

On discretely entropy conservative and entropy stable discontinuous Galerkin methods

Jesse Chan^a,

^a*Department of Computational and Applied Mathematics, Rice University, 6100 Main St, Houston, TX, 77005*

Abstract

High order methods based on diagonal-norm summation by parts operators can be shown to satisfy a discrete conservation or dissipation of entropy for nonlinear systems of hyperbolic PDEs [1, 2]. These methods can also be interpreted as nodal discontinuous Galerkin methods with diagonal mass matrices [3, 4, 5, 6]. In this work, we describe how use flux differencing, quadrature-based projections, and SBP-like operators to construct discretely entropy conservative schemes for DG methods under more arbitrary choices of volume and surface quadrature rules. The resulting methods are semi-discretely entropy conservative or entropy stable with respect to the volume quadrature rule used. Numerical experiments confirm the stability and high order accuracy of the proposed methods for the compressible Euler equations in one and two dimensions.

1. Introduction

Numerical simulations in engineering increasingly require higher accuracy without sacrificing computational efficiency. Because they are more accurate than low order methods per degree of freedom for sufficiently regular solutions, high order methods provide one avenue towards improving fidelity in numerical simulations while maintaining reasonable computational costs. High order methods which can accommodate unstructured meshes are desirable for problems with complex geometries, and 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 of high order methods can be attributed in part to their low numerical dissipation and dispersion compared to low order schemes [9]. This accuracy has made them advantageous for the simulation of wave propagation [7, 10]. However, while high order methods can be applied in a stable manner to linear wave propagation problems, instabilities are observed when applying them to nonlinear hyperbolic problems. This is contrast to low order schemes, whose high numerical dissipation tends to apply a stabilizing effect [11]. As a result, most high order schemes for nonlinear conservation laws typically require additional stabilization procedures, including filtering [7], slope limiting [12], artificial viscosity [13], and polynomial de-aliasing through over-integration [14]. Moreover, stabilized numerical methods can still fail, requiring user intervention or heuristic modifications to achieve non-divergent solutions.

For linear wave propagation problems, semi-discretely energy stable numerical methods can be constructed, even in the presence of curvilinear coordinates or variable coefficients [15, 16, 17, 18]. This semi-discrete stability implies that, under a stable timestep restriction (CFL condition), discrete solutions do not suffer from non-physical growth in time. However, for nonlinear systems of conservation laws, traditional methods do not admit theoretical proofs of semi-discrete stability. This was addressed for low order methods with the introduction of discretely entropy conservative and entropy stable finite volume schemes by Tadmor [19]. These schemes rely on a specific entropy conservative flux which satisfies a condition involving the entropy variables and entropy potential, and were extended to low order finite volume methods on un-

Email address: Jesse.Chan@caam.rice.edu (Jesse Chan)

structured grids in [20]. High order entropy stable methods were also developed for structured grids in [21] based on an entropy conservative essentially non-oscillatory (ENO) reconstruction.

The extension of entropy conservative schemes to unstructured high order methods 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 also 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, the summation by parts (SBP) property [22], and a concept referred to as *flux differencing*. Similar entropy-stable schemes have been constructed for the shallow water and MHD equations [4, 5, 23]. Finally, high order entropy conservative and entropy stable schemes have been extended to unstructured triangular meshes in [24, 6].

It is possible to construct energy preserving schemes for certain conservation laws based on split forms of conservation laws, which involve both conservative and non-conservative derivative terms. Split formulations have been shown to recover kinetic energy preserving schemes for the compressible Euler and Navier-Stokes equations under diagonal norm SBP operators [25, 26, 27]. Additionally, for dense norm and generalized SBP operators, stable schemes for Burgers' equation can be constructed based on the split form of the underlying equations [22, 28, 29]. However, entropy conservative and entropy stable schemes for the compressible Euler or Navier-Stokes equations do not correspond to split formulations [6], and (to the authors knowledge) the construction of unstructured high order entropy conservative and entropy stable schemes for these equations has required diagonal norm SBP operators.¹ We refer to DG methods with these properties as diagonal norm SBP-DG methods.

Appropriate diagonal norm SBP operators are straightforward to construct on tensor product elements based on a DG-SEM discretization. Diagonal-norm SBP operators can also be constructed for triangles and tetrahedra [31, 32, 6]; however, the number of nodal points for such operators is typically greater than the dimension of the natural polynomial approximation space, and the resulting diagonal norm SBP-DG operators do not correspond to any basis [32]. Furthermore, to the author's knowledge, appropriate point sets have only been constructed for $N \leq 4$ in three dimensions [33], and the construction of high order diagonal norm SBP-DG methods has not yet been performed for uncommon elements such as pyramids [34].

This work focuses on the construction of entropy conservative high order DG schemes for systems of conservation laws. In order to generalize beyond diagonal norm SBP-DG methods, we will consider DG discretizations using over-integrated quadrature rules with more points than the dimension of the approximation space, which are commonly used for non-tensor product elements in two and three dimensions [35]. These quadrature rules induce DG schemes which are related to dense norm and generalized SBP operators [36, 28, 37], for which discretely entropy stable schemes for the compressible Euler equations have not yet been constructed. We present proofs of discrete entropy stability using both a matrix formulation involving a "decoupled" SBP-like operator and continuous formulations involving projection and lifting operators. In both cases, the proofs rely only on properties which hold under quadrature-based integration. We also focus on ensuring discrete entropy stability for conservation laws which do not admit a nonlinearly stable split formulation.

The outline of the paper is as follows: Section 2 will briefly review the construction of entropy conservative diagonal norm SBP-DG methods on a single element. Section 4 will describe how to construct analogous entropy conservative methods on single element in a continuous setting. Section 5 will discuss extensions to multiple elements, including comparisons of different coupling terms and entropy stable fluxes. Finally, Section 6 presents numerical results which verify the high order accuracy and discrete entropy conservation of the proposed methods in one and two spatial dimensions.

¹Entropy stable high order finite element and DG methods which do not fall under the diagonal norm SBP-DG category have been proposed [30], but the proofs are often given at the continuous level, relying on exact integration or the chain rule, which do hold at the discrete level.

2. Entropy stability for systems of hyperbolic PDEs

We will begin by reviewing continuous entropy theory. We consider systems of nonlinear conservation laws in one dimension with n variables

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = 0, \quad \mathbf{u}(x, t) = (u_1(x, t), \dots, u_n(x, t)). \quad (1)$$

where the fluxes $\mathbf{f}(\mathbf{u})$ are smooth functions of the vector of conservative variables $\mathbf{u}(x, t)$. We are interested in systems for which there exists a convex entropy function $U(\mathbf{u})$ such that

$$U''(\mathbf{u})\mathbf{A}(\mathbf{u}) = (U''(\mathbf{u})\mathbf{A}(\mathbf{u}))^T, \quad (\mathbf{A}(\mathbf{u}))_{ij} = \left(\frac{\partial \mathbf{f}(\mathbf{u})}{\partial u_j} \right)_i, \quad (2)$$

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

It can be shown (see, for example, [38]) that (2) is equivalent to the existence of an entropy flux function $F(\mathbf{u})$ and entropy potential ψ such that

$$\mathbf{v}^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \frac{\partial F(\mathbf{u})}{\partial \mathbf{u}}^T, \quad \psi(\mathbf{v}) = \mathbf{v}^T \mathbf{f}(\mathbf{u}(\mathbf{v})) - F(\mathbf{u}(\mathbf{v})), \quad \psi'(\mathbf{v}) = \mathbf{f}(\mathbf{u}(\mathbf{v})). \quad (3)$$

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

$$\frac{\partial U(\mathbf{u})}{\partial t} + \frac{\partial F(\mathbf{u})}{\partial x} = 0. \quad (4)$$

We assume now that the domain is the interval $[-1, 1]$. Integrating (4) over this interval and using the definition of the entropy potential then yields a statement of entropy conservation for smooth solutions

$$\int_{-1}^1 \frac{\partial U(\mathbf{u})}{\partial t} + (\mathbf{v}^T \mathbf{f}(\mathbf{u}(\mathbf{v})) - \psi(\mathbf{v})) \Big|_{-1}^1 = 0. \quad (5)$$

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 inequality

$$\frac{\partial U(\mathbf{u})}{\partial t} + \frac{\partial F(\mathbf{u})}{\partial x} \leq 0. \quad (6)$$

Integrating over $[-1, 1]$ then yields a more general statement of entropy inequality

$$\int_{-1}^1 \frac{\partial U(\mathbf{u})}{\partial t} + (\mathbf{v}^T \mathbf{f}(\mathbf{u}(\mathbf{v})) - \psi(\mathbf{v})) \Big|_{-1}^1 \leq 0. \quad (7)$$

The focus of this work is the construction of high order polynomial DG methods which satisfy a discrete analogue of the conservation of entropy (5) and the dissipation of entropy (7).

3. Discrete differential operators and quadrature-based matrices

3.1. Mathematical assumptions and notations

We begin with a d -dimensional reference element \widehat{D} with boundary $\partial \widehat{D}$. We denote the i th component of the outward normal vector on the boundary of the reference element $\partial \widehat{D}$ as \widehat{n}_i . For this work, we assume

that \hat{n}_i is constant; i.e., that the faces of the reference element are planar, which is true for most commonly used reference elements in two and three dimensions [39].

We define an approximation space using degree N polynomials on the reference element. In one dimension, this space is defined as

$$P^N(\hat{D}) = \left\{ \hat{x}^i, \quad \hat{x} \in \hat{D}, \quad 0 \leq i \leq N \right\}. \quad (8)$$

In higher dimensions, the choice of approximation space depends on the type of element [39], but generally contains the space of total degree N polynomials. We denote the dimension of the approximation space P^N as $N_p = \dim(P^N(\hat{D}))$ (with $N_p = N + 1$ in one dimension).

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

$$(\mathbf{u}, \mathbf{v})_{\hat{D}} = \int_{\hat{D}} \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x} = \int_{\hat{D}} \mathbf{u} \cdot \mathbf{v} J^k \, d\hat{x}, \quad \|\mathbf{u}\|_{\hat{D}}^2 = (\mathbf{u}, \mathbf{u})_{\hat{D}}, \quad \langle \mathbf{u}, \mathbf{v} \rangle_{\partial\hat{D}} = \int_{\partial\hat{D}} \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x}, \quad (9)$$

3.2. Interpolation and differentiation matrices

In most implementations, integrals and L^2 inner products are approximated using a quadrature rule which is exact for polynomials of a certain degree. This defines a discrete L^2 inner product, which in turn can be used to construct operators which obey a property resembling summation-by-parts for diagonal norm SBP operators.

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

$$u(\mathbf{x}) = \sum_{j=1}^{N_p} \mathbf{u}_j \phi_j(\hat{\mathbf{x}}), \quad P^N(\hat{D}) = \text{span} \{ \phi_i(\hat{\mathbf{x}}) \}_{i=1}^{N_p}. \quad (10)$$

The construction of these quadrature-based matrices utilizes ϕ_i , as well as volume and surface quadrature rules with N_q and N_q^f points, respectively. We make the following assumptions on the strength of the quadrature rules:

Assumption 1. *The volume quadrature rule $\{(\mathbf{x}_i, w_i)\}_{i=1}^{N_q}$ exactly integrates polynomials of degree at least $(2N-1)$ on the reference element \hat{D} , and the surface quadrature $\{(\mathbf{x}_i^f, w_i^f)\}_{i=1}^{N_q^f}$ exactly integrates polynomials of at least degree $2N$ on the boundary of the reference element $\partial\hat{D}$.*

These assumptions impose a minimum strength of the quadrature rule; however, the polynomial degree for which these quadrature rules are accurate can be taken arbitrarily high. These conditions are imposed to ensure that integration by parts holds for any two polynomials of degree N when integrals are approximated using quadrature.²

Let $\mathbf{W} \in \mathbb{R}^{N_q \times N_q}$ denote the diagonal matrix whose entries are quadrature weights

$$\mathbf{W}_{ij} = \begin{cases} w_i & i = j \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

We also define \mathbf{W}_f as the diagonal matrix of surface quadrature weights. We define the quadrature interpolation matrix \mathbf{V}_q

$$(\mathbf{V}_q)_{ij} = \phi_j(\mathbf{x}_i), \quad 1 \leq j \leq N_p, \quad 1 \leq i \leq N_q, \quad (12)$$

²We note that these conditions are sufficient, but not always necessary. For example, appropriate SBP operators can be constructed using tensor product Gauss-Legendre-Lobatto quadratures, though the GLL surface quadrature is only exact for degree $2N-1$ polynomials. More general conditions can be formulated by requiring that integration by parts holds when volume and surface integrals are approximated using volume and surface quadrature rules.

which maps coefficients \mathbf{u} to evaluations of u at quadrature points

$$\mathbf{u}_q = \mathbf{V}_q \mathbf{u}, \quad (\mathbf{u}_q)_i = u(x_i), \quad 1 \leq i \leq N_q. \quad (13)$$

Let \mathbf{V}_f denote the matrix which interpolates to boundary values ³

$$(\mathbf{V}_f)_{ij} = \phi_j(\hat{\mathbf{x}}_i^f), \quad 1 \leq j \leq N_p, \quad 1 \leq i \leq N_q^f, \quad (15)$$

Next, let \mathbf{D}_i denote the differentiation matrix with respect to the i th coordinate, defined implicitly through

$$u(\hat{\mathbf{x}}) = \sum_{j=1}^{N_p} \mathbf{u}_j \phi_j(\hat{\mathbf{x}}), \quad \frac{\partial u}{\partial \hat{\mathbf{x}}_i} = \sum_{j=1}^{N_p} (\mathbf{D}_i \mathbf{u})_j \phi_j(\hat{\mathbf{x}}) \quad (16)$$

In other words, \mathbf{D}_i maps basis coefficients of some polynomial $u \in P^N$ to coefficients of its i th derivative with respect to the reference coordinate $\hat{\mathbf{x}}$, and is sometimes referred to as a “modal”⁴ differentiation matrix with respect to a general non-nodal basis [32].

3.3. Quadrature-based projection matrices and lifting matrices

Given \mathbf{V}_q , we introduce the mass matrix, whose entries are the evaluation of inner products of different basis functions using quadrature

$$\mathbf{M} = \mathbf{V}_q^T \mathbf{W} \mathbf{V}_q, \quad \mathbf{M}_{ij} = \sum_{k=1}^{N_q} w_k \phi_j(\hat{\mathbf{x}}_k) \phi_i(\hat{\mathbf{x}}_k) \approx \int_{\hat{D}} \phi_j \phi_i d\hat{\mathbf{x}} = (\phi_j, \phi_i)_{\hat{D}}. \quad (17)$$

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, and is also referred to in the SBP literature as a “norm” matrix, with distinctions made between dense and diagonal norm matrices. We do not make any distinctions between diagonal and dense \mathbf{M} in this work.

The mass matrix appears in the computation of the L^2 projection both when integrals are computed exactly and when integrals are computed using quadrature. The continuous L^2 projection operator is defined as $\Pi_N : L^2(\hat{D}) \rightarrow P^N(\hat{D})$ such that

$$(\Pi_N f, v)_{\hat{D}} = (f, v)_{\hat{D}}, \quad \forall v \in P^N(\hat{D}). \quad (18)$$

In other words, Π_N is the operator which maps an L^2 integrable function f to a polynomial $\Pi_N f \in P^N(\hat{D})$. Assuming some polynomial basis $\phi_j(\hat{\mathbf{x}})$ for P^N , the L^2 projection reduces to the determination of coefficients of $\Pi_N f$ in the $\phi_j(\hat{\mathbf{x}})$. Additionally, when integrals within the L^2 inner products present in (18) are computed using quadrature, the **discrete quadrature-based** L^2 projection of a function $f(x)$ (18) can be reduced to the following matrix problem:

$$\mathbf{M} \mathbf{u} = \mathbf{V}_q^T \mathbf{W} \mathbf{f}, \quad \mathbf{f}_i = f(\hat{\mathbf{x}}_i), \quad i = 1, \dots, N_q. \quad (19)$$

³In one dimension, $\hat{D} = [-1, 1]$, and the surface quadrature rule (\mathbf{x}_i^f, w_i^f) consists only of boundary points $-1, 1$, with normal directions $\hat{n} = \pm 1$ and both weights $w_1^f = w_2^f = 1$, such that \mathbf{W}_f is simply the 2×2 identity matrix. In one dimension, \mathbf{V}_f reduces to

$$(\mathbf{V}_f)_{1j} = \phi_j(-1), \quad (\mathbf{V}_f)_{2j} = \phi_j(1), \quad 1 \leq j \leq N_p. \quad (14)$$

⁴The term “modal” refers to bases which are not necessarily defined in terms of nodal points, and should not be confused with modal bases which are orthonormal with respect to an L^2 inner product.

where u is the vector of coefficients of the **quadrature-based** L^2 projection of \mathbf{f} . Inverting the mass matrix allows us to define the quadrature-based L^2 projection matrix \mathbf{P}_q as a discretization of the L^2 projection operator Π_N

$$\mathbf{P}_q = \mathbf{M}^{-1} \mathbf{V}_q^T \mathbf{W}. \quad (20)$$

The matrix \mathbf{P}_q which maps a function (in terms of its evaluation at quadrature points) to coefficients of the L^2 projection in the basis $\phi_j(\hat{\mathbf{x}})$. Note that, since $\mathbf{M} = \mathbf{V}_q^T \mathbf{W} \mathbf{V}_q$,

$$\mathbf{P}_q \mathbf{V}_q = \mathbf{M}^{-1} \mathbf{V}_q^T \mathbf{W} \mathbf{V}_q = \mathbf{I}. \quad (21)$$

In other words, the projection operator reduces to the identity matrix when applied to any linear combination of basis functions $\phi_j(\hat{\mathbf{x}})$ evaluated at quadrature points. This implies that, when applied to the evaluation of any polynomial at volume quadrature points, the matrix \mathbf{P}_q simply returns the coefficients of the polynomial in the basis $\phi_j(\hat{\mathbf{x}})$.⁵

We also introduce the quadrature-based lifting matrix

$$\mathbf{L}_q = \mathbf{M}^{-1} \mathbf{V}_f^T \mathbf{W}_f, \quad (22)$$

which “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. This is a quadrature-based discretization of the lifting operator $L : L^2(\partial\hat{D}) \rightarrow P^N$ [7, 40]

$$(Lu, v)_{\hat{D}} = \langle u, v \rangle_{\partial\hat{D}}, \quad \forall v \in P^N. \quad (23)$$

3.4. Quadrature-based differentiation matrices and a “decoupled” SBP-like operator

The projection and lifting matrices map between function values at volume or surface quadrature points and approximations which can be represented in the basis $\{\phi_i\}_{i=1}^{N_p}$. By combining them with the polynomial differentiation matrix \mathbf{D}_i , we can construct a differencing operator for functions defined at quadrature points. For example, given function evaluations at volume quadrature points, we can project the function to P^N using \mathbf{P}_q , differentiate the resulting polynomial, and evaluate the result at quadrature points. This sequence of operations can be concisely expressed as the product of three matrices

$$\mathbf{D}_q^i = \mathbf{V}_q \mathbf{D}_i \mathbf{P}_q. \quad (24)$$

Define $\text{diag}(\mathbf{u})$ as the diagonal matrix with the entries of \mathbf{u} on the diagonal. The quadrature-based differentiation matrix \mathbf{D}_q^i obeys the following lemma:

Lemma 1. *The operator $\mathbf{D}_q^i = \mathbf{V}_q \mathbf{D}_i \mathbf{P}_q$ satisfies the SBP property with respect to the diagonal matrix \mathbf{W} such that*

$$\mathbf{W} \mathbf{D}_q^i + (\mathbf{W} \mathbf{D}_q^i)^T = \mathbf{P}_q^T \mathbf{V}_f^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}_i) \mathbf{V}_f \mathbf{P}_q. \quad (25)$$

Additionally, \mathbf{D}_q^i is a degree N approximation to the derivative $\frac{\partial}{\partial \hat{\mathbf{x}}_i}$.

