AIAA Aviation
13-17 June 2016, Washington, D.C.
46th AIAA Fluid Dynamics Conference

Simultaneous Approximation Terms for Multi-dimensional Summation-by-parts Operators

Jason E. Hicken*, [a] David C. Del Rey Fernández†, [b]
David W. Zingg‡, [b]
Rensselaer Polytechnic Institute[a]
University of Toronto Institute for Aerospace Studies[b]

This paper continues our effort to generalize summation-by-parts (SBP) finite-difference methods beyond tensor-products in multiple dimensions. In this work, we focus on the accurate and stable coupling of elements in the context of discontinuous solution spaces. We show how penalty terms — simultaneous approximation terms (SATs) — can be adapted to discretizations based on multi-dimensional SBP operators. We show that the part of the SBP operator corresponding to boundary integration can be decomposed into an interpolation/extrapolation operator and a boundary cubature. The SBP operators themselves are independent of the boundary cubature, and no additional degrees of freedom are introduced. The resulting decomposition facilitates the construction of SATs between arbitrary elements, and we prove that the resulting SBP-SAT discretizations are conservative and stable for divergence-free linear advection. The SATs are illustrated using triangular-element SBP operators with and without nodes that lie on the boundary. The solution accuracy of the resulting SBP-SAT discretizations is verified, and functional accuracy is shown to be superconvergent. The conservation and stability properties of the discretizations are confirmed on a divergence-free linear advection problem.

I. Introduction

Classical summation-by-parts (SBP) operators [1, 2] are high-order finite-difference methods that mimic integration by parts and, thereby, facilitate the construction of time-stable discretizations [3–9]. Like all one-dimensional finite-difference methods, classical SBP operators are typically applied to multi-dimensional problems using tensor-product operators defined on Cartesian reference domains. This approach, while adequate for many applications, has limitations in the context of complex geometries and localized mesh adaptation. This motivates our interest in generalizing SBP operators to more general multi-dimensional subdomains, such as simplicial elements.

In [10], Del Rey Fernández *et al.* considered a generalization of classical SBP operators. Motivated by this one-dimensional generalization, Ref. [11] proposed an SBP definition suitable for arbitrary (bounded) subdomains with piecewise smooth boundaries (see also [12]). We refer to these finite-difference operators as multi-dimensional SBP operators.

For diagonal-norm, multi-dimensional SBP operators[a] that are exact for polynomials of total degree p, it was shown in [11] that the norm and corresponding nodes define a strong cubature rule that is exact for polynomials of degree $2p-1$. This connection to cubature rules greatly simplifies the construction of multi-dimensional SBP operators, since many suitable cubature rules have already been identified in the literature [13].

*Assistant Professor, Department of Mechanical, Aerospace, and Nuclear Engineering, 110 8th Street, JEC Rm 2049, and AIAA Member
†Postdoctoral Fellow, Institute for Aerospace Studies, 4925 Dufferin St., and AIAA Student Member (dcdelrey@gmail.com)
‡Professor and Director, J. Armand Bombardier Foundation Chair in Aerospace Flight, Institute for Aerospace Studies, 4925 Dufferin St., and Associate Fellow AIAA.
[a]The norm matrix can be equivalently viewed as a mass matrix.

Multi-dimensional SBP operators, like their classical counterparts, provide a means of discretizing partial differential equations (PDEs), but they do not incorporate boundary conditions. Most SBP-based discretizations rely on simultaneous approximation terms (SATs) [14, 15] to impose boundary conditions and, when the solution space is discontinuous, couple elements. SATs are penalty terms that impose boundary data in a weak sense, and they provide stability without adversely impacting the asymptotic order of the discretization.

One way to apply SATs to multi-dimensional SBP discretizations is to use boundary-integrated numerical flux functions, similar to those used in the nodal discontinuous Galerkin method [16]. This was the approach adopted for the discontinuous SBP discretizations presented in [11]. A drawback of these SATs is that they rely on dense mass matrices over the faces where the penalty is applied. These face-based mass matrices are either dense with respect to the nodes on the face, in the case of the SBP operators in [11], or dense with respect to all of the nodes on an element, in the case of the multi-dimensional SBP operators presented in Section IV.

While dense SATs of the type described above produce provably stable SBP-SAT schemes for constant-coefficient advection, these schemes are not stable for more general advection velocities. The instability arises because the face-based mass matrix does not commute with the nodal advection velocity, in general. This is analogous to the issues experienced by dense norms and coordinate transformations [17].

Therefore, our primary objective in the present work is to address the stable and conservative implementation of SATs with multi-dimensional SBP operators for general, spatially varying advection velocities.

The remainder of the paper is organized as follows. After introducing some notation, Section II reviews the definition of multi-dimensional SBP operators from [11] and shows how the symmetric part of those operators can be related to auxiliary nodes on the boundary. Section III provides the theory for multi-dimensional SATs, including conservation and stability theorems. To illustrate SATs on a concrete example, Section IV presents two families of SBP operators for the triangle and describes how SATs are constructed for these operators. Some numerical results are given in Section V to verify the theory, and conclusions are provided in Section VI.

II. Some groundwork

A. Notation

Notation is consistent with that used in [11], and, as in that work, we focus on two-dimensional operators to simplify the presentation. We also restrict the definitions and theorems to operators defined in the x coordinate direction only, since operators defined in the other directions are analogous.

