Entropy-stable, high-order discretizations using continuous summation-by-parts operators

Jason E. Hicken∗
Rensselaer Polytechnic Institute, 110 8th Street Troy NY 12180, U.S.A.

We present summation-by-parts (SBP) discretizations of the linear advection and Euler equations that use a continuous solution space. These continuous SBP discretizations assemble global operators from element operators, and they are the SBP analog, or generalization, of continuous Galerkin finite-element methods. Consequently, a stabilization method is needed to suppress high-frequency oscillations and ensure optimal convergence rates. The proposed stabilization, which is a form of local-projection stabilization, leads to energy-stable discretizations of the linear advection equation and entropy-stable discretizations of the Euler equations. Furthermore, the stabilization is element local, which keeps the stencil compact, and the stabilized discretizations have favorable time-step restrictions. Results are provided to demonstrate the accuracy and efficiency of the continuous SBP discretizations.

I. Introduction

The aerospace industry has used computational fluid dynamics (CFD) to model the cruise performance of aircraft for decades. For such simulations, second-order discretizations have proven to be reliable. However, engineers are increasingly turning to CFD to inform the design of aircraft over the broader flight envelope. Many of these non-cruise flight conditions are inherently transient or have proven difficult to accurately model using a steady-state assumption. Consequently, the industrial use of unsteady CFD simulations has risen. Given the computational cost of large-scale unsteady simulations, the efficiency of the CFD method becomes even more paramount.

High-order discretizations are considered by many as a possible means of improving the efficiency of CFD simulations. Advocates of high-order discretizations cite their improved accuracy-per-degree-of-freedom and cache efficiency on modern and future architectures. Despite these potential advantages, the industrial use of high-order CFD has been limited. There are many issues that hinder the widespread adoption of high-order methods, but lack of robustness is certainly one of them. High-order discretizations have inherently less numerical dissipation, which makes them prone to instabilities, particularly on under-resolved meshes.

In order to address the robustness of high-order discretizations, there has been growing interest in semi-discrete and fully-discrete schemes that are provably entropy stable. This interest was sparked by a series of publications — in particular, Fisher [1], Fisher and Carpenter [2], and Fisher et al. [3] — that showed that summation-by-parts (SBP) finite difference methods could be combined with entropy-conservative flux functions [4–7] to produce high-order entropy-stable schemes. This was later extended to tensor-product spectral-element methods [8] that also possess the summation-by-parts property [9]. SBP operators for simplex elements were presented in [10] and subsequently used to construct entropy-stable discretizations on triangular and tetrahedral grids [11]; see also [12].

Entropy stability is not a new idea: over thirty years ago, Hughes et al. [13] presented a finite-element discretization of the compressible Navier-Stokes equations that satisfied the second-law of thermodynamics; and twenty years ago Barth [14] extended this work to cover Galerkin-least-squares stabilizations and discontinuous Galerkin (DG) schemes. However, unlike the new generation of entropy-stable schemes, the schemes in [13] and [14] rely on exact integration of the finite-element semi-linear forms, which is not possible for the Euler and Navier-Stokes equations. In contrast, more recent entropy-stable schemes explicitly account for inexact integration and can be made entropy stable in a semi-discrete or fully-discrete sense.

The goal of this work is to develop an entropy-stable discretization that is based on a continuous solution space. Existing entropy-stable SBP discretizations use discontinuous solution spaces, such as the spectral collocation discontinuous Galerkin method in [8]. Even the finite-difference methods in [1], [3], and [2] used numerical flux functions embedded in penalty terms to couple blocks in multi-block grids. In contrast, we present high-order continuous SBP (C-SBP) discretizations that are entropy-stable and free of interface penalties; the class of C-SBP discretizations, which were first proposed in [10], are a generalization of continuous Galerkin finite-element methods.

∗Associate Professor, Department of Mechanical, Aerospace, and Nuclear Engineering, 110 8th Street, Troy, NY, AIAA Member
It is well-known that continuous Galerkin finite-element methods produce oscillatory solutions for hyperbolic partial-differential equations (PDEs); they require a suitable stabilization method, such as streamline upwind Petrov-Galerkin (SUPG) [15], Galerkin-least-squares (GLS) [16], or edge stabilization [17]. However, most conventional stabilization methods are not well-suited to entropy-stable C-SBP discretizations. For example, SUPG cannot be made entropy stable; GLS requires an expensive space-time discretization, and; edge-stabilization produces an overly stringent time-step restriction or high condition number; see, for example, [18]. Therefore, one of the principle contributions of this work is a stabilization method that is entropy-stable, suitable for semi-discretizations, and has a time-step restriction comparable to, or better than, discontinuous SBP discretizations of the same order.

The remaining sections of the paper are structured as follows. In Section II we review C-SBP discretizations in the context of the linear advection equation and use this simplified setting to introduce the proposed stabilization method. We then generalize this stabilization for use in an entropy-stable discretization of the Euler equations in Section III. The properties of the C-SBP discretizations are verified using numerical experiments in Section IV. The paper concludes with a summary and discussion in Section V.

II. Continuous SBP discretization of the linear advection equation and its stabilization

Before describing the entropy-stable discretization of the Euler equations, we first present the proposed stabilization in the the simplified context of the constant-coefficient advection equation. This section also provides the opportunity to review the SBP definition and introduce some notation.

