interpolation at nodal points in those cases.

Finally, we discuss how to recover a computational form of flux differencing analogous to the approaches presented in [3, 6]. We will describe how to compute the three non-standard terms which appear within (15):

$$\left(\left.\frac{\partial \Pi_N f_S(u(x), u(y))}{\partial x}\right|_{y=x}, w\right)_{D^k} \tag{16}$$

$$\left\langle f_S(u^+, u) - \left.\Pi_N^x f_S(u(x), u(y))\right|_{y=x}, w\boldsymbol{n}_x\right\rangle_{\partial D^k} \tag{17}$$

$$\left(\left.\left(L^x\left(f_S(u(x), u(y)) - \Pi_N^x f_S(u(x), u(y))\right)\boldsymbol{n}_x\right)\right|_{y=x}, w\right)_{D^k} \tag{18}$$

where we have assumed a scalar variable $u(x)$ and flux function f_S for simplicity.

We first introduce matrices defined over $[-1, 1]$. We assume some discrete quadrature rule on the reference element consisting of N_q points $\boldsymbol{x}_q = \left\{x_1, \ldots, x_{N_q}\right\}$ and quadrature weights $\boldsymbol{w}_q = \left\{w_1, \ldots, w_{N_q}\right\}$. Let $\boldsymbol{W}$

denote the diagonal matrix whose entries are quadrature weights. Then, let $u(x)$ be represented in some basis $\phi_j(x)$ of dimension N_p which spans V_h

$$V_h = \operatorname{span}\left\{\phi_i(\widehat{x})\right\}_{i=1}^{N_p},$$

we can define the quadrature interpolation matrix $\boldsymbol{V}_q$

$$\left(\boldsymbol{V}_q\right)_{ij} = \phi_j(\widehat{x}_i), \qquad 1 \leq j \leq N_p, \qquad 1 \leq i \leq N_q.$$

Given $\boldsymbol{V}_q$, we introduce the mass matrix $\boldsymbol{M} = \boldsymbol{V}_q^T \boldsymbol{W} \boldsymbol{V}_q$, as well as the quadrature-based L^2 projection operator $\boldsymbol{P}_q$

$$\boldsymbol{P}_q = \boldsymbol{M}^{-1} \boldsymbol{V}_q^T \boldsymbol{W},$$

which maps a function (in terms of its evaluation at quadrature points) to coefficients of the basis $\phi_j(x)$. Next, let $\boldsymbol{D}_x$ denote the differentiation matrix defined implicitly through

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

In other words, $\boldsymbol{D}_x$ maps basis coefficients of some function $u \in V_h$ to coefficients of its derivative on $[-1, 1]$.

We first describe how to evaluate the volume term (16) using these matrices. Let $\boldsymbol{F}_S$ denote the $N_q \times N_q$ matrix constructed by evaluating f_S at volume quadrature points

$$\left(\boldsymbol{F}_S\right)_{ij} = f_S(u(x_i), u(x_j)).$$

Then, integrating (16) using quadrature is equivalent to computing

$$\left(\left. \frac{\partial \Pi_N f_S(u(x), u(y))}{\partial x} \right|_{y=x}, w \right)_{D^k} = \boldsymbol{w}^T \boldsymbol{V}_q^T \boldsymbol{W} \operatorname{diag}\left(\boldsymbol{V}_q \boldsymbol{D}_x \boldsymbol{P}_q \boldsymbol{F}_S\right),$$

where $\boldsymbol{w}$ denote the vector of coefficients for the weighting function $w(x)$.

The diagonal operation can be replaced by a more convenient evaluation. Let $\boldsymbol{A} \circ \boldsymbol{B}$ denote the Hadamard product of two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{N_q \times N_q}$, such that

$$\left(\boldsymbol{A} \circ \boldsymbol{B}\right)_{ij} = \boldsymbol{A}_{ij} \boldsymbol{B}_{ij}, \qquad 1 \leq i, j \leq N_q.$$

The Hadamard product of two matrices can also be used to conveniently represent the diagonal of a product of matrices [40]:

$$\operatorname{diag}\left(\boldsymbol{A}\boldsymbol{B}^T\right) = \left(\boldsymbol{A} \circ \boldsymbol{B}\right) \boldsymbol{1}.$$

Using this property and the symmetry of $\boldsymbol{F}_S$, we can rewrite the diagonal operation as follows

$$\operatorname{diag}\left(\boldsymbol{V}_q \boldsymbol{D}_q \boldsymbol{P}_q \boldsymbol{F}_S\right) = \left(\left(\boldsymbol{V}_q \boldsymbol{D}_q \boldsymbol{P}_q\right) \circ \boldsymbol{F}_S\right) \boldsymbol{1}.$$

A computational advantage of evaluation in terms of the Hadamard product is that $\boldsymbol{F}_S$ does not need to be formed explicitly and stored.