Functions of space are denoted with capital letters with a script type; e.g., $\mathcal{U}(x) \in L^2(\Omega)$ denotes a function on the domain Ω that is square-integrable. Functions and operators are discretized on a set of n nodes, $S = \{(x_i, y_i)\}_{i=1}^n$. The restriction of a function to the nodes is a column vector, which is represented using lower-case bold font; in the case of $\mathcal{U}$, we would write

$$\boldsymbol{u} = \left[\mathcal{U}(x_1, y_1), \ldots, \mathcal{U}(x_n, y_n)\right]^{\mathrm{T}}.$$

A basis for polynomials of total degree p has a cardinality of

$$n_p^* \equiv \binom{p+d}{d},$$

where d is the spatial dimension; for $d = 2$ we have $n_p^* = (p+1)(p+2)/2$. Several definitions and theorems rely on the monomial basis, defined below in (partial) order of nondecreasing degree.

$$\mathcal{P}_k(x, y) \equiv x^i y^{j-i}, \qquad k = j(j+1)/2 + i + 1, \quad \forall\, j \in \{0, 1, \ldots, p\}, \quad i \in \{0, 1, \ldots, j\}.$$

The monomials and their derivatives evaluated at the nodes are represented by

$$\boldsymbol{p}_k \equiv \left[\mathcal{P}_k(x_1, y_1), \ldots, \mathcal{P}_k(x_n, y_n)\right]^{\mathrm{T}},$$

$$\text{and} \qquad \boldsymbol{p}_k' \equiv \left[\frac{\partial \mathcal{P}_k}{\partial x}(x_1, y_1), \ldots, \frac{\partial \mathcal{P}_k}{\partial x}(x_n, y_n)\right]^{\mathrm{T}}.$$

B. Review of multi-dimensional summation-by-parts operators

To keep the presentation self-contained, we include the definition of a multi-dimensional SBP operator that was proposed in [11].

Definition 1. Two-dimensional summation-by-parts operator: *Consider an open and bounded domain $\Omega \subset \mathbb{R}^2$ with a piecewise-smooth boundary Γ. The matrix D_x is a degree p SBP approximation to the first derivative $\frac{\partial}{\partial x}$ on the nodes $S_\Omega = \{(x_i, y_i)\}_{i=1}^n$ if*

I. $\mathsf{D}_x \boldsymbol{p}_k = \boldsymbol{p}_k', \qquad \forall\, k \in \{1, 2, \ldots, n_p^*\}$;

II. $\mathsf{D}_x = \mathsf{H}^{-1}\mathsf{Q}_x$, *where* H *is symmetric positive-definite; and*

III. $\mathsf{Q}_x = \mathsf{S}_x + \frac{1}{2}\mathsf{E}_x$, *where* $\mathsf{S}_x^{\mathrm{T}} = -\mathsf{S}_x$, $\mathsf{E}_x^{\mathrm{T}} = \mathsf{E}_x$, *and* E_x *satisfies*

$$\boldsymbol{p}_k^{\mathrm{T}} \mathsf{E}_x \boldsymbol{p}_m = \oint_\Gamma \mathcal{P}_k \mathcal{P}_m n_x \mathrm{d}\Gamma, \qquad \forall\, k, m \in \{1, 2, \ldots, n_\tau^*\},$$

where $\tau \geq p$, and n_x is the x component of $\boldsymbol{n} = [n_x, n_y]^{\mathrm{T}}$, the outward pointing unit normal on Γ.

As mentioned in the introduction, SBP operators are closely linked to cubature rules. Under mild assumptions on a generalized Vandermonde matrix, the existence of a cubature implies the existence of a diagonal-norm SBP operator, and vice versa [11]. In particular,

$$\boldsymbol{p}_k^{\mathrm{T}} \mathsf{H} \boldsymbol{p}_m = \int_\Omega \mathcal{P}_k \mathcal{P}_m \mathrm{d}\Omega,$$

for $\mathcal{P}_k\mathcal{P}_m$ at most degree $2p-1$. Thus, the norm $\|\boldsymbol{u}\|_{\mathsf{H}}^2 \equiv \boldsymbol{u}^{\mathrm{T}}\mathsf{H}\boldsymbol{u}$ is a $2p$-order approximation to the integral L^2 norm over Ω. In addition, the matrices Q_x and S_x satisfy

$$\boldsymbol{p}_k^{\mathrm{T}} \mathsf{Q}_x \boldsymbol{p}_m = \int_\Omega \mathcal{P}_k \frac{\partial \mathcal{P}_m}{\partial x} \mathrm{d}\Omega,$$

$$\text{and} \qquad \boldsymbol{p}_k^{\mathrm{T}} \mathsf{S}_x \boldsymbol{p}_m = \int_\Omega \mathcal{P}_k \frac{\partial \mathcal{P}_m}{\partial x} \mathrm{d}\Omega - \frac{1}{2} \oint_\Gamma \mathcal{P}_k \mathcal{P}_m n_x \mathrm{d}\Gamma,$$

where $k, m \leq n_\tau^*$ and $\mathcal{P}_k\mathcal{P}_m$ is at most degree $2p$.

C. Decomposing E_x

Recall that the symmetric matrix $\mathsf{E}_x = 2(\mathsf{Q}_x - \mathsf{S}_x)$ approximates boundary integration in the multi-dimensional SBP definition:

$$\boldsymbol{v}^{\mathrm{T}} \mathsf{E}_x \boldsymbol{u} = \oint_\Gamma \mathcal{V}\mathcal{U} n_x \mathrm{d}\Gamma + \mathrm{O}(h^{2p}),$$

where h is the nominal element size. Not surprisingly, E_x has an important role to play in the weak imposition of boundary conditions. For example, in [11] E_x was decomposed into three matrices, one for each face on the triangle, and simultaneous approximation terms (SATs) were constructed using these face-based matrices; such SATs are equivalent to the approach used in the nodal DG method [16].

With the approach described above, the SATs for enforcing homogeneous boundary conditions in constant-coefficient advection are propoortional to

$$\mathsf{H}^{-1}(\mathsf{E}_x)_\nu \lambda_\nu \boldsymbol{u}$$

where $(\mathsf{E}_x)_\nu$ is E_x restricted to the face with index ν, and the scalar λ_ν is the normal component of the advection velocity along face ν. Now, consider how this SAT changes for a spatially varying advection:

$$\mathsf{H}^{-1}(\mathsf{E}_x)_\nu \Lambda \boldsymbol{u}$$

where $\Lambda = \mathrm{diag}(\lambda_\nu)$ is the matrix containing the spatially varying advection velocity evaluated at the SBP nodes. Whereas $(\mathsf{E}_x)_\nu \lambda$ is symmetric, $(\mathsf{E}_x)_\nu \Lambda$ is not symmetric, in general. This loss of symmetry presents

a significant challenge to stability analysis. We address it here by considering a more general decomposition of E_x.