Proof. We first note that the “modal” differentiation matrix \mathbf{D}_i satisfies the following property with respect to the mass matrix

$$\mathbf{M} \mathbf{D}_i + \mathbf{D}_i^T \mathbf{M} = \mathbf{V}_f^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}_i) \mathbf{V}_f. \quad (26)$$

This is simply a restatement of integration by parts for polynomials [7, 39, 32]

$$\int_{\hat{D}} \frac{\partial \phi_j}{\partial \hat{\mathbf{x}}_i} \phi_i \, d\hat{\mathbf{x}} = \int_{\partial\hat{D}} \phi_j \phi_i \hat{\mathbf{n}}_i \, d\hat{\mathbf{x}} - \int_{\hat{D}} \phi_j \frac{\partial \phi_i}{\partial \hat{\mathbf{x}}_i} \, d\hat{\mathbf{x}} \quad (27)$$

⁵It is worth noting that $\mathbf{V}_q \mathbf{P}_q \neq \mathbf{I}$ in general. However, if both matrices are square such that $N_p = N_q$ and the number of basis functions coincides with the number of quadrature points, then the mass matrix inverse can be explicitly written as $\mathbf{V}_q^{-T} \mathbf{W}^{-1} \mathbf{V}_q^{-1}$, and we can simplify $\mathbf{V}_q \mathbf{P}_q = \mathbf{V}_q \mathbf{M}^{-1} \mathbf{V}_q^T \mathbf{W} = \mathbf{I}$. When $N_q > N_p$, the matrix \mathbf{V}_q cannot be inverted and the mass matrix does not have an explicit inverse in terms of \mathbf{V}_q .

using the fact that the above integrals are exact for volume quadratures of degree $2N - 1$ and surface quadratures of degree $2N$, which we have from Assumption 1. Multiplying equation (26) by \mathbf{P}_q^T on the left, \mathbf{P}_q on the right, and using $\mathbf{P}_q = \mathbf{M}^{-1}\mathbf{V}_q^T\mathbf{W}$ yields

$$\begin{aligned} \mathbf{P}_q^T \mathbf{M} \mathbf{D}_i \mathbf{P}_q + \mathbf{P}_q^T \mathbf{D}_i^T \mathbf{M} \mathbf{P}_q^T &= \mathbf{W} \mathbf{V}_q \mathbf{D}_i \mathbf{P}_q + \mathbf{P}_q^T \mathbf{D}_i^T \mathbf{V}_q^T \mathbf{W} \\ &= \mathbf{W} \mathbf{D}_q^i + (\mathbf{W} \mathbf{D}_q^i)^T = \mathbf{P}_q^T \mathbf{V}_f^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}_i) \mathbf{V}_f \mathbf{P}_q. \end{aligned} \quad (28)$$

The accuracy of \mathbf{D}_q^i results from the fact that \mathbf{D}_q^i recovers the exact derivative of polynomials up to degree N . Let \mathbf{u}_q be the values of some polynomial $u \in P^N$ at quadrature points, such that $\mathbf{u}_q = \mathbf{V}_q \mathbf{u}$ for coefficients \mathbf{u} . Then by (21),

$$\mathbf{D}_q^i \mathbf{u}_q = \mathbf{V}_q \mathbf{D}_i \mathbf{P}_q \mathbf{V}_q^T \mathbf{u} = \mathbf{V}_q \mathbf{D}_i \mathbf{u} \quad (29)$$

which is the evaluation of the exact derivative of $u(\mathbf{x})$ at quadrature points. \square

Lemma 1 shows how the projection matrix \mathbf{P}_q can transform a “modal” SBP property (involving the norm matrix \mathbf{M} , which can be dense) to a quadrature-based SBP property involving the diagonal norm matrix \mathbf{W} . The boundary term in the resulting SBP property includes the matrix $\mathbf{V}_f \mathbf{P}_q$, which can be interpreted as taking the projection of a function (defined through values at volume quadrature points) and evaluating the result at surface quadrature points.

We can also use the quadrature-based differentiation operator \mathbf{D}_q^i to define a “decoupled” operator \mathbf{D}_N^i which maps from a vector of both volume and surface quadrature points to values at both volume and surface quadrature points

$$\mathbf{D}_N^i = \begin{bmatrix} \mathbf{D}_q^i - \frac{1}{2} \mathbf{V}_q \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}_i) \mathbf{V}_f \mathbf{P}_q & \frac{1}{2} \mathbf{V}_q \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}_i) \\ -\frac{1}{2} \text{diag}(\hat{\mathbf{n}}_i) \mathbf{V}_f \mathbf{P}_q & \frac{1}{2} \text{diag}(\hat{\mathbf{n}}_i) \end{bmatrix}, \quad (30)$$

where $\hat{\mathbf{n}}_i$ is the vector containing the i th normal component \hat{n}_i evaluated at surface quadrature points.

Let \mathbf{W}_N be defined as the matrix of both volume and surface quadrature weights

$$\mathbf{W}_N = \begin{pmatrix} \mathbf{W} & \\ & \mathbf{W}_f \end{pmatrix}. \quad (31)$$

We can show that the “weak” or integrated version of the differentiation matrix \mathbf{D}_N^i satisfies the following properties:

Theorem 1. *The matrix $\mathbf{Q}_N^i = \mathbf{W}_N \mathbf{D}_N^i$ satisfies the SBP-like property*

$$\mathbf{Q}_N^i + (\mathbf{Q}_N^i)^T = \mathbf{B}_N^i, \quad \mathbf{B}_N^i = \begin{pmatrix} 0 & \\ & \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}_i) \end{pmatrix}. \quad (32)$$

Additionally, $\mathbf{Q}_N^i \mathbf{1} = 0$, where $\mathbf{1}$ is the vector of all ones.

Proof. The matrix \mathbf{Q}_N^i is given explicitly as

$$\mathbf{Q}_N^i = \begin{bmatrix} \mathbf{W} \mathbf{D}_q^i - \frac{1}{2} \mathbf{W} \mathbf{V}_q \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}_i) \mathbf{V}_f \mathbf{P}_q & \frac{1}{2} \mathbf{W} \mathbf{V}_q \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}_i) \\ -\frac{1}{2} \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}_i) \mathbf{V}_f \mathbf{P}_q & \frac{1}{2} \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}_i) \end{bmatrix}. \quad (33)$$

The bottom right block of $\mathbf{Q}_N^i + (\mathbf{Q}_N^i)^T$ is $\mathbf{W}_f \text{diag}(\hat{\mathbf{n}}_i)$, as both \mathbf{W}_f and $\text{diag}(\hat{\mathbf{n}}_i)$ are diagonal and symmetric. We will show that all remaining blocks of $\mathbf{Q}_N^i + (\mathbf{Q}_N^i)^T$ are zero. We first deal with the off-diagonal blocks, which are equal to

$$\frac{1}{2} \mathbf{W} \mathbf{V}_q \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}_i) - \frac{1}{2} (\mathbf{W}_f \text{diag}(\hat{\mathbf{n}}_i) \mathbf{V}_f \mathbf{P}_q)^T \quad (34)$$

and its transpose. These off-diagonal blocks reduce to zero by noting that

$$\begin{aligned}\mathbf{W}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i) &= \mathbf{W}\mathbf{V}_q\mathbf{M}^{-1}\mathbf{V}_f\mathbf{W}_f\text{diag}(\hat{\mathbf{n}}_i) \\ &= \mathbf{P}_q^T\mathbf{V}_f\mathbf{W}_f\text{diag}(\hat{\mathbf{n}}_i) = (\mathbf{W}_f\text{diag}(\hat{\mathbf{n}})\mathbf{V}_f\mathbf{P}_q)^T.\end{aligned}\quad (35)$$

To show the top left block of $\mathbf{Q}_N^i + (\mathbf{Q}_N^i)^T$ is zero, we first use the fact that $\mathbf{L}_q = \mathbf{M}^{-1}\mathbf{V}_f^T\mathbf{W}_f$ to rewrite Lemma 1 as

$$\begin{aligned}\mathbf{W}\mathbf{D}_q^i &= \mathbf{P}_q^T\mathbf{V}_f^T\mathbf{W}_f\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q - (\mathbf{W}\mathbf{D}_q^i)^T \\ &= \mathbf{W}\mathbf{V}_q\mathbf{M}^{-1}\mathbf{V}_f^T\mathbf{W}_f\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q - (\mathbf{W}\mathbf{D}_q^i)^T = \mathbf{W}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q - (\mathbf{W}\mathbf{D}_q^i)^T.\end{aligned}\quad (36)$$

Next, we show that

$$\mathbf{W}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q = \mathbf{W}\mathbf{V}_q\mathbf{M}^{-1}\mathbf{V}_f^T\mathbf{W}_f\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{M}^{-1}\mathbf{V}_q^T\mathbf{W}, \quad (37)$$

from which we can see that $\mathbf{W}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q$ is symmetric. Combining (36) and (37), we have that the top left block of $\mathbf{Q}_N^i + (\mathbf{Q}_N^i)^T$ is also zero

$$\begin{aligned}\mathbf{W}\mathbf{D}_q^i - \frac{1}{2}\mathbf{W}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q + \left(\mathbf{W}\mathbf{D}_q^i - \frac{1}{2}\mathbf{W}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q\right)^T \\ = \mathbf{W}\mathbf{D}_q^i - \mathbf{W}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q + (\mathbf{W}\mathbf{D}_q^i)^T = 0.\end{aligned}\quad (38)$$

Showing $\mathbf{Q}_N^i\mathbf{1} = 0$ is equivalent to showing $\mathbf{D}_N^i\mathbf{1} = 0$. Direct multiplication gives

$$\mathbf{D}_N^i\mathbf{1} = \begin{bmatrix} \mathbf{D}_q^i\mathbf{1} - \frac{1}{2}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q\mathbf{1} + \frac{1}{2}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i)\mathbf{1} \\ -\frac{1}{2}\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q\mathbf{1} + \frac{1}{2}\text{diag}(\hat{\mathbf{n}}_i) \end{bmatrix} = 0. \quad (39)$$

Here, we have used that (21) implies $\mathbf{V}_f\mathbf{P}_q\mathbf{1} = \mathbf{1}$, and that $\mathbf{D}_q^i\mathbf{1} = 0$ by Lemma 1. \square

3.5. Discrete differentiation operators and approximating weighted derivatives

While \mathbf{D}_N^i satisfies an SBP-like property, it cannot be used directly as a differentiation operator. To see this, let \mathbf{u} denote the coefficients of some function $u \in P^N$. Then, $\mathbf{V}_q\mathbf{u}$ and $\mathbf{V}_f\mathbf{u}$ are the evaluations of $u(\mathbf{x})$ at volume and surface quadrature points, respectively. We expect a high order accurate derivative operator to exactly differentiate polynomials. However, applying \mathbf{D}_N^i to the concatenated vector containing evaluations at both volume and surface quadrature points yields

$$\begin{aligned}\mathbf{D}_N^i \begin{bmatrix} \mathbf{V}_q\mathbf{u} \\ \mathbf{V}_f\mathbf{u} \end{bmatrix} &= \begin{bmatrix} \mathbf{D}_q^i - \frac{1}{2}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q & \frac{1}{2}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i) \\ -\frac{1}{2}\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q & \frac{1}{2}\text{diag}(\hat{\mathbf{n}}_i) \end{bmatrix} \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \mathbf{u} \\ &= \begin{bmatrix} \mathbf{V}_q\mathbf{D}_i\mathbf{P}_q\mathbf{V}_q - \frac{1}{2}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q\mathbf{V}_q + \frac{1}{2}\mathbf{V}_q\mathbf{L}_q\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f \\ -\frac{1}{2}\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f\mathbf{P}_q\mathbf{V}_q + \frac{1}{2}\text{diag}(\hat{\mathbf{n}}_i)\mathbf{V}_f \end{bmatrix} \mathbf{u} = \begin{bmatrix} \mathbf{V}_q\mathbf{D}_i\mathbf{u} \\ 0 \end{bmatrix}.\end{aligned}\quad (40)$$

where we have used that $\mathbf{P}_q\mathbf{V}_q = \mathbf{I}$ from (21). This indicates that the rows of \mathbf{D}_N^i corresponding to volume quadrature points recover the exact derivatives of a polynomial, but that the rows of \mathbf{D}_N^i corresponding to surface quadrature points return zero when applied to a polynomial. Thus, while \mathbf{D}_N^i has an SBP-like property, it is not an SBP operator due to the fact that it is not a high order accurate approximation of the derivative [36].

However, while \mathbf{D}_N^i by itself is not a differentiation operator, we can recover the polynomial differentiation operator \mathbf{D}_i by contracting the output of \mathbf{D}_N^i with projection and lifting matrices. Straightforward computations show that

$$\mathbf{D}_i = \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} \mathbf{D}_N^i \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}. \quad (41)$$

Motivated by this fact, let $u(\mathbf{x})$ be a non-polynomial function, and let $[\mathbf{u}_q, \mathbf{u}_f]^T$ be the vector whose entries $\mathbf{u}_q, \mathbf{u}_f$ are the evaluations of u at volume and surface quadrature points respectively. A polynomial approximation of the derivative of u can be computed by first applying \mathbf{D}_N^i to $[\mathbf{u}_q, \mathbf{u}_f]^T$, then applying the projection and lifting matrices to the result

$$\tilde{\mathbf{u}} = \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} \mathbf{D}_N^i \begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix}, \quad \sum_{j=1}^{N_p} \tilde{u}_j \phi_j(\mathbf{x}) \approx \frac{\partial u}{\partial \hat{\mathbf{x}}_i}. \quad (42)$$

In the above case, the application of the projection and lifting matrices can be combined with the application of \mathbf{D}_N^i into a single matrix-vector product. However, applying \mathbf{P}_q and \mathbf{L}_q separately from \mathbf{D}_N^i makes it possible to approximate the product of a function and a function derivative while maintaining an SBP-like property. Let $u(\hat{\mathbf{x}})$ and $w(\hat{\mathbf{x}})$ denote two non-polynomial functions, whose evaluations at volume and surface quadrature points are denoted $[\mathbf{u}_q, \mathbf{u}_f]^T$ and $[\mathbf{w}_q, \mathbf{w}_f]^T$, respectively. The utility of the operator \mathbf{D}_N^i is that, when combined with $\mathbf{P}_q, \mathbf{L}_q$, it can be used to construct some polynomial with coefficients $\tilde{\mathbf{u}}_w$ which approximates the product $w(\hat{\mathbf{x}}) \frac{\partial u}{\partial \hat{\mathbf{x}}_i}$

$$\tilde{\mathbf{u}}_w = \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} \text{diag} \left(\begin{bmatrix} \mathbf{w}_q \\ \mathbf{w}_f \end{bmatrix} \right) \mathbf{D}_N^i \begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix}, \quad \sum_{j=1}^{N_p} (\tilde{\mathbf{u}}_w)_j \phi_j(\hat{\mathbf{x}}) \approx w(\hat{\mathbf{x}}) \frac{\partial u}{\partial \hat{\mathbf{x}}_i}. \quad (43)$$

We note that the polynomial approximation resulting from (43) is equivalent to a quadrature discretization of the following approximation of $w(\mathbf{x}) \frac{\partial u}{\partial \hat{\mathbf{x}}_i}$ involving the continuous L^2 projection and lifting operators Π_N and L

$$w(\hat{\mathbf{x}}) \frac{\partial u}{\partial \hat{\mathbf{x}}_i} \approx \Pi_N \left(w(\hat{\mathbf{x}}) \left(\frac{\partial \Pi_N u}{\partial \hat{\mathbf{x}}_i} + L(u - \Pi_N u) \right) \right) + L(w(u - \Pi_N u)). \quad (44)$$

In the context of discretizations for (1), (43) provides a way to approximate the spatial term involving the nonlinear flux functions when it is in non-conservative form. The connection to the entropy conservative discretization of (1) is made explicit using Burgers' equation as an example in Appendix A.

4. Entropy conservative DG methods on a single element

In this section, we focus on constructing an entropy conservative DG scheme on a single element using the matrix operators defined in Section 3. We first prove that the proposed scheme is entropy conservative on a single one dimensional element, then generalize the proof to higher dimensions.

