AIAA AVIATION Forum
17-21 June 2019, Dallas, Texas
AIAA Aviation 2019 Forum

10.2514/6.2019-3206

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.

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206

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

Copyright © 2019 by Jason Hicken. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

It is well-known that continuous Galkerin 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 κ with domain Ω_κ. Functions on κ are represented using their values at n_κ nodes whose coordinates are given by $S_\kappa = \{(x_i, y_i)\}_{i=1}^{n_\kappa}$. For example, if $\mathcal{U} \in L^2(\Omega_\kappa)$ is a generic function on the element κ, then we use $\boldsymbol{u}_\kappa \in \mathbb{R}^{n_\kappa}$ to denote the function sampled at the nodes S_κ; that is

$$(\boldsymbol{u}_\kappa)_i = \mathcal{U}(x_i, y_i), \qquad \forall i = 1, 2, \ldots, n_\kappa.$$

The matrix $\mathsf{D}_{x,\kappa}$ is a degree p SBP approximation to $\partial/\partial x$ on S_κ if the following three conditions are met.

1) $\mathsf{D}_{x,\kappa}\boldsymbol{p}$ is equal to $\partial\mathcal{P}/\partial x$ at the nodes S_κ, for all polynomials $\mathcal{P} \in \mathbb{P}_p(\Omega_\kappa)$, where $\mathbb{P}_p(\Omega_\kappa)$ denotes the space of polynomials of total degree p on Ω_κ, and $\boldsymbol{p}$ denotes $\mathcal{P}$ evaluated at the nodes S_κ.
2) $\mathsf{D}_{x,\kappa} = \mathsf{H}_\kappa^{-1}\mathsf{Q}_x$, where H_κ is symmetric positive-definite.
3) $\mathsf{Q}_{x,\kappa} = \mathsf{S}_{x,\kappa} + \frac{1}{2}\mathsf{E}_{x,\kappa}$, where $\mathsf{S}_{x,\kappa}^T = -\mathsf{S}_{x,\kappa}$, $\mathsf{E}_{x,\kappa}^T = \mathsf{E}_{x,\kappa}$, and $\mathsf{E}_{x,\kappa}$ satisfies

$$\boldsymbol{p}^T\mathsf{E}_{x,\kappa}\boldsymbol{q} = \int_{\partial\Omega_\kappa} \mathcal{P}\,\mathcal{Q}\,n_x d\Gamma,$$

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

Similar definitions holds for $\mathsf{D}_{y,\kappa}$ and $\mathsf{D}_{z,\kappa}$, but all difference operators should be based on the same mass, or norm, matrix H_κ. Note that H_κ is dense in general; however, in this work, we consider only diagonal-norm SBP operators, that is, operators for which H_κ is a diagonal matrix. In this case, the nodes S_κ and diagonal entries in H_κ define a quadrature rule on Ω_κ 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-Lobbato 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_κ 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 Ω. 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_κ denote the restriction operator for element κ in a mesh $\mathcal{T}_h \equiv \{\Omega_\kappa\}_{\kappa=1}^K$ with K elements. Then, a first-derivative SBP operator on Ω is given by

$$\mathsf{D}_x = \mathsf{H}^{-1}\mathsf{Q}_x$$

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206

where

$$\mathsf{Q}_x \equiv \sum_{\kappa=1}^{K} \mathsf{R}_\kappa^T \mathsf{Q}_{x,\kappa} \mathsf{R}_\kappa, \qquad \text{and} \qquad \mathsf{H} \equiv \sum_{\kappa=1}^{K} \mathsf{R}_\kappa^T \mathsf{H}_\kappa \mathsf{R}_\kappa.$$

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 $\mathcal{T}_h$ was 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 $\mathsf{D}_x = \mathsf{H}^{-1}\mathsf{Q}_x$. 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 $\mathsf{D}_x\mathbf{1} = \mathbf{0}$ is satisfied, D_x is not exact for higher degree polynomials in $\mathbb{P}_p(\Omega)$, 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 Ω:

$$\begin{aligned} \frac{\partial \mathcal{U}}{\partial t} + \lambda_x \frac{\partial \mathcal{U}}{\partial x} + \lambda_y \frac{\partial \mathcal{U}}{\partial y} = 0, \qquad & \forall x \in \Omega \\ \mathcal{U}(x,t) = \mathcal{U}_{\text{bc}}(x,t), \quad & \forall x \in \Gamma^-, \\ \mathcal{U}(x,0) = \mathcal{U}_0(x), \quad & \forall x \in \Omega, \end{aligned} \tag{1}$$

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

$$\begin{aligned} \frac{d\boldsymbol{u}_h}{dt} + \lambda_x \mathsf{D}_x \boldsymbol{u}_h + \lambda_y \mathsf{D}_y \boldsymbol{u}_h = \boldsymbol{s}_h(\boldsymbol{u}_h, \mathcal{U}_{\text{bc}}(x,t)), \\ \boldsymbol{u}_h(0) = \mathcal{U}_0(\boldsymbol{x}_h). \end{aligned} \tag{2}$$

In the above semi-discretization, $\boldsymbol{x}_h$ denotes the coordinates of the global node set in $\mathcal{T}_h$, and $\boldsymbol{s}_h$ 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 $\mathcal{U} \in L^2(\Omega_\kappa)$ be a given function. The L^2 projection of $\mathcal{U}$ onto the space $\mathbb{P}_q(\Omega_\kappa)$ of total degree q polynomials is denoted by $\tilde{\mathcal{U}} \in \mathbb{P}_q(\Omega_\kappa)$ and is the solution to the (element-local) problem

$$\int_{\Omega_\kappa} \tilde{\mathcal{V}}(\mathcal{U} - \tilde{\mathcal{U}})\, d\Omega = 0, \qquad \forall\, \tilde{\mathcal{V}} \in \mathbb{P}_q(\Omega_\kappa). \tag{3}$$

The projection $\tilde{\mathcal{U}}$ can be thought of as the low-frequency content in the solution; consequently, $\mathcal{U} - \tilde{\mathcal{U}}$ represents the high-frequency content that we want to dissipate. To express the projection operator in matrix form, we first write $\tilde{\mathcal{U}}$ in

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206

terms of orthogonal polynomials evaluated at the nodes of κ:

$$\tilde{\boldsymbol{u}}_\kappa = \mathsf{L}\boldsymbol{y}$$

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

$$\mathsf{L}^T\mathsf{H}_\kappa\left(\boldsymbol{u}_\kappa - \mathsf{L}\boldsymbol{y}\right) = \boldsymbol{0}, \qquad \Rightarrow \qquad \boldsymbol{y} = \mathsf{L}^T\mathsf{H}_\kappa\boldsymbol{u}_\kappa, \qquad \Rightarrow \qquad \tilde{\boldsymbol{u}}_\kappa = \mathsf{L}\mathsf{L}^T\mathsf{H}_\kappa\boldsymbol{u}_\kappa.$$