The same approach may also be applied to (17) and (18). We focus first on (17) and introduce matrices used to compute boundary terms. Let $\boldsymbol{V}_f$ denote the matrix which interpolates to boundary values[2]

$$\left(\boldsymbol{V}_f\right)_{1j} = \phi_j(-1), \qquad \left(\boldsymbol{V}_f\right)_{2j} = \phi_j(1), \qquad 1 \leq j \leq N_p.$$

Furthermore, let $\boldsymbol{W}_f$ denote the diagonal matrix of surface quadrature weights. In one dimension, this is simply the 2×2 identity matrix. We now define $\boldsymbol{F}_S^f$ to be the matrix constructed by evaluating $\boldsymbol{F}_S$ at both volume quadrature and boundary points

$$\left(\boldsymbol{F}_S^f\right)_{ij} = f_S\left(u(x_i), u\left(x_j^f\right)\right), \qquad 1 \leq i \leq N_q, \quad j = 1, 2.$$

[2] In generalizing to higher dimensions, $\boldsymbol{V}_f$ would be the matrix which interpolates to boundary points.

We also define $\boldsymbol{F}_S^+$ to be the vector of evaluations of the flux function $f_S\left(u^+,u\right)$ at the boundary points of D^k. Using these matrices, we can then evaluate

$$\begin{aligned}\left\langle f_S(u^+,u) - \Pi_N^x f_S(u(x),u(y))|_{y=x}\,, w\boldsymbol{n}_x\right\rangle_{\partial D^k} &= \boldsymbol{w}^T\boldsymbol{V}_f^T\boldsymbol{W}_f\operatorname{diag}(\boldsymbol{n}_x)\left(\boldsymbol{F}_S^+ - \operatorname{diag}\left(\boldsymbol{V}_f\boldsymbol{P}_q\boldsymbol{F}_S^f\right)\right)\\ &= \boldsymbol{w}^T\boldsymbol{V}_f^T\boldsymbol{W}_f\operatorname{diag}(\boldsymbol{n}_x)\left(\boldsymbol{F}_S^+ - \left((\boldsymbol{V}_f\boldsymbol{P}_q)\circ\left(\boldsymbol{F}_S^f\right)^T\right)\boldsymbol{1}\right).\end{aligned}$$

This step can be interpreted as coupling together evaluations of the flux function at volume and boundary points. We note that this is the only step in evaluating (15) which requires non-local data (to compute $\boldsymbol{F}_S^+$).

In order to evaluate the final term (18), we need to introduce the quadrature-based lift matrix

$$\boldsymbol{L} = \boldsymbol{M}^{-1}\boldsymbol{V}_f^T\boldsymbol{W}_f,$$

which (as the name implies) "lifts" a function from the boundary (in terms of its evaluation at surface quadrature points) to coefficients of a basis defined on the interior of the element. Then, we have that

$$\begin{aligned}&\left(\left(L^x\left(f_S(u(x),u(y)) - \Pi_N^x f_S(u(x),u(y))\right)\boldsymbol{n}_x\right)|_{y=x}\,, w\right)_{D^k}\\ &= \boldsymbol{w}^T\boldsymbol{V}_q^T\boldsymbol{W}\operatorname{diag}\left(\boldsymbol{V}_q\boldsymbol{L}\operatorname{diag}(\boldsymbol{n}_x)\left(\left(\boldsymbol{F}_S^f\right)^T - \boldsymbol{V}_f\boldsymbol{P}_q\boldsymbol{F}_S^f\right)\right)\\ &= \boldsymbol{w}^T\boldsymbol{V}_q^T\boldsymbol{W}\left((\boldsymbol{V}_q\boldsymbol{L}\operatorname{diag}(\boldsymbol{n}_x))\circ\boldsymbol{F}_S^f - (\boldsymbol{V}_q\boldsymbol{L}\operatorname{diag}(\boldsymbol{n}_x)\boldsymbol{V}_f\boldsymbol{P}_q)\circ\boldsymbol{F}_S\right)\boldsymbol{1}.\end{aligned}$$

This expression is (in a loose sense) analogous to the evaluation of (17) in that it couples together volume and boundary evaluations of the flux function, but in the reverse order.