4.1. A continuous interpretation of flux differencing

In this section, we discuss a continuous interpretation of flux differencing [1, 41, 6], which also encompasses the split form methodology proposed in [3]. This interpretation will guide the construction of entropy conservative DG schemes. We first introduce the definition of an entropy conservative finite volume numerical flux given by Tadmor [19]:

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

$$\mathbf{f}_S(\mathbf{u}_L, \mathbf{u}_R) = \mathbf{f}_S(\mathbf{u}_R, \mathbf{u}_L), \quad \mathbf{f}_S(\mathbf{u}, \mathbf{u}) = \mathbf{f}(\mathbf{u}) \quad (45)$$

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

$$(\mathbf{v}_L - \mathbf{v}_R)^T \mathbf{f}_S(\mathbf{u}_L, \mathbf{u}_R) = (\psi_L - \psi_R), \quad \psi_L = \psi(\mathbf{v}(\mathbf{u}_L)), \quad \psi_R = \psi(\mathbf{v}(\mathbf{u}_R)). \quad (46)$$

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

This numerical flux can be used to construct entropy conservative finite volume methods, and was generalized in [21] for the construction of high order finite volume schemes. This flux was later used for the construction of discretely entropy conservative schemes using an approach referred to as *flux differencing* [1, 2, 41, 6]. The key factor enabling the construction of discretely entropy conservative schemes is that (unlike the continuous proof of entropy conservation), the proof of discrete entropy conservation avoids the use of the chain rule, which does not hold in general at the discrete level. Entropy conservation can also be extended beyond a single element by defining interface fluxes between two elements using the entropy conservative flux $f_S(u_L, u_R)$ [2, 41, 6]. Similarly, employing an entropy stable flux at element interfaces results in an entropy stable method which satisfies a global entropy inequality.

Let $\mathbf{f}_S(\mathbf{u}_L, \mathbf{u}_R)$ now be the symmetric and consistent two-point flux defined in (1). Consider the bivariate function

$$\mathbf{f}_S(\mathbf{u}(x), \mathbf{u}(y)), \quad x, y \in \mathbb{R}. \quad (47)$$

We consider an interpretation of flux differencing motivated by the derivative projection operator introduced by Gassner, Winters, Hindenlang, and Kopriva in [41]. Here, the two-point flux $\mathbf{f}_S = \mathbf{f}_S(\mathbf{u}_1, \mathbf{u}_2)$ is a bivariate function of $\mathbf{u}_1, \mathbf{u}_2$. Then, using the consistency of \mathbf{f}_S ,

$$\frac{\partial \mathbf{f}(\mathbf{u})}{\partial \mathbf{u}} = \frac{\partial \mathbf{f}_S(\mathbf{u}, \mathbf{u})}{\partial \mathbf{u}_1} = \left(\frac{\partial \mathbf{f}_S(\mathbf{u}_1, \mathbf{u}_2)}{\partial \mathbf{u}_1} + \frac{\partial \mathbf{f}_S(\mathbf{u}_1, \mathbf{u}_2)}{\partial \mathbf{u}_2} \right) \Big|_{\mathbf{u}_1, \mathbf{u}_2 = \mathbf{u}} = 2 \frac{\partial \mathbf{f}_S(\mathbf{u}_1, \mathbf{u}_2)}{\partial \mathbf{u}_1} \Big|_{\mathbf{u}_1, \mathbf{u}_2 = \mathbf{u}}, \quad (48)$$

where we have used the consistency of \mathbf{f}_S in the first step and the symmetry of \mathbf{f}_S in the last. Let $\mathbf{u}_1 = \mathbf{u}(x)$ and $\mathbf{u}_2 = \mathbf{u}(y)$, where x, y are independent spatial coordinates over the domain. Combining (48) with the chain rule yields that

$$\frac{\partial \mathbf{f}(\mathbf{u}(x))}{\partial x} = 2 \frac{\partial \mathbf{f}_S(\mathbf{u}(x), \mathbf{u}(y))}{\partial x} \Big|_{y=x}. \quad (49)$$

The accuracy of (49) was shown using this same approach in [6, 42]. Flux differencing was first used to systematically recover split formulations [3], which we describe in more detail in Appendix A for Burgers' equation.

4.2. Quadrature-based operators and flux differencing

We first simplify the evaluation of (49) using quadrature-based operators. For simplicity of notation, we have dropped the superscript i from one-dimensional operators. The matrix $\mathbf{D}_q = \mathbf{V}_q \mathbf{D} \mathbf{P}_q$ is a discretization of the one-dimensional operator $\frac{\partial}{\partial \hat{x}} \Pi_N$, and maps from volume quadrature points to volume quadrature points. Let $f_S(\mathbf{u}(x), \mathbf{u}(y))$ denote a scalar bivariate function. Define the matrix \mathbf{F}_S as the evaluation of $f_S(\mathbf{u}(x), \mathbf{u}(y))$ at quadrature points

$$(\mathbf{F}_S)_{ij} = f_S(\mathbf{u}(\hat{x}_i), \mathbf{u}(\hat{x}_j)), \quad 1 \leq i, j \leq N_q. \quad (50)$$

The columns of the matrix-matrix product $\mathbf{D}_q \mathbf{F}_S$ then correspond to the projection and differentiation of the univariate function $f_S(\mathbf{u}(\hat{x}), \mathbf{u}(\hat{x}_j))$ for fixed quadrature points \hat{x}_j for $j = 1, \dots, N_q$. Thus, evaluating (49) at different quadrature points is equivalent to evaluating

$$\frac{\partial \Pi_N f_S(\mathbf{u}(\hat{x}), \mathbf{u}(\hat{y}))}{\partial \hat{x}} \Big|_{\hat{y} = \hat{x}_i} = (\mathbf{D}_q \mathbf{F}_S)_{ii} = (\text{diag}(\mathbf{D}_q \mathbf{F}_S))_i. \quad (51)$$

In other words, performing flux differencing and evaluating (49) reduces to the computation of the diagonal entries of a matrix-matrix product when using quadrature-based matrices.

We can further simplify (51) using the Hadamard product, which is defined as the matrix operation \circ such that

$$(\mathbf{A} \circ \mathbf{B})_{ij} = \mathbf{A}_{ij} \mathbf{B}_{ij}, \quad (52)$$

where \mathbf{A}, \mathbf{B} are two matrices with the same row and column dimensions. The Hadamard product obeys the following properties, whose proofs can be found in [43].

Lemma 2. *Let $\mathbf{A}, \mathbf{B}, \mathbf{C}$ be square matrices of the same dimension.*

1. *The Hadamard product is commutative, with $\mathbf{A} \circ \mathbf{B} = \mathbf{B} \circ \mathbf{A}$.*
2. *The Hadamard product is linear with respect to addition and the transpose operation*

$$(\mathbf{A} + \mathbf{B}) \circ \mathbf{C} = \mathbf{A} \circ \mathbf{C} + \mathbf{B} \circ \mathbf{C}, \quad (\mathbf{A} \circ \mathbf{B})^T = \mathbf{A}^T \circ \mathbf{B}^T. \quad (53)$$

3. *The Hadamard product is related to the “diag” operation as follows:*

$$\text{diag}(\mathbf{A}\mathbf{B}^T) = (\mathbf{A} \circ \mathbf{B}) \mathbf{1}. \quad (54)$$

where $\mathbf{1}$ is the vector of all ones.

Using part 2 of Lemma 2 and the fact that \mathbf{F}_S is symmetric, we rewrite the “diag” operation as the row sum of a Hadamard product, or equivalently the multiplication of a Hadamard product with the vector of all ones

$$\text{diag}(\mathbf{D}_q \mathbf{F}_S) = (\mathbf{D}_q \circ \mathbf{F}_S) \mathbf{1}. \quad (55)$$

This yields a generalization of the derivative projection operator introduced in [41].

4.3. A one-dimensional entropy conservative DG method on a single element

Motivated by the observations in the previous section, we now construct a semi-discretely entropy conservative formulation on a single element in one spatial dimension based on the interpretation of flux differencing (49) in the previous section. This formulation involves projection and lifting matrices along with the decoupled SBP-like operator \mathbf{D}_N used in (43). We seek degree N polynomial approximations $\mathbf{u}_N(x, t)$ to the conservative variables $\mathbf{u}(x, t)$, with coefficients $\mathbf{u}_h(t)$ such that

$$\mathbf{u}_N(\hat{\mathbf{x}}, t) = \sum_{j=1}^{N_p} (\mathbf{u}_h(t))_j \phi_j(\hat{\mathbf{x}}), \quad (\mathbf{u}_h(t))_j \in \mathbb{R}^n. \quad (56)$$

Because \mathbf{u}_h consists of vectors of coefficients for each scalar component of $\mathbf{u}_N(\hat{\mathbf{x}}, t)$, from this point onward we understand the product of matrices applied to component-wise vectors like \mathbf{u}_h in a Kronecker product sense (as in [6]), e.g. $\mathbf{A}\mathbf{u}_h$ should be understood as applying \mathbf{A} to each component of \mathbf{u}_h , or applying $\mathbf{A} \otimes \mathbf{I}$ (with \mathbf{I} the $n \times n$ identity matrix) to the full vector \mathbf{u}_h .

We now introduce \mathbf{v}_h as the L^2 projection of the entropy variables and $\tilde{\mathbf{u}}$ as the evaluation (at volume and surface quadrature points) of the conservative variables in terms of the L^2 projected entropy variables.

$$\begin{aligned} \mathbf{u}_q &= \mathbf{V}_q \mathbf{u}_h, & \mathbf{v}_q &= \mathbf{v}(\mathbf{u}_q), & \mathbf{v}_h &= \mathbf{P}_q \mathbf{v}_q, \\ \tilde{\mathbf{v}} &= \begin{bmatrix} \tilde{\mathbf{v}}_q \\ \tilde{\mathbf{v}}_f \end{bmatrix} = \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \mathbf{v}_h, & \tilde{\mathbf{u}} &= \begin{bmatrix} \tilde{\mathbf{u}}_q \\ \tilde{\mathbf{u}}_f \end{bmatrix} = \mathbf{u}(\tilde{\mathbf{v}}). \end{aligned} \quad (57)$$

Here, $\mathbf{u}_q, \mathbf{v}_q$ denote the conservative variables and entropy variables (as a function of the conservation variables) evaluated at volume quadrature points. The vector $\tilde{\mathbf{v}}$ denotes the evaluations of the L^2 projection of the entropy variables at volume and surface quadrature points, while $\tilde{\mathbf{u}}$ denotes the evaluation of the conservative variables in terms of the projected entropy variables $\mathbf{u}(\Pi_N \mathbf{v})$, which will be crucial to proving discrete conservation of entropy. For the remainder of this paper, $\mathbf{u}(\Pi_N \mathbf{v})$ and $\tilde{\mathbf{u}}$ will be referred to as entropy-projected conservative variables. We note that this approach closely resembles that of [44], where entropy stable schemes were constructed for generalized SBP operators based on Gauss nodes. In [44], the flux is computed by first interpolating the entropy variables at Gauss nodes, then evaluating the conservative variables in terms of the interpolated entropy variables at a separate set of nodes.

Let \mathbf{f}^* be a numerical flux which is used for the imposition of boundary conditions. Motivated by the quadrature discretization of flux differencing (51) and its reformulation using the Hadamard product (55), we introduce the semi-discrete formulation for $\mathbf{u}_h(t)$ as

$$\begin{aligned} \frac{d\mathbf{u}_h}{dt} &= - \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} (2\mathbf{D}_N \circ \mathbf{F}_S) \mathbf{1} - \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}) (\mathbf{f}^* - \mathbf{f}(\tilde{\mathbf{u}}_f)), \\ (\mathbf{F}_S)_{ij} &= \mathbf{f}_S(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j), \quad 1 \leq i, j \leq N_q + N_f^f, \end{aligned} \quad (58)$$

where \mathbf{F}_S is a symmetric matrix (symmetry is a result of the symmetry condition of Definition 1) and $\tilde{\mathbf{u}}_j$ denotes the evaluation of the entropy-projected conservative variables at the i th quadrature point, where i indexes into the combined set of both volume and surface quadrature points.

We have the following theorem:

Theorem 2. *Let \mathbf{f}_S be an entropy conservative flux from Definition 1. Then, assuming continuity in time, the semi-discrete formulation defined by (57) and (58) satisfies*

$$\mathbf{1}^T \mathbf{W} \frac{dU(\mathbf{u}_q)}{dt} = \mathbf{1}^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) (\psi(\tilde{\mathbf{u}}_f) - \tilde{\mathbf{v}}_f^T \mathbf{f}^*) \quad (59)$$

which is an approximation of the statement of conservation of entropy (5) involving numerical quadrature and the numerical flux \mathbf{f}^*

$$\int_{\hat{D}} \frac{\partial U(\mathbf{u}_N)}{\partial t} d\hat{x} = \int_{\partial \hat{D}} \left(\psi(\mathbf{u}(\Pi_N \mathbf{v})) - (\Pi_N \mathbf{v})^T \mathbf{f}^* \right) \hat{n} d\hat{x}. \quad (60)$$

Proof. The proof borrows concepts from [41, 6] in combination with properties of the decoupled SBP-like operator \mathbf{D}_N . We first note that we can rewrite the projection and lifting operation as

$$\begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T \begin{bmatrix} \mathbf{W} & \mathbf{W}_f \end{bmatrix}. \quad (61)$$

Multiplying (58) by \mathbf{M} yields the variational form of the equation

$$\mathbf{M} \frac{d\mathbf{u}_h}{dt} = - \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T (2\mathbf{Q}_N \circ \mathbf{F}_S) \mathbf{1} - \mathbf{V}_f^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) (\mathbf{f}^* - \mathbf{f}(\tilde{\mathbf{u}}_f)). \quad (62)$$

where we have moved the matrix of quadrature weights \mathbf{W} inside the Hadamard product because multiplication of a matrix by a diagonal matrix is identical to taking the Hadamard product of both matrices [43]. We now test with the projection of the entropy variables \mathbf{v}_h^T on both sides and note that

$$\mathbf{v}_h^T \mathbf{M} = \mathbf{v}_q^T \mathbf{P}_q^T \mathbf{M} = \mathbf{v}_q^T \mathbf{W} \mathbf{V}_q \mathbf{M}^{-1} \mathbf{M} = \mathbf{v}_q^T \mathbf{W} \mathbf{V}_q. \quad (63)$$

Plugging this into the left hand side of (62), assuming continuity in time (such that the chain rule holds), and using that $\mathbf{v} = \frac{\partial U(\mathbf{u})}{\partial \mathbf{u}}$ then yields

$$\begin{aligned} \mathbf{v}_h^T \mathbf{M} \frac{d\mathbf{u}_h}{dt} &= \mathbf{v}_q^T \mathbf{W} \frac{d\mathbf{u}_q}{dt} = \sum_{i=1}^{N_q} w_i \mathbf{v}(\mathbf{u}_N(x_i, t))^T \frac{\partial \mathbf{u}_N(x_i, t)}{\partial t} = \sum_{i=1}^{N_q} w_i \frac{\partial U(\mathbf{u}_N(x_i, t))}{\partial t} \\ &= \mathbf{1}^T \mathbf{W} \frac{dU(\mathbf{u}_q)}{dt} \approx \int_{\hat{D}} \frac{\partial U(\mathbf{u}_N)}{\partial t} d\hat{x}. \end{aligned} \quad (64)$$

We treat the right hand side next. The contribution involving the numerical flux \mathbf{f}^* yields

$$-\mathbf{v}_h^T \mathbf{V}_f^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) (\mathbf{f}^* - \mathbf{f}(\tilde{\mathbf{u}})) = -\tilde{\mathbf{v}}_f^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) (\mathbf{f}^* - \mathbf{f}(\tilde{\mathbf{u}}_f)). \quad (65)$$

For the right hand side terms involving \mathbf{F}_S , testing with \mathbf{v}_h^T and using Theorem 1 yields

$$-\left(\begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix} \mathbf{v}_h\right)^T ((\mathbf{B}_N + \mathbf{Q}_N - \mathbf{Q}_N^T) \circ \mathbf{F}_S) \mathbf{1} = -\tilde{\mathbf{v}}^T ((\mathbf{B}_N + \mathbf{Q}_N - \mathbf{Q}_N^T) \circ \mathbf{F}_S) \mathbf{1}. \quad (66)$$

The first term involving \mathbf{B}_N is simplified by noting that \mathbf{B}_N is diagonal, and that $\mathbf{B}_N \circ \mathbf{F}_S$ extracts the values of \mathbf{F}_S at face quadrature points. These values correspond to evaluations of the entropy conservative numerical flux $\mathbf{f}_S(\tilde{\mathbf{u}}_f, \tilde{\mathbf{u}}_f)$, where $\tilde{\mathbf{u}}_f = \mathbf{u}(\mathbf{V}_f \mathbf{v}_h)$ is the evaluation of the entropy-projected conservative variables at surface quadrature points. By the consistency condition of Definition 1, $\mathbf{f}_S(\tilde{\mathbf{u}}_f, \tilde{\mathbf{u}}_f) = \mathbf{f}(\tilde{\mathbf{u}}_f)$. As a result,

$$(\mathbf{B}_N \circ \mathbf{F}_S) \mathbf{1} = \left(\begin{bmatrix} 0 & \\ & \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) \text{diag}(\mathbf{f}(\tilde{\mathbf{u}}_f)) \end{bmatrix} \right) \mathbf{1} = \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) \mathbf{f}(\tilde{\mathbf{u}}_f). \quad (67)$$