Before we proceed further, we need to introduce some assumptions regarding the boundary, Γ, of the element domain Ω. The most restrictive of these is the requirement that Γ be piecewise linear. While this assumption holds for most reference elements (right triangle, right tetrahedron, etc.), it does not hold for more general curvilinear elements; future efforts will seek to remove this assumption.

Assumption 1. *$\Gamma = \bigcup_\nu \Gamma_\nu$ is orientable, $\bigcap_\nu \Gamma_\nu = \emptyset$, and each face Γ_ν is smooth, i.e. Γ is piecewise smooth. Furthermore, for each Γ_ν there is a linear bijective mapping $\mathcal{M} : \hat{\Gamma}_\nu \to \Gamma_\nu$, where the reference domains satisfy $\hat{\Gamma}_\nu \subset \mathbb{R}^{d-1}$ and have piecewise linear boundaries. Finally, for each reference face $\hat{\Gamma}_\nu$ and for each degree $p \geq 0$, there exists a strong cubature rule of degree $q \geq 2p$.*

Assumption 1 ensures that we can evaluate boundary fluxes with sufficient accuracy. For example, in two-dimensions and for each Γ_ν, we have a linear mapping $\mathcal{M}(\xi)$ and cubature nodes $S_{\hat{\Gamma}_\nu} = \{\xi_i\}_{i=1}^{n_\nu}$ and weights $\{b_i\}_{i=1}^{n_\nu}$ such that

$$\int_{\Gamma_\nu} \mathcal{U}(x,y) n_x \, \mathrm{d}\Gamma = \int_{\hat{\Gamma}_\nu} \mathcal{U}(\xi) \frac{\partial y}{\partial \xi} \, \mathrm{d}\hat{\Gamma} = \frac{\partial y}{\partial \xi} \sum_{i=1}^{n_\nu} b_i \mathcal{U}(\xi_i)$$

for all polynomials $\mathcal{U}(\xi)$ of degree $q \geq 2p$ or less. A similar expression holds for fluxes in the y direction. Note that, because $\mathcal{M}$ is linear, the term $\partial y / \partial \xi$ is constant over $\hat{\Gamma}_\nu$, which is necessary for the face cubature to remain exact for degree $2p$ polynomials. If $\mathcal{M}$ is not linear, but still polynomial, then the accuracy of the face cubature can be increased accordingly.

We are now in a position to show that E_x can be expressed in terms of auxiliary nodes that lie on the boundary Γ.

Theorem 1. *Let Assumption 1 hold and let D_x be a degree p SBP approximation of the first derivative on the domain Ω. Then the symmetric part of Q_x can be written as*

$$\frac{1}{2}\mathsf{E}_x = \frac{1}{2} \sum_\nu \left[\frac{\partial y}{\partial \xi} \right]_\nu \mathsf{R}_\nu^{\mathrm{T}} \mathsf{B}_\nu \mathsf{R}_\nu, \tag{1}$$

where $\mathsf{B}_\nu \in \mathbb{R}^{n_\nu \times n_\nu}$ is a diagonal matrix with the cubature weights of the reference domain $\hat{\Gamma}_\nu$ along the main diagonal, and $\mathsf{R}_\nu \in \mathbb{R}^{n_\nu \times n}$ is a degree $r \geq p$ interpolation/extrapolation operator from the nodes of the reference volume domain, $S_{\hat{\Omega}}$, to the nodes of the reference boundary domain, $S_{\hat{\Gamma}_\nu}$. That is,

$$(\mathsf{R}_\nu \boldsymbol{p}_k)_i = \mathcal{P}_k(\xi_i), \qquad \forall\, k \in \{1, 2, \ldots, n_r^*\}, \quad \forall\, \xi_i \in S_{\hat{\Gamma}_\nu}.$$

Proof. The terms $\mathsf{R}_\nu^{\mathrm{T}} \mathsf{B}_\nu \mathsf{R}_\nu$ are clearly symmetric by construction, as is E_x. Therefore, we need only show that the accuracy conditions of Property III hold. Since the R_ν are exact for degree $r \geq p$ polynomials, we have, $\forall k, m \in \{1, 2, \ldots, n_p^*\}$,

$$\begin{aligned}
\boldsymbol{p}_k^{\mathrm{T}} \mathsf{E}_x \boldsymbol{p}_m &= \sum_\nu \left[\frac{\partial y}{\partial \xi} \right]_\nu \boldsymbol{p}_k^{\mathrm{T}} \mathsf{R}_\nu^{\mathrm{T}} \mathsf{B}_\nu \mathsf{R}_\nu \boldsymbol{p}_m \\
&= \sum_\nu \left[\frac{\partial y}{\partial \xi} \right]_\nu \sum_{i=1}^{n_\nu} b_i \mathcal{P}_k \left(x(\xi_i), y(\xi_i) \right) \mathcal{P}_m \left(x(\xi_i), y(\xi_i) \right) \\
&= \sum_\nu \int_{\hat{\Gamma}_\nu} \mathcal{P}_k(\xi) \mathcal{P}_m(\xi) \frac{\partial y}{\partial \xi} \, \mathrm{d}\hat{\Gamma},
\end{aligned}$$

where we have used the linearity of the mappings (see Assumption 1) and the fact that the product $\mathcal{P}_k(\xi)\mathcal{P}_m(\xi)$ has total degree less than or equal to $2p$. The result follows by the additive property of integrals. □

III. Multi-dimensional simultaneous approximation terms

In this section, we show how multi-dimensional SBP elements can be coupled in an accurate, conservative, and stable manner using simultaneous approximation terms (SATs). The same approach can be used to impose boundary conditions weakly. We consider upwind SATs, since they are the most widely used; however, Ref. [18] includes an analysis of more general, parameterized SATs.

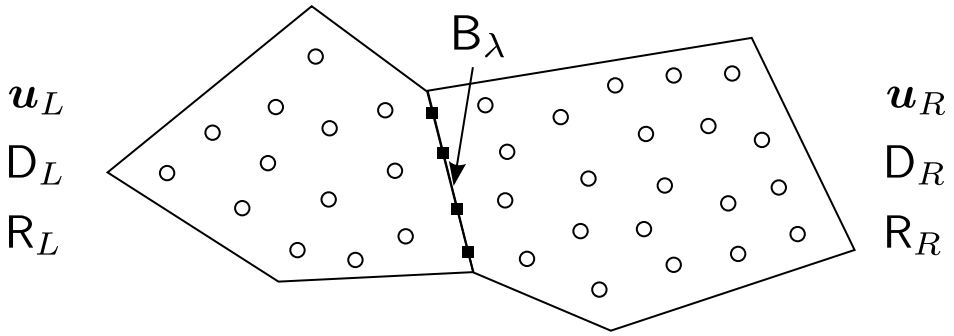


Figure 1. Illustration of two generic SBP elements and their common interface used for the analysis of SATs.