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

The high-frequency in $\boldsymbol{u}_\kappa$ can be obtained by applying the matrix operator $\mathsf{P}_\kappa \equiv \mathsf{I} - \mathsf{L}\mathsf{L}^T\mathsf{H}_\kappa$, as shown by the following equalities:

$$\boldsymbol{u}_\kappa - \tilde{\boldsymbol{u}}_\kappa = \boldsymbol{u}_\kappa - \mathsf{L}\mathsf{L}^T\mathsf{H}_\kappa\boldsymbol{u}_\kappa = \left(\mathsf{I} - \mathsf{L}\mathsf{L}^T\mathsf{H}_\kappa\right)\boldsymbol{u}_\kappa = \mathsf{P}_\kappa\boldsymbol{u}_\kappa.$$

We want to dissipate the high-frequency content, $\mathsf{P}_\kappa\boldsymbol{u}_\kappa$, 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

$$\int_{\Omega_\kappa} (\mathcal{V} - \tilde{\mathcal{V}})\mathcal{A}(x,y)(\mathcal{U} - \tilde{\mathcal{U}})\, d\Omega \approx \boldsymbol{v}_\kappa^T\left(\mathsf{P}_\kappa\right)^T\mathsf{H}_\kappa\mathsf{A}_\kappa\mathsf{P}_\kappa\boldsymbol{u}_\kappa, \tag{4}$$

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

The function $\mathcal{A}(x,y)$ — 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, $\mathcal{A}(x,y)$ is defined using the magnitude of the advection velocity in reference space:

$$\mathcal{A}(\xi,\eta) = \sqrt{\lambda_\xi^2 + \lambda_\eta^2},$$

where

$$\begin{aligned}\lambda_\xi &= J\lambda_x\partial_x\xi + J\lambda_y\partial_y\xi = \quad \lambda_x\partial_\eta y - \lambda_y\partial_\eta x,\\ \lambda_\eta &= J\lambda_x\partial_x\eta + J\lambda_y\partial_y\eta = -\lambda_x\partial_\xi y + \lambda_y\partial_\xi x,\end{aligned}$$

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

$$\mathsf{A}_\kappa = \mathrm{diag}\left[\mathcal{A}(x_1,y_1), \mathcal{A}(x_2,y_2), \ldots, \mathcal{A}(x_{n_\kappa},y_{n_\kappa})\right].$$

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_κ, much like the SBP operators H_κ and $\mathsf{D}_{x,\kappa}$ are assembled using the R_κ to obtain global operators. Thus, the global LPS dissipation operator is given by

$$\mathsf{M} \equiv \sum_{\kappa=1}^{K}\mathsf{R}_\kappa^T\left[\left(\mathsf{P}_\kappa\right)^T\mathsf{H}_\kappa\mathsf{A}_\kappa\mathsf{P}_\kappa\right]\mathsf{R}_\kappa.$$

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

2. Energy Stability

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

$$-\boldsymbol{u}_h^T\mathsf{M}\boldsymbol{u}_h = -\sum_{\kappa=1}^{K}\boldsymbol{u}_h^T\mathsf{R}_\kappa^T\left[\left(\mathsf{P}_\kappa\right)^T\mathsf{H}_\kappa\mathsf{A}_\kappa\mathsf{P}_\kappa\right]\mathsf{R}_\kappa\boldsymbol{u}_h = -\sum_{\kappa=1}^{K}\left(\mathsf{P}_\kappa\boldsymbol{u}_\kappa\right)^T\mathsf{H}_\kappa\mathsf{A}_\kappa\mathsf{P}\boldsymbol{u}_\kappa = -\sum_{\kappa=1}^{K}\underbrace{\tilde{\boldsymbol{u}}_\kappa^T\mathsf{H}_\kappa\mathsf{A}_\kappa\tilde{\boldsymbol{u}}_\kappa}_{\geq 0} \leq 0$$

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206

where $\mathsf{R}_\kappa \boldsymbol{u}_\kappa$ denotes $\boldsymbol{u}_h$ restricted to element κ. Thus, $-\boldsymbol{u}_h^T \mathsf{M} \boldsymbol{u}_h \leq 0$, 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 $\boldsymbol{s}_h(\boldsymbol{u}_h, \mathcal{U}_{\text{bc}}(x,t))$ are zero — and left multiply (2) by $\boldsymbol{u}_h^T \mathsf{H}$:

$$\begin{aligned} & \boldsymbol{u}_h^T \mathsf{H} \frac{d\boldsymbol{u}_h}{dt} + \boldsymbol{u}_h^T \mathsf{H} \left(\lambda_x \mathsf{D}_x \boldsymbol{u}_h + \lambda_y \mathsf{D}_y \boldsymbol{u}_h \right) = -\boldsymbol{u}_h^T \mathsf{H} \left(\mathsf{H}^{-1} \mathsf{M} \boldsymbol{u}_h \right) \\ \Rightarrow \quad & \frac{1}{2} \frac{d}{dt} \|\boldsymbol{u}_h\|_{\mathsf{H}}^2 + \underbrace{\lambda_x \boldsymbol{u}_h^T \mathsf{Q}_x \boldsymbol{u}_h + \lambda_y \boldsymbol{u}_h^T \mathsf{Q}_y \boldsymbol{u}_h}_{=0} = -\boldsymbol{u}_h^T \mathsf{M} \boldsymbol{u}_h \leq 0, \end{aligned}$$

where $\|\boldsymbol{u}_h\|_{\mathsf{H}}^2 = \boldsymbol{u}_h^T \mathsf{H} \boldsymbol{u}_h$ approximates the L^2 norm of $\boldsymbol{u}_h$. Note that, Q_x and Q_y are skew symmetric on a periodic domain, so $\boldsymbol{u}_h^T \mathsf{Q}_x \boldsymbol{u}_h = 0$ and $\boldsymbol{u}_h^T \mathsf{Q}_y \boldsymbol{u}_h = 0$. The bottom line is that $d\|\boldsymbol{u}_h\|^2/dt \leq 0$, 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 $\mathsf{D}_{x,\kappa}$ 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 $\tilde{\mathcal{U}} = \mathcal{U}$ because $\mathbb{P}_p = \mathbb{P}_q$. Thus, the high-frequency content is trivial, $\mathcal{U} - \tilde{\mathcal{U}} = 0$, and we get no dissipation. This shows that the projection operator P_κ 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 $\tilde{\mathcal{U}}$ 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 $\mathsf{P}\boldsymbol{u}_\kappa = 0$ if $\boldsymbol{u}_\kappa$ is a linear polynomial, while $\mathsf{P}\boldsymbol{u}_\kappa \neq 0$ for quadratics. Consequently, the dissipation is zero for linear polynomials and $\mathsf{M}\boldsymbol{u}_h \neq \boldsymbol{0}$ 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

$$\begin{aligned} & \frac{\partial \mathcal{U}}{\partial t} + \frac{\partial \mathcal{F}_x}{\partial x} + \frac{\partial \mathcal{F}_y}{\partial y} = 0, \qquad \forall x \in \Omega \\ & \mathsf{A}_n^- \mathcal{U}(x,t) = \mathsf{A}_n^- \mathcal{U}_{\text{bc}}(x,t), \quad \forall x \in \Gamma, \\ & \mathcal{U}(x,0) = \mathcal{U}_0(x), \quad \forall x \in \Omega. \end{aligned} \tag{5}$$

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.