Combining this with (66) and (65) yields

$$\begin{aligned} -\tilde{\mathbf{v}}^T (\mathbf{B}_N \circ \mathbf{F}_S) \mathbf{1} - \tilde{\mathbf{v}}_f^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) (\mathbf{f}^* - \mathbf{f}(\tilde{\mathbf{u}})) &= -\tilde{\mathbf{v}}_f^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) \mathbf{f}(\tilde{\mathbf{u}}_f) - \tilde{\mathbf{v}}_f^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) (\mathbf{f}^* - \mathbf{f}(\tilde{\mathbf{u}}_f)) \\ &= -\mathbf{1}^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) \tilde{\mathbf{v}}_f^T \mathbf{f}^* \approx -\int_{\partial \hat{D}} (\Pi_N \mathbf{v})^T \mathbf{f}^* \hat{\mathbf{n}} d\hat{x}. \end{aligned} \quad (68)$$

where we have used the fact that $\mathbf{W}_f \text{diag}(\hat{\mathbf{n}})$ is diagonal to commute $\tilde{\mathbf{v}}_f^T$. The remaining terms in (66) involving \mathbf{Q}_N are

$$\tilde{\mathbf{v}}^T ((\mathbf{Q}_N - \mathbf{Q}_N^T) \circ \mathbf{F}_S) \mathbf{1} = \tilde{\mathbf{v}}^T (\mathbf{Q}_N \circ \mathbf{F}_S - \mathbf{Q}_N^T \circ \mathbf{F}_S) \mathbf{1} = \tilde{\mathbf{v}}^T (\mathbf{Q}_N \circ \mathbf{F}_S) \mathbf{1} - \mathbf{1}^T (\mathbf{Q}_N \circ \mathbf{F}_S) \tilde{\mathbf{v}}, \quad (69)$$

where we have used

$$\tilde{\mathbf{v}}^T (\mathbf{Q}_N^T \circ \mathbf{F}_S) \mathbf{1} = \mathbf{1}^T (\mathbf{Q}_N^T \circ \mathbf{F}_S)^T \tilde{\mathbf{v}} = \mathbf{1}^T (\mathbf{Q}_N \circ \mathbf{F}_S) \tilde{\mathbf{v}} \quad (70)$$

by the transpose property of Lemma 2 and symmetry of \mathbf{F}_S . Writing out (69) using sum notation

$$\begin{aligned} \tilde{\mathbf{v}}^T (\mathbf{Q}_N \circ \mathbf{F}_S) \mathbf{1} - \mathbf{1}^T (\mathbf{Q}_N \circ \mathbf{F}_S) \tilde{\mathbf{v}} &= \sum_{i,j=1}^{N_q} (\mathbf{Q}_N)_{ij} \tilde{\mathbf{v}}_i^T (\mathbf{F}_S)_{ij} - (\mathbf{Q}_N)_{ij} \tilde{\mathbf{v}}_j^T (\mathbf{F}_S)_{ij} \\ &= \sum_{i,j=1}^{N_q} (\mathbf{Q}_N)_{ij} (\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j)^T \mathbf{f}_S(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j). \end{aligned} \quad (71)$$

Using the conservation condition in Definition 1 of an entropy conservative flux and the fact that $\tilde{\mathbf{u}}_i = \mathbf{u}(\tilde{\mathbf{v}}_i)$,

$$(\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j)^T \mathbf{f}_S(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j) = \psi(\tilde{\mathbf{u}}_i) - \psi(\tilde{\mathbf{u}}_j). \quad (72)$$

Let $\psi_i = \psi(\tilde{\mathbf{u}}_i)$. Inserting (72) into (71) yields

$$\begin{aligned} \sum_{i,j=1}^{N_q} (\mathbf{Q}_N)_{ij} (\tilde{\mathbf{v}}_i - \tilde{\mathbf{v}}_j)^T \mathbf{f}_S(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j) &= \mathbf{1}^T \mathbf{Q}_N \psi - \psi^T \mathbf{Q}_N \mathbf{1} = \mathbf{1}^T \mathbf{Q}_N \psi \\ &= \mathbf{1}^T (\mathbf{B}_N - \mathbf{Q}_N^T) \psi = \mathbf{1}^T \mathbf{B}_N \psi \\ &= \mathbf{1}^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) \psi(\tilde{\mathbf{u}}_f) \approx \int_{\partial \hat{D}} \hat{\mathbf{n}} \psi(\tilde{\mathbf{u}}_f) d\hat{x}, \end{aligned} \quad (73)$$

where Theorem 1 implies that $\mathbf{Q}_N \mathbf{1} = 0$.

□

It is important to emphasize that conservation of entropy does not, in general, imply stability of the numerical scheme unless the solution satisfies additional constraints. For example, the transformation between conservative and entropy variables for the compressible Euler and Navier-Stokes equations is well-defined only if the density and internal energy are positive, and steps must be taken in any numerical scheme to guarantee that the discrete solution satisfies such constraints. For DG methods, this is most commonly done using bound and positivity-preserving limiters [45, 46].

It is also worth pointing out that, for specific choices of basis and quadrature, the formulation (58) can be significantly simplified. 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 $\mathbf{V}_q, \mathbf{P}_q$ to identity matrices. Furthermore, if quadrature points coincide with boundary points (as with DG-SEM methods), the lift matrix \mathbf{L}_q 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.

4.4. Local conservation

We next show that (58) is locally conservative. The proof is very similar to that of [6]. Recall the matrix variational form (62)

$$\mathbf{M} \frac{d\mathbf{u}_h}{dt} = - \begin{bmatrix} \mathbf{V}_q \\ \mathbf{V}_f \end{bmatrix}^T (2\mathbf{Q}_N \circ \mathbf{F}_S) \mathbf{1} - \mathbf{V}_f^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) (\mathbf{f}^* - \mathbf{f}(\tilde{\mathbf{u}}_f)). \quad (74)$$

To prove local conservation, we test with $(\mathbf{P}_q \mathbf{1})^T$, which gives

$$\mathbf{1}^T \mathbf{W} \frac{d\mathbf{u}_q}{dt} = -\mathbf{1}^T (2\mathbf{Q}_N \circ \mathbf{F}_S) \mathbf{1} - \mathbf{1}^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) (\mathbf{f}^* - \mathbf{f}(\tilde{\mathbf{u}}_f)). \quad (75)$$

Using the commutativity of the Hadmard product (Lemma 2), symmetry of \mathbf{F}_S , and Theorem 1,

$$\begin{aligned} \mathbf{1}^T (2\mathbf{Q}_N \circ \mathbf{F}_S) \mathbf{1} &= \mathbf{1}^T (\mathbf{Q}_N \circ \mathbf{F}_S) \mathbf{1} + \mathbf{1}^T (\mathbf{Q}_N \circ \mathbf{F}_S) \mathbf{1} = \mathbf{1}^T (\mathbf{Q}_N \circ \mathbf{F}_S) \mathbf{1} + \mathbf{1}^T (\mathbf{Q}_N^T \circ \mathbf{F}_S) \mathbf{1} \\ &= \mathbf{1}^T ((\mathbf{Q}_N + \mathbf{Q}_N^T) \circ \mathbf{F}_S) \mathbf{1} = \mathbf{1}^T (\mathbf{B}_N \circ \mathbf{F}_S) \mathbf{1} = \mathbf{1}^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) \mathbf{f}(\tilde{\mathbf{u}}_f). \end{aligned} \quad (76)$$

Plugging this back into (75) yields

$$\mathbf{1}^T \mathbf{W} \frac{d\mathbf{u}_q}{dt} + \mathbf{1}^T \mathbf{W}_f \text{diag}(\hat{\mathbf{n}}) \mathbf{f}^* = 0 \quad (77)$$

which is a quadrature approximation to the conservation condition

$$\int_{\hat{D}} \frac{\partial \mathbf{u}}{\partial t} d\hat{x} + \int_{\partial \hat{D}} \mathbf{f}^* \text{diag}(\hat{\mathbf{n}}) d\hat{x} = 0. \quad (78)$$

5. Entropy stable DG methods on multiple elements and in higher dimensions

In this section, we discuss the extension of entropy conservative DG schemes to multiple elements, the addition of interface dissipation, and the construction of entropy stable schemes in higher dimensions.

5.1. Multiple elements

We now extend entropy conservative schemes to multiple elements in one dimension, where the domain Ω is broken up into K non-overlapping intervals D^k with outward normals n . Each interval can be represented as the affine mapping Φ^k of the reference interval \hat{D} . Because this mapping is affine, J^k (the determinant of the Jacobian of Φ^k) is constant over each element.

An entropy conservative formulation can be constructed by modifying the lifting matrix for dimensional consistency. Let $\mathbf{L}_q = \frac{1}{J^k} \hat{\mathbf{L}}_q$, where $\hat{\mathbf{L}}_q$ is the lifting matrix over the reference element, and let $\mathbf{n} = [-1, 1]$

be the vector containing values of the outward normal at face quadrature points on the element D^k . An entropy stable formulation over D^k is given by

$$\begin{aligned} \frac{d\mathbf{u}_h}{dt} &= - \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} (2\mathbf{D}_N \circ \mathbf{F}_S) \mathbf{1} - \mathbf{L}_q \text{diag}(\mathbf{n}) (\mathbf{f}^* - \mathbf{f}(\tilde{\mathbf{u}}_f)), \\ (\mathbf{F}_S)_{ij} &= \mathbf{f}_S(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j), \quad 1 \leq i, j \leq N_q + N_q^f, \\ \mathbf{f}^* &= \mathbf{f}_S(\tilde{\mathbf{u}}_f^+, \tilde{\mathbf{u}}_f), \quad \text{at interior interfaces,} \end{aligned} \quad (79)$$

where $\tilde{\mathbf{u}}_f^+$ is the value of the entropy-projected conservative variables on the neighboring element.

Let $\hat{\mathbf{x}}_i^f$ denote face quadrature points on the reference element \hat{D} , and let $\mathbf{x}_i^f = \Phi^k(\hat{\mathbf{x}}_i^f)$ denote the mapping of the reference face points to physical face points of D^k . Let $\mathbf{W}_{\partial\Omega}$ be the diagonal boundary matrix such that

$$(\mathbf{W}_{\partial\Omega})_{ii} = \begin{cases} (\mathbf{W}_f)_{ii}, & \text{if } \mathbf{x}_i^f \text{ is on the domain boundary } \partial\Omega \\ 0, & \text{otherwise.} \end{cases} \quad (80)$$

In other words, $\mathbf{W}_{\partial\Omega}$ is zero for any interior elements and is equal to \mathbf{W}_f at face quadrature points \mathbf{x}_i^f which coincide with the boundary $\partial\Omega$. Then, (79) satisfies the following theorem:

Theorem 3. *Let \mathbf{f}_S be an entropy conservative flux from Definition 1. The scheme (79) is locally conservative and satisfies*

$$\sum_{k=1}^K \mathbf{1}^T J^k \mathbf{W} \frac{d\mathbf{u}_q}{dt} = \sum_{k=1}^K \mathbf{1}^T \mathbf{W}_{\partial\Omega} \text{diag}(\mathbf{n}) (\psi(\tilde{\mathbf{u}}_f) - \tilde{\mathbf{v}}_f^T \mathbf{f}^*), \quad (81)$$

which is an approximation of (5) involving quadrature and the numerical flux on the boundary \mathbf{f}^*

$$\int_{\Omega} \frac{\partial U(\mathbf{u}_N)}{\partial t} dx = \int_{\partial\Omega} \left(\psi(\mathbf{u}(\Pi_N \mathbf{v})) - (\Pi_N \mathbf{v})^T \mathbf{f}^* \right) n dx. \quad (82)$$

Proof. The proof of local conservation over each element D^k is the same as the one-element case shown in Section 4.4. For multiple elements, showing conservation of entropy is done by first applying the one-element proof of conservation of entropy over each element D^k , summing over all elements, then cancelling shared interface terms. Without loss of generality, we assume a periodic domain such that all interfaces are interior interfaces. Scaling by J^k on both sides, applying the one-element proof of conservation of entropy in Theorem 2, and summing the results gives

$$\sum_{k=1}^K \mathbf{1}^T J^k \mathbf{W} \frac{d\mathbf{u}_q}{dt} = \sum_{k=1}^K (\mathbf{1}^T \mathbf{W}_f \text{diag}(\mathbf{n}) \psi(\tilde{\mathbf{u}}_f) - \mathbf{1}^T \mathbf{W}_f \text{diag}(\mathbf{n}) \tilde{\mathbf{v}}_f^T \mathbf{f}_S(\tilde{\mathbf{u}}^+, \tilde{\mathbf{u}})). \quad (83)$$

Note that $n^+ = -n$, where n^+ denotes the outward normal on a neighboring element. Then, splitting interface contributions between neighboring elements yields

$$\begin{aligned} - \sum_{k=1}^K \tilde{\mathbf{v}}_f^T \mathbf{W}_f \text{diag}(\mathbf{n}) \mathbf{f}_S(\tilde{\mathbf{u}}^+, \tilde{\mathbf{u}}) &= \sum_{k=1}^K \frac{1}{2} \mathbf{1}^T \mathbf{W}_f \text{diag}(\mathbf{n}) \left((\tilde{\mathbf{v}}_f)^+ - \tilde{\mathbf{v}}_f \right)^T \mathbf{f}_S(\tilde{\mathbf{u}}^+, \tilde{\mathbf{u}}), \\ &= \sum_{k=1}^K \frac{1}{2} \mathbf{1}^T \mathbf{W}_f \text{diag}(\mathbf{n}) (\psi(\tilde{\mathbf{u}}^+) - \psi(\tilde{\mathbf{u}})), \end{aligned} \quad (84)$$

where we have used the symmetry and conservation conditions of Definition 1. Returning contributions involving $\psi(\tilde{\mathbf{u}}^+)$ to neighboring elements of D^k cancels interface terms, such that

$$\sum_{k=1}^K \mathbf{1}^T J^k \mathbf{W} \frac{d\mathbf{u}_q}{dt} = 0. \quad (85)$$

When the domain is not periodic, (84) is required only on interior interfaces, such that contributions from \mathbf{f}^* remain on the boundaries of the domain. \square

We note that, since \mathbf{f}^* is a function of $\tilde{\mathbf{u}}$ and not \mathbf{u} , it is not immediately clear that this statement of local conservation satisfies the conditions of the classic Lax-Wendroff theorem. However, this form of local conservation does satisfy a generalized definition of local conservation, which is sufficient to guarantee convergence to a weak solution under mesh refinement [47].

We also 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 matrix dissipation terms [48, 23]) or Lax-Friedrichs penalization in terms of either the entropy variables [2] or the conservative variables [6]. For the numerical experiments presented in Section 6, we apply a local Lax-Friedrichs penalization in terms of the entropy-projected conservative variables, augmenting the flux function at element interfaces with the additional term

$$\mathbf{f}_S(\mathbf{u}_L, \mathbf{u}_R) \Rightarrow \mathbf{f}_S(\mathbf{u}_L, \mathbf{u}_R) - \frac{\lambda}{2} \llbracket \tilde{\mathbf{u}} \rrbracket, \quad (86)$$

where λ is an estimate of the maximum eigenvalue of the flux Jacobian. It is not immediately obvious that the Lax-Friedrichs penalization dissipate entropy when multiplied by the entropy variables; however, it was shown in [6, Corollary 3.2] that the local Lax-Friedrichs flux is entropy dissipative when λ is an appropriately chosen estimate of the average wave-speed. The result can be extended to the current setting by noting that, since conservation of entropy requires testing with the projection of the entropy variables, the jump term should involve the evaluations of the entropy-projected conservative variables $\tilde{\mathbf{u}} = \mathbf{u}(\Pi_N \mathbf{v})$ in order to guarantee entropy dissipation.

Finally, Theorem 3 implies that if boundary conditions are enforced in such a way that \mathbf{f}^* is entropy stable at the boundaries (see, for example, [49, 50, 6]), then the semi-discrete scheme will satisfy a discrete analogue of the global entropy inequality

$$\int_{\Omega} \frac{\partial U(\mathbf{u}_N)}{\partial t} dx \leq \int_{\partial\Omega} \left(\psi(\mathbf{u}(\Pi_N \mathbf{v})) - (\Pi_N \mathbf{v})^T \mathbf{f}^* \right) n dx \leq 0. \quad (87)$$

A similar statement of global entropy dissipation also holds for periodic boundary conditions.

5.2. Higher dimensions

In this section, we describe the construction of entropy stable DG methods for nonlinear conservation laws in d dimensions on a domain Ω

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{i=1}^d \frac{\partial \mathbf{f}_i(\mathbf{u})}{\partial x_i} = 0. \quad (88)$$

We assume that the d -dimensional domain $\Omega \in \mathbb{R}^d$ is decomposed into non-overlapping elements D^k , and that D^k is the image of the reference element \hat{D} under an affine mapping $\mathbf{x} = \Phi^k(\hat{\mathbf{x}})$ (where $\hat{\mathbf{x}}$ denotes coordinates on the reference element). We approximate the solution over a physical element by mapping $P^N(\hat{D})$ to D^k under Φ^k

$$P^N(D^k) = \Phi^k \circ P^N(\hat{D}). \quad (89)$$

Volume integrals over each physical element D^k can be mapped to the reference element. In two and three dimensions, we also assume that each face of the element D^k is the image of some reference face, such that surface integrals can be mapped from the physical element boundary ∂D^k to reference element boundary $\partial \hat{D}$. Thus, we have

$$\int_{D^k} f(\mathbf{x}) d\mathbf{x} = \int_{\hat{D}} f(\mathbf{x}) J^k d\hat{\mathbf{x}}, \quad \int_{\partial D^k} f(\mathbf{x}) d\mathbf{x} = \int_{\partial \hat{D}} f(\mathbf{x}) J_f^k d\hat{\mathbf{x}}, \quad (90)$$

where J^k is the determinant of the Jacobian of Φ^k and J_f^k is the Jacobian factor of the face mapping. We assume both mappings to be affine in this work, such that both J^k is constant over each element D^k and J_f^k is constant over each face.

To construct entropy stable schemes in higher dimensions, we require a generalization of the entropy conservative fluxes defined in Definition 1.

Definition 2. Let $\mathbf{f}_{i,S}(\mathbf{u}_L, \mathbf{u}_R)$ be a bivariate function which is symmetric and consistent with the i th coordinate flux function $\mathbf{f}_i(\mathbf{u})$. The numerical flux $\mathbf{f}_{i,S}(\mathbf{u}_L, \mathbf{u}_R)$ is entropy conservative if, for entropy variables $\mathbf{v}_L = \mathbf{v}(\mathbf{u}_L)$, $\mathbf{v}_R = \mathbf{v}(\mathbf{u}_R)$

$$\begin{aligned} (\mathbf{v}_L - \mathbf{v}_R)^T \mathbf{f}_{i,S}(\mathbf{u}_L, \mathbf{u}_R) &= (\psi_{i,L} - \psi_{i,R}), \\ \psi_{i,L} &= \psi_i(\mathbf{v}(\mathbf{u}_L)), \quad \psi_{i,R} = \psi_i(\mathbf{v}(\mathbf{u}_R)), \quad i = 1, \dots, d. \end{aligned} \quad (91)$$

We now define SBP-like operators on mapped elements in multiple dimensions. Let $\mathbf{G}^k = \frac{\partial \Phi^k}{\partial \mathbf{x}}$ be the matrix of geometric factors. Since the mapping Φ^k is assumed to be affine, the entries of \mathbf{G}^k are constant over each element. Using these factors, we can define

$$\mathbf{D}_N^i = \sum_{j=1}^d \mathbf{G}_{ij}^k \widehat{\mathbf{D}}_N^j, \quad \mathbf{L}_q = \widehat{\mathbf{L}}_q \text{diag} \left(\frac{\mathbf{J}_f^k}{J^k} \right), \quad (92)$$

where $\widehat{\mathbf{D}}_N^j$ and $\widehat{\mathbf{L}}_q$ are SBP-like operators defined on the reference element \widehat{D} , and \mathbf{J}_f^k is a vector of the values of J_f^k at surface quadrature points. One can show the relation between the geometric factors \mathbf{G}_{ij} and components of the outward normals \mathbf{n}_i

$$\sum_{j=1}^d J^k \mathbf{G}_{ij} \widehat{\mathbf{n}}_i \widehat{J}_f = \mathbf{n}_i J_f^k, \quad (93)$$

where \widehat{J}_f is the face Jacobian factor of the mapping from faces of the reference element to the reference face [7].⁶ We assume \widehat{J}_f is pre-multiplied into the surface quadrature weights.