It is worth pointing out that these sets of expressions can be simplified; for example, straightforward computations show that $\boldsymbol{W}\boldsymbol{V}_q\boldsymbol{L} = \left(\boldsymbol{W}_f\operatorname{diag}(\boldsymbol{n}_x)\boldsymbol{V}_f\boldsymbol{P}_q\right)^T$, and the computation of the volume term $\left((\boldsymbol{V}_q\boldsymbol{D}_q\boldsymbol{P}_q)\circ\boldsymbol{F}_S\right)\boldsymbol{1}$ and the surface term $\left((\boldsymbol{V}_q\boldsymbol{L}\operatorname{diag}(\boldsymbol{n}_x)\boldsymbol{V}_f\boldsymbol{P}_q)\circ\boldsymbol{F}_S\right)\boldsymbol{1}$ can be combined into one single operation. Furthermore, for specific choices of basis and quadrature, one can eliminate certain terms altogether. For example, the co-location of nodal and quadrature points (assuming that the dimension of the approximation space is identical to the number of quadrature points) reduces $\boldsymbol{V}_q, \boldsymbol{P}_q$ to identity matrices. Furthermore, if quadrature points coincide with boundary points (as with DG-SEM methods), the lift matrix $\boldsymbol{L}$ is zero except for entries which correspond to those boundary points. These assumptions greatly simplify both the formulation and implementation of entropy conservative/stable DG methods, and will be discussed more thoroughly in a future manuscript.

5 Numerical experiments

In this section, we illustrate the entropy conservation and accuracy of the proposed schemes for the one dimensional compressible Euler equations. All numerical experiments utilize the fourth order five-stage low-storage Runge-Kutta method Carpenter and Kennedy [42]. Finally, following the derivation of stable timestep restrictions in [10], we define the timestep Δt to be

$$\Delta t = \text{CFL}\times\frac{h}{C_N}, \qquad C_N = \frac{(N+1)^2}{2},$$

where C_N is the one-dimensional constant in the trace inequality [41], and CFL is a user-defined constant.

5.1 Compressible Euler equations

The one-dimensional compressible Euler equations correspond to the inviscid limit of the compressible Navier-Stokes equations and are given as follows:

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

We assume an ideal gas, such that the pressure satisfies the constitutive relation $p=(\gamma-1)\left(E-\frac{1}{2}\rho u^2\right)$, where $\gamma=1.4$ is the ratio of specific heat for a diatomic gas.

The choice of convex entropy for the Euler equations is non-unique [43]. 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 [26]. This leads to $U(\boldsymbol{u})$ of the form

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

where $s=\log\left(\frac{p}{\rho^\gamma}\right)$ is the physical specific entropy. The entropy variables under this choice of entropy are then

$$q_1=\frac{\gamma-s}{\gamma-1}-\frac{\rho u^2}{2p},\qquad q_2=\frac{\rho u}{p},\qquad q_3=-\frac{\rho}{p}.$$

The inverse mapping is given by

$$\rho=-(\rho e)q_3,\qquad m=(\rho e)v_2,\qquad E=(\rho e)\left(1-\frac{q_2^2}{2q_3}\right),$$

where ρe and s in terms of the entropy variables are

$$\rho e=\left(\frac{(\gamma-1)}{\left(-q_3\right)^\gamma}\right)^{1/(\gamma-1)}e^{\frac{-s}{\gamma-1}},\qquad s=\gamma-q_1+\frac{q_2^2}{2q_3}.$$

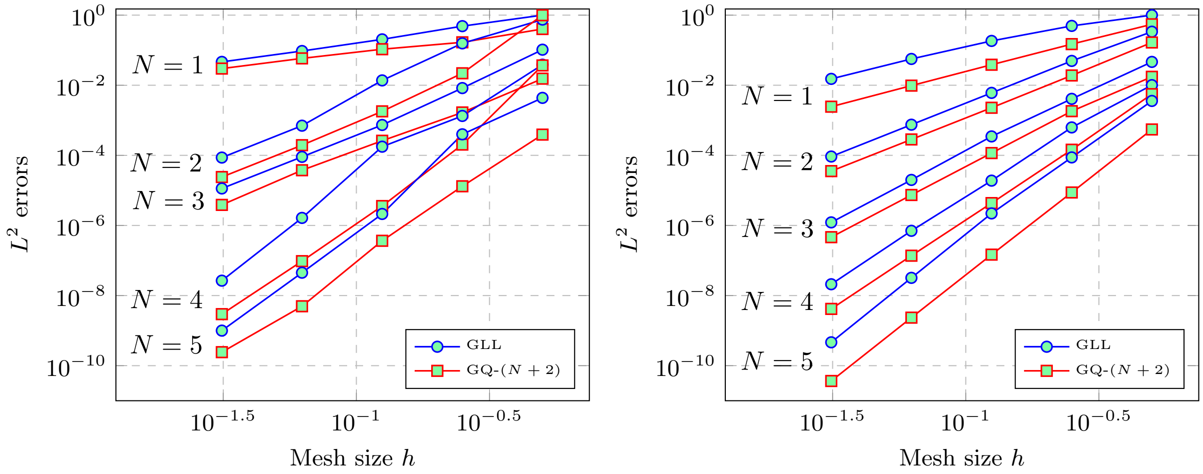
In order for the entropy $U(\boldsymbol{u})$ to be well defined, we require the assumption that the discrete density and pressure solutions are bounded away from zero