A. Element-based SBP operators and assembly of global operators

Like its discontinuous counterpart, a C-SBP discretization relies on element-level operators that satisfy the multi-dimensional summation-by-parts definition [10]. To describe these operators, we consider an element \( \kappa \) with domain \( \Omega_{\kappa} \). Functions on \( \kappa \) are represented using their values at \( n_{\kappa} \) nodes whose coordinates are given by \( S_{\kappa} = \{(x_i, y_i)\}_{i=1}^{n_{\kappa}} \). For example, if \( u \in L^2(\Omega_{\kappa}) \) is a generic function on the element \( \kappa \), then we use \( u_\kappa \in \mathbb{R}^{n_{\kappa}} \) to denote the function sampled at the nodes \( S_{\kappa} \); that is,

\[
(u_\kappa)_i = u(x_i, y_i), \quad \forall i = 1, \ldots, n_{\kappa}.
\]

The matrix \( D_{x,\kappa} \) is a degree \( p \) SBP approximation to \( \partial/\partial x \) on \( S_{\kappa} \) if the following three conditions are met.

1. \( D_{x,\kappa} p \) is equal to \( \partial P / \partial x \) at the nodes \( S_{\kappa} \), for all polynomials \( P \in \mathbb{P}_p(\Omega_{\kappa}) \), where \( \mathbb{P}_p(\Omega_{\kappa}) \) denotes the space of polynomials of total degree \( p \) on \( \Omega_{\kappa} \), and \( p \) denotes \( P \) evaluated at the nodes \( S_{\kappa} \).

2. \( D_{x,\kappa} = H^{-1}_{\kappa}Q_{\kappa} \), where \( H_{\kappa} \) is symmetric positive-definite.

3. \( Q_{x,\kappa} = S_{x,\kappa} + \frac{1}{2}E_{x,\kappa} \), where \( S_{x,\kappa}^T = -S_{x,\kappa}, E_{x,\kappa}^T = E_{x,\kappa} \), and \( E_{x,\kappa} \) satisfies

\[
p^T E_{x,\kappa} q = \int_{\partial \Omega_{\kappa}} P Q n_x d\Gamma,
\]

for all polynomials \( P, Q \in \mathbb{P}_r(\Omega_{\kappa}) \), where \( r \geq p \). In the above integral, \( n_x \) is the x component of \( n = [n_x, n_y]^T \), the outward pointing unit normal on \( \partial \Omega_{\kappa} \).

Similar definitions holds for \( D_{y,\kappa} \) and \( D_{z,\kappa} \), but all difference operators should be based on the same mass, or norm, matrix \( H_{\kappa} \). Note that \( H_{\kappa} \) is dense in general; however, in this work, we consider only diagonal-norm SBP operators, that is, operators for which \( H_{\kappa} \) is a diagonal matrix. In this case, the nodes \( S_{\kappa} \) and diagonal entries in \( H_{\kappa} \) define a quadrature rule on \( \Omega_{\kappa} \) that is at least \( 2p - 1 \) exact [10, 19].

At this point, it is helpful to highlight the similarities and differences between SBP operators and finite-element bilinear forms. In particular, collocation finite-element methods, such as those based on the Legendre-Gauss-Lobatto quadrature points, are a subset of SBP operators [9]. However, unlike finite-element methods, SBP methods do not, in general, have a unique underlying basis. Typically the number of SBP nodes \( n_{\kappa} \) exceeds the number of elements in a basis for \( \mathbb{P}_p(\Omega_{\kappa}) \). Furthermore, to the best of our knowledge, the additional degrees of freedom cannot be associated with bubble functions sometimes used in finite element methods [20, 21].

Similar to the finite-element method, the element-level SBP operators can be assembled into SBP operators on the global domain \( \Omega \). The assembly process can be described with the help of restriction operators that map global degrees of freedom to element-local degrees of freedom. Let \( R_{\kappa} \) denote the restriction operator for element \( \kappa \) in a mesh \( \mathcal{T}_h = \{\Omega_{\kappa}\}_{\kappa=1}^{K} \) with \( K \) elements. Then, a first-derivative SBP operator on \( \Omega \) is given by

\[
D_x = H^{-1}\mathbb{Q}_{\kappa}.
\]
where

\[
Q_x \equiv \sum_{k=1}^{K} R_x^T Q_{x,k} R_x, \quad \text{and} \quad H \equiv \sum_{k=1}^{K} R_x^T H_x R_x.
\]

See [22], or any textbook on finite-element methods, for additional details on the assembly process. The fact that \(D_x\) defines a degree \(p\) SBP operator at the nodes of \(T_h\), as established in [10].

In the above assembly, we have tacitly assumed that the element-level SBP operators are defined in physical space. In practice, an SBP operator is defined on a reference element and a diffeomorphism is used to map between this reference element and the physical elements. For affine mappings, it is easy to show that the SBP definition is satisfied by the global operator \(D_x = H^{-1}Q_x\). When more general, curvilinear mappings are used, it was shown in Crean et al. [11] how to compute the mapping Jacobian such that the SBP properties are maintained. In the curvilinear case, while \(D_x\) is satisfied, \(D_x\) is not exact for higher degree polynomials in \(P_p(\Omega)\), although it remains accurate to the order of scheme.

B. Preliminary discretization of the linear advection equation

In this section we describe the preliminary (unstabilized) C-SBP semi-discretization of the linear advection equation. The subsequent section introduces the proposed stabilization method.

Consider the two-dimensional, constant-coefficient linear advection equation on the domain \(\Omega\):

\[
\begin{align*}
\frac{\partial U}{\partial t} + \lambda_x \frac{\partial U}{\partial x} + \lambda_y \frac{\partial U}{\partial y} &= 0, \quad \forall x \in \Omega, \\
U(x,t) &= U_{bc}(x,t), \quad \forall x \in \Gamma^-, \\
U(x,0) &= U_0(x), \quad \forall x \in \Omega,
\end{align*}
\]

(1)

where \(\Gamma^- = \{ x \in \partial \Omega \mid \lambda_x n_x + \lambda_y n_y \leq 0 \}\) is the in-flow boundary and \(\Gamma^+ = \partial \Omega \setminus \Gamma^-\) is the out-flow boundary. We assume the advection velocity, \([\lambda_x, \lambda_y]^T\), boundary conditions, and initial condition are such that (1) is well posed.