Let \mathbf{W}_N^k be the Jacobian-weighted diagonal matrix of volume and surface quadrature points

$$\mathbf{W}_N^k = \begin{pmatrix} \mathbf{W} J^k & \\ & \mathbf{W}_f \text{diag} \left(\mathbf{J}_f^k / \widehat{\mathbf{J}}_f \right) \end{pmatrix}, \quad (94)$$

where $\widehat{\mathbf{J}}_f$ is the vector containing values of \widehat{J}_f at surface quadrature points, and $\mathbf{J}_f^k / \widehat{\mathbf{J}}_f$ is a vector corresponding to the entry-wise division of \mathbf{J}_f^k by $\widehat{\mathbf{J}}_f$. Then, \mathbf{D}_N^i and \mathbf{L}_q satisfy the following analogue of Theorem 1 with respect to \mathbf{W}_N^k

$$\mathbf{W}_N^k \mathbf{D}_N^i + (\mathbf{W}_N^k \mathbf{D}_N^i)^T = \mathbf{B}_N^i, \quad \mathbf{B}_N^i = \begin{pmatrix} 0 & \\ & \mathbf{W}_f \text{diag} \left(\mathbf{J}_f^k / \widehat{\mathbf{J}}_f \right) \text{diag}(\mathbf{n}_i) \end{pmatrix}. \quad (95)$$

A semi-discrete scheme can then be constructed over each element using (92)

$$\begin{aligned} \frac{d\mathbf{u}_h}{dt} &= - \sum_{i=1}^d \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} (2\mathbf{D}_N^i \circ \mathbf{F}_{i,S}) \mathbf{1} - \mathbf{L}_q \text{diag}(\mathbf{n}_i) (\mathbf{f}_i^* - \mathbf{f}_i(\tilde{\mathbf{u}}_f)), \\ (\mathbf{F}_{i,S})_{ij} &= \mathbf{f}_{i,S}(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j), \quad 1 \leq i, j \leq N_q + N_q^f. \end{aligned} \quad (96)$$

⁶The factor \widehat{J}_f appears, for example, for triangles, where the reference triangle is usually taken to be a right triangle. For this reference triangle, two faces are of the same size, but the hypotenuse face is larger and will thus have a different value of \widehat{J}_f from the other two faces when mapping to the reference face.

where \mathbf{f}_i^* is the i th component of the numerical flux and $\tilde{\mathbf{u}}$ is again the evaluation of the entropy-projected conservative variables.

Let $\mathbf{W}_{\partial\Omega}^k = \mathbf{W}_{\partial\Omega} \text{diag}(\mathbf{J}_f^k / \hat{\mathbf{J}}_f)$, where $\mathbf{W}_{\partial\Omega}$ is the diagonal boundary matrix defined in (80). The multi-dimensional scheme satisfies the following theorem:

Theorem 4. *Let \mathbf{f}_S be a higher dimensional entropy conservative flux from Definition 2. The scheme (96) is locally conservative and satisfies*

$$\sum_{k=1}^K \mathbf{1}^T J^k \mathbf{W} \frac{dU(\mathbf{u}_q)}{dt} = \sum_{k=1}^K \sum_{i=1}^d \mathbf{1}^T \mathbf{W}_{\partial\Omega} \text{diag}(\mathbf{n}_i) (\psi_i(\tilde{\mathbf{u}}_f) - \tilde{\mathbf{v}}_f^T \mathbf{f}_i^*), \quad (97)$$

which is an approximation of the higher dimensional generalization of the conservation of entropy (5) involving numerical quadrature and the boundary numerical fluxes \mathbf{f}_i^*

$$\int_{\Omega} \frac{\partial U(\mathbf{u}_N)}{\partial t} dx = \int_{\partial\Omega} \left(\psi_i(\mathbf{u}(\Pi_N \mathbf{v})) - (\Pi_N \mathbf{v})^T \mathbf{f}_i^* \right) \cdot \mathbf{n} d\mathbf{x}. \quad (98)$$

Proof. The proofs of both stability and conservation are shown by applying (95) and the one-dimensional proofs of Theorem 2 and Theorem 3 along each of the i th coordinate directions. \square

6. Numerical experiments: the compressible Euler equations

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 [51]. Following the derivation of stable timestep restrictions in [39], we define the timestep Δt to be

$$\Delta t = \text{CFL} \times \frac{h}{C_N}, \quad C_N = \frac{(N+1)^2}{2}, \quad (99)$$

where C_N is the one-dimensional constant in the trace inequality [52], and CFL is a user-defined constant.

We note that the numerical implementations used here are oblivious to the choice of basis, and the discretization is specified completely by the choice of quadrature. For example, when GLL quadratures are used, an entropy conservative/stable DG-SEM discretization is recovered [2, 6, 41], and when a Gauss quadrature with $(N+1)$ points is used, generalized SBP-DG methods are recovered [28, 37].

6.1. One-dimensional experiments

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

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} &= 0, \\ \frac{\partial \rho u}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} &= 0, \\ \frac{\partial E}{\partial t} + \frac{\partial(u(E + p))}{\partial x} &= 0. \end{aligned} \quad (100)$$

We assume an ideal gas, such that the pressure satisfies the constitutive relation $p = (\gamma - 1)(E - \frac{1}{2}\rho u^2)$, 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 [53]. 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 [30]. This leads to $U(\mathbf{u})$ of the form

$$U(\mathbf{u}) = -\frac{\rho s}{\gamma - 1}, \quad (101)$$

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

$$v_1 = \frac{E - \rho e(\gamma + 1 - s)}{\rho e}, \quad v_2 = \frac{\rho u}{\rho e}, \quad v_3 = -\frac{\rho}{\rho e}, \quad (102)$$

where $\rho e = E - \frac{1}{2}\rho u^2$ is the specific internal energy. The inverse mapping is given by

$$\rho = -(\rho e)v_3, \quad \rho u = (\rho e)v_2, \quad E = (\rho e)\left(1 - \frac{v_2^2}{2v_3}\right), \quad (103)$$

where ρe and s in terms of the entropy variables are

$$\rho e = \left(\frac{(\gamma - 1)}{(-v_3)}\right)^{1/(\gamma-1)} e^{\frac{-s}{\gamma-1}}, \quad s = \gamma - v_1 + \frac{v_2^2}{2v_3}. \quad (104)$$

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

$$\rho \geq \rho_0 > 0, \quad p \geq p_0 > 0. \quad (105)$$

This can be achieved using positivity-preserving limiters [45, 46]. However, these have not been implemented in our numerical simulations.

Examples of entropy conservative flux functions can be found in [54, 48]. In this work, we utilize the flux function $\mathbf{f}_S(\mathbf{u}_L, \mathbf{u}_R)$ introduced by Chandreshekar [48], whose components are given as

$$\begin{aligned} f_S^1(\mathbf{u}_L, \mathbf{u}_R) &= \{\{\rho\}\}^{\log} \{\{u\}\} \\ f_S^2(\mathbf{u}_L, \mathbf{u}_R) &= \frac{\{\{\rho\}\}}{2\{\{\beta\}\}} + \{\{u\}\} f_S^1 \\ f_S^3(\mathbf{u}_L, \mathbf{u}_R) &= f_S^1 \left(\frac{1}{2(\gamma - 1)\{\{\beta\}\}^{\log}} - \frac{1}{2} \{\{u^2\}\} \right) + \{\{u\}\} f_S^2, \end{aligned} \quad (106)$$

where we have introduced the inverse temperature β

$$\beta = \frac{\rho}{2p} \quad (107)$$

and the logarithmic mean

$$\{\{u\}\}^{\log} = \frac{u_L - u_R}{\log u_L - \log u_R}. \quad (108)$$

We note that, because the direct evaluation of the logarithmic mean is numerically sensitive for $u_L \approx u_R$, when $|u_L - u_R| < \epsilon$ we switch to evaluation using a high order accurate expansion introduced by Ismail and Roe [54].

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.

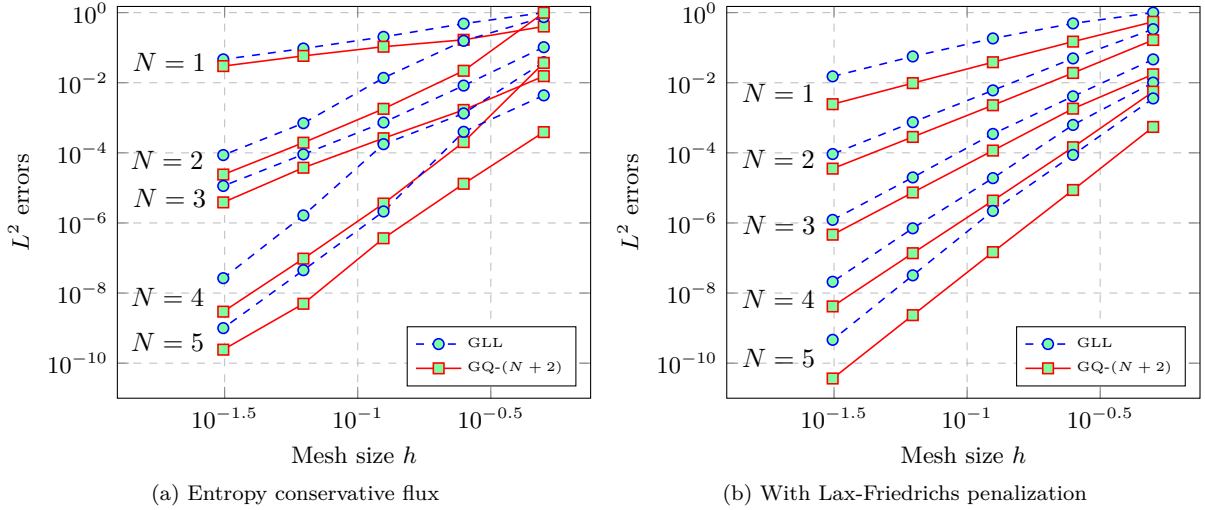


Figure 1: L^2 errors under mesh refinement for entropy conservative and Lax-Friedrichs fluxes under both Gauss-Legendre-Lobatto (GLL) and over-integrated ($N+2$) point Gauss quadrature (GQ-($N+2$)). Both sets of errors are evaluated using a GQ-($N+5$) quadrature rule.

We also present results which utilize the dissipative local Lax-Friedrichs flux described in Section 5, where the value of λ is estimated by

$$\lambda = \max_{\mathbf{u}^+, \mathbf{u}} \{|\mathbf{u}| + c\}, \quad c = \sqrt{\frac{\gamma p}{\rho}}. \quad (109)$$

We will refer to the combination of the entropy conservative flux with Lax-Friedrichs dissipation as the “Lax-Friedrichs” flux.

6.1.1. Smooth entropy wave solution

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

$$\rho(x, t) = 2 + \sin(\pi(x - t)), \quad u(x, t) = 1, \quad p(x, t) = 1. \quad (110)$$

We compute the L^2 error in the conservative variables

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2}^2 = \|\rho - \rho_h\|_{L^2}^2 + \|\rho u - \rho u_h\|_{L^2}^2 + \|E - E_h\|_{L^2}^2 \quad (111)$$

at final time $T = .7$ using both the entropy conservative and local Lax-Friedrichs fluxes and a CFL of .125. For these experiments, we compare two quadrature rules:

1. the ($N+1$) point Gauss-Legendre-Lobatto rule (referred to as “GLL”),
2. an over-integrated ($N+2$) point Gauss quadrature rule (referred to as “GQ-($N+2$)”).

The L^2 error is evaluated using a more accurate $N+5$ point Gauss quadrature rule.

Figure 1 shows computed L^2 errors under mesh refinement for both GLL and GQ-($N+2$) quadrature rules with and without Lax-Friedrichs penalization. We do not compare against the ($N+1$) point Gauss quadrature rule or quadrature rules with $N_q > (N+2)$, as the L^2 errors are very similar to the GQ-($N+2$) case. 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 the optimal $O(h^{N+1})$ convergence rate for both quadrature choices.

	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$
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. Even-odd decoupling of the convergence rates are observed for the entropy conservative flux, while optimal convergence rates are observed for both GLL and GQ- $(N + 2)$ quadratures when using Lax-Friedrichs penalization.

6.1.2. Discontinuous profile on a periodic domain

Next, we examine the discrete evolution of entropy by evolving a discontinuous initial profile to final time $T = 2$ on the domain $[-1, 1]$. We initialize the density and velocity to be

$$\rho(x, t) = \begin{cases} 3 & |x| < 1/2 \\ 2 & \text{otherwise,} \end{cases} \quad u(x, t) = 0, \quad p(x, t) = \rho^\gamma. \quad (112)$$

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 in the figure as “EC” and “LF”, respectively) 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 next examine the change in entropy over time. 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 in the observation that the numerical change in entropy

$$\Delta U(t) = U(\mathbf{u}(x, t)) - U(\mathbf{u}(x, 0)) \quad (113)$$

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.

Figure 2b shows the evolution of the integral of $\Delta U(t)$ to final time $T = 2$ for both entropy conservative and Lax-Friedrichs fluxes at various CFL numbers using a GQ- $(N + 2)$ quadrature rule. We focus on this rule, as the conservation of entropy for the GLL quadrature rule has been established in the literature theoretically and numerically for the compressible Euler equations [1, 41, 6]. Additionally, we note that the semi-discrete conservation of the integrated entropy depends on the strength of the quadrature rule utilized. Comparisons between GLL and GQ- $(N + 2)$ quadrature rules introduce inconsistency due to the fact that the former rule is exact for degree $2N - 1$ polynomials, while the latter is a stronger rule and is exact for $2N + 3$ polynomials.

For the entropy conservative flux, we observe that $\Delta U(t)$ decreases as the CFL and timestep 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 $\Delta t \rightarrow 0$. We also observe that $\Delta U(t)$ does not change significantly as a function of the CFL for the Lax-Friedrichs flux, indicating that the non-zero change in entropy in this case is due to the effect of the dissipative flux rather than the time discretization. We also compute the convergence rate of $\Delta U(T)$ to zero with respect to the timestep Δt , as shown in Figure 2c. Despite the fact that a fourth order time-stepper is used, we observe nearly fifth order convergence. This phenomena is not observed in two dimensions, as we show in Section 6.2.

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

$$\delta(t) = \left| \left(D_h^x \mathbf{f}_S(\tilde{\mathbf{u}}(x), \tilde{\mathbf{u}}(y)) \right) \Big|_{y=x}, \Pi_N \mathbf{v} \right) \Big|_\Omega, \quad 0 \leq t \leq T. \quad (114)$$

We utilize the non-dissipative entropy conservative flux and evolve the initial profile (123) until time $T = 1$ using a GQ- $(N + 2)$ quadrature rule. From the proof of entropy conservation, we expect $\delta_{\max} = \max_{t \in (0, T)} \delta$

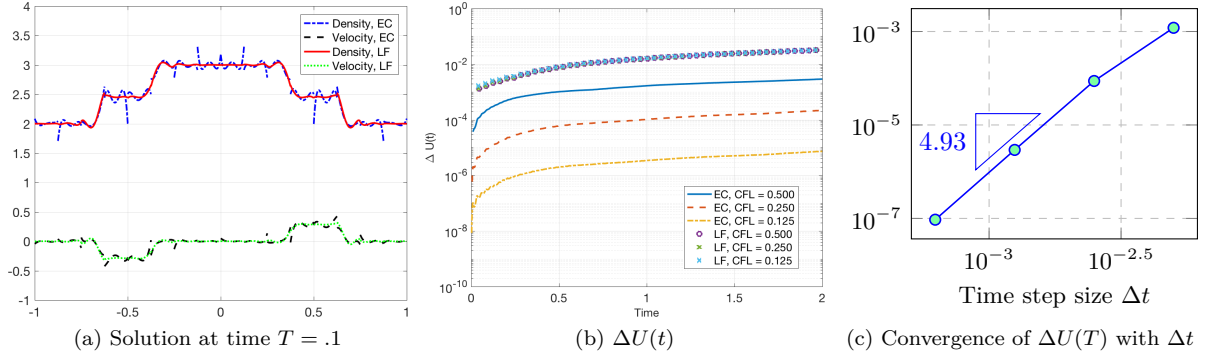


Figure 2: Solution snapshot and change in entropy $\Delta U(t)$ for both entropy conservative (EC) and Lax-Friedrichs (LF) fluxes using a $GQ-(N+2)$ quadrature rule. The convergence of the change in entropy $\Delta U(t)$ at the final time $T = 2$ converges to zero as $O(\Delta t^{4.93})$, which is greater than the order of the 4th order time-stepper used.

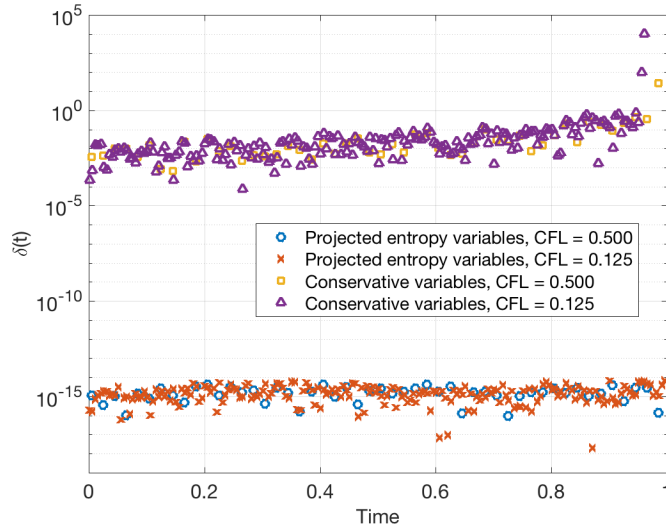


Figure 3: Comparison of $\delta(t)$ when evaluating the flux function f_S using conservative variables and projected entropy variables. A $GQ-(N+2)$ quadrature rule is used.

to be machine zero. In practice, we have found that δ_{\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 [54]) results in $\delta_{\max} = O(10^{-10})$. Decreasing ϵ to 10^{-3} reduces δ_{\max} to $O(10^{-14})$, and decreasing ϵ further to 1×10^{-4} reduces δ_{\max} to 10^{-15} . Smaller values of ϵ do result in observable changes to δ_{\max} , and we do not observe any significant dependence of δ_{\max} on other discretization parameters.