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206

The Euler fluxes are given by

$$\mathcal{F}_x = \begin{bmatrix} \rho u & \rho u^2 + p & \rho uv & (e+p)u \end{bmatrix}^T, \qquad \text{and} \qquad \mathcal{F}_y = \begin{bmatrix} \rho v & \rho vu & \rho v^2 + p & (e+p)v \end{bmatrix}^T,$$

where $p = (\gamma - 1)(e - \frac{\rho}{2}(u^2 + v^2))$ is the pressure. In the boundary condition in (5), the matrix $\mathsf{A}_n^- = \frac{1}{2}\mathsf{X}(\Lambda_n - |\Lambda_n|)\mathsf{X}^{-1}$ denotes the inflow part of the flux Jacobian, where $\mathsf{A}_n = \mathsf{X}\Lambda_n\mathsf{X}^{-1}$ 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 $\mathcal{W} \equiv \partial\mathcal{S}/\partial\mathcal{U}$ denotes the entropy variables, then for smooth solutions one can show that [5]

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

where $\mathcal{G}_x = -\rho u \mathcal{S}$ and $\mathcal{G}_y = -\rho v \mathcal{S}$ are the entropy fluxes in the x- and y-coordinate directions, respectively. Note that the domain Ω' in (6) is any compact subset of Ω 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 $\mathcal{F}_x^\star : \mathbb{R}^4 \times \mathbb{R}^4 \to \mathbb{R}^4$ is an EC flux function in the x direction if it satisfies the following conditions:

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

Furthermore, a necessary condition for high-order discretizations is that $\mathcal{F}_x^\star$ be continuously differentiable [11]. Analogous conditions hold for $\mathcal{F}_y^\star$. 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 $\boldsymbol{u}_h \in \mathbb{R}^{4n}$, where n denotes the number of nodes in the global mesh, then the EC discretization takes the form

$$\frac{d\boldsymbol{u}_h}{dt} + \left[\bar{\mathsf{D}}_x \circ \mathsf{F}_x(\boldsymbol{u}_h)\right]\mathbf{1} + \left[\bar{\mathsf{D}}_y \circ \mathsf{F}_y(\boldsymbol{u}_h)\right]\mathbf{1} = \boldsymbol{s}_h(\boldsymbol{u}_h, \mathcal{U}_{\text{bc}}(x,t)), \tag{7}$$

where, as before, the term $\boldsymbol{s}_h(\boldsymbol{u}_h, \mathcal{U}_{\text{bc}}(x,t))$ 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 $\bar{\mathsf{D}}_x = \mathsf{D}_x \otimes \mathsf{I}_4$ and $\bar{\mathsf{D}}_y = \mathsf{D}_y \otimes \mathsf{I}_4$, 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 $\mathsf{F}_x(\boldsymbol{u}_h)$ and $\mathsf{F}_y(\boldsymbol{u}_h)$, which appear in Hadamard products with $\bar{\mathsf{D}}_x$ and $\bar{\mathsf{D}}_y$, respectively. The matrices $\mathsf{F}_x(\boldsymbol{u}_h)$ and $\mathsf{F}_y(\boldsymbol{u}_h)$ are sparse matrices that evaluate the EC flux function between "neighboring" nodes. For example, the (i,j)th 4×4 block in $\mathsf{F}_x(\boldsymbol{u}_h)$ is defined by

$$[\mathsf{F}_x(\boldsymbol{u}_h)]_{ij} = \begin{cases} \text{diag}\left[\mathcal{F}_x^\star(\boldsymbol{u}_i, \boldsymbol{u}_j)\right], & \text{if } [\mathsf{D}_x]_{i,j} \neq 0, \\ \mathsf{0}_{4\times 4}, & \text{if } [\mathsf{D}_x]_{ij} = 0, \end{cases},$$

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206

where diag$(\cdot)$ produces a diagonal matrix with the entries of its argument along the diagonal, and $0_{4\times4}$ is the 4×4 zero matrix.

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

$$\boldsymbol{w}_h^T \bar{\mathsf{H}} \frac{d\boldsymbol{u}_h}{dt} + \boldsymbol{w}_h^T \bar{\mathsf{H}} \left[\bar{\mathsf{D}}_x \circ \mathsf{F}_x(\boldsymbol{u}_h)\right] \mathbf{1} + \boldsymbol{w}_h^T \bar{\mathsf{H}} \left[\bar{\mathsf{D}}_y \circ \mathsf{F}_y(\boldsymbol{u}_h)\right] \mathbf{1} = \frac{d}{dt} \mathbf{1}^T \bar{\mathsf{H}} \boldsymbol{S}_h = 0,$$

where $\bar{\mathsf{H}} = \mathsf{H} \otimes \mathsf{I}_4$, and $\boldsymbol{S}_h \in \mathbb{R}^n$ 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 discreization of the Euler equations. The key idea is that the dissipation should be applied to the entropy variables, $\boldsymbol{w}_h$. Thus, including entropy-stable LPS, the discretization (7) becomes

$$\frac{d\boldsymbol{u}_h}{dt} + \left[\bar{\mathsf{D}}_x \circ \mathsf{F}_x(\boldsymbol{u}_h)\right] \mathbf{1} + \left[\bar{\mathsf{D}}_y \circ \mathsf{F}_y(\boldsymbol{u}_h)\right] \mathbf{1} = \boldsymbol{s}_h(\boldsymbol{u}_h, \mathcal{U}_{\text{bc}}(x,t)) - \bar{\mathsf{H}}^{-1} \bar{\mathsf{M}}_h \boldsymbol{w}_h, \tag{8}$$

where the LPS dissipation operator on the right is

$$-\bar{\mathsf{M}}_h \boldsymbol{w}_h = -\left[\sum_{\kappa=1}^{K} \bar{\mathsf{R}}_\kappa^T \left(\bar{\mathsf{P}}_\kappa\right)^T \bar{\mathsf{H}}_\kappa \mathsf{A}_\kappa(\boldsymbol{u}_\kappa) \bar{\mathsf{P}}_\kappa \bar{\mathsf{R}}_\kappa\right] \boldsymbol{w}_h. \tag{9}$$

The barred matrices $\bar{\mathsf{R}}_\kappa$, $\bar{\mathsf{P}}_\kappa$, and $\bar{\mathsf{H}}_\kappa$ denote the scalar operators in Kronecker products with I_4.