Let \(u_h \in \mathbb{R}^n\) denote the discrete solution approximating \(U(x,t)\) on the global mesh \(T_h\). Each entry in \(u_h\) corresponds to a node in \(T_h\) and the entries are ordered to be consistent with the projections, \(P_x\). Thus, \(D_x\) and \(D_x\) can be applied directly to \(u_h\) to approximate the spatial derivatives at the nodes, and the C-SBP semi-discretization of (1) that governs \(u_h\) is

\[
\begin{align*}
\frac{du_h}{dt} + \lambda_x D_x u_h + \lambda_y D_y u_h &= s_h(u_h, U_{bc}(x,t)), \\
u_h(0) &= U_0(x_h).
\end{align*}
\]

(2)

In the above semi-discretization, \(x_h\) denotes the coordinates of the global node set in \(T_h\), and \(s_h\) denotes penalty terms that enforce the boundary conditions weakly. At the element level, the C-SBP discretization (2) uses the same boundary-condition penalties as a discontinuous SBP discretization; see, for example, [23]. Therefore, to keep the presentation concise we will not review the penalties here.

C. Stabilization method

Numerical dissipation must be added to (2) to suppress high-frequency oscillations that pollute the solution and prevent optimal \((p+1)\) rates of convergence. Such oscillations were observed in the \(p = 2\) and \(p = 4\) degree C-SBP discretization in [10]. In this work we add dissipation to the discretization (2) in the form of local-projection stabilization, or LPS.

1. Local-projection stabilization

The idea behind LPS is to isolate high-frequency content in the solution and then dissipate only these high-frequency modes. To describe the method more precisely, let \(\tilde{U} \in L^2(\Omega_x)\) be a given function. The \(L^2\) projection of \(\tilde{U}\) onto the space \(P_q(\Omega_x)\) of total degree \(q\) polynomials is denoted by \(\tilde{U} \in P_q(\Omega_x)\) and is the solution to the (element-local) problem

\[
\int_{\Omega_x} \tilde{V}(U - \tilde{U}) \, d\Omega = 0, \quad \forall \tilde{V} \in P_q(\Omega_x).
\]

(3)

The projection \(\tilde{U}\) can be thought of as the low-frequency content in the solution; consequently, \(U - \tilde{U}\) represents the high-frequency content that we want to dissipate. To express the projection operator in matrix form, we first write \(\tilde{U}\) in
terms of orthogonal polynomials evaluated at the nodes of $\kappa$:

$$\tilde{u}_k = L y$$

where the $j$th column in $L$ is the $j$th orthogonal polynomial (e.g. Legendre polynomial in 1D) evaluated at the nodes $S_k$, and $y$ are the coefficients for $\tilde{u}_k$ in terms of the basis. Since we can also use $L$ for the test function space, the discretization of the projection is

$$L^T H_k (u_k - y) = 0, \quad \Rightarrow \quad y = L^T H_k u_k, \quad \Rightarrow \quad \tilde{u}_k = L L^T H_k u_k.$$ 

In the above derivation of $\tilde{u}_k$, we have used the orthonormality of the basis polynomials in $L$, that is $L^T H_k L = 1$, where $1$ is the identify matrix. This identity also relies on $H_k$ being exact for degree $2q$ polynomials, a requirement we will discuss in more detail below.

The high-frequency in $u_k$ can be obtained by applying the matrix operator $P_k \equiv 1 - L L^T H_k$, as shown by the following equalities:

$$u_k - \tilde{u}_k = u_k - L L^T H_k u_k = \left(1 - L L^T H_k\right) u_k = P_k u_k.$$ 

We want to dissipate the high-frequency content, $P_k u_k$, in a symmetric manner; as we will show below, having a symmetric dissipation operator facilitates energy stability. To this end, the LPS continuous operator in two dimensions is given by

$$\int_{\Omega_k} (V - \tilde{V}) A(x, y)(\tilde{u} - \tilde{U}) \, d\Omega \approx v_k^T (P_k)^T H_k A_k P_k u_k,$$

where the right-hand side gives the discretization of LPS.

The function $A(x, y)$ — which must be non-negative for energy stability — is used to scale the dissipation to ensure it is dimensionally consistent and comparable in magnitude to the appropriate inviscid flux. In this work, $A(x, y)$ is defined using the magnitude of the advection velocity in reference space:

$$A(\xi, \eta) = \sqrt{\alpha_x^2 + \alpha_y^2},$$

where

$$\alpha_x = J \lambda_x \partial_x \xi + J \lambda_y \partial_x \eta = \lambda_x \partial_x y - \lambda_y \partial_x x,$$

$$\alpha_y = J \lambda_x \partial_x \eta + J \lambda_y \partial_x \eta = -\lambda_x \partial_y y + \lambda_y \partial_x x,$$

where $J$ denotes the determinant of the mapping Jacobian $\partial (x, y)/\partial (\xi, \eta)$. The matrix $A_k$ is a diagonal, positive definite matrix with $A(\xi, \eta)$, evaluated at the nodes, along the diagonal:

$$A_k = \text{diag} \left[ A(x_1, y_1), A(x_2, y_2), \ldots, A(x_n, y_n) \right].$$

The right-hand side of (4) is the discretization of LPS for a single element. The LPS operator for the entire solution domain is obtained using the restriction operators $R_k$, much like the SBP operators $H_k$ and $D_{x,k}$ are assembled using the $R_k$ to obtain global operators. Thus, the global LPS dissipation operator is given by

$$\mathbf{M} = \sum_{k=1}^{K} R_k^T \left( (P_k)^T H_k A_k P_k \right) R_k.$$

The dissipation term $-H^{-1} M u_h$ would then be added to the right-hand side of (2).