A. SBP-SAT discretization of the linear advection equation with a divergence-free velocity

By making use of the decomposition (1), we can derive a conservative and stable SBP discretization based on simultaneous approximation terms (SATs). To illustrate the process, we consider a scalar advection equation with a spatially-varying, divergence-free velocity $\boldsymbol{\lambda} = (\lambda_x, \lambda_y)$:

$$\frac{\partial \mathcal{U}}{\partial t} + \frac{1}{2}\nabla \cdot (\boldsymbol{\lambda}\mathcal{U}) + \frac{1}{2}\boldsymbol{\lambda} \cdot \nabla \mathcal{U} = 0, \tag{2}$$

$$\text{where} \qquad \nabla \cdot \boldsymbol{\lambda} = 0. \tag{3}$$

Note that we have written the PDE in skew-symmetric form. Our subsequent analysis focuses on coupling adjacent elements, so we ignore boundary conditions and discretize (2) on two elements with a common face (see Figure 1). Subscripts L and R denote quantities associated with the "left" and "right" domains, respectively. For example, $\boldsymbol{u}_L$ and $\boldsymbol{u}_R$ denote the discrete solutions on the left and right elements. To keep the notation simple, we assume that both SBP elements have the same number of nodes, i.e. $n_L = n_R = n$, but this is not necessary in general.

The following matrices will be useful in the discretization of (2) and (3). Recall that $\boldsymbol{n} = [n_x, n_y]$ is the unit normal on interface.

$$\begin{aligned}
&\Lambda_x \equiv \operatorname{diag}(\lambda_{x,1}, \lambda_{x,2}, \ldots, \lambda_{x,n}), && \Lambda_y \equiv \operatorname{diag}(\lambda_{y,1}, \lambda_{y,2}, \ldots, \lambda_{y,n}) \\
&\mathsf{B}_x \equiv n_x \operatorname{diag}(b_1, b_2, \ldots, b_{n_\nu}), && \mathsf{B}_y \equiv n_y \operatorname{diag}(b_1, b_2, \ldots, b_{n_\nu}), \\
&\mathsf{B}_\lambda \equiv \operatorname{diag}(b_1(\boldsymbol{\lambda}\cdot\boldsymbol{n})_1, b_2(\boldsymbol{\lambda}\cdot\boldsymbol{n})_2, \ldots, b_{n_\nu}(\boldsymbol{\lambda}\cdot\boldsymbol{n})_{n_\nu}),
\end{aligned}$$

To be clear, the diagonal matrices $\Lambda_x, \Lambda_y \in \mathbb{R}^{n\times n}$ hold the x and y components of the advection velocity, respectively, evaluated at the nodes of the SBP operator. The matrices B_x and B_y are diagonal matrices holding the face-cubature weights scaled by n_x and n_y, respectively. Finally, B_λ holds the product of the face-cubature weights with the normal component of the velocity. In the subsequent analysis, B_λ uses the outward normal with respect to the left domain in all cases, i.e. $\mathsf{B}_\lambda \equiv (\mathsf{B}_\lambda)_L = -(\mathsf{B}_\lambda)_R$.

An SBP-SAT discretization of (2) on the two-element domain is given by

$$\begin{aligned}
\frac{\mathrm{d}\boldsymbol{u}_L}{\mathrm{d}t} + \mathsf{D}_L\boldsymbol{u}_L &= \frac{1}{2}\mathsf{H}_L^{-1}\mathsf{R}_L^{\mathrm{T}}(\mathsf{B}_\lambda - |\mathsf{B}_\lambda|)(\mathsf{R}_L\boldsymbol{u}_L - \mathsf{R}_R\boldsymbol{u}_R), \\
\frac{\mathrm{d}\boldsymbol{u}_R}{\mathrm{d}t} + \mathsf{D}_R\boldsymbol{u}_R &= \frac{1}{2}\mathsf{H}_R^{-1}\mathsf{R}_R^{\mathrm{T}}(-\mathsf{B}_\lambda - |\mathsf{B}_\lambda|)(\mathsf{R}_R\boldsymbol{u}_R - \mathsf{R}_L\boldsymbol{u}_L),
\end{aligned} \tag{4}$$

where the matrix operators D_L and D_R are defined by

$$\begin{aligned}
\mathsf{D}_L &\equiv \frac{1}{2}\mathsf{H}_L^{-1}\left(-\mathsf{Q}_{xL}^{\mathrm{T}}\Lambda_{xL} - \mathsf{Q}_{yL}^{\mathrm{T}}\Lambda_{yL} + \mathsf{R}_L^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_L + \Lambda_{xL}\mathsf{Q}_{xL} + \Lambda_{yL}\mathsf{Q}_{yL}\right), \\
\mathsf{D}_R &\equiv \frac{1}{2}\mathsf{H}_R^{-1}\left(-\mathsf{Q}_{xR}^{\mathrm{T}}\Lambda_{xR} - \mathsf{Q}_{yR}^{\mathrm{T}}\Lambda_{yR} - \mathsf{R}_R^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_R + \Lambda_{xR}\mathsf{Q}_{xR} + \Lambda_{yR}\mathsf{Q}_{yR}\right),
\end{aligned}$$

and the diagonal, positive-semidefinite matrix $|\mathsf{B}_\lambda|$ is given by

$$|\mathsf{B}_\lambda| \equiv \operatorname{diag}(b_1|(\boldsymbol{\lambda}\cdot\boldsymbol{n})_1|, b_2|(\boldsymbol{\lambda}\cdot\boldsymbol{n})_2|, \ldots, b_{n_\nu}|(\boldsymbol{\lambda}\cdot\boldsymbol{n})_{n_\nu}|).$$