Next, in order to illustrate the importance of evaluating the flux in terms of the projected entropy variables $\Pi_N \mathbf{v}$, we compute $\delta(t)$ while evaluating the flux function f_S directly in terms of the conservative variables $\mathbf{u}(x)$ and in terms of the entropy-projected conservative variables $\mathbf{u}((\Pi_N \mathbf{v})(x))$. 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 directly in terms of the conservative variables, $\delta(t)$ begins near 10^{-4} but grows steadily, blowing up exponentially near $t = 1$.

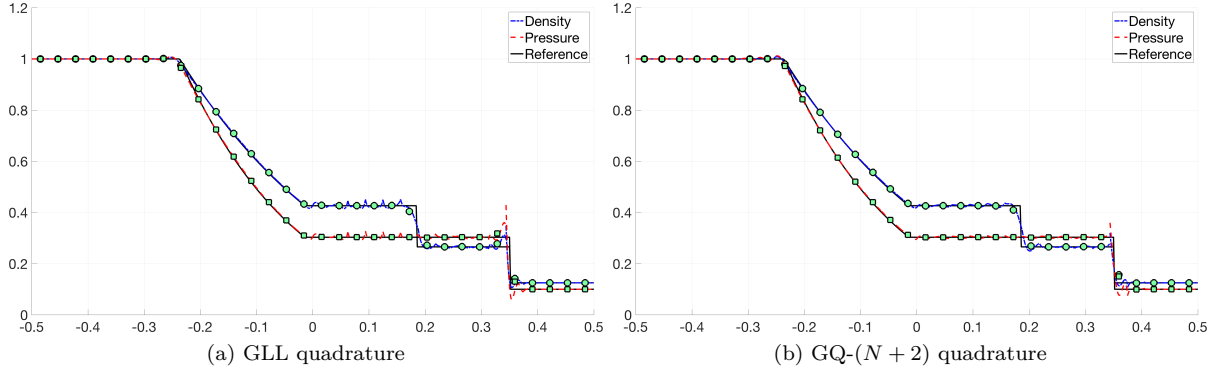


Figure 4: Density and pressure solutions for the Sod shock tube at $T = .2$ for $N = 4$ and $K = 32$ elements. Results for both GLL and GQ- $(N + 2)$ quadratures are shown. Cell averages are overlaid as filled circles. Both simulations are run using a CFL of .125.

6.1.3. Sod shock tube

We now examine the behavior of the proposed DG methods for some common one-dimensional test problems. We begin with the Sod shock tube, which is posed on the domain $[-1/2, 1/2]$ with initial conditions

$$\rho = \begin{cases} 1 & x < 0 \\ .125 & x \geq 0, \end{cases} \quad u = 0, \quad p = \begin{cases} 1 & x < 0 \\ .1 & x \geq 0. \end{cases} \quad (115)$$

Boundary conditions are enforced by taking the external value \mathbf{u}^+ in the numerical flux to be that of the initial condition at $x = \pm 1$. The solution develops a left-moving rarefaction, as well as a right moving shock wave and a contact discontinuity. We simulate the solution until time $T = .2$, without the use of positivity preserving or TVD-type limiters. For all choices of quadrature tested, the solution diverges when using the entropy conservative flux, which is a result of oscillations in the solution and density and temperature becoming negative. This is remedied when using the dissipative Lax-Friedrichs flux, for which we do not observe blowup of the solution.

Figure 4 shows both the exact solution and the computed density and pressure along with their cell averages. These results were obtained using a CFL of .125 and GLL and GQ- $(N + 2)$ quadrature rules. For both quadratures, the cell averages agree relatively well with the exact solution. Both solutions contain spurious oscillations, though the oscillations under GQ- $(N + 2)$ quadrature appear qualitatively smoother and smaller.

6.1.4. Sine-shock interaction

The next benchmark problem we consider is the sine-shock interaction problem, which is posed on domain $[-5, 5]$ with initial conditions

$$\rho(x, 0) = \begin{cases} 3.857143 & x < -4 \\ 1 + .2 \sin(5x) & x \geq -4, \end{cases} \quad (116)$$

$$u(x, 0) = \begin{cases} 2.629369 & x < -4 \\ 0 & x \geq -4, \end{cases} \quad p(x, 0) = \begin{cases} 10.3333 & x < -4 \\ 1 & x \geq -4. \end{cases}$$

We simulate the solution using different quadrature rules with $N = 4$, $K = 40$ elements, and a Lax-Friedrichs flux. No TVD or positivity-preserving limiters are applied. A smaller CFL of .05 is used, and is necessary to avoid solution divergence when using GQ- $(N + 2)$ quadrature. It should be pointed out that, for GLL quadrature, it is possible to use a larger CFL of .125 without observing solution blowup. The reason for

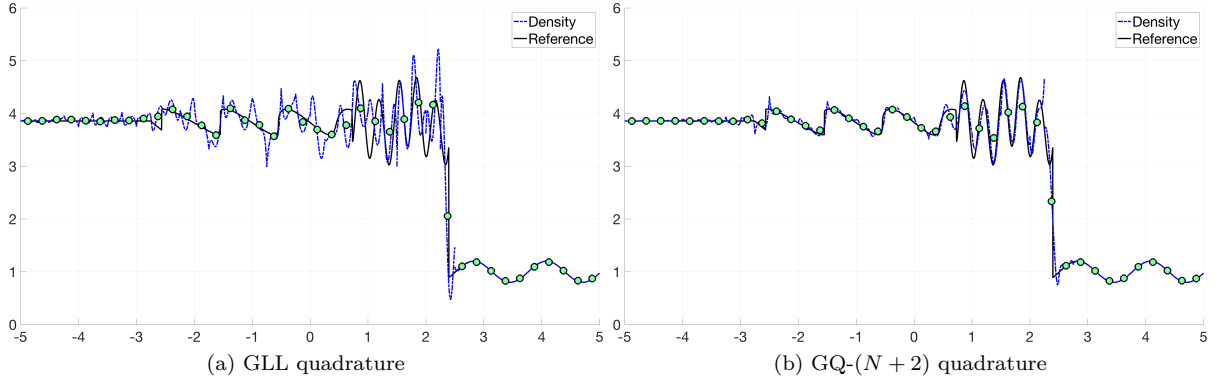


Figure 5: Density solutions for the sine-shock interaction problem at $T = 1.8$ for $N = 4$ and $K = 40$ elements. Results for both GLL and GQ- $(N + 2)$ quadratures are shown. Cell averages are overlaid as filled circles. Both simulations are run using a CFL of .05, though the GLL simulation is stable for a larger CFL of .125.

this discrepancy is the sensitivity of the evaluation $\mathbf{u}(\Pi_N \mathbf{v})$ when $\Pi_N \mathbf{v}$ differs significantly from \mathbf{v} , and is described in more detail in Section 6.1.5.

Figure 5 shows snapshots of the density at final time $T = 1.8$ for both GLL and GQ- $(N + 2)$ quadrature, along with a reference solution computed using a 5th order WENO scheme with 25000 cells [55]. For both GLL and GQ- $(N + 2)$ quadrature, the cell averages are close to the reference solution, though the solutions still contain spurious oscillations resulting from the presence of shocks and discontinuities. However, as with the Sod shock tube, we observe that these oscillations are smoother and smaller in amplitude for the choice of GQ- $(N + 2)$ quadrature.

6.1.5. Sensitivity of evaluation in terms of the projected entropy variables

The numerical experiments in previous sections suggest that solutions computed using Gauss quadrature rules can be more accurate than those computed using GLL quadratures. However, we also observe that, for the sine-shock interaction problem, a larger CFL of .125 can be taken when using GLL quadrature, whereas a much smaller CFL of .05 is required to prevent solution blowup when using GQ- $(N + 1)$ and GQ- $(N + 2)$ quadratures. For GQ- $(N + 1)$ and GQ- $(N + 2)$ quadratures, solution spikes can occur when evaluating the entropy-projected conservative variables at surface points. The use of GLL quadrature avoids this phenomena because of two factors: the equivalence between interpolation and projection under a $(N + 1)$ point quadrature rule, and the presence of boundary points in GLL quadrature.

Figure 6 shows snapshots of different variables for $N = 2$ and $K = 20$ elements at the fifth Runge-Kutta stage of the first timestep (just prior to the detection of negative density and pressure values) using a GQ- $(N + 2)$ quadrature rule. Discrepancies between the evaluated and projected values of the entropy variables v_1, v_2, v_3 at element boundaries are present, and while these discrepancies are not extremely large, they produce large spikes in the conservative variables due to the sensitivity of the nonlinear evaluation $\mathbf{u}(\Pi_N \mathbf{v})$. In particular, because v_3 appears in the denominators of entropy and ρe (as functions of the entropy variables), values of v_3 near zero result in large values of density and energy. These spikes result in large oscillations in the solution if the CFL is too large, which eventually cause the density and pressure to become negative at quadrature or boundary points.

Qualitatively similar behavior is observed when using a GQ- $(N + 1)$ quadrature rule, though negative density and pressure values occur at a slightly later time. In this case, the L^2 projection reduces to interpolation at interior Gauss points. This still implies that $\mathbf{u}(\Pi_N \mathbf{v})$ does not necessarily agree with \mathbf{u} at element boundaries, and this discrepancy again leads to spikes in the conservative variables similar to those presented in Figure 6. These experiments suggest that the lack of control of boundary values in $\Pi_N \mathbf{v}$ leads to large oscillations or spikes in the boundary values of $\mathbf{u}(\Pi_N \mathbf{v})$. The presence of boundary nodes in the $(N + 1)$ point GLL quadrature avoids this issue: $\mathbf{u}(\Pi_N \mathbf{v})$ and \mathbf{u} agree at element boundaries due to the fact that

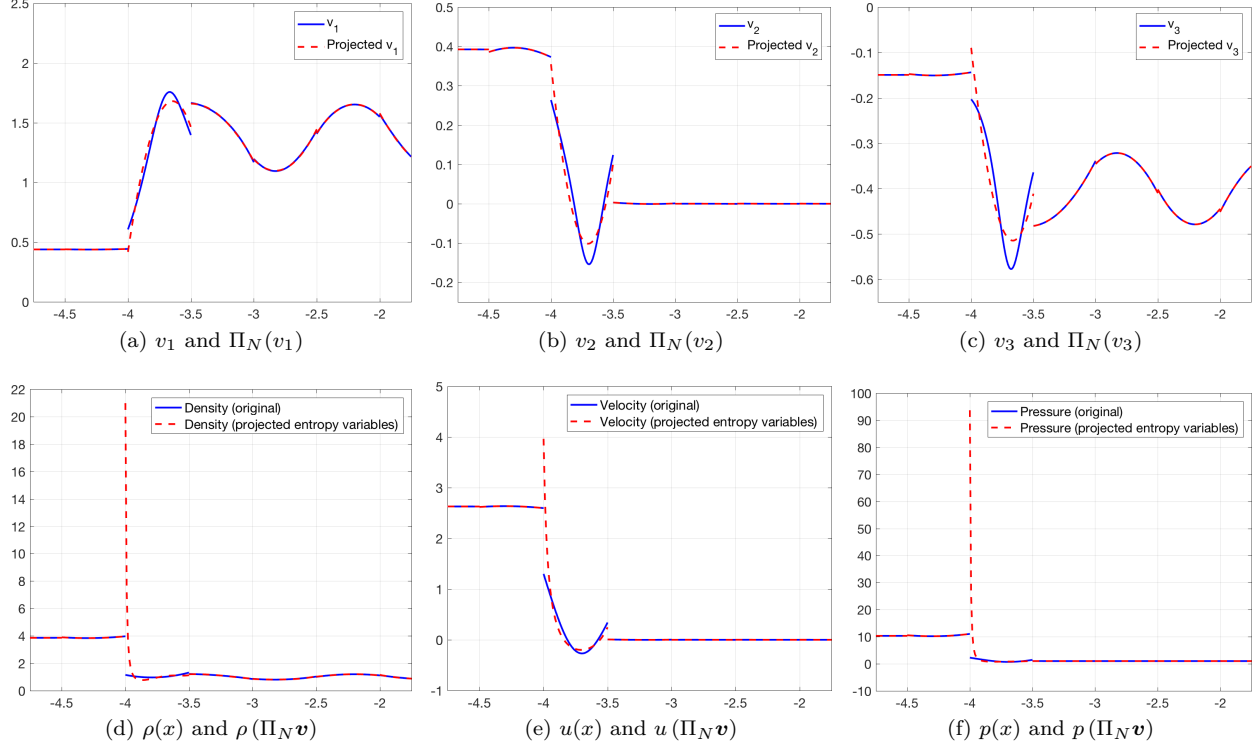


Figure 6: Top row: comparisons of entropy variables (evaluated directly as functions of conservative variables) and projected entropy variables (using $N = 2$, $K = 20$ elements, and a GQ- $(N + 2)$ quadrature rule) for the sine-shock interaction before the detection of negative density and pressures. Bottom row: comparisons of the conservative variables and their evaluations in terms of the projected entropy variables. The overshoot in the L^2 projection of v_3 produces a large spike in the entropy-projected conservation variables $\mathbf{u}(\Pi_N \mathbf{v})$.

quadrature-based L^2 projection under a GLL quadrature is equivalent to interpolation at GLL points.⁷

This scenario illustrates some of the more subtle differences between the use of GLL and GQ quadratures. We note that bound-preserving and TVD limiters [45, 46] are typically implemented in practice to detect and control spikes of the kind observed in these numerical experiments. However, slope-limiting the conservative variables \mathbf{u} may still result in spikes being generated when evaluating $\mathbf{u}(\Pi_N \mathbf{v})$ (unless the slope is set to zero such that \mathbf{u} is constant). A more nuanced approach will likely require modifying the conservative variables \mathbf{u} based on the projected entropy variables $\Pi_N \mathbf{v}$ to control oscillations while maintaining entropy conservation. These strategies will be the focus in future work.

⁷We note that the presence of boundary nodes in a quadrature does not necessarily reduce the sensitivity of the evaluation of the entropy-projected conservative variables $\mathbf{u}(\Pi_N \mathbf{v})$. This is due to the fact that the projection is computed using weighted averages of the points: while GLL quadrature rules contain endpoints, the quadrature weights are small at boundary points, implying that boundary values factor less strongly into the quadrature-based projection than interior points. Additionally, while Gauss quadrature rules do not contain endpoints, the variation of the Gauss quadrature weights is slightly less than that of GLL quadrature weights.

6.2. Two-dimensional experiments

We now present numerical experiments in two dimensions to verify the semi-discrete entropy conservation and accuracy of the presented schemes for the two-dimensional compressible Euler equations:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} &= 0, \\
\frac{\partial \rho u}{\partial t} + \frac{\partial (\rho u^2 + p)}{\partial x} + \frac{\partial (\rho uv)}{\partial y} &= 0, \\
\frac{\partial \rho v}{\partial t} + \frac{\partial (\rho uv)}{\partial x} + \frac{\partial (\rho v^2 + p)}{\partial y} &= 0, \\
\frac{\partial E}{\partial t} + \frac{\partial (u(E + p))}{\partial x} + \frac{\partial (v(E + p))}{\partial y} &= 0.
\end{aligned} \tag{117}$$

The following experiments use a triangular volume quadrature rule from [35] which is exact for polynomials of degree $2N$, with $(N + 1)$ point one-dimensional Gauss quadrature rules on the faces. L^2 errors are evaluated using a volume quadrature rule of degree $2N + 2$.