2. **Energy Stability**

It is easy to show that the LPS dissipation operator $-\mathbf{M}$ is negative semi-definite:

$$-u_h^T M u_h = - \sum_{k=1}^{K} u_h^T R_k \left( (P_k)^T H_k A_k P_k \right) R_k u_h = - \sum_{k=1}^{K} (P_k u_k)^T H_k A_k P_k u_k = - \sum_{k=1}^{K} \tilde{u}_k^T H_k A_k \tilde{u}_k \leq 0$$
where $P_h u_x$ denotes $u_h$ restricted to element $\kappa$. Thus, $-u_h^T M u_h \leq 0$, which ensures that LPS decreases the solution norm over time. To see this more clearly, consider constant-coefficient advection on a periodic domain — so that the boundary penalties $s_h(u_h, U_{bc}(x, t))$ are zero — and left multiply (2) by $u_h^T H$:

$$u_h^T H \frac{d u_h}{d t} + u_h^T H (\lambda_x D_x u_h + \lambda_y D_y u_h) = -u_h^T H (H^{-1} M u_h)$$

$$\Rightarrow \quad \frac{1}{2} \frac{d}{d t} \|u_h\|_H^2 + \lambda_x u_h^T Q_x u_h + \lambda_y u_h^T Q_y u_h = -u_h^T H M u_h \leq 0,$$

where $\|u_h\|_H^2 = u_h^T H u_h$ approximates the $L^2$ norm of $u_h$. Note that, $Q_x$ and $Q_y$ are skew symmetric on a periodic domain, so $u_h^T Q_x u_h = 0$ and $u_h^T Q_y u_h = 0$. The bottom line is that $d\|u_h\|^2/dt \leq 0$, which shows that LPS encourages a non-increasing solution norm.

3. Accuracy of the dissipation

We end this section by discussing the accuracy of the LPS dissipation and the implications this has for the number of nodes on each element. For simplicity, we will consider one-dimensional SBP operators, specifically Gauss-Legendre-Lobatto spectral collocation operators.

Suppose we use elements with 3 nodes, i.e. $p = 2$, for which the SBP operator $D_{x, e}$ is exact for quadratic polynomials. Now, consider the $L^2$ projection (3) using $q = p$. With this choice of projection space, the projected solution is $\tilde{U} = U$ because $P_p = P_q$. Thus, the high-frequency content is trivial, $\tilde{U} - U = 0$, and we get no dissipation. This shows that the projection operator $P_k$ is limited to $q < p$ for non-zero dissipation.

What are the implications of $q < p$ on accuracy? Consider again a degree $p = 2$ element and set $q = 1$. Then $\check{U}$ is a degree 2 polynomial. A necessary condition for the discretization to be order $p + 1$ is that the residual is zero for degree $p$ polynomials. Here we find that $P_h u_e = 0$ if $u_e$ is a linear polynomial, while $P_h u_e \neq 0$ for quadratics. Consequently, the dissipation is zero for linear polynomials and $Mu_h \neq 0$ for a degree 2 polynomial. The discretization is at best second-order accurate in this case. More generally, the discretization is order $q + 1$ when $q < p$.

In summary, for a degree $p$ basis, we need to use a degree $q < p$ projection to ensure the proposed LPS is non-trivial; however, this then limits the accuracy to degree $q + 1 < p + 1$. Thus, this form of element-local dissipation is not “optimal” for C-SBP operators, since more than $p + 1$ nodes are required for an operator to be order $p + 1$. One could devise a method to overcome this issue, but we believe any such method will either i) require a larger stencil; ii) fail to be energy stable; or iii) have a large spectral radius.

Note that our method is not the only one that requires more than $p + 1$ nodes: a finite-difference stencil that is order $p + 1$ is typically applied on a grid with many more than $p + 1$ nodes. Finite-difference methods tradeoff optimal approximation for sparsity and conditioning, and a similar tradeoff is necessary for C-SBP discretizations using the proposed stabilization.

III. Entropy-stable discretization of the Euler equations

In this section we briefly review continuous (i.e. not discretized) entropy analysis for the Euler equations, and we present a semi-discrete entropy-conservative discretization for the Euler equations. We then generalize the LPS dissipation to the context of the Euler equations to obtain an entropy-stable discretization.

A. Entropy analysis of the Euler equations

We consider the Euler equations with the pressure defined by the ideal-gas-law equation of state. Furthermore, for simplicity, we restrict the presentation to the two-dimensional Euler equations. Thus, the governing PDE is

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \mathcal{F}_x}{\partial x} + \frac{\partial \mathcal{F}_y}{\partial y} = 0, \quad \forall x \in \Omega$$

$$A^\omega \mathcal{U}(x, t) = A^\omega \mathcal{U}_{bc}(x, t), \quad \forall x \in \Gamma,$n

$$\mathcal{U}(x, 0) = \mathcal{U}_0(x), \quad \forall x \in \Omega.$$

(5)

The state $\mathcal{U}$ here denotes the conservative variables density, momentum per unit volume, and energy per unit volume: $\mathcal{U} = [\rho \quad \rho u \quad \rho v \quad e]^T$. The variables are non-dimensionalized based on the free-stream density and speed of sound.
The Euler fluxes are given by

\[ F_x = \left[ \begin{array}{cccc} \rho u & \rho u^2 + p & \rho uv & (e + p)u \end{array} \right]^T, \quad \text{and} \quad F_y = \left[ \begin{array}{cccc} \rho v & \rho vu & \rho v^2 + p & (e + p)v \end{array} \right]^T, \]

where \( p = (\gamma - 1)(e - \frac{1}{2}(u^2 + v^2)) \) is the pressure. In the boundary condition in (5), the matrix \( A_n = \frac{1}{2}(\Lambda_L - |\Lambda_n|)X^{-1} \) denotes the inflow part of the flux Jacobian, where \( \Lambda_n = X\Lambda_xX^{-1} \) is the eigen decomposition of the normal-flux Jacobian with respect to the outward pointing normal vector on \( \Gamma = \partial \Omega \). Thus, boundary conditions are applied to the incoming characteristics only.