Like the matrix A_κ used in the stabilization of the linear-advection equation, the scaling matrix $\mathsf{A}_\kappa(\boldsymbol{u}_\kappa)$ in (9) is proportional to the flow velocity. In the case of the Euler equations, $\mathsf{A}_\kappa(\boldsymbol{u}_\kappa)$ is a block diagonal matrix in which the block corresponding to node i is given by the 4×4 matrix

$$[\mathsf{A}_\kappa(\boldsymbol{u}_\kappa)]_i = \left[\frac{1}{2}(\sigma_\xi + \sigma_\eta)\frac{\partial \mathcal{U}}{\partial \mathcal{W}}\right]_i.$$

The scalars σ_ξ and σ_η are the spectral radii of the flux Jacobians in reference space:

$$\begin{aligned} \sigma_\xi &= J\left(|\nabla_x \xi \cdot (u,v)| + a\|\nabla_x \xi\|\right), \\ \text{and} \qquad \sigma_\eta &= J\left(|\nabla_x \eta \cdot (u,v)| + a\|\nabla_x \eta\|\right), \end{aligned}$$

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

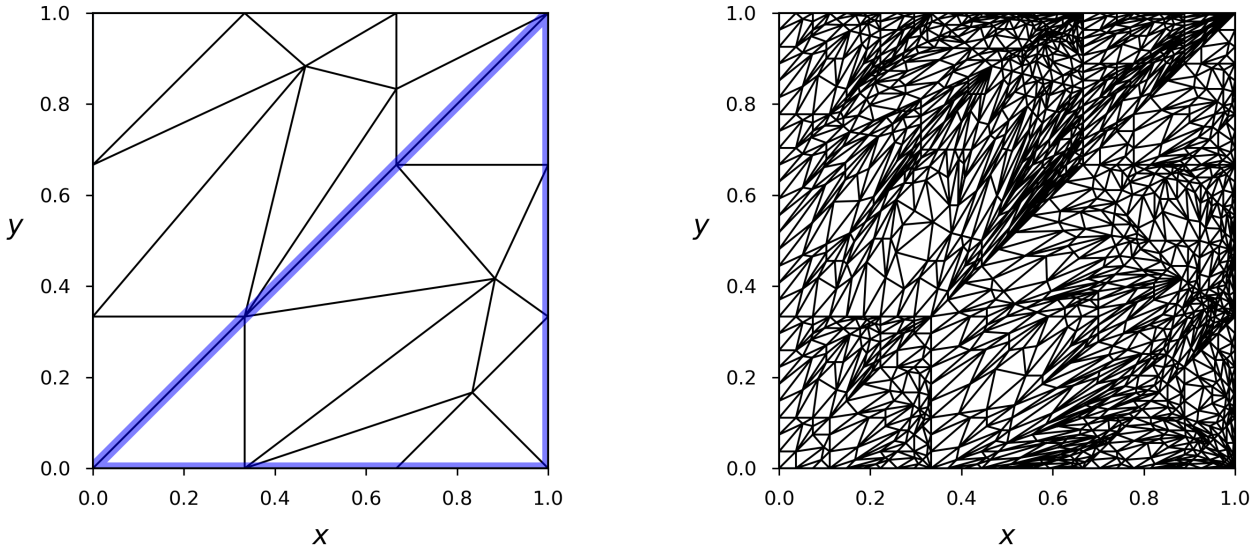
We can now show that the discretization (8) is entropy stable on periodic domains. This follows from the entropy conservation of (7) and the positive definiteness of $\bar{\mathsf{M}}_h$:

$$\begin{aligned} \boldsymbol{w}_h^T \bar{\mathsf{H}} \frac{d\boldsymbol{u}_h}{dt} + \boldsymbol{w}_h^T \bar{\mathsf{H}} \left[\bar{\mathsf{D}}_x \circ \mathsf{F}_x(\boldsymbol{u}_h)\right] \mathbf{1} + \boldsymbol{w}_h^T \bar{\mathsf{H}} \left[\bar{\mathsf{D}}_y \circ \mathsf{F}_y(\boldsymbol{u}_h)\right] \mathbf{1} &= -\boldsymbol{w}_h^T \bar{\mathsf{H}} \left(\bar{\mathsf{H}}^{-1} \bar{\mathsf{M}}_h \boldsymbol{w}_h\right) \\ \Rightarrow \qquad \frac{d}{dt} \mathbf{1}^T \bar{\mathsf{H}} \boldsymbol{S}_h &= -\boldsymbol{w}_h^T \bar{\mathsf{M}}_h \boldsymbol{w}_h \\ &= -\sum_{\kappa=1}^{K} \boldsymbol{w}_h^T \bar{\mathsf{R}}_\kappa^T \left(\bar{\mathsf{P}}_\kappa\right)^T \bar{\mathsf{H}}_\kappa \mathsf{A}_\kappa(\boldsymbol{u}_\kappa) \bar{\mathsf{P}}_\kappa \bar{\mathsf{R}}_\kappa \boldsymbol{w}_h \\ &= -\sum_{\kappa=1}^{K} \underbrace{\boldsymbol{w}_\kappa^T \left(\bar{\mathsf{P}}_\kappa\right)^T \bar{\mathsf{H}}_\kappa \mathsf{A}_\kappa(\boldsymbol{u}_\kappa) \bar{\mathsf{P}}_\kappa \boldsymbol{w}_\kappa}_{\geq 0} \\ &\leq 0. \end{aligned}$$

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

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

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206



(a) level-1 grid

(b) level-3 grid

Fig. 1 Example grids used for the linear-advection convergence study.

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206

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

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

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.