In two dimensions, the pressure is $p = (\gamma - 1) (E - \frac{1}{2}\rho(u^2 + v^2))$, and the specific internal energy is $\rho e = E - \frac{1}{2}\rho(u^2 + v^2)$. The formula for the entropy $U(\mathbf{u})$ is the same as in one dimension. The entropy variables in two dimensions are

$$v_1 = \frac{\rho e(\gamma + 1 - s) - E}{\rho e}, \quad v_2 = \frac{\rho u}{\rho e}, \quad v_3 = \frac{\rho v}{\rho e}, \quad v_4 = -\frac{\rho}{\rho e}. \tag{118}$$

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

$$\rho = -(\rho e)v_4, \quad \rho u = (\rho e)v_2, \quad \rho v = (\rho e)v_3, \quad E = (\rho e) \left(1 - \frac{v_2^2 + v_3^2}{2v_4} \right), \tag{119}$$

where ρe and s in terms of the entropy variables are

$$\rho e = \left(\frac{(\gamma - 1)}{(-v_4)^\gamma} \right)^{1/(\gamma-1)} e^{\frac{-s}{\gamma-1}}, \quad s = \gamma - v_1 + \frac{v_2^2 + v_3^2}{2v_4}. \tag{120}$$

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

$$\begin{aligned}
f_{1,S}^1(\mathbf{u}_L, \mathbf{u}_R) &= \{\{\rho\}\}^{\log} \{\{u\}\}, & f_{2,S}^1(\mathbf{u}_L, \mathbf{u}_R) &= \{\{\rho\}\}^{\log} \{\{v\}\}, \\
f_{1,S}^2(\mathbf{u}_L, \mathbf{u}_R) &= f_{1,S}^1 \{\{u\}\} + p_{\text{avg}}, & f_{2,S}^2(\mathbf{u}_L, \mathbf{u}_R) &= f_{2,S}^1 \{\{u\}\}, \\
f_{1,S}^3(\mathbf{u}_L, \mathbf{u}_R) &= f_{2,S}^2, & f_{2,S}^3(\mathbf{u}_L, \mathbf{u}_R) &= f_{2,S}^1 \{\{v\}\} + p_{\text{avg}}, \\
f_{1,S}^4(\mathbf{u}_L, \mathbf{u}_R) &= \left(\frac{p_{\text{avg}}^{\log}}{\gamma - 1} + p_{\text{avg}} + \frac{\|\mathbf{u}\|_{\text{avg}}^2}{2} \right) \{\{u\}\}, & f_{2,S}^4(\mathbf{u}_L, \mathbf{u}_R) &= \left(\frac{p_{\text{avg}}^{\log}}{\gamma - 1} + p_{\text{avg}} + \frac{\|\mathbf{u}\|_{\text{avg}}^2}{2} \right) \{\{v\}\},
\end{aligned} \tag{121}$$

where we have defined

$$p_{\text{avg}} = \frac{\{\{\rho\}\}}{2 \{\{\beta\}\}}, \quad p_{\text{avg}}^{\log} = \frac{\{\{\rho\}\}^{\log}}{2 \{\{\beta\}\}^{\log}}, \quad \|\mathbf{u}\|_{\text{avg}}^2 = 2(\{\{u\}\}^2 + \{\{v\}\}^2) - (\{\{u^2\}\} + \{\{v^2\}\}). \tag{122}$$

6.2.1. Discontinuous profile on a periodic domain

We again examine semi-discrete conservation of entropy by evolving a discontinuous initial profile to final time $T = 2$ on the square domain $[-1, 1]^2$. We set the initial velocities to be zero, and initialize the density and pressure as a discontinuous square pulse

$$\rho(\mathbf{x}, t) = \begin{cases} 3 & |x| < 1/2 \text{ and } |y| < 1/2 \\ 2 & \text{otherwise,} \end{cases} \quad u(\mathbf{x}, t) = v(\mathbf{x}, t) = 0, \quad p(\mathbf{x}, t) = \rho^\gamma. \tag{123}$$

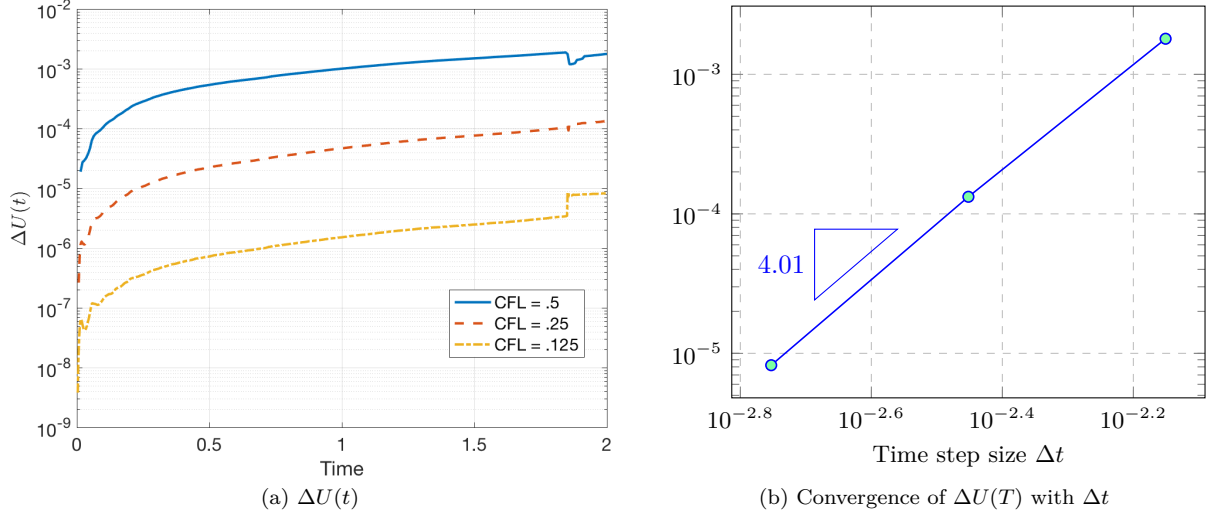


Figure 7: Change in entropy $\Delta U(t) = |U(t) - U(0)|$ over time and convergence of $\Delta U(T)$ at the final time T for a $N = 4$ and a mesh of 8×8 subdivided quadrilaterals. These results use a volume quadrature rule exact for degree $2N$ polynomials, and surface quadrature rules constructed using one-dimensional $(N + 1)$ point Gauss quadratures. The convergence of the change in entropy $\Delta U(t)$ at the final time $T = 2$ converges to zero as $O(\Delta t^4)$, matching the order of the 4th order time-stepper used.

Figure 7 shows the evolution of entropy over time for an entropy conservative scheme without additional interface dissipation. As in the one-dimensional case, we observe that the entropy increases over time, but decreases with the time-step. Unlike the one-dimensional case, we observe a jump in the change in entropy near time $t = 1.86$, though this does not appear to affect the order of convergence of the change in entropy at the final time $\Delta U(T)$, which converges to zero as $O(\Delta t^4)$ (which corresponds to the order of the time-stepping scheme used).

6.2.2. Isentropic vortex problem

Next, we examine high order convergence in two dimensions using the vortex problem as set up in [56, 24]. The analytical solution is given as

$$\begin{aligned} \rho(\mathbf{x}, t) &= \left(1 - \frac{\frac{1}{2}(\gamma - 1)(\beta e^{1-r(\mathbf{x}, t)^2})^2}{8\gamma\pi^2} \right)^{\frac{1}{\gamma-1}}, & p &= \rho^\gamma, \\ u(\mathbf{x}, t) &= 1 - \frac{\beta}{2\pi} e^{1-r(\mathbf{x}, t)^2} (y - y_0), & v(\mathbf{x}, t) &= \frac{\beta}{2\pi} e^{1-r(\mathbf{x}, t)^2} (y - y_0), \end{aligned} \quad (124)$$

where u, v are the x and y velocity and $r(\mathbf{x}, t) = \sqrt{(x - x_0 - t)^2 + (y - y_0)^2}$. Here, we take $x_0 = 5, y_0 = 0$ and $\beta = 5$.

We solve on a periodic rectangular domain $[0, 20] \times [-5, 5]$ until final time $T = 5$. The mesh is constructed by first building a mesh of uniform quadrilateral elements and subdividing each quadrilateral into two uniform triangles. We estimate the L^2 errors and their rates of convergence, which are shown in Figure 8. We observe L^2 optimal $O(h^{N+1})$ rates of convergence for $N = 1, \dots, 3$, while for $N = 4$ we observe a rate of convergence between $O(h^{N+1})$ and $O(h^{N+1/2})$. We note that the rate of $O(h^{N+1/2})$ is the theoretically proven rate of convergence for upwind DG methods on general meshes applied to linear hyperbolic problems [57, 58]. We note that this observed rate does not change significantly if the time-step is halved (improving from 4.785 to 4.8), suggesting that this slight degradation in convergence rate is not due to temporal errors.

6.2.3. A two-dimensional Riemann problem

Finally, we present numerical results for a two-dimensional Riemann problem [59, 60, 61] using an entropy stable high order DG scheme using Lax-Friedrichs penalization. No additional stabilization, artificial

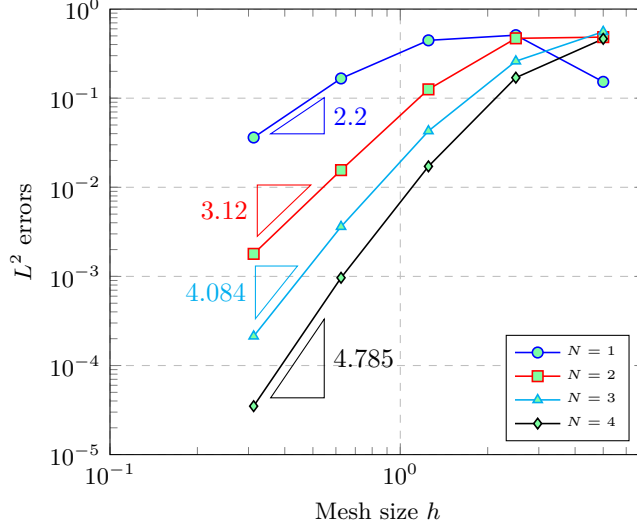


Figure 8: Convergence of L^2 errors for the isentropic vortex problem at final time $T = 5$ for various N .

dissipation, or limiting is applied. The problem is posed on the square domain $[-.5, .5]^2$ with piecewise constant initial conditions

$$\begin{aligned}
 \rho(\mathbf{x}) &= .5313, & u(\mathbf{x}) &= 0, & v(\mathbf{x}) &= 0, & p(\mathbf{x}) &= .4, & \mathbf{x} &\in [0, .5] \times [0, .5], \\
 \rho(\mathbf{x}) &= 1, & u(\mathbf{x}) &= .7276, & v(\mathbf{x}) &= 0, & p(\mathbf{x}) &= 1, & \mathbf{x} &\in [-.5, 0] \times [0, .5], \\
 \rho(\mathbf{x}) &= .8, & u(\mathbf{x}) &= 0, & v(\mathbf{x}) &= 0, & p(\mathbf{x}) &= 1, & \mathbf{x} &\in [-.5, 0] \times [-.5, 0], \\
 \rho(\mathbf{x}) &= 1, & u(\mathbf{x}) &= 0, & v(\mathbf{x}) &= .7276, & p(\mathbf{x}) &= 1, & \mathbf{x} &\in [0, .5] \times [-.5, 0].
 \end{aligned} \tag{125}$$

In [6], the boundary condition is imposed by computing the exact shock speed using a one-dimensional Riemann solver on each boundary. Here, we solve instead on an enlarged quadrilateral domain $[-1, 1]^2$ with periodic boundary conditions, but consider the solution only on the smaller physical domain $[-.5, .5]^2$. The size of the enlarged domain is chosen such that solution within the fictitious portion of the domain $[-1, 1]^2 \setminus [-.5, .5]^2$ does not pollute the solution within the smaller subdomain $[-.5, .5]^2$ at the final time $T = 1/4$. The piecewise constant initial conditions are initialized as piecewise constants in each quadrant of the enlarged domain.

The enlarged domain $[-1, 1]^2$ is meshed using 64 uniform quadrilaterals per direction. A uniform triangular mesh of 8192 elements is constructed by subdividing each quadrilateral into two triangles. Figure 9 shows the numerical solution on this mesh for $N = 3$ using a CFL of .125 on both the enlarged and physical domains. The result on the physical domain shows qualitative agreement with results presented in the literature.

6.3. On accuracy and computational cost

While we have shown how to construct high order schemes which are discretely entropy conservative or entropy stable, we have not shown theoretically that these methods are high order accurate for general conservation laws. We first note that, for conservation laws where $\mathbf{v} = \mathbf{u}$ (i.e. the entropy variables are the same as the conservative variables), the entropy-projected conservation variables $\tilde{\mathbf{u}} = \mathbf{u}(\Pi_N \mathbf{v})$ reduce to the conservative variables. By (41),

$$[\mathbf{P}_q \quad \mathbf{L}_q] \mathbf{D}_N \tag{126}$$

is at least a degree N approximation to the derivative. Using arguments in [6], the truncation error of the proposed schemes is then at least $O(h^N)$ in smooth regions.

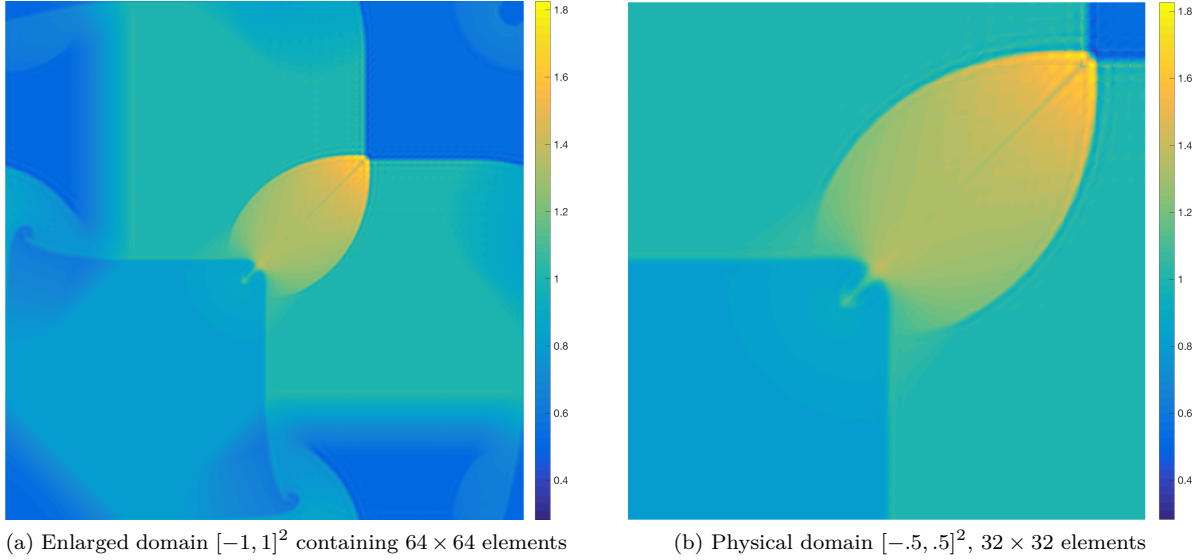


Figure 9: Numerical solution of the Riemann problem at time $T = .25$ using an entropy stable DG method with $N = 3$. These results use a volume quadrature rule of degree $2N$ and surface quadrature rules using one-dimensional GQ- $(N + 1)$ quadratures.

When $\mathbf{v} \neq \mathbf{u}$, we require additional conditions for high order accuracy. Ensuring that the truncation error is degree r accurate in this more general case would require at least that

$$|\mathbf{u} - \mathbf{u}(\Pi_N \mathbf{v})| = O(h^r) \quad (127)$$

in smooth regions. In other words, we would require that the difference between $\tilde{\mathbf{u}}$ and \mathbf{u} is high order accurate where \mathbf{u} is a high order accurate approximation to the solution. While we have not yet proven this, numerical experiments indicate that this is satisfied with $r = N + 1$ in both one and two dimensions.

We compute the L^2 error between \mathbf{u} and $\mathbf{u}(\Pi_N \mathbf{v})$ on the one-dimensional interval $[-1, 1]$ and the two-dimensional domain $[-1, 1]^2$. The conservation variables are set to the polynomial L^2 projections of smooth functions. In one dimension, we set

$$\rho(\mathbf{x}) = \rho_0 + e^{x/2} \sin(\pi x), \quad \rho u(\mathbf{x}) = \sin(\pi x), \quad E(\mathbf{x}) = E_0 + \frac{\rho}{2} u^2, \quad (128)$$

while in two dimensions, we set

$$\begin{aligned} \rho(\mathbf{x}) &= \rho_0 + e^{(x+y)/2} \sin(\pi x) \sin(\pi y), \\ \rho u(\mathbf{x}) &= \rho v(\mathbf{x}) = \sin(\pi x) \sin(\pi y), \quad E(\mathbf{x}) = E_0 + \frac{\rho}{2} (u^2 + v^2). \end{aligned} \quad (129)$$

Uniform meshes of K_{1D} elements are used in both cases. The two-dimensional domain is first meshed using a uniform quadrilateral mesh of $K_{1D} \times K_{1D}$ elements in each direction, and a uniform triangular mesh is constructed by bisecting each quadrilateral element into two triangles. In one dimension, a $(N + 2)$ point Gauss quadrature rule is used, while in two dimensions, a quadrature rule exact for degree $2N$ polynomials is used. The error is evaluated using a quadrature rule of two degrees higher.

Figure 10 plots the L^2 error $\|\mathbf{u} - \mathbf{u}(\Pi_N \mathbf{v})\|_{L^2(\Omega)}$ between the conservative and entropy-projected conservative variables in one and two dimensions for $\rho_0 = E_0 = 2$. Convergence rates of $O(h^{N+1})$ are observed for $N = 1, \dots, 5$. This implies that the entropy-projected conservative variables approximate the conservative variables with high order accuracy, and that the flux is approximated with high order accuracy. Numerical experiments also suggest that the constant in the $O(h^{N+1})$ error rate increases if ρ_0 and E_0 are

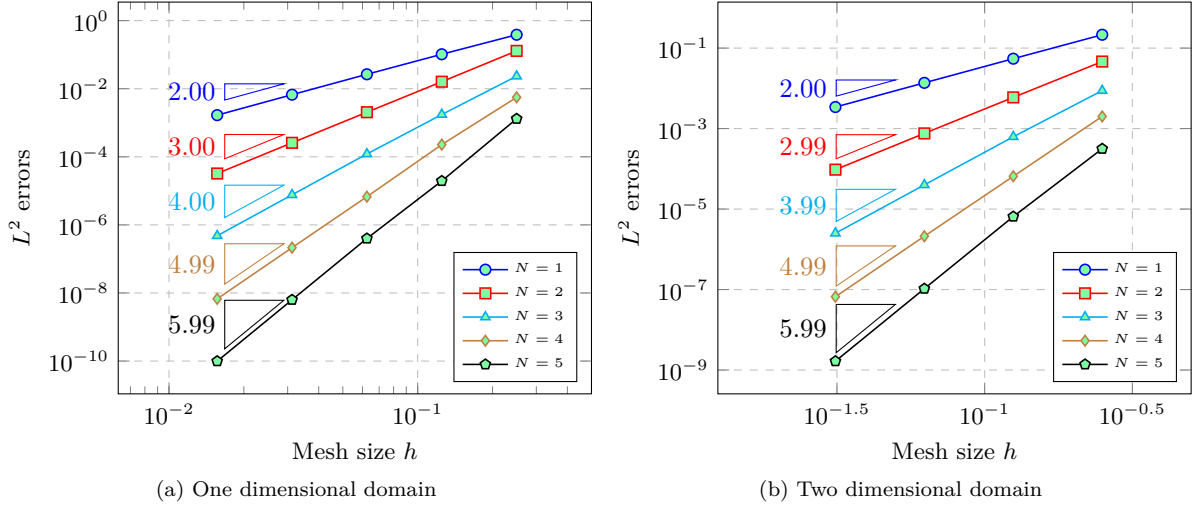


Figure 10: Convergence of the L^2 difference $\|\mathbf{u} - \mathbf{u}(\Pi_N \mathbf{v})\|_{L^2(\Omega)}$ between conservative variables and conservative variables as a function of the projected entropy variables with mesh refinement. A GQ- $(N+2)$ quadrature rule is used in one dimension, while a volume quadrature rule of degree $2N$ is used in two dimensions. The L^2 norm is evaluated using a quadrature of two degrees higher.

decreased. This is likely due to the fact that the error $\|\mathbf{u} - \mathbf{u}(\Pi_N \mathbf{v})\|_{L^2(\Omega)}$ depends on the nonlinearity of the mapping between conservation and entropy variables. Because this mapping is non-invertible when $\rho \leq 0$ or $E - \frac{1}{2}\rho|\mathbf{u}|^2 \leq 0$, we expect that it becomes more nonlinear the closer the minimum values of the thermodynamic variables are to 0.