Notice that the divergence-free condition (3) can be obtained from (2) by setting $\mathcal{U}$ to the constant two. Similarly, the discrete divergence-free condition can be obtained from (4) by setting the discrete solution to a vector of twos. Doing this in the first equation of (4) produces the discrete divergence-free condition on the left element:

$$\left(\mathsf{D}_{xL}\Lambda_{xL} + \mathsf{D}_{yL}\Lambda_{yL}\right)\mathbf{1}_L = \mathsf{H}_L^{-1}\left(\mathsf{R}_L^{\mathrm{T}}\mathsf{B}_{xL}\mathsf{R}_L\Lambda_{xL} + \mathsf{R}_L^{\mathrm{T}}\mathsf{B}_{yL}\mathsf{R}_L\Lambda_{yL} - \mathsf{R}_L^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_L\right)_L \mathbf{1}_L. \tag{5}$$

The constant vectors $\mathbf{1}_L$ and $\mathbf{1}_R$ are composed of ones and have lengths equal to the number of nodes in their respective elements. The left-hand side of (5) is a direct discretization of the divergence-free condition using SBP operators; note that $\Lambda_{xL}\mathbf{1}_L$ and $\Lambda_{yL}\mathbf{1}_L$ are the x and y components of the advection velocity. The right-hand side is a penalty term that vanishes for all velocity fields of polynomial degree p or less; for such velocity fields R_L is exact, i.e. $(\mathsf{R}_L\boldsymbol{u})_i = \mathcal{U}(\xi_i),\ \forall \xi_i \in S_{\hat{\Gamma}_\nu}$, and we have

$$\left(\mathsf{R}_L^{\mathrm{T}}\mathsf{B}_{xL}\mathsf{R}_L\Lambda_{xL} + \mathsf{R}_L^{\mathrm{T}}\mathsf{B}_{yL}\mathsf{R}_L\Lambda_{yL} - \mathsf{R}_L^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_L\right)\mathbf{1}_L = \mathsf{R}_L^{\mathrm{T}}\begin{bmatrix} b_1(\boldsymbol{\lambda}\cdot\boldsymbol{n} - \boldsymbol{\lambda}\cdot\boldsymbol{n})_1 \\ \vdots \\ b_{n_\nu}(\boldsymbol{\lambda}\cdot\boldsymbol{n} - \boldsymbol{\lambda}\cdot\boldsymbol{n})_{n_\nu}\end{bmatrix}_L = \mathbf{0}.$$

Thus, the right-hand side term does not impact the asymptotic error; however, the right-hand side is necessary to obtain conservation in the overall scheme.

The following useful identities follow from (5) and its variant for the right domain by left multiplying by $\boldsymbol{v}_{L/R}^T\mathsf{H}_{L/R}$ and transposing the result; this operation has the effect of multiplying the divergence-free equation by an arbitrary function and integrating the result. The identities relate the discretized volume integral of divergence to the boundary integral of the flux.

$$\begin{aligned} \mathbf{1}_L^T\left(\Lambda_{xL}\mathsf{Q}_{xL} + \Lambda_{yL}\mathsf{Q}_{yL}\right)\boldsymbol{v}_L &= \mathbf{1}_L^T\left(\mathsf{R}_L^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_L\right)\boldsymbol{v}_L, \qquad \forall\, \boldsymbol{v}_L \in \mathbb{R}^n, \\ \mathbf{1}_R^T\left(\Lambda_{xR}\mathsf{Q}_{xR} + \Lambda_{yR}\mathsf{Q}_{yR}\right)\boldsymbol{v}_R &= -\mathbf{1}_R^T\left(\mathsf{R}_R^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_R\right)\boldsymbol{v}_R, \qquad \forall\, \boldsymbol{v}_R \in \mathbb{R}^n. \end{aligned} \tag{6}$$

The difference in sign between the two right-hand sides of (6) comes about because of the difference in orientation of the outward-pointing normals.

B. Conservation analysis

Theorem 2. *The discretization (4) is conservative in the sense that*

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{1}_L^{\mathrm{T}}\mathsf{H}_L\boldsymbol{u}_L + \mathbf{1}_R^{\mathrm{T}}\mathsf{H}_R\boldsymbol{u}_R\right) = 0.$$

Proof. Multiplying the equation for $\boldsymbol{u}_L$, i.e. the first equation of (4), by $\mathbf{1}^{\mathrm{T}}\mathsf{H}_L$ from the left, we find

$$\begin{aligned} &\mathbf{1}_L^{\mathrm{T}}\mathsf{H}_L\frac{\mathrm{d}\boldsymbol{u}_L}{\mathrm{d}t} + \mathbf{1}_L^{\mathrm{T}}\frac{1}{2}\left(-\mathsf{Q}_{xL}^{\mathrm{T}}\Lambda_{xL} - \mathsf{Q}_{yL}^{\mathrm{T}}\Lambda_{yL} + \mathsf{R}_L^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_L + \Lambda_{xL}\mathsf{Q}_{xL} + \Lambda_{yL}\mathsf{Q}_{yL}\right)\boldsymbol{u}_L \\ &\qquad\qquad = \frac{1}{2}\mathbf{1}_L^{\mathrm{T}}\mathsf{R}_L^{\mathrm{T}}(\mathsf{B}_\lambda - |\mathsf{B}_\lambda|)\left(\mathsf{R}_L\boldsymbol{u}_L - \mathsf{R}_R\boldsymbol{u}_R\right) \\ \Rightarrow \qquad &\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{1}_L^{\mathrm{T}}\mathsf{H}_L\boldsymbol{u}_L\right) + \mathbf{1}_L^{\mathrm{T}}\mathsf{R}_L^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_L\boldsymbol{u}_L = \frac{1}{2}\mathbf{1}_L^{\mathrm{T}}\mathsf{R}_L^{\mathrm{T}}(\mathsf{B}_\lambda - |\mathsf{B}_\lambda|)\left(\mathsf{R}_L\boldsymbol{u}_L - \mathsf{R}_R\boldsymbol{u}_R\right) \end{aligned}$$

where we have used identity (6) and the fact that SBP operators are exact for constants, i.e. $\mathbf{1}^{\mathrm{T}}\mathsf{Q}_x^{\mathrm{T}} = (\mathsf{D}_x\mathbf{1})^{\mathrm{T}}\mathsf{H} = \mathbf{0}^{\mathrm{T}}$. Adding the above expression to the analogous one for $\boldsymbol{u}_R$ we have (recall the difference in sign in identity (6))