$$\rho\geq\rho_0>0,\qquad p\geq p_0>0. \tag{19}$$

This can be achieved using positivity-preserving limiters [31, 32]. These were not found to be necessary for the experiments in this work, and have thus not been implemented in our numerical simulations.

Examples of entropy conservative flux functions can be found in [44, 38]. In this work, we utilize the flux function $\boldsymbol{f}_S(\boldsymbol{u}_L,\boldsymbol{u}_R)$ introduced by Chandreshekar [38], whose components are given as

$$\begin{aligned}f_S^1(\boldsymbol{u}_L,\boldsymbol{u}_R)&=\{\{\rho\}\}^{\log}\{\{u\}\}\\ f_S^2(\boldsymbol{u}_L,\boldsymbol{u}_R)&=\frac{\{\{\rho\}\}}{2\{\{\beta\}\}}+\{\{u\}\}f_S^1\\ f_S^3(\boldsymbol{u}_L,\boldsymbol{u}_R)&=f_S^1\left(\frac{1}{2(\gamma-1)\{\{\beta\}\}^{\log}}-\frac{1}{2}\{\{u^2\}\}\right)+\{\{u\}\}f_S^2,\end{aligned}$$



(a) Entropy conservative flux

(b) With Lax-Friedrichs penalization

Figure 1: L^2 errors under mesh refinement for entropy conservative and Lax-Friedrichs fluxes under both Gauss-Legendre-Lobatto (GLL) and over-integrated $(N + 2)$ point Gauss quadrature (GQ-$(N + 2)$).

where we have introduced the inverse temperature β

$$\beta = \frac{\rho}{2p}$$

and the logarithmic mean

$$\{\{u\}\}^{\log} = \frac{u_L - u_R}{\log u_L - \log u_R}.$$

We note that, because the direct evaluation of the logarithmic mean is numerically sensitive for $u_L \approx u_R$, we switch to evaluation using a high order accurate expansion introduced by Ismail and Roe [44] when $|u_L - u_R| < \epsilon$.

These fluxes are both entropy conservative and kinetic energy preserving. Unlike the shallow water equations, these entropy conservative fluxes do not correspond to stable split formulations of the Euler equations. Thus, the compressible Euler equations serve as a test of the flux differencing formulation, as entropy stability cannot be achieved through skew-symmetry.

We also present results which utilize the dissipative local Lax-Friedrichs flux described in Section 4.2, where the value of λ is estimated by

$$\lambda = \max_{\boldsymbol{u}^+, \boldsymbol{u}} \left\{|u| + c\right\}, \qquad c = \sqrt{\frac{\gamma p}{\rho}}.$$

We begin by verifying the high order accuracy of the proposed methods using a periodic entropy wave solution

$$\rho(x,t) = 2 + \sin\left(\pi(x - t)\right), \qquad u(x,t) = 1, \qquad p(x,t) = 1.$$

We compute the L^2 error in the conservation variables

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2}^2 = \|\rho - \rho_h\|_{L^2}^2 + \|m - m_h\|_{L^2}^2 + \|E - E_h\|_{L^2}^2$$

at final time $T = .7$ using both the entropy conservative and local Lax-Friedrichs fluxes. For these experiments, we compare two quadrature rules: the $(N + 1)$ point Gauss-Legendre-Lobatto rule (which induces a DG-SEM discretization and is referred to in Figure 1 as "GLL") and an $(N+2)$ point Gauss quadrature rule

	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$
GLL (entropy conservative)	1.0672	3.6561	3.0086	6.3486	5.5278
GQ-$(N + 2)$ (entropy conservative)	0.9175	3.1132	3.0388	5.1221	5.2779
GLL (Lax-Friedrichs)	1.8887	3.0164	4.0241	5.0650	6.1095
GQ-$(N + 2)$ (Lax-Friedrichs)	1.9950	3.0018	3.9980	5.0419	6.0034

Table 1: Computed asymptotic convergence rates of L^2 errors under mesh refinement.