We will now review the entropy conservation implied by (5), since this motivates the discretization described in the subsequent sections. Let \( \mathcal{S} \equiv -\rho s/(\gamma - 1) \) be the mathematical entropy, where \( s = \ln(p/\rho^\gamma) \) is the physical entropy. If \( \mathcal{W} \equiv \partial S/\partial \mathcal{U} \) denotes the entropy variables, then for smooth solutions one can show that [5]

\[ \int_{\Omega'} \mathcal{W}^T \left[ \frac{\partial \mathcal{U}}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} \right] d\Omega = \frac{d}{dt} \int_{\Omega'} \mathcal{S} d\Omega + \int_{\partial \Omega} \mathcal{G}_x \partial_x n_x + \mathcal{G}_y \partial_y n_y \, d\Gamma = 0, \tag{6} \]

where \( \mathcal{G}_x = -\rho u S \) and \( \mathcal{G}_y = -\rho v S \) are the entropy fluxes in the \( x \)- and \( y \)-coordinate directions, respectively. Note that the domain \( \Omega' \) in (6) is any compact subset of \( \Omega \) with a piecewise-smooth boundary. Thus, (6) implies that entropy is conserved for a smooth flow. More generally, for discontinuous solutions, the time-rate-of-change of the integral of (mathematical) entropy should be non-increasing.

The objective when constructing an entropy-conservative scheme is to mimic (6) either semi-discretely or fully-discretely. The motivation for doing so is that one can then bound the \( L^2 \) norm of the solution \( \mathcal{U} \) [24]. A scheme that ensures the integral of entropy over the domain is constant is said to be entropy conservative. A scheme that ensures the integral is non-increasing is said to be entropy stable.

### B. Entropy-conservative discretization

As has been shown elsewhere [1–3], an entropy conservative scheme can be constructed by combining SBP operators with so-called entropy-conservative (EC) flux functions [4]. The function \( F^*_x : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}^4 \) is an EC flux function in the \( x \) direction if it satisfies the following conditions:

- \( F^*_x \) is symmetric in its arguments, \( F^*_x(\mathcal{U}_L, \mathcal{U}_R) = F^*_x(\mathcal{U}_R, \mathcal{U}_L) \);
- \( F^*_x \) is consistent with the Euler flux, \( F^*_x(\mathcal{U}, \mathcal{U}) = F_x(\mathcal{U}) \), and;
- \( F^*_x \) satisfies the entropy-consistency conditions \( (\mathcal{W}_L - \mathcal{W}_R)^T F^*_x(\mathcal{U}_L, \mathcal{U}_R) = (\psi_x)_L - (\psi_x)_R \), where \( \psi_x \) is the potential flux in the \( x \) direction [5].

Furthermore, a necessary condition for high-order discretizations is that \( F^*_x \) be continuously differentiable [11]. Analogous conditions hold for \( F^*_y \). Examples of computationally practical entropy-conservative flux functions include the flux of Ismail and Roe [6] and the flux of Chandrashekar [7].

While previous work on entropy-conservative SBP discretizations has focused on discontinuous-solution discretizations, we can still leverage this prior work in the case of a C-SBP discretization. Indeed, we can build an entropy conservative scheme directly from the global SBP operators \( D_x \) and \( D_y \) and any EC flux function. If the discrete solution is denoted by \( \mathcal{U}_h \in \mathbb{R}^{4n} \), where \( n \) denotes the number of nodes in the global mesh, then the EC discretization takes the form

\[ \frac{d\mathcal{U}_h}{dt} + \left[ \mathcal{D}_x \circ F_x(\mathcal{U}_h) \right] \mathbf{1} + \left[ \mathcal{D}_y \circ F_y(\mathcal{U}_h) \right] \mathbf{1} = s_h(\mathcal{U}_h, \mathcal{U}_h^{bc}(x, t)), \tag{7} \]

where, as before, the term \( s_h(\mathcal{U}_h, \mathcal{U}_h^{bc}(x, t)) \) denotes penalties that enforce the boundary conditions weakly. These penalties are similar to those implemented for discontinuous SBP discretizations, and so they will not be described further here; entropy-stable boundary conditions remain an active area of research.

In (7) we have introduced \( \mathcal{D}_x = D_x \otimes \mathcal{I}_4 \) and \( \mathcal{D}_y = D_y \otimes \mathcal{I}_4 \), where \( \otimes \) is the Kronecker product, to account for the four conservation equations present in the Euler PDE. In addition, we have introduced the matrices \( F_x(\mathcal{U}_h) \) and \( F_y(\mathcal{U}_h) \), which appear in Hadamard products with \( \mathcal{D}_x \) and \( \mathcal{D}_y \), respectively. The matrices \( F_x(\mathcal{U}_h) \) and \( F_y(\mathcal{U}_h) \) are sparse matrices that evaluate the EC flux function between “neighboring” nodes. For example, the \((i, j)\)th \( 4 \times 4 \) block in \( F_x(\mathcal{U}_h) \) is defined by

\[ [F_x(\mathcal{U}_h)]_{ij} = \begin{cases} \text{diag} \left[ F^*_x(\mathcal{U}_i, \mathcal{U}_j) \right], & \text{if } [\mathcal{D}_x]_{i,j} \neq 0, \\ 0_{4 \times 4}, & \text{if } [\mathcal{D}_x]_{i,j} = 0, \end{cases} \]
where \( \text{diag}(\cdot) \) produces a diagonal matrix with the entries of its argument along the diagonal, and \( 0_{4 \times 4} \) is the \( 4 \times 4 \) zero matrix.