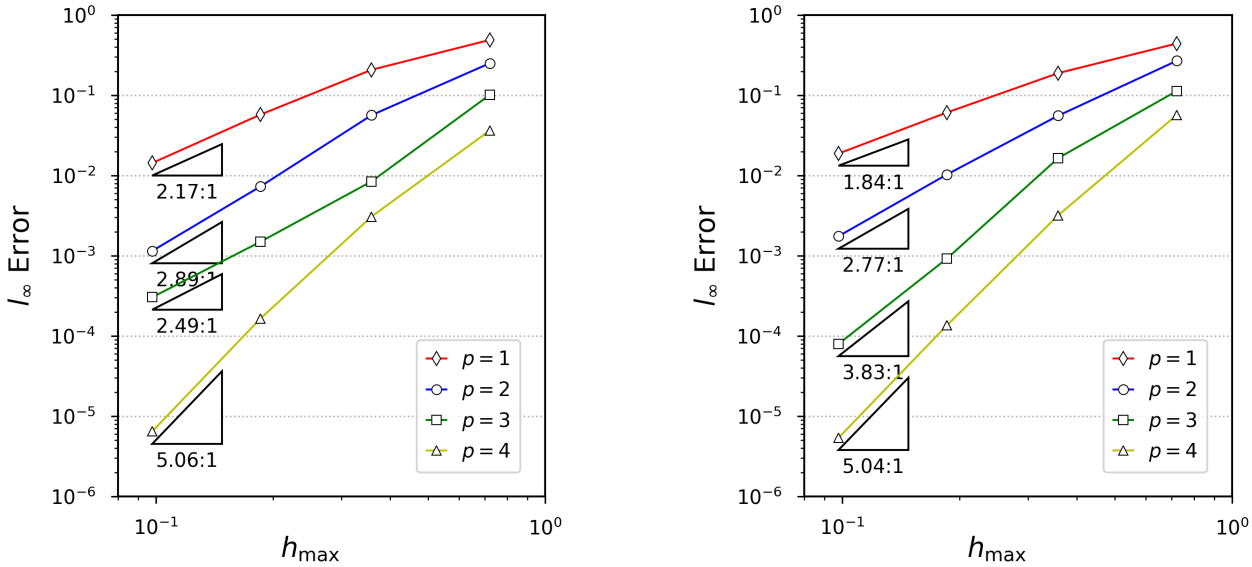
A sequence of grids were constructed for the mesh refinement study using the "kernel" method described in [23]. The coarsest, level-1, grid is shown in Figure 1(a). The topological pattern in this coarsest mesh is repeated recursively on each triangle to obtain the finer grids in the sequence. The level-3 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 Ω 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 $\mathsf{D}_{x,\kappa}$ and $\mathsf{D}_{y,\kappa}$ for a total degree p basis, so the remaining degrees of freedom in $\mathsf{D}_{x,\kappa}$ in $\mathsf{D}_{y,\kappa}$ are used to minimize the Frobenius norms of $\mathsf{H}_\kappa^{-1}\mathsf{S}_{x,\kappa}$ and $\mathsf{H}_\kappa^{-1}\mathsf{S}_{y,\kappa}$; 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.

	p=1	p=2	p=3	p=4
C-SBP	$1.00\Delta t_{\text{ref}}$	$0.47\Delta t_{\text{ref}}$	$0.28\Delta t_{\text{ref}}$	$0.19\Delta t_{\text{ref}}$
D-SBP	$0.44\Delta t_{\text{ref}}$	$0.30\Delta t_{\text{ref}}$	$0.20\Delta t_{\text{ref}}$	$0.17\Delta t_{\text{ref}}$



(a) D-SBP accuracy (b) C-SBP accuracy

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

element-level bilinear form, so minimizing the Frobenius norm helps to increase the maximum stable time step. The details regarding the construction of $\mathsf{D}_{x,\kappa}$ and $\mathsf{D}_{y,\kappa}$ are provided in [22].

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 Δt_{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_∞ solution error is plotted versus the maximum edge length in the mesh, denoted $h_{\max}$. The l_∞ error is based on the SBP nodal values; note that computing the L^∞ 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_∞ 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, or 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

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206

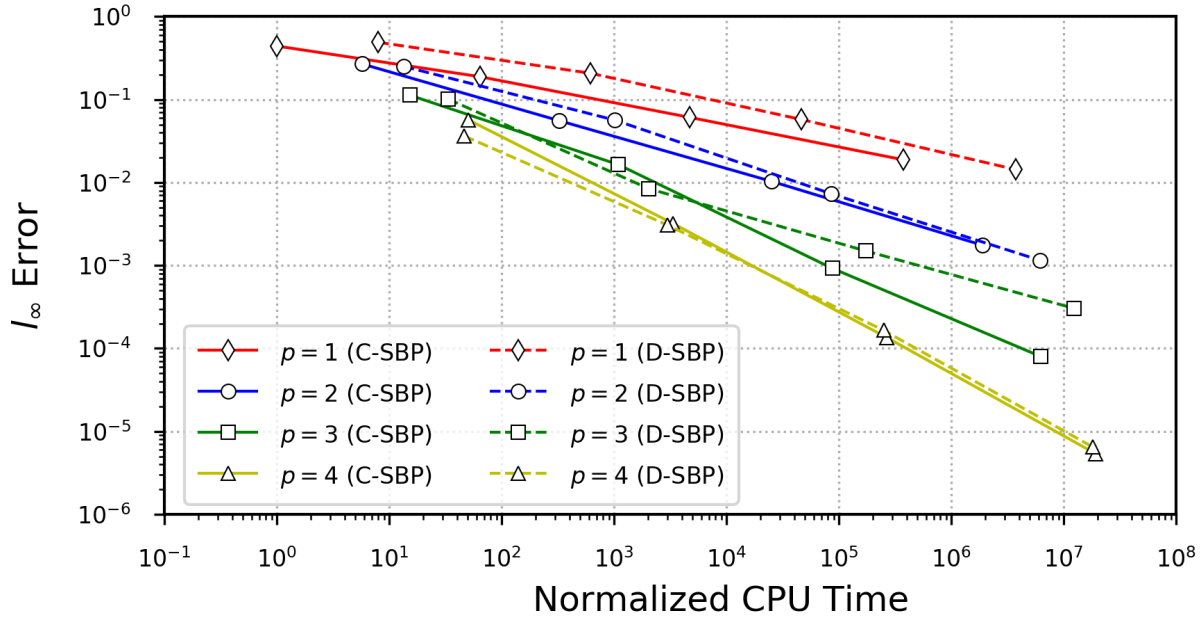


Fig. 3 Efficiency study; l_∞ error versus CPU time for the C-SBP and D-SBP discretizations.