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{1}_L^{\mathrm{T}}\mathsf{H}_L\boldsymbol{u}_L + \mathbf{1}_R^{\mathrm{T}}\mathsf{H}_R\boldsymbol{u}_R\right) + \mathbf{1}_L^{\mathrm{T}}\mathsf{R}_L^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_L\boldsymbol{u}_L - \mathbf{1}_R^{\mathrm{T}}\mathsf{R}_R^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_R\boldsymbol{u}_R \\ &\qquad = \frac{1}{2}\mathbf{1}_L^{\mathrm{T}}\mathsf{R}_L^{\mathrm{T}}(\mathsf{B}_\lambda - |\mathsf{B}_\lambda|)\left(\mathsf{R}_L\boldsymbol{u}_L - \mathsf{R}_R\boldsymbol{u}_R\right) + \frac{1}{2}\mathbf{1}_R^{\mathrm{T}}\mathsf{R}_R^{\mathrm{T}}(-\mathsf{B}_\lambda - |\mathsf{B}_\lambda|)\left(\mathsf{R}_R\boldsymbol{u}_R - \mathsf{R}_L\boldsymbol{u}_L\right) \end{aligned}$$

Finally, we take advantage of the accuracy of the interpolation/extrapolation operator, namely that $\mathbf{1}_L^{\mathrm{T}}\mathsf{R}_L^{\mathrm{T}} = \mathbf{1}_R^{\mathrm{T}}\mathsf{R}_R^{\mathrm{T}} = \mathbf{1}_\nu^{\mathrm{T}}$ where $\mathbf{1}_\nu \in \mathbb{R}^{n_\nu}$ is a vector of ones. Thus,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{1}_L^{\mathrm{T}}\mathsf{H}_L\boldsymbol{u}_L + \mathbf{1}_R^{\mathrm{T}}\mathsf{H}_R\boldsymbol{u}_R\right) + \mathbf{1}_\nu^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_L\boldsymbol{u}_L - \mathbf{1}_\nu^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_R\boldsymbol{u}_R$$
$$= \frac{1}{2}\mathbf{1}_\nu^{\mathrm{T}}(\mathsf{B}_\lambda - |\mathsf{B}_\lambda|)\left(\mathsf{R}_L\boldsymbol{u}_L - \mathsf{R}_R\boldsymbol{u}_R\right) + \frac{1}{2}\mathbf{1}_\nu^{\mathrm{T}}(-\mathsf{B}_\lambda - |\mathsf{B}_\lambda|)\left(\mathsf{R}_R\boldsymbol{u}_R - \mathsf{R}_L\boldsymbol{u}_L\right)$$

The result now follows after some algebra. □

C. Stability analysis

Theorem 3. *The discretization* (4) *satisfies*

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{u}_L^{\mathrm{T}}\mathsf{H}_L\boldsymbol{u}_L + \boldsymbol{u}_R^{\mathrm{T}}\mathsf{H}_R\boldsymbol{u}_R\right) \leq 0,$$

and is therefore time stable.

Proof. Multiplying the first equation in (4) by $\boldsymbol{u}_L^{\mathrm{T}}\mathsf{H}_L$ from the left, we find

$$\boldsymbol{u}_L^{\mathrm{T}}\mathsf{H}_L\frac{\mathrm{d}\boldsymbol{u}_L}{\mathrm{d}t} + \boldsymbol{u}_L^{\mathrm{T}}\frac{1}{2}\left(-\mathsf{Q}_{xL}^{\mathrm{T}}\Lambda_{xL} - \mathsf{Q}_{yL}^{\mathrm{T}}\Lambda_{yL} + \mathsf{R}_L^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_L + \Lambda_{xL}\mathsf{Q}_{xL} + \Lambda_{yL}\mathsf{Q}_{yL}\right)\boldsymbol{u}_L$$
$$= \frac{1}{2}\boldsymbol{u}_L^{\mathrm{T}}\mathsf{R}_L^{\mathrm{T}}(\mathsf{B}_\lambda - |\mathsf{B}_\lambda|)\left(\mathsf{R}_L\boldsymbol{u}_L - \mathsf{R}_R\boldsymbol{u}_R\right)$$

$$\Rightarrow \qquad \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{u}_L^{\mathrm{T}}\mathsf{H}_L\boldsymbol{u}_L\right) + \frac{1}{2}\boldsymbol{u}_L^{\mathrm{T}}\mathsf{R}_L^{\mathrm{T}}\mathsf{B}_\lambda\mathsf{R}_L\boldsymbol{u}_L = \frac{1}{2}\boldsymbol{u}_L^{\mathrm{T}}\mathsf{R}_L^{\mathrm{T}}(\mathsf{B}_\lambda - |\mathsf{B}_\lambda|)\left(\mathsf{R}_L\boldsymbol{u}_L - \mathsf{R}_R\boldsymbol{u}_R\right)$$

where we have used $\boldsymbol{u}_L^{\mathrm{T}}\Lambda_{xL}\mathsf{Q}_{xL}\boldsymbol{u}_L = \boldsymbol{u}_L^{\mathrm{T}}\mathsf{Q}_{xL}^{\mathrm{T}}\Lambda_{xL}\boldsymbol{u}_L$ and $\boldsymbol{u}_L^{\mathrm{T}}\Lambda_{yL}\mathsf{Q}_{yL}\boldsymbol{u}_L = \boldsymbol{u}_L^{\mathrm{T}}\mathsf{Q}_{yL}^{\mathrm{T}}\Lambda_{yL}\boldsymbol{u}_L$. Adding the analogous expression for the right domain to the above and simplifying, we obtain