One can show that (7) conserves entropy semi-discretely over a periodic domain where \( s_h = 0 \). Let \( w_h \in \mathbb{R}^{4n} \) denote the entropy variables corresponding to \( u_h \). Then,

\[
w_h^T \Phi \frac{d}{dt}u_h + w_h^T \Phi \left[ \bar{D}_x \circ F_x(u_h) \right] 1 + w_h^T \Phi \left[ \bar{D}_y \circ F_y(u_h) \right] 1 = \frac{d}{dt}1^T \Phi S_h = 0,
\]

where \( \Phi = H \otimes I_4 \), and \( S_h \in \mathbb{R}^n \) holds the (mathematical) entropy at the nodes of the mesh. The proof of the above statement is similar to that used for discontinuous-solution SBP discretizations [11], so it is omitted here. Interested readers are directed to [22] for the proof in the case of C-SBP.

C. Entropy-stable discretization

LPS dissipation is easily adapted to (7) to obtain an entropy-stable discretization of the Euler equations. The key idea is that the dissipation should be applied to the entropy variables, \( w_h \). Thus, including entropy-stable LPS, the discretization (7) becomes

\[
\frac{d}{dt}u_h + \left[ \bar{D}_x \circ F_x(u_h) \right] 1 + \left[ \bar{D}_y \circ F_y(u_h) \right] 1 = s_h(u_h, \ell_{bc}(x,t)) - \Phi^{-1} \bar{M}_h w_h,
\]

where the LPS dissipation operator on the right is

\[
- \bar{M}_h w_h = - \sum_{k=1}^{K} \bar{N}_k (\bar{P}_k)^T \bar{H}_x A_k(u_k) \bar{P}_k \bar{R}_k w_h.
\]

The barred matrices \( \bar{R}_k, \bar{P}_k, \) and \( \bar{H}_k \) denote the scalar operators in Kronecker products with \( I_4 \).

Like the matrix \( A_k \) used in the stabilization of the linear-advection equation, the scaling matrix \( A_k(u_k) \) in (9) is proportional to the flow velocity. In the case of the Euler equations, \( A_k(u_k) \) is a block diagonal matrix in which the block corresponding to node \( i \) is given by the \( 4 \times 4 \) matrix

\[
[A_k(u_k)]_i = \left[ \frac{1}{\rho} (\sigma_{\xi} + \sigma_{\eta}) \frac{\partial U}{\partial \mathcal{W}} \right]_i.
\]

The scalars \( \sigma_{\xi} \) and \( \sigma_{\eta} \) are the spectral radii of the flux Jacobians in reference space:

\[
\sigma_{\xi} = J \left( |\nabla_{x,\xi} \cdot (u, v)| + a ||\nabla_{x,\xi}|| \right),
\]

and \( \sigma_{\eta} = J \left( |\nabla_{x,\eta} \cdot (u, v)| + a ||\nabla_{x,\eta}|| \right),
\]

where \( J \) is the mapping Jacobian and \( a = \sqrt{\gamma p/\rho} \) is the speed of sound. The matrix \( \partial U/\partial \mathcal{W} \) is the inverse of the Hessian \( \partial^2 S/\partial U^2 \) and, therefore, symmetric positive definite [5].

We can now show that the discretization (8) is entropy stable on periodic domains. This follows from the entropy conservation of (7) and the positive definiteness of \( \bar{M}_h \):

\[
w_h^T \Phi \frac{d}{dt}u_h + w_h^T \Phi \left[ \bar{D}_x \circ F_x(u_h) \right] 1 + w_h^T \Phi \left[ \bar{D}_y \circ F_y(u_h) \right] 1 = -w_h^T \Phi \left( \Phi^{-1} \bar{M}_h w_h \right)
\]

\[
\Rightarrow \quad \frac{d}{dt}1^T \Phi S_h = -w_h^T \bar{M}_h w_h
\]

\[
= - \sum_{k=1}^{K} w_h^T \bar{R}_k (\bar{P}_k)^T \bar{H}_x A_k(u_k) \bar{P}_k \bar{R}_k w_h
\]

\[
= - \sum_{k=1}^{K} w_h^T (\bar{P}_k)^T \bar{H}_x A_k(u_k) \bar{P}_k w_h
\]

\[
geq 0.
\]

Thus, the LPS dissipation is guaranteed to decrease* the global (mathematical) entropy.

---

*More precisely, LPS is guaranteed to not increase the entropy.
IV. Numerical experiments

A. 2-dimensional linear advection

We begin by verifying the accuracy and stability of the LPS method for the constant-coefficient linear advection equation (1). We consider a square domain, $\Omega = [0, 1]^2$, with periodic boundary conditions. The advection velocity is $(\lambda_x, \lambda_y) = (1, 0)$, and the initial condition is given by the bump-shaped function

$$U(x, y, 0) = \begin{cases} 
1 - (4R^2 - 1)^{5/2} & \text{if } R \leq \frac{1}{2} \\
1 & \text{otherwise},
\end{cases}$$

where $R(x, y) \equiv \sqrt{(x - \frac{1}{2})^2 + (y - \frac{1}{2})^2}$. A similar problem was also used in [23] to verify the discontinuous SBP (D-SBP) discretization of the linear advection equation, and part of the motivation for choosing this problem is so we can benchmark the C-SBP discretization against the established D-SBP method.