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

$$\Delta\text{Energy} = (\boldsymbol{u}_h^T \mathrm{H} \boldsymbol{u}_h)\Big|_{t=0} - (\boldsymbol{u}_h^T \mathrm{H} \boldsymbol{u}_h)\Big|_{t=1},$$

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 ΔEnergy 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 6×6 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

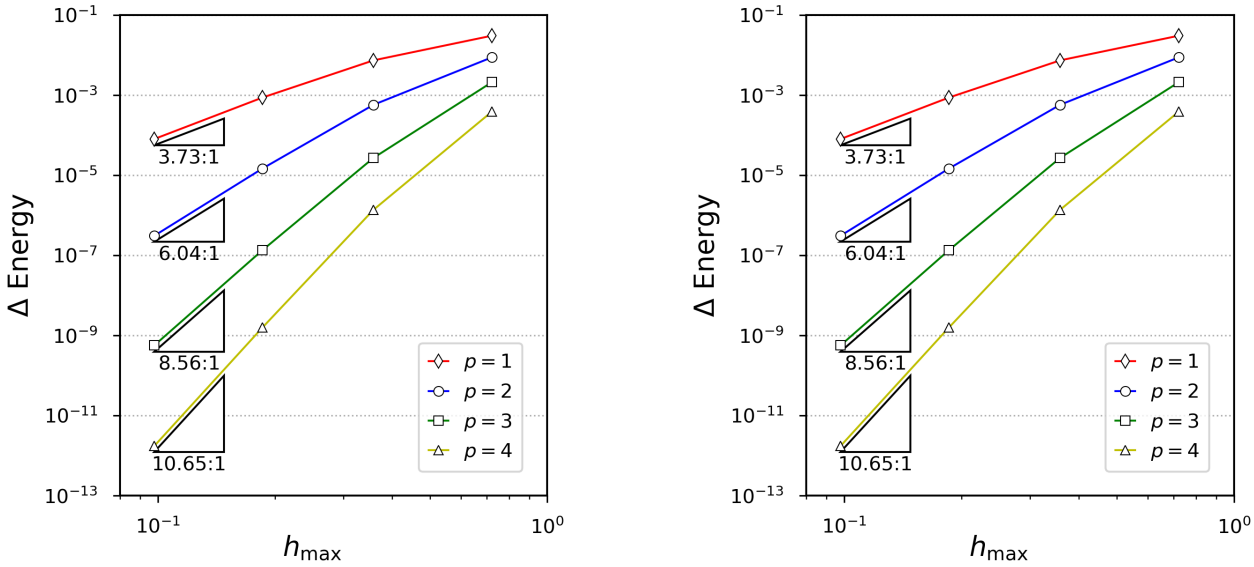
$$x = \xi + \frac{1}{20}\sin(3\pi\xi)\sin(3\pi\eta), \qquad y = \eta - \frac{1}{20}\sin(3\pi\xi)\sin(3\pi\eta).$$

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 discreitzation 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

$$\mathcal{U}(\boldsymbol{x}, 0)^T = \Big[\rho, \quad \rho u, \quad \rho v, \quad e\Big]\Big|_{t=0} = \begin{cases} \Big[1.1, \quad 0, \quad 0, \quad 5.1\Big], & \text{if } \frac{1}{3} \leq x, y \leq \frac{2}{3}, \\ \Big[1.0, \quad 0, \quad 0, \quad 5.0\Big], & \text{otherwise.} \end{cases}$$

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206



(a) D-SBP energy change

(b) C-SBP energy change

Fig. 4 Change in solution energy versus maximum element edge length, $h_{\max}$.

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

$$\Delta \boldsymbol{s}_h^{(k)} \equiv \mathbf{1}^T \mathsf{H}(\boldsymbol{s}_h^{(k)} - \boldsymbol{s}_h^{(k-1)}),$$

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

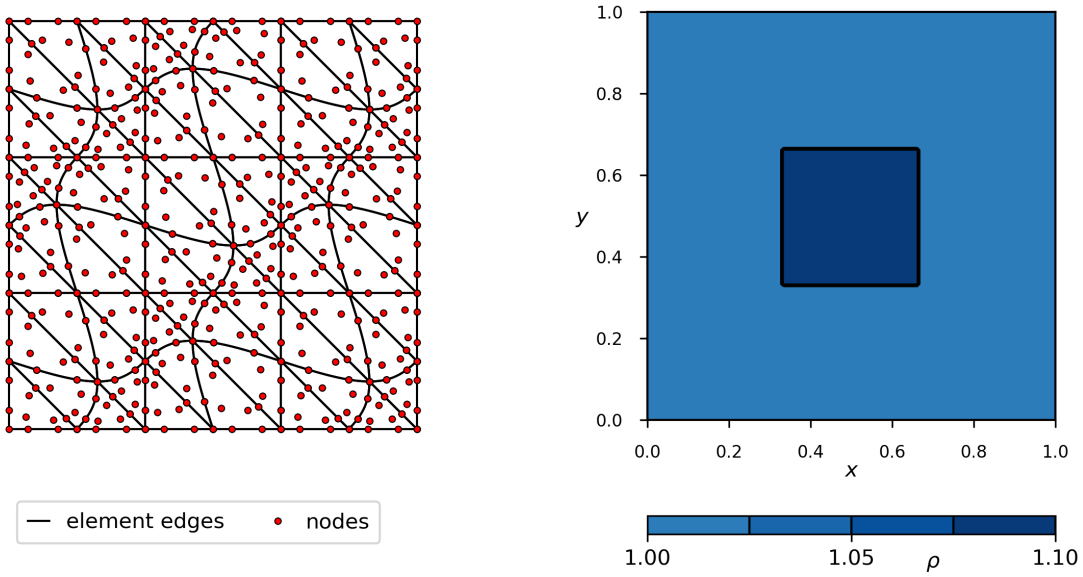
Our final study examines the effect that LPS dissipation has on total entropy. According to the analysis presented in Section III, the entropy-stable C-SBP discretization should satisfy $\Delta \boldsymbol{s}_h^{(k)} \leq 0$; that is, the LPS terms should cause the total entropy to decrease or remain 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 $\Delta \boldsymbol{s}_h^{(k)}$ 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

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206



(a) example mesh ($p = 2$)

(b) initial density

Fig. 5 Example curvilinear mesh and initial condition for the entropy-conservation and entropy-stability studies.

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206

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

[1] Fisher, T. C., "High-order L2 stable multi-domain finite difference method for compressible flows," Ph.D. thesis, Purdue University, 2012.

[2] Fisher, T. C., and Carpenter, M. H., "High-order entropy stable finite difference schemes for nonlinear conservation laws: Finite domains," *Journal of Computational Physics*, Vol. 252, No. 1, 2013, pp. 518–557.