(referred to in the figure as "GQ-$(N+2)$"). We do not compare against the $(N+1)$ point Gauss quadrature rule, which behaves very similarly to the GQ-$(N+2)$ case. The L^2 error is evaluated using a more accurate $N+5$ point Gauss quadrature rule, and a CFL of .125 is used.

Figure 1 shows computed L^2 errors under mesh refinement for both GLL and GQ-$(N+2)$ quadrature rules. All experiments utilize the entropy conservative flux of Chandreshekar, and we compare errors with and without Lax-Friedrichs penalization. Computed asymptotic convergence rates are reported in Table 1. It can be observed that, for both GLL and GQ-$(N+2)$ quadratures, the entropy conservative flux exhibits suboptimal convergence rates for odd orders, while the inclusion of Lax-Friedrichs penalization restores exhibits the optimal $O(h^{N+1})$ convergence rate for both quadrature choices.

Next, we examine the discrete evolution of entropy by evolving a discontinuous initial profile to final time $T = 4$. We initialize the density and velocity to be

$$\rho(x,t) = \begin{cases} 3 & \|x\| < 1/2 \\ 2 & \text{otherwise,} \end{cases} \qquad u(x,t) = 0, \qquad p(x,t) = \rho^{\gamma}. \tag{20}$$

Non-physical periodic boundary conditions are enforced in order to examine the evolution of entropy over longer time periods. Figure 2a shows ρ, u at time $T = 1/10$ using both entropy conservative and Lax-Friedrichs fluxes (referred to as "EC" and "LF", respectively in the figure) and GQ-$(N+2)$ quadrature. As expected, the entropy conservative flux results in spurious high frequency oscillations, which are significantly damped under Lax-Friedrichs penalization.

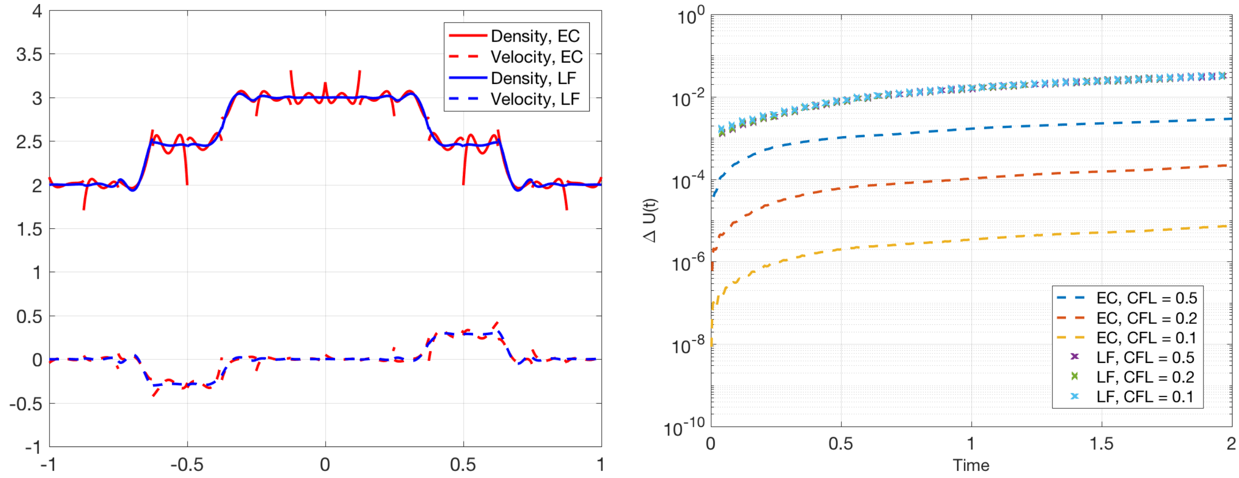
We note that, while the proof of conservation of entropy holds at the semi-discrete level, it does not take into account the time discretization and thus does not hold at the fully discrete level. This is reflected into numerical observation that the change in entropy $\Delta U(t) = U(\boldsymbol{u}(t)) - U(\boldsymbol{u}(t))$ increases as t increases. However, numerical experiments in [4] suggest that the discrete change in entropy over time should converge to zero as the timestep decreases, which (under appropriate assumptions) implies that the fully discrete scheme converges to the semi-discrete scheme.

Figure 2b shows the evolution of $\Delta U(t)$ to final time $T = 2$ for both entropy conservative and Lax-Friedrichs fluxes at various CFL numbers. For the entropy conservative flux, we observe that $\Delta U(t)$ to decreases as the CFL and Δt decrease. This is expected, since the discrete time problem should converge to the continuous semi-discrete problem (for which $\Delta U(t) = 0$) as $\Delta t \to 0$. We also observe that, as expected, $\Delta U(t)$ does not change significantly with CFL for the Lax-Friedrichs flux, indicating that the change in entropy is due to the effect of the dissipative flux rather than the time discretization.