A sequence of grids were constructed for the mesh refinement study using the “kernel” method described in [23]. The coarsest, level-1, grid is shown in Figure 1(a). The topological pattern in this coarsest mesh is repeated recursively on each triangle to obtain the finer grids in the sequence. The level-3 grid is shown in Figure 1(b) as an example. The elements in the mesh do not vary smoothly in size, so the C-SBP discretization has no advantage due to systematic error cancellations. Note that the kernel mesh, outlined in blue in Figure 1(a), differs from the one used in [23], and it produces a much larger ratio between the smallest and largest elements.

The benchmark D-SBP discretization is based on the one described in [23] and uses the so-called SBP operators. These operators produce discretizations that are analogous to tensor-product Legendre-Gauss collocation methods. However, the $p = 3$ and $p = 4$ operators used in [23] had only $2p - 1$ exact cubatures. In contrast, all of the operators used in the current work have cubatures that are $2p$ exact.

The C-SBP discretization, described in Section II, uses so-called diagonal-E operators. As the name suggests, these operators have diagonal $E_x$ and $E_y$ matrices. A degree $p$ diagonal-E operator has $p + 2$ nodes on its edges whose locations are given by the degree $p + 1$ Legendre-Gauss-Lobatto quadrature nodes. Furthermore, the node locations define a $2p$ exact cubature rule with (positive) weights given by the diagonal entries in $H$. The resulting cubature rules have more nodes than necessary to define $D_{x,k}$ and $D_{y,k}$ for a total degree $p$ basis, so the remaining degrees of freedom in $D_{x,k}$ and $D_{y,k}$ are used to minimize the Frobenius norms of $H^{-1}_{x,k}S_{x,k}$ and $H^{-1}_{y,k}S_{y,k}$; this bounds the spectral radius of the
Table 1  Maximally stable time step for the constant-coefficient advection problem, relative to the C-SBP $p = 1$ discretization.

<table>
<thead>
<tr>
<th></th>
<th>p=1</th>
<th>p=2</th>
<th>p=3</th>
<th>p=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-SBP</td>
<td>1.00$\Delta t_{\text{ref}}$</td>
<td>0.47$\Delta t_{\text{ref}}$</td>
<td>0.28$\Delta t_{\text{ref}}$</td>
<td>0.19$\Delta t_{\text{ref}}$</td>
</tr>
<tr>
<td>D-SBP</td>
<td>0.44$\Delta t_{\text{ref}}$</td>
<td>0.30$\Delta t_{\text{ref}}$</td>
<td>0.20$\Delta t_{\text{ref}}$</td>
<td>0.17$\Delta t_{\text{ref}}$</td>
</tr>
</tbody>
</table>

Fig. 2  Mesh convergence study; $l_\infty$ error versus maximum element edge length, $h_{\text{max}}$.

Both the D-SBP and C-SBP discretizations use the classical fourth-order Runge-Kutta scheme to advance the solution one period, i.e. so that the bump-shaped solution returns to its initial position. Table 1 lists the maximum stable time steps that were used with each degree $p$ discretization. The maximum time step is given in terms of the reference time step for the $p = 1$ C-SBP scheme, which we denote by $\Delta t_{\text{ref}}$. Table 1 shows that the D-SBP scheme has a more restrictive time step for this particular problem.

Figure 2 presents the results of the mesh convergence study. The $l_\infty$ solution error is plotted versus the maximum edge length in the mesh, denoted $h_{\text{max}}$. The $l_\infty$ error is based on the SBP nodal values; note that computing the $L_\infty$ error would be ambiguous for SBP operators, since there is no basis in general. Overall, the errors produced by the two discretizations behave similarly, and both the C-SBP and D-SBP discretizations achieve optimal or near optimal convergence rates; only the $p = 3$ D-SBP discretization has sub-optimal convergence.

Next we consider efficiency. Figure 3 plots the $l_\infty$ solution error versus normalized CPU time; the times are normalized by the C-SBP $p = 1$ discretization on the coarsest mesh. For both the C-SBP and D-SBP discretizations, increasing the degree of the scheme increases efficiency. Comparing the two discretizations, we see that the C-SBP discretization tends to produce a more accurate solution for the same computational cost when $p = 1, 2, 3$. For the $p = 4$ schemes, the two discretizations are comparable. Note that the Julia code that these results are based on used pre-allocated work arrays to avoid expensive memory allocation, and was run on Julia version 0.6.2 with array-bound checking turned off.

Finally, Figures 4(b) and 4(a) show the difference between the initial and final solution “energy” versus the maximum...
Fig. 3  Efficiency study; $l_\infty$ error versus CPU time for the C-SBP and D-SBP discretizations.

element edge length. The difference, or change, in energy is defined as

$$\Delta \text{Energy} = \left| \left( \mathbf{u}_h^T H \mathbf{u}_h \right)_{t=0} - \left( \mathbf{u}_h^T H \mathbf{u}_h \right)_{t=1} \right|,$$

which is strictly positive for these energy-stable schemes. Furthermore, since the change in energy is an integral functional and the discretizations are adjoint consistent, we expect $\Delta \text{Energy}$ to converge to the exact value of zero at a superconvergent rate. This is confirmed by the results in Figures 4(b) and 4(a), which indicate convergence rates of approximately $2p + 2$.

B. Euler equations

Our last set of results are intended to verify the entropy conservation (without LPS) and entropy stability (with LPS) for the C-SBP discretization of the Euler equations that was described in Section III. We do not investigate the accuracy of the scheme here; the high-order convergence of the C-SBP discretization of the Euler equations is verified in [22].