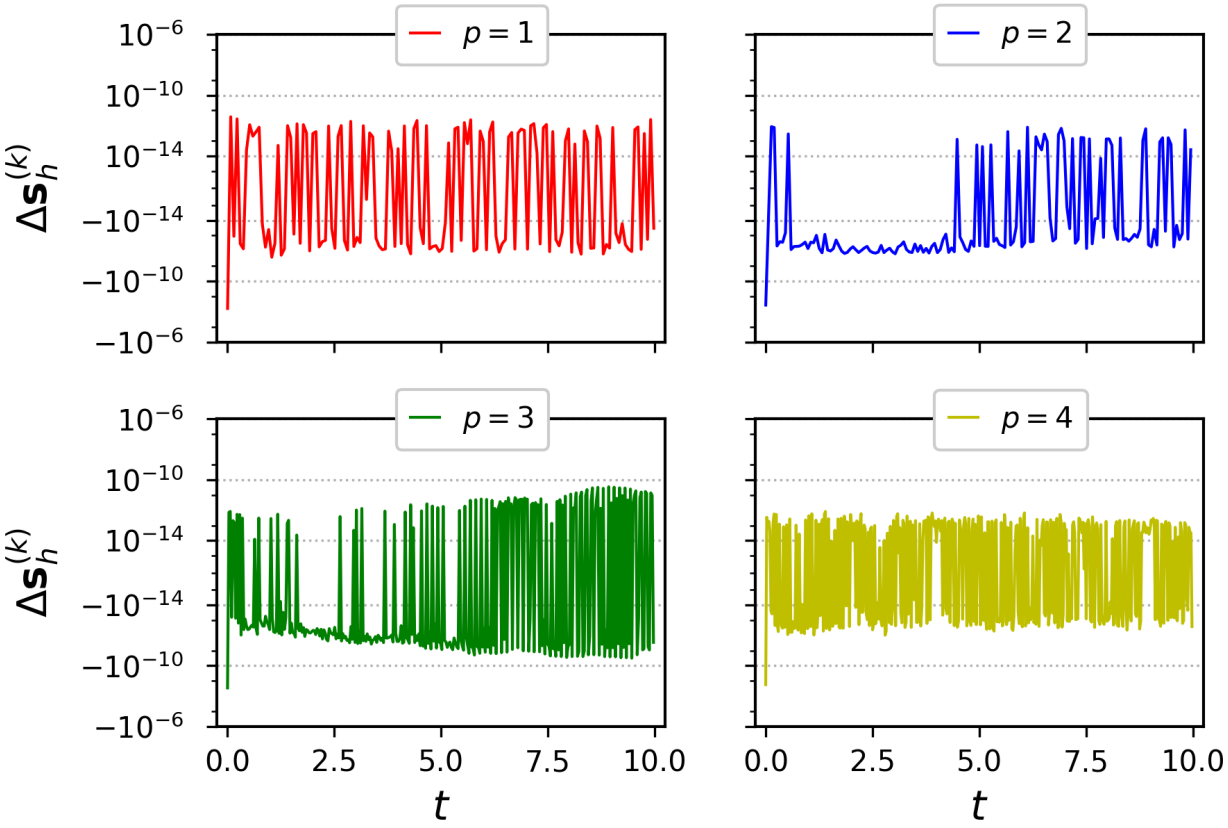


Fig. 6 Change in entropy between steps for the entropy-conservative discretizations.

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206

[3] Fisher, T. C., Carpenter, M. H., Nordström, J., Yamaleev, N. K., and Swanson, C., "Discretely conservative finite-difference formulations for nonlinear conservation laws in split form: Theory and boundary conditions," *Journal of Computational Physics*, Vol. 234, 2013, pp. 353 – 375. doi:http://dx.doi.org/10.1016/j.jcp.2012.09.026.

[4] Tadmor, E., "The numerical viscosity of entropy stable schemes for systems of conservation laws I," *Mathematics of Computation*, Vol. 49, No. 179, 1987, pp. 91–103.

[5] Tadmor, E., "Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems," *Acta Numerica*, Vol. 12, 2003, pp. 451–512. doi:10.1017/S0962492902000156.

[6] Ismail, F., and Roe, P. L., "Affordable, entropy-consistent Euler flux functions II: entropy production at shocks," *Journal of Computational Physics*, Vol. 228, No. 15, 2009, pp. 5410–5436. doi:http://dx.doi.org/10.1016/j.jcp.2009.04.021.

[7] Chandrashekar, P., "Kinetic Energy Preserving and Entropy Stable Finite Volume Schemes for Compressible Euler and Navier-Stokes Equations," *Communications in Computational Physics*, Vol. 14, No. 5, 2013, p. 1252–1286. doi:10.4208/cicp.170712.010313a.

[8] Carpenter, M. H., Fisher, T. C., Nielsen, E. J., and Frankel, S. H., "Entropy Stable Spectral Collocation Schemes for the Navier–Stokes Equations: Discontinuous Interfaces," *SIAM Journal on Scientific Computing*, Vol. 36, No. 5, 2014, p. B835–B867.

[9] Gassner, G. J., "A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods," *SIAM Journal on Scientific Computing*, Vol. 35, No. 3, 2013, pp. A1233–A1253.

[10] Hicken, J. E., Del Rey Fernández, D. C., and Zingg, D. W., "Multi-dimensional Summation-By-Parts Operators: General Theory and Application to Simplex Elements," *SIAM Journal on Scientific Computing*, Vol. 38, No. 4, 2016, pp. A1935–A1958.