We also numerically evaluate the spatial formulation tested against the projected the entropy variables

$$\delta(t) = \left| \left(\left(D_h^x \boldsymbol{f}_S(\boldsymbol{u}_{\boldsymbol{v}}(x), \boldsymbol{u}_{\boldsymbol{v}}(y))\right)\Big|_{y=x}, \Pi_N \boldsymbol{v} \right)_{\Omega} \right|, \qquad 0 \leq t \leq T.$$

We utilize the non-dissipative entropy conservative flux and evolve the initial profile (20) until time $T = 1$ using a GQ-$(N+2)$ quadrature rule. From Theorem 3, we expect $\delta_{\max} = \max_{t\in(0,T)} \delta$ to be machine zero. In practice, we have found that $\delta_{\max}$ depends on the tolerance ϵ used in evaluation of the logarithmic mean. For a simulation to time $T = 4$ using $N = 4$ and $K = 16$ with a CFL of $1/2$, we observe that using $\epsilon = 10^{-2}$ (as recommended in [44] results in $\delta_{\max} = O\left(10^{-10}\right)$. Decreasing ϵ to 10^{-3} reduces $\delta_{\max}$ to $O\left(10^{-14}\right)$, and decreasing ϵ further to 1×10^{-4} reduces $\delta_{\max}$ to 10^{-15}. Smaller values of ϵ do result in observable changes to $\delta_{\max}$, and we do not observe any significant dependence of $\delta_{\max}$ on other discretization parameters.



(a) Solution at time $T = .1$

(b) $\Delta U(t)$

Figure 2: Solution snapshot and change in entropy $\Delta U(t)$ for both entropy conservative (EC) and Lax-Friedrichs (LF) fluxes.

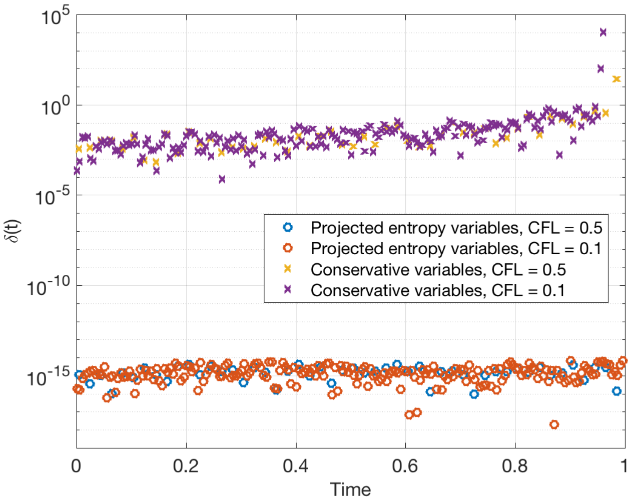


Figure 3: Comparison of $\delta(t)$ when evaluating the flux function $\boldsymbol{f}_S$ using conservative variables and projected entropy variables.

Finally, in order to illustrate the importance of evaluating the flux in terms of the projected entropy variables $\Pi_N \boldsymbol{v}$, we compute $\delta(t)$ while evaluating the flux function $\boldsymbol{f}_S$ directly in terms of the conservative variables $\boldsymbol{u}(x)$ and in terms of $\boldsymbol{u}\left((\Pi_N \boldsymbol{v})(x)\right)$. It can be observed from Figure 3 that $\delta(t)$ is near machine precision when evaluating the flux in terms of the projected entropy variables. When evaluating the flux function using the conservative variables directly, $\delta(t)$ begins near 10^{-4} but grows steadily, blowing up exponentially near $t = 1$.

6 Conclusions

This work presents a generalization of discretely entropy conservative methods from diagonal-norm SBP-DG methods to a more general class of high order DG methods which include dense norm SBP methods and over-integrated quadrature rules. The resulting DG methods satisfy a discrete conservation of entropy while maintaining a reasonable computational cost and small stencil. Numerical experiments indicate the resulting methods deliver high order accuracy for smooth solutions while displaying greatly improved stability for under-resolved and shock solutions.

While entropy conservative and stability improve the robustness of solvers for nonlinear hyperbolic conservation laws, they do not address problems such as spurious oscillations in high order approximations of shock solutions or positivity preservation of density and pressure variables [6]. These issues can be addressed through the use of regularization (e.g. filtering, artificial viscosity) and/or limiting; however, the goal in constructing discretely entropy stable methods is to avoid using excessive regularization and limiting in the pursuit of stability.