The discretized Euler equations were solved on a unit-square domain with periodic boundary conditions. To generate the mesh, a uniform $6 \times 6$ quadrilateral grid was subdivided into triangles in the computational space $0 \leq \xi, \eta \leq 1$. This uniform mesh was then mapped to curvilinear elements in physical space using the transformation

$$x = \xi + \frac{1}{20} \sin(3\pi \xi) \sin(3\pi \eta), \quad y = \eta - \frac{1}{20} \sin(3\pi \xi) \sin(3\pi \eta).$$

Figure 5(a) shows the $p = 2$ mesh. The mapping was applied to the nodes of standard Lagrange elements with $(p + 2)(p + 3)/2$ nodes for a degree $p$ SBP operator. The mapping Jacobian used in the discretization was computed following the method described in Crean et al. [11], which ensured that the SBP discretization remained entropy conservative/stable on the curvilinear grid.

A discontinuous initial condition, similar to the one used in [25], was chosen in order to demonstrate that the discretization is highly robust despite its lack of shock-capturing mechanism; however, we emphasize that the baseline scheme is not monotonicity preserving and should not be used for discontinuous problems in general. To be precise, the initial condition was defined as

$$\mathbf{U}(x, 0)^T = \left[ \rho, \, \rho u, \, \rho v, \, e \right]_{t=0} = \begin{cases} [1.1, \ 0, \ 0, \ 5.1], & \text{if } \frac{1}{3} \leq x, y \leq \frac{2}{3}, \\
[1.0, \ 0, \ 0, \ 5.0], & \text{otherwise}. \end{cases}$$
The analytical initial condition is shown in Figure 5(b).

We used the implicit midpoint rule to advance the solution in time from \( t = 0 \) to \( t = 10 \). Although the midpoint rule is only second-order accurate in time, we found that it produced a relatively small entropy conservation error. This is important, because the current scheme is only semi-discretely entropy conservative/stable, and a time-marching method may generate or dissipate entropy, in general [5]. To keep such time-discretization errors reasonably small, we adopted a CFL number of 0.01.

If the time-discretization errors are sufficiently small, the entropy conservative C-SBP discretization should produce no change in total entropy from one time step to the next. This is confirmed by the results in Figure 6, which shows the change in total entropy as a function of time for discretizations of degree \( p = 1, 2, 3, \) and 4. The change in entropy is defined as

\[
\Delta s_h^{(k)} \equiv T \mathcal{H}(s_h^{(k)} - s_h^{(k-1)}),
\]

where \( s_h^{(k)} \) denotes the nodal values of the (mathematical) entropy at time step \( k \). Note that the magnitude of the fluctuations in \( \Delta s_h^{(k)} \) decrease at a rate of \( \Delta t^2 \) [22], which is consistent with these entropy-conservation errors being caused by the (second-order accurate) time marching scheme.

Our final study examines the effect that LPS dissipation has on total entropy. According to the analysis presented in Section III, the entropy-stable C-SBP discretization should satisfy \( \Delta s_h^{(k)} \leq 0 \); that is, the LPS terms should cause the total entropy to decrease or remain constant. This is precisely what we observe in Figure 7, which shows the change in entropy using the degree one through four entropy-stable discretizations. Comparing the magnitude of \( \Delta s_h^{(k)} \) across the four discretizations for times \( t > 2.5 \), we see that the higher order discretizations dissipate less entropy.

V. Conclusions

Summation-by-parts operators permit the construction of robust high-order discretizations. In the case of the Euler (and Navier-Stokes) equations, robustness can be achieved by constructing entropy-stable SBP discretizations. Over the past few years, considerable effort has been devoted to entropy-stable methods based on discontinuous-solution spaces. In contrast, there has been limited work on entropy-stable SBP discretizations based on continuous-solution spaces. Thus, one of the goals of this work was to help mature such continuous SBP methods.

We have presented continuous SBP discretizations of the linear advection equation and the Euler equations. The discretizations require stabilization to ensure optimal rates of convergence on sufficiently smooth problems. To this
end, we adapted the local-projection-stabilization (LPS) to SBP operators. The LPS-based dissipation offers several attractive properties in the context of SBP discretizations:

1) LPS can be constructed to be energy stable (e.g. linear advection) and entropy stable (e.g. Euler equations);
2) The proposed LPS dissipation is element local, so the stencil is the same as the unstabilized scheme and it avoid potentially expensive interface flux functions; and
3) The spectral radius of discretized conservation laws is small compared to other stabilization methods for C-SBP discretizations [22], and the CFL number is comparable to or better than D-SBP discretizations.

The results demonstrate that the C-SBP discretization achieves optimal convergence rates for the constant-coefficient linear-advection equation; optimal rates for the Euler equations are presented in [22]. The present results also indicate that the stabilized C-SBP discretizations have comparable accuracy to D-SBP methods on the same mesh. Furthermore, the C-SBP methods tend to be more efficient, at least for lower order discretizations when using explicit time-marching schemes. We also verified the energy stability of the C-SBP discretization of the linear advection equation and the entropy conservation/stability of the C-SBP discretization of the Euler equations. Overall, the C-SBP discretization using LPS dissipation offers a compelling alternative to D-SBP discretizations for robust, high-order CFD.

Acknowledgments

Thanks Anthony Ashley, Jared Crean, Sharanjeet Kaur, and Ge Yan for proofreading the paper. All the results in this paper were obtained using software written in Julia [26]. The plots were generated using Matplotlib [27], with help from the Numpy [28, 29] and Scipy [30] libraries.

References


Fig. 5 Example curvilinear mesh and initial condition for the entropy-conservation and entropy-stability studies.
Fig. 6  Change in entropy between steps for the entropy-conservative discretizations.


Fig. 7  Change in entropy between steps for the entropy-stable C-SBP discretizations.