Finally, we briefly discuss the computational cost of the entropy stable method described in this work. It was mentioned previously that the method proposed in this work can utilize high order polynomial approximation spaces with $N_p \leq N_q$, i.e. fewer degrees of freedom than the dimension of the underlying quadrature space. However, this comes at the additional cost of computing L^2 projections of entropy variables and applying projection and lifting matrices for each right hand side evaluation. Additionally, the main computational steps involving the Hadamard product and the application of the operator \mathbf{D}_N^i depend only on the number of quadrature points N_q and not N_p .

Thus, the computational cost of the proposed methods is higher than that of SBP methods based on under-integrated quadrature rules. However, the proposed work also allows for the construction of entropy conservative and entropy stable schemes for more general pairings of approximation spaces and quadrature. Additionally, numerical experiments indicate that, in certain cases, the use of over-integrated quadrature rules improves solution accuracy. Future work will compare the qualitative and quantitative behavior of entropy stable diagonal-norm SBP methods with the method proposed in this work.

7. Conclusions

This work presents a generalization of discretely entropy conservative methods, allowing for the combination of more general approximation spaces and (sufficiently accurate) quadrature rules. We introduce SBP-like operators using quadrature-based projection and lifting matrices, and construct high order DG schemes for conservation laws. The resulting schemes satisfy a discrete conservation of entropy. Numerical results for the compressible Euler equations indicate that the resulting methods deliver high order accuracy for smooth solutions while improving stability and robustness for under-resolved and shock solutions compared to non-entropy conservative and non-entropy stable schemes. We also describe differences between the sensitivity of methods based on Gauss-Lobatto quadrature and methods based on Gauss quadrature rules.

We note that, while entropy conservative and entropy stable schemes 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, this can lead to a reliance on regularization techniques as ad-hoc stabilization mechanisms. Addressing this issue requires constructing methods which avoid the need for excessive regularization and limiting in the pursuit of stability, and is a significant motivation for the development of entropy conservative and entropy stable discretizations.

Future work will focus on generalizations to three dimensions (including tetrahedra, prisms, and pyramidal elements), as well as analytical estimates for the error between the conservative and entropy-conservative variables. These estimates will be necessary to construct error estimates for the proposed entropy stable methods.

8. Acknowledgments

The author thanks Lucas Wilcox, Andrew Winters, David M. Williams, and Weifeng Qiu for helpful discussions. The author also thanks David C. Del Rey Fernandez for the name “decoupled SBP operator”. Jesse Chan is supported by the National Science Foundation under awards DMS-1719818 and DMS-1712639.

Appendix A. Recovery of known schemes for Burgers’ equation

In this appendix, we show how the framework presented recovers some existing entropy conservative (stable) schemes, and describe in more detail the connection between flux differencing and split formulations [3] in the context of the Burgers’ equation. The one-dimensional Burgers equation has flux $f(u) = u^2/2$, and can be rewritten in split form as

$$\frac{\partial u}{\partial t} + \frac{1}{3} \left(\frac{\partial u^2}{\partial x} + u \frac{\partial u}{\partial x} \right) = 0. \quad (\text{A.1})$$

Assuming smooth u , one can show that discretizing this form of Burgers’ equation results in a scheme which conserves the square entropy $U(u) = u^2/2$ [28, 29]. As pointed out in [3, 41], this split formulation may also be recovered using flux differencing under the following two-point flux

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

Applying (49) yields the split form of Burgers’ equation

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

Recovering the split-form of Burgers’ equation relies on the property that $\frac{\partial(u(y)^2)}{\partial x} = u(y)^2 \frac{\partial(1)}{\partial x} = 0$. More generally, recovery of the split form can be achieved by using flux differencing in combination with any differential operator D such that $D1 = 0$.

Given the entropy conservative flux (A.2) for Burgers’ equation, we can show that the decoupled SBP operator recovers existing entropy stable schemes for Burgers’ equation [28, 29]. We will assume periodic boundary conditions for simplicity and use the energy conserving flux

$$f_S(u_L, u_R) = \frac{1}{6} (u_L^2 + u_L u_R + u_R^2). \quad (\text{A.4})$$

Define the entries of the matrix $(F_S)_{ij} = u(\hat{x}_i)^2 + u(\hat{x}_i)u(\hat{x}_j) + u(\hat{x}_j)^2$. Then, we have that

$$F_S = \frac{1}{6} (U_L \circ U_L + U_L \circ U_R + U_R \circ U_R), \quad U_L = \begin{bmatrix} u_q \\ u_f \end{bmatrix} e^T, \quad U_R = e \begin{bmatrix} u_q \\ u_f \end{bmatrix}^T, \quad (\text{A.5})$$

where \mathbf{e} is the vector of all ones and $\mathbf{u}_q, \mathbf{u}_f$ denote the vector of $u(x)$ evaluated at volume and surface quadrature points, respectively. Applying the one-dimensional multi-element formulation (79) to (A.2) gives

$$\frac{d\mathbf{u}}{dt} + \frac{1}{3} \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} (\mathbf{D}_N \circ (\mathbf{U}_L \circ \mathbf{U}_L + \mathbf{U}_L \circ \mathbf{U}_R + \mathbf{U}_R \circ \mathbf{U}_R)) \mathbf{1} + \mathbf{L}_q \text{diag}(\mathbf{n}) (\mathbf{f}^* - \mathbf{f}(\tilde{\mathbf{u}}_f)) = 0. \quad (\text{A.6})$$

Using the definition of the Hadamard product and that $\mathbf{D}_N \mathbf{1} = 0$ from Theorem 1, we have that

$$((\mathbf{D}_N \circ \mathbf{U}_L) \mathbf{1})_i = \sum_{j=1}^{N_q+N_q^f} (\mathbf{D}_N)_{ij} \left(\begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix} \right)_j = \left(\mathbf{D}_N \begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix} \right)_i, \quad (\text{A.7})$$

$$((\mathbf{D}_N \circ \mathbf{U}_R) \mathbf{1})_i = \sum_{j=1}^{N_q+N_q^f} (\mathbf{D}_N)_{ij} \left(\begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix} \right)_i = \left(\text{diag} \left(\begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix} \right) \mathbf{D}_N \mathbf{1} \right)_i = 0 \quad (\text{A.8})$$

We can similarly show that

$$(\mathbf{D}_N \circ \mathbf{U}_L \circ \mathbf{U}_L) \mathbf{1} = \mathbf{D}_N \begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix}^2, \quad (\text{A.9})$$

$$(\mathbf{D}_N \circ \mathbf{U}_L \circ \mathbf{U}_R) \mathbf{1} = \text{diag} \left(\begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix} \right) \mathbf{D}_N \begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix}, \quad (\mathbf{D}_N \circ \mathbf{U}_R \circ \mathbf{U}_R) \mathbf{1} = 0.$$

where the entries are $\begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix}^2$ are the squared entries of $\begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix}$. Substituting these into (A.6) simplifies the formulation

$$\begin{aligned} \frac{d\mathbf{u}}{dt} + \frac{1}{3} \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} \left(\mathbf{D}_N \begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix}^2 + \text{diag} \left(\begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix} \right) \mathbf{D}_N \begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix} \right) \\ + \frac{1}{3} \mathbf{L}_q \text{diag}(\mathbf{n}) \left(\frac{1}{6} \left(\mathbf{u}_f^2 + \mathbf{u}_f^+ \mathbf{u}_f + (\mathbf{u}_f^+)^2 \right) - \frac{1}{2} \mathbf{u}_f^2 \right) = 0, \end{aligned} \quad (\text{A.10})$$

where \mathbf{u}_f^+ denotes the values of the solution at face quadrature points on a neighboring elements.

We can now use (21) and the structure of \mathbf{D}_N to simplify (A.11)

$$\begin{aligned} \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} \mathbf{D}_N \begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix}^2 &= \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} \begin{bmatrix} \mathbf{D}_q \mathbf{u}_q^2 + \frac{1}{2} \mathbf{V}_q \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}) (\mathbf{u}_f^2 - \mathbf{V}_f \mathbf{P}_q \mathbf{u}_q^2) \\ \frac{1}{2} \text{diag}(\hat{\mathbf{n}}) (\mathbf{u}_f^2 - \mathbf{V}_f \mathbf{P}_q \mathbf{u}_q^2) \end{bmatrix} \\ &= \mathbf{D}_i \mathbf{P}_q \mathbf{u}_q^2 + \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}) (\mathbf{u}_f^2 - \mathbf{V}_f \mathbf{P}_q \mathbf{u}_q^2) \end{aligned} \quad (\text{A.11})$$

We can similarly simplify the second part of (A.11)

$$\begin{aligned} \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} \text{diag} \left(\begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix} \right) \mathbf{D}_N \begin{bmatrix} \mathbf{u}_q \\ \mathbf{u}_f \end{bmatrix} \\ = \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} \begin{bmatrix} \text{diag}(\mathbf{u}_q) (\mathbf{D}_q \mathbf{u}_q + \frac{1}{2} \mathbf{V}_q \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}) (\mathbf{u}_f - \mathbf{V}_f \mathbf{P}_q \mathbf{u}_q)) \\ \frac{1}{2} \text{diag}(\mathbf{u}_f \hat{\mathbf{n}}) (\mathbf{u}_f - \mathbf{V}_f \mathbf{P}_q \mathbf{u}_q) \end{bmatrix} \\ = \mathbf{P}_q \text{diag}(\mathbf{u}_q) (\mathbf{D}_q \mathbf{u}_q), \end{aligned} \quad (\text{A.12})$$

where we have used that, because $u \in P^N$, $\mathbf{u} = \mathbf{P}_q \mathbf{V}_q \mathbf{u} = \mathbf{P}_q \mathbf{u}_q$ and $\mathbf{u}_f = \mathbf{V}_f \mathbf{u} = \mathbf{V}_f \mathbf{P}_q \mathbf{u}_q$. Combining these two yields a simplified formulation

$$\frac{d\mathbf{u}}{dt} + \frac{1}{3} (\mathbf{D}_i \mathbf{P}_q \mathbf{u}_q^2 + \mathbf{P}_q \text{diag}(\mathbf{u}_q) (\mathbf{D}_q \mathbf{u}_q)) \quad (\text{A.13})$$

$$+ \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}) \left(\frac{1}{6} \left(\mathbf{u}_f^2 + \mathbf{u}_f^+ \mathbf{u}_f + (\mathbf{u}_f^+)^2 \right) - \frac{1}{2} \mathbf{u}_f^2 + \frac{1}{3} (\mathbf{u}_f^2 - \mathbf{V}_f \mathbf{P}_q \mathbf{u}_q^2) \right) = 0. \quad (\text{A.14})$$

Cancelling out flux terms gives

$$\begin{aligned} \frac{d\mathbf{u}}{dt} + \frac{1}{3} (\mathbf{D}_i \mathbf{P}_q \mathbf{u}_q^2 + \mathbf{P}_q \text{diag}(\mathbf{u}_q) (\mathbf{D}_q \mathbf{u}_q)) \\ + \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}) \left(\frac{1}{6} \left(\mathbf{u}_f^+ \mathbf{u}_f + \left(\mathbf{u}_f^+ \right)^2 \right) - \frac{1}{3} \mathbf{V}_f \mathbf{P}_q \mathbf{u}_q^2 \right) = 0. \end{aligned} \quad (\text{A.15})$$

If Gauss quadrature with $(N + 1)$ points is used, and the basis is chosen to be the co-located Lagrange basis at Gauss nodes, then $\mathbf{V}_q = \mathbf{P}_q = \mathbf{I}$, $\mathbf{u} = \mathbf{u}_q$, and the formulation reduces to

$$\frac{d\mathbf{u}}{dt} + \frac{1}{3} (\mathbf{D}_i \mathbf{u}^2 + \text{diag}(\mathbf{u}) (\mathbf{D}_i \mathbf{u})) + \mathbf{L}_q \text{diag}(\hat{\mathbf{n}}) \left(\frac{1}{6} \left(\mathbf{u}_f^+ \mathbf{u}_f + \left(\mathbf{u}_f^+ \right)^2 \right) - \frac{1}{3} \mathbf{V}_f (\mathbf{u}^2) \right) = 0. \quad (\text{A.16})$$

This is equivalent to the weak form of the split formulation and the correction terms introduced in [28] for Burgers' equation with an entropy conservative flux. Entropy conservative SBP-SATs were introduced for generalized SBP operators and the compressible Euler equations in [37]; however, we have not yet determined if (79) recovers these formulations as well.

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] Lucas C Wilcox, Georg Stadler, Carsten Burstedde, and Omar Ghattas. A high-order discontinuous Galerkin method for wave propagation through coupled elastic–acoustic media. *Journal of Computational Physics*, 229(24):9373–9396, 2010.

- [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] 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.
- [15] T Warburton. A low-storage curvilinear discontinuous Galerkin method for wave problems. *SIAM Journal on Scientific Computing*, 35(4):A1987–A2012, 2013.
- [16] Jesse Chan, Russell J Hewett, and T Warburton. Weight-adjusted discontinuous Galerkin methods: wave propagation in heterogeneous media. *arXiv preprint arXiv:1608.01944*, 2016.
- [17] Jesse Chan, Russell J Hewett, and T Warburton. Weight-adjusted discontinuous Galerkin methods: curvilinear meshes. *arXiv preprint arXiv:1608.03836*, 2016.
- [18] Jesse Chan. Weight-adjusted discontinuous Galerkin methods: matrix-valued weights and elastic wave propagation in heterogeneous media. *arXiv preprint arXiv:1701.00215*, 2017.
- [19] Eitan Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Mathematics of Computation*, 49(179):91–103, 1987.
- [20] 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.
- [21] Ulrik S Fjordholm, Siddhartha Mishra, and Eitan Tadmor. Arbitrarily high-order accurate entropy stable essentially nonoscillatory schemes for systems of conservation laws. *SIAM Journal on Numerical Analysis*, 50(2):544–573, 2012.
- [22] 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.
- [23] 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.
- [24] Jared Crean, Jason E Hicken, David C Del Rey Fernández, David W Zingg, and Mark H Carpenter. High-Order, Entropy-Stable Discretizations of the Euler Equations for Complex Geometries. In *23rd AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2017.
- [25] Gregor J Gassner. A kinetic energy preserving nodal discontinuous Galerkin spectral element method. *International Journal for Numerical Methods in Fluids*, 76(1):28–50, 2014.
- [26] Sigrun Ortleb. Kinetic energy preserving DG schemes based on summation-by-parts operators on interior node distributions. *PAMM*, 16(1):857–858, 2016.
- [27] Sigrun Ortleb. A Kinetic Energy Preserving DG Scheme Based on Gauss–Legendre Points. *Journal of Scientific Computing*, 71(3):1135–1168, 2017.

- [28] 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.
- [29] Hendrik Ranocha. Generalised summation-by-parts operators and variable coefficients. *arXiv preprint arXiv:1705.10541*, 2017.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] Jesse Chan and T Warburton. Orthogonal bases for vertex-mapped pyramids. *SIAM Journal on Scientific Computing*, 38(2):A1146–A1170, 2016.
- [35] H Xiao and Zydrunas Gimbutas. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Comput. Math. Appl.*, 59:663–676, 2010.
- [36] David C Del Rey Fernández, Pieter D Boom, and David W Zingg. A generalized framework for nodal first derivative summation-by-parts operators. *Journal of Computational Physics*, 266:214–239, 2014.
- [37] Hendrik Ranocha. Comparison of some entropy conservative numerical fluxes for the Euler equations. *arXiv preprint arXiv:1701.02264*, 2017.
- [38] Michael S Mock. Systems of conservation laws of mixed type. *Journal of Differential equations*, 37(1):70–88, 1980.
- [39] 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.
- [40] Daniele Antonio Di Pietro and Alexandre Ern. *Mathematical aspects of discontinuous Galerkin methods*, volume 69. Springer Science & Business Media, 2011.
- [41] 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.
- [42] 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, 2018.
- [43] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.
- [44] 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.

- [45] 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.
- [46] 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.
- [47] Cengke Shi and Chi-Wang Shu. On local conservation of numerical methods for conservation laws. *Computers and Fluids*, 2017.
- [48] 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.
- [49] Magnus Svård and Hatice Özcan. Entropy-stable schemes for the Euler equations with far-field and wall boundary conditions. *Journal of Scientific Computing*, 58(1):61–89, 2014.
- [50] Matteo Parsani, Mark H Carpenter, and Eric J Nielsen. Entropy stable wall boundary conditions for the three-dimensional compressible Navier–Stokes equations. *Journal of Computational Physics*, 292:88–113, 2015.
- [51] 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.
- [52] 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.
- [53] Amiram Harten. On the symmetric form of systems of conservation laws with entropy. *Journal of computational physics*, 49(1):151–164, 1983.
- [54] 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.
- [55] Chi-Wang Shu. High order weighted essentially nonoscillatory schemes for convection dominated problems. *SIAM review*, 51(1):82–126, 2009.
- [56] 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, 1998.
- [57] Claes Johnson and Juhani Pitkäranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Mathematics of computation*, 46(173):1–26, 1986.
- [58] Bernardo Cockburn, Bo Dong, and Johnny Guzmán. Optimal convergence of the original DG method for the transport-reaction equation on special meshes. *SIAM Journal on Numerical Analysis*, 46(3):1250–1265, 2008.
- [59] Peter D Lax and Xu-Dong Liu. Solution of two-dimensional Riemann problems of gas dynamics by positive schemes. *SIAM Journal on Scientific Computing*, 19(2):319–340, 1998.
- [60] Alexander Kurganov and Eitan Tadmor. Solution of two-dimensional Riemann problems for gas dynamics without Riemann problem solvers. *Numerical Methods for Partial Differential Equations*, 18(5):584–608, 2002.
- [61] Richard Liska and Burton Wendroff. Comparison of several difference schemes on 1D and 2D test problems for the Euler equations. *SIAM Journal on Scientific Computing*, 25(3):995–1017, 2003.