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{u}_L^{\mathrm{T}}\mathsf{H}_L\boldsymbol{u}_L + \boldsymbol{u}_R^{\mathrm{T}}\mathsf{H}_R\boldsymbol{u}_R\right) = -\frac{1}{2}\boldsymbol{u}_L^{\mathrm{T}}\mathsf{R}_L^{\mathrm{T}}|\mathsf{B}_\lambda|\left(\mathsf{R}_L\boldsymbol{u}_L - \mathsf{R}_R\boldsymbol{u}_R\right) - \frac{1}{2}\boldsymbol{u}_R^{\mathrm{T}}\mathsf{R}_R^{\mathrm{T}}|\mathsf{B}_\lambda|\left(\mathsf{R}_R\boldsymbol{u}_R - \mathsf{R}_L\boldsymbol{u}_L\right)$$
$$= -\frac{1}{2}\begin{bmatrix}\boldsymbol{u}_L^{\mathrm{T}}\mathsf{R}_L^{\mathrm{T}} & \boldsymbol{u}_R^{\mathrm{T}}\mathsf{R}_R^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}|\mathsf{B}_\lambda| & -|\mathsf{B}_\lambda| \\ -|\mathsf{B}_\lambda| & |\mathsf{B}_\lambda|\end{bmatrix}\begin{bmatrix}\mathsf{R}_L\boldsymbol{u}_L \\ \mathsf{R}_R\boldsymbol{u}_R\end{bmatrix}.$$

The quadratic form on the right-hand side is negative semi-definite since the matrix composed of $|\mathsf{B}_\lambda|$ blocks is positive semi-definite; its eigenvalues are $\{2b_i|(\boldsymbol{\lambda}\cdot\hat{n})_i|\}_{i=1}^{n_\nu}$ together with 0, which has multiplicity n_ν. It follows that the discretization is time stable. □

IV. Example operators on the triangle

In this section, we describe the construction, for triangular elements, of multi-dimensional SBP operators in conjunction with the matrices R and B that define the SATs.

As shown in [11], a multi-dimensional SBP operator of degree p exists for any domain that has 1) a cubature rule of degree at least $2p-1$ with 2) strictly positive weights and 3) a full-rank generalized Vandermonde matrix. In general, such cubature rules are not unique, so there exist many different SBP operators for a given domain. To illustrate this fact, we present two families of SBP operators for the triangle.

The first family of SBP operators on the triangle was presented previously in [11]. This family consists of operators with $p+1$ nodes on each face and will be referred to as the SBP-Γ family. Figure 2 shows the $p=1$ through $p=4$ operators from this family. The second family of triangular-element SBP operators has strictly interior nodes. This family will be referred to as the SBP-Ω family, and the first four operators in this family[b] are shown in Figure 3.

The algorithmic steps involved in constructing the operators are listed below. The process is similar to that outlined in [11] for SBP-Γ, with a few minor changes that are highlighted.

[b] We did not consider the $p=0$ operator in this work

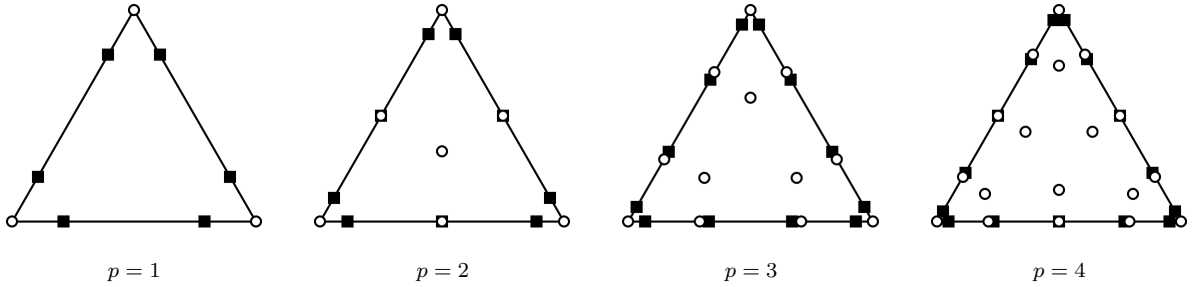


Figure 2. Nodes of the SBP-Γ family of operators that include $p + 1$ nodes on each face. The open circles denote the SBP operator nodes, while the black squares denote the face cubature points used for the SATs.

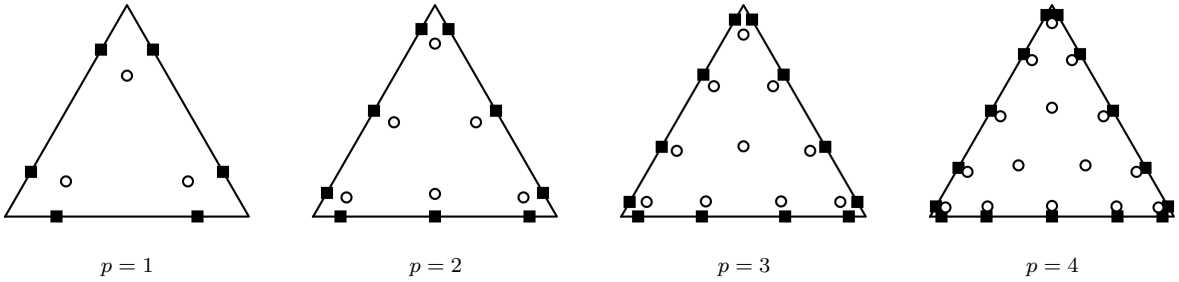


Figure 3. Nodes of the SBP-Ω family of operators whose nodes are strictly interior to Ω. The open circles denote the SBP operator nodes, while the black squares denote the face cubature points used for the SATs.

1. For a given design accuracy p, a symmetric cubature rule is selected or constructed that is exact for polynomials of total degree $2p-1$ and has at least $(p+1)(p+2)/2$ nodes. The nodes for the SBP-Ω family are required to be strictly interior, and the SBP-Γ family is required to have $p+1$ nodes on each face, including the vertices. For all SBP-Ω operators considered here ($p = 1, \ldots, 4$), there are exactly $(p+1)(p+2)/2$ cubature nodes, whereas the SBP-Γ operators generally have more nodes for the same value of p.

2. A Legendre-Gauss quadrature rule with $p + 1$ nodes is used to define B on all faces, i.e. the same quadrature rule is used for all three sides, although this is not strictly necessary.