Finally, we note that the analysis and numerical experiments in this work are presented in one dimension for clarity and conciseness of presentation. The main concepts are directly extendable to higher dimensions and simplicial and pyramidal elements, which will be the focus of future work.

7 Acknowledgments

The author thanks Andrew Winters, David M. Williams, and Paul Hand for informative discussions. Jesse Chan is supported by NSF awards DMS-1719818 and DMS-1712639.

A Assumptions on the flux function

In this work, we assume that the flux function $\boldsymbol{f}_S$ admits an absolutely convergent expansion of the form

$$\boldsymbol{f}_S(x,y) = \sum_{i=1}^{\infty} \boldsymbol{f}_i(x)\boldsymbol{g}_i(y),$$

where $\boldsymbol{f}_i, \boldsymbol{g}_i$ are real and bounded over each element D^k. This holds, for example, for analytic $\boldsymbol{f}_S$, for which the Taylor series can be rearranged to yield the form of the desired expansion. This assumption is trivially satisfied for the Burgers' and shallow water equations, as the flux functions consist only of finite sums of products of variables. However, for the compressible Euler equations, this expansion is less clear due to the presence of the logarithmic mean in the entropy conservative Chandreshekar flux and the presence of inverses of averages and logarithmic means of the variable $\beta = \rho/(2p)$.

We first address the logarithmic mean by noting that it can be expanded out into series form using multi-index notation [45]. Let $\mu = (\mu_1, \mu_2)$ be positive integers and define $|\mu| = \mu_1 + \mu_2$; then, the logarithmic mean can be rewritten as

$$\frac{x-y}{\log(x)-\log(y)} = \sum_{n=0}^{\infty} \frac{x_n}{n!}, \qquad x_n = \sum_{|\mu|=n} \frac{\log(x)^{\mu_1}\log(y)^{\mu_2}}{n+1},$$

where division by $(n+1)$ averages over the $(n+1)$ different permutations of $\mu_1, \mu_2 \geq 0$ which sum to n. Since x_n is the average of degree n powers of $\log(x), \log(y)$, we have that

$$\left|\frac{x_n}{n!}\right| \leq \frac{\max|\log(x), \log(y)|^n}{n!}.$$

The latter term is simply the nth term of the exponential Taylor series evaluated with the argument $\max|\log(x), \log(y)|$. Because the exponential Taylor series converges absolutely everywhere, the series for the logarithmic mean also converges absolutely for x, y bounded away from 0 and ∞ (i.e. x, y are in the logarithmic interval of convergence).

The entropy conservative Chandreshekar flux also requires multiplication by the terms

$$\frac{1}{\{\{\beta\}\}}, \qquad \frac{1}{\{\{\beta\}\}^{\log}}, \qquad \beta = \frac{\rho}{2p}.$$

These terms are analytic if $\{\{\beta\}\}, \{\{\beta\}\}^{\log}$ are bounded away from zero, which is satisfied if the density and pressure are bounded from below as assumed in (19). Because the composition, sum, and product of analytic functions are also analytic, the Chandreshekar flux also is analytic under positivity assumptions.

Finally, we note that the application of the operator D_N to convergent infinite sums should be rigorously justified. This requires D_N to be a bounded operator from H^1 to P^N, which can be shown using boundedness of the L^2 projection operator and discrete inverse and trace inequalities [10].

References

[1] 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, 2013.

[2] 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.

[3] 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, 2016.

[4] 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, 2016.

[5] Niklas Wintermeyer, Andrew R Winters, Gregor J Gassner, and David A Kopriva. An entropy stable nodal discontinuous Galerkin method for the two dimensional shallow water equations on unstructured curvilinear meshes with discontinuous bathymetry. *Journal of Computational Physics*, 340:200–242, 2017.

[6] 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, 2017.

[7] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, volume 54. Springer, 2007.

[8] Andreas Klöckner, Tim Warburton, Jeff Bridge, and Jan S Hesthaven. Nodal discontinuous Galerkin methods on graphics processors. *Journal of Computational Physics*, 228(21):7863–7882, 2009.

[9] Mark Ainsworth. Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. *Journal of Computational Physics*, 198(1):106–130, 2004.

[10] 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, 2016.

[11] Zhijian J Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, Hung T Huynh, et al. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[12] Lilia Krivodonova. Limiters for high-order discontinuous Galerkin methods. *Journal of Computational Physics*, 226(1):879–896, 2007.

[13] Per-Olof Persson and Jaime Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. *AIAA*, 112, 2006.