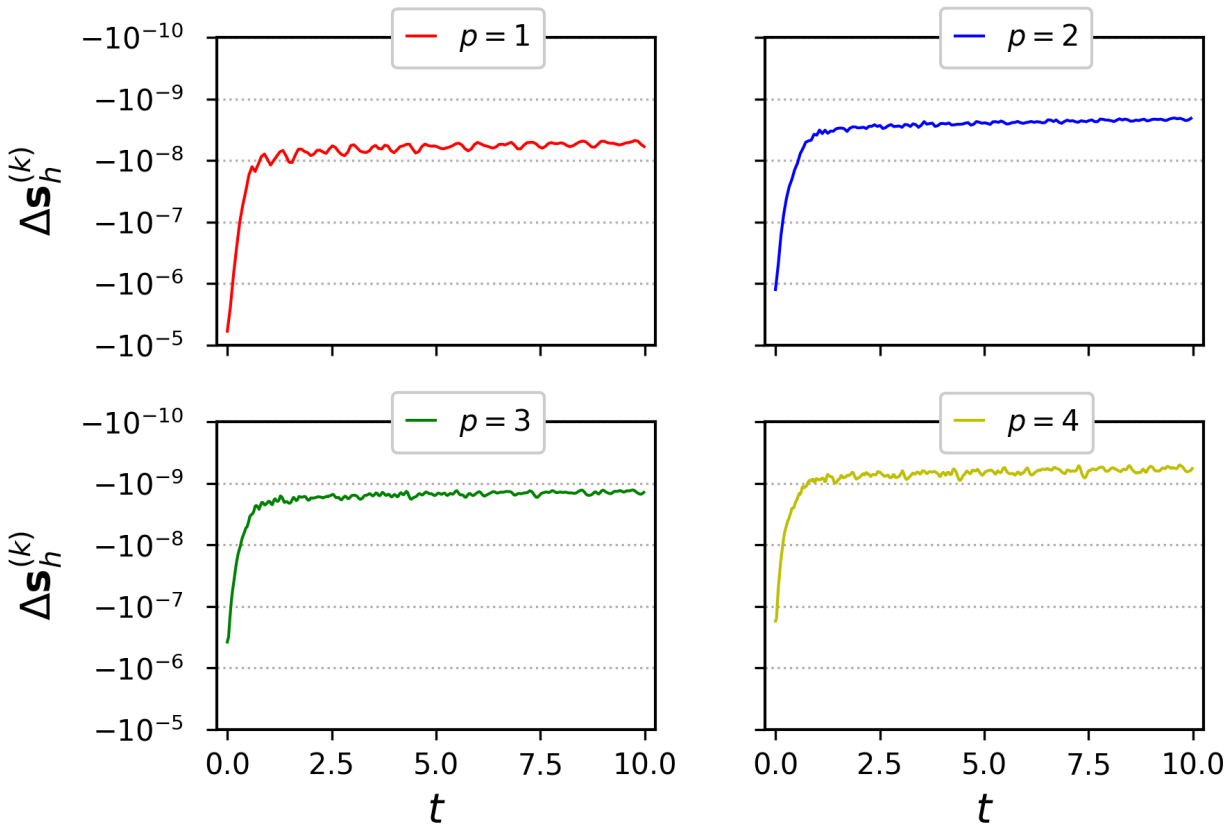


Fig. 7 Change in entropy between steps for the entropy-stable C-SBP discretizations.

[11] Crean, J., Hicken, J. E., Del Rey Fernández, D. C., Zingg, D. W., and Carpenter, M. H., "Entropy-stable summation-by-parts discretization of the Euler equations on general curved elements," *Journal of Computational Physics*, Vol. 356, 2018, pp. 410–438. doi:10.1016/j.jcp.2017.12.015.

[12] Chen, T., and Shu, C.-W., "Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws," *Journal of Computational Physics*, Vol. 345, 2017, pp. 427–461. doi:10.1016/j.jcp.2017.05.025.

[13] Hughes, T. J. R., Franca, L. P., and Mallet, M., "A new finite element formulation for computational fluid dynamics: I. symmetric forms of the compressible Navier-Stokes equations and the second law of thermodymaics," *Computer Methods in Applie Mechanics and Engineering*, Vol. 54, No. 2, 1986, pp. 223–234.

[14] Barth, T. J., "Numerical methods for gasdynamic systems on unstructured meshes," *An introduction to recent developments in theory and numerics for conservation laws*, Springer, 1999, pp. 195–285.

[15] Brooks, A. N., and Hughes, T. J. R., "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, No. 1-3, 1982, pp. 199–259. doi:10.1016/0045-7825(82)90071-8.

[16] Hughes, T. J. R., Franca, L. P., and Hulbert, G. M., "A new finite element formulation for computational fluid dynamics: VIII. The galerkin/least-squares method for advective-diffusive equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 73, No. 2, 1989, pp. 173–189. doi:10.1016/0045-7825(89)90111-4.

[17] Burman, E., Fernández, M. A., and Hansbo, P., "Continuous Interior Penalty Finite Element Method for Oseen's Equations," *SIAM Journal on Numerical Analysis*, Vol. 44, No. 3, 2006, pp. 1248–1274. doi:10.1137/040617686.

[18] Crean, J., Panda, K., Ashley, A., and Hicken, J. E., "Investigation of stabilization methods for multi-dimensional summation-by-parts discretizations of the Euler equations," *54th AIAA Aerospace Sciences Meeting*, San Diego, California, United States, 2016, p. 13. doi:10.2514/6.2016-1328, aIAA 2016-1328.

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206

[19] Hicken, J. E., and Zingg, D. W., "Summation-by-parts operators and high-order quadrature," *Journal of Computational and Applied Mathematics*, Vol. 237, No. 1, 2013, pp. 111–125.

[20] Baiocchi, C., Brezzi, F., and Franca, L. P., "Virtual bubbles and Galerkin-least-squares type methods (Ga.L.S.)," *Computer Methods in Applied Mechanics and Engineering*, Vol. 105, No. 1, 1993, pp. 125–141. doi:10.1016/0045-7825(93)90119-i.

[21] Franca, L. P., and Russo, A., "Deriving upwinding, mass lumping and selective reduced integration by residual-free bubbles," *Applied Mathematics Letters*, Vol. 9, No. 5, 1996, pp. 83–88. doi:10.1016/0893-9659(96)00078-x.

[22] Hicken, J. E., "Entropy-stable, high-order summation-by-parts discretizations without interface penalties," *Journal of Scientific Computing*, 2019. Submitted.

[23] Del Rey Fernández, D. C., Hicken, J. E., and Zingg, D. W., "Simultaneous Approximation Terms for Multi-dimensional Summation-by-Parts Operators," *Journal of Scientific Computing*, 2017, pp. 1–28. doi:10.1007/s10915-017-0523-7.

[24] Dafermos, C. M., *Hyperbolic Conservation Laws in Continuum Physics*, Vol. 325, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. doi:10.1007/978-3-642-04048-1.

[25] Del Rey Fernández, D. C., Crean, J., Carpenter, M. H., and Hicken, J. E., "Staggered-Grid Entropy-Stable Multidimensional Summation-By-Parts Discretizations on Curvilinear Coordinates," *Journal of Scientific Computing*, 2019. doi:10.1016/j.jcp.2019.04.029, in press, accepted manuscript.

[26] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B., "Julia: A fresh approach to numerical computing," *SIAM review*, Vol. 59, No. 1, 2017, pp. 65–98.

[27] Hunter, J. D., "Matplotlib: A 2D graphics environment," *Computing In Science & Engineering*, Vol. 9, No. 3, 2007, pp. 90–95. doi:10.1109/MCSE.2007.55.

[28] Oliphant, T. E., *A guide to NumPy*, Vol. 1, Trelgol Publishing USA, 2006.

[29] Van Der Walt, S., Colbert, S. C., and Varoquaux, G., "The NumPy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, Vol. 13, No. 2, 2011, p. 22.

[30] Jones, E., Oliphant, T., Peterson, P., et al., "SciPy: Open source scientific tools for Python," , 2001–. URL http://www.scipy.org/, [Online; accessed <today>].

Downloaded by Jason Hicken on June 8, 2020 | http://arc.aiaa.org | DOI: 10.2514/6.2019-3206