[14] George Karniadakis and Spencer Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.

[15] Robert M Kirby and George Em Karniadakis. De-aliasing on non-uniform grids: algorithms and applications. *Journal of Computational Physics*, 191(1):249–264, 2003.

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

[17] Jesse Chan, Russell J Hewett, and T Warburton. Weight-adjusted discontinuous Galerkin methods: wave propagation in heterogeneous media. *arXiv preprint arXiv:1608.01944*, 2016.

[18] Jesse Chan, Russell J Hewett, and T Warburton. Weight-adjusted discontinuous Galerkin methods: curvilinear meshes. *arXiv preprint arXiv:1608.03836*, 2016.

[19] Jesse Chan. Weight-adjusted discontinuous Galerkin methods: matrix-valued weights and elastic wave propagation in heterogeneous media. *arXiv preprint arXiv:1701.00215*, 2017.

[20] Eitan Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Mathematics of Computation*, 49(179):91–103, 1987.

[21] Eitan Tadmor. Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. *Acta Numerica*, 12:451–512, 2003.

[22] Deep Ray, Praveen Chandrashekar, Ulrik S Fjordholm, and Siddhartha Mishra. Entropy stable scheme on two-dimensional unstructured grids for Euler equations. *Communications in Computational Physics*, 19(5):1111–1140, 2016.

[23] Gregor J Gassner. A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods. *SIAM Journal on Scientific Computing*, 35(3):A1233–A1253, 2013.

[24] 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, 2017.

[25] Jason E Hicken, David C Del Rey Fernndez, 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.

[26] 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, 1986.

[27] MJS Chin-Joe-Kong, WA Mulder, and M Van Veldhuizen. Higher-order triangular and tetrahedral finite elements with mass lumping for solving the wave equation. *Journal of Engineering Mathematics*, 35(4):405–426, 1999.

[28] Elena Zhebel, Sara Minisini, Alexey Kononov, and Wim A Mulder. A comparison of continuous mass-lumped finite elements with finite differences for 3-D wave propagation. *Geophysical Prospecting*, 62(5):1111–1125, 2014.

[29] Gregor J Gassner, Andrew R Winters, Florian J Hindenlang, and David A Kopriva. The BR1 Scheme is Stable for the Compressible Navier-Stokes Equations. *arXiv preprint arXiv:1704.03646*, 2017.

[30] Michael S Mock. Systems of conservation laws of mixed type. *Journal of Differential equations*, 37(1):70–88, 1980.

[31] Xiangxiong Zhang and Chi-Wang Shu. On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *Journal of Computational Physics*, 229(23):8918–8934, 2010.

[32] 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, 2012.

[33] Hendrik Ranocha, Philipp Öffner, and Thomas Sonar. Extended skew-symmetric form for summation-by-parts operators and varying Jacobians. *Journal of Computational Physics*, 342:13–28, 2017.

[34] David C Fernández, Jason E Hicken, and David W Zingg. Simultaneous approximation terms for multi-dimensional summation-by-parts operators. *arXiv preprint arXiv:1605.03214*, 2016.

[35] Hendrik Ranocha. Comparison of some entropy conservative numerical fluxes for the Euler equations. *arXiv preprint arXiv:1701.02264*, 2017.

[36] Daniele Antonio Di Pietro and Alexandre Ern. *Mathematical aspects of discontinuous Galerkin methods*, volume 69. Springer Science & Business Media, 2011.

[37] Junping Wang and Xiu Ye. A weak Galerkin finite element method for second-order elliptic problems. *Journal of Computational and Applied Mathematics*, 241:103–115, 2013.

[38] 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, 2013.

[39] Cengke Shi and Chi-Wang Shu. On local conservation of numerical methods for conservation laws. *Computers and Fluids*, 2017.

[40] George PH Styan. Hadamard products and multivariate statistical analysis. *Linear algebra and its applications*, 6:217–240, 1973.

[41] T Warburton and Jan S Hesthaven. On the constants in *hp*-finite element trace inverse inequalities. *Computer methods in applied mechanics and engineering*, 192(25):2765–2773, 2003.

[42] 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.

[43] Amiram Harten. On the symmetric form of systems of conservation laws with entropy. *Journal of computational physics*, 49(1):151–164, 1983.

[44] Farzad Ismail and Philip L Roe. Affordable, entropy-consistent Euler flux functions II: Entropy production at shocks. *Journal of Computational Physics*, 228(15):5410–5436, 2009.

[45] Seppo Mustonen. Logarithmic mean for several arguments. http://www.survo.fi/papers/logmean.pdf, 2002.