3. Let $\hat{\Gamma}_\nu$ denote one of the faces of the triangle. Then the volume-to-face interpolation/extrapolation operator for this face is defined by $\mathsf{R} = \mathsf{V}_{\hat{\Gamma}_\nu}(\mathsf{V}_{\hat{\Omega}})^\dagger$, where $\mathsf{V}_{\hat{\Gamma}_\nu}$ denotes an orthogonal polynomial basis evaluated at the quadrature nodes of $\hat{\Gamma}_\nu$, and the superscript † denotes the Moore-Penrose pseudoinverse. The definition of $\mathsf{V}_{\hat{\Omega}}$ depends on whether we are constructing the SBP-Γ or SBP-Ω family. For the latter, $\mathsf{V}_{\hat{\Omega}}$ is an orthogonal polynomial basis evaluated at all of the nodes in the volume cubature. In contrast, for the SBP-Γ family, $\mathsf{V}_{\hat{\Omega}}$ is the basis evaluated at the $p + 1$ volume cubature nodes that lie on the face of $\hat{\Gamma}_\nu$.

   Although we have considered only the face $\hat{\Gamma}_\nu$, symmetry allows the same R matrix to be used on all three faces simply by permuting indices of the volume nodes.

4. The boundary operator E_x is constructed from the face cubature B and interpolation/extrapolation operator R using equation (1). An analogous equation is used for E_y.

5. The skew-symmetric operators S_x and S_y are determined using the accuracy conditions, Property I of Definition 1. For the SBP-Ω operators considered here, the S_x and S_y operators are fully determined by the accuracy conditions; in contrast, the SBP-Γ operators are underdetermined by the accuracy conditions, so the minimum-norm solution is used for those operators.

Table 1. Summary of cubature accuracy, node counts, and operator dimensions for the two different families of SBP operators on the triangle.

family	degree (p)	# nodes (n)	H degree	R matrix size
SBP-Γ	1	3	1	2×2
SBP-Ω	1	3	2	2×3
SBP-Γ	2	7	3	3×3
SBP-Ω	2	6	4	3×6
SBP-Γ	3	12	5	4×4
SBP-Ω	3	10	5	4×10
SBP-Γ	4	18	7	5×5
SBP-Ω	4	15	7	5×15

Table 1 summarizes the accuracy and node-set properties of both the SBP-Ω and SBP-Γ families. Beyond the fact that SBP-Γ includes boundary nodes and SBP-Ω excludes boundary nodes, a few other differences between the families are worth highlighting. First, the SBP-Γ family generally requires more nodes than the SBP-Ω family for the same design accuracy p; this translates into D_x and D_y operators that require more storage and computation. Second, the cubature accuracy is higher for the SBP-Ω family; the $p = 1$ and $p = 2$ operators have cubatures that are more accurate than $2p - 1$, and the $p = 3$ and $p = 4$ operators appear to have smaller error constants. Finally, the volume-to-face interpolation/extrapolation operators used by the SBP-Γ operators are smaller, giving them a computational advantage when it comes to evaluating the SATs.

V. Results

A. Steady, constant-coefficient advection

The steady, constant-coefficient advection equation was chosen to verify the accuracy of the SATs and the newly derived SBP-Ω operators. The SBP-Γ operators were previously verified in [11], but we repeat their verification here, because the SAT implementation has changed. We also use this test case to investigate the convergence rate of boundary functionals.

The specific PDE and boundary conditions are given by

$$\frac{\partial \mathcal{U}}{\partial x} + \frac{\partial \mathcal{U}}{\partial y} = \mathcal{S}, \qquad \forall\ (x, y) \in \Omega = [0, 1]^2,$$
$$\mathcal{U}(x, 0) = 1, \qquad \text{and} \qquad \mathcal{U}(0, y) = 1,$$

where $\mathcal{S}(x, y) = ye^{xy} + xe^{xy}$. The exact solution to this PDE is $\mathcal{U}(x, y) = e^{xy}$.

To verify order of accuracy, we conduct a mesh refinement study using uniform grid refinement. Let N denote the number of element edges along the x and y coordinates. The vertices of the elements are located at $(x_i, y_j) = (ih, jh),\ \forall i, j = 0, 1, \ldots, N$, where $h = 1/N$. For each of the N^2 quadrilaterals, two triangles are generated from the vertices $\{(x_i, y_j), (x_{i+1}, y_j), (x_i, y_{j+1})\}$ and $\{(x_{i+1}, y_{j+1}), (x_i, y_{j+1}), (x_{i+1}, y_j)\}$. For the solution- and functional-accuracy studies below we use $N \in \{2, 4, 6, 8, 10, 12\}$. We also use the $N = 12$ grid in the subsequent conservation and stability studies.

1. *Solution Accuracy*

To assess the accuracy of the discrete solutions, we evaluate the SBP-based L^2 norm of the difference between the numerical and exact solutions. We then normalize by the exact solution's norm; that is,

$$\text{Normalized } L^2 \text{ Error} = \frac{\sqrt{(\boldsymbol{u} - \boldsymbol{u}_\mathrm{e})^\mathrm{T}\, \mathsf{H}\, (\boldsymbol{u} - \boldsymbol{u}_\mathrm{e})}}{\sqrt{\boldsymbol{u}_\mathrm{e}^\mathrm{T} \mathsf{H} \boldsymbol{u}_\mathrm{e}}},$$

where $\boldsymbol{u}$ is the discrete solution and $\boldsymbol{u}_\mathrm{e}$ the exact solution at the element nodes. Figure 4 and 5 shows this error for the SBP-Γ the SBP-Ω families, respectively.

From the perspective of accuracy alone, we see a significant difference between the two $p = 1$ schemes. On the finest grid, the second-order SBP-Ω scheme is more than 7.8 times more accurate than the second-order SBP-Γ scheme. This difference in accuracy is also present in the higher order schemes, but it is less pronounced. For example, the third-, fourth-, and fifth-order SBP-Ω schemes are 4.1, 1.4, and 2.0 times more accurate, respectively, than their corresponding SBP-Γ schemes on the finest grid. This difference in accuracy is also observed in spectral-element schemes based on Legendre-Gauss nodes (interior-only nodes) versus Legendre-Gauss-Lobbato nodes (boundary nodes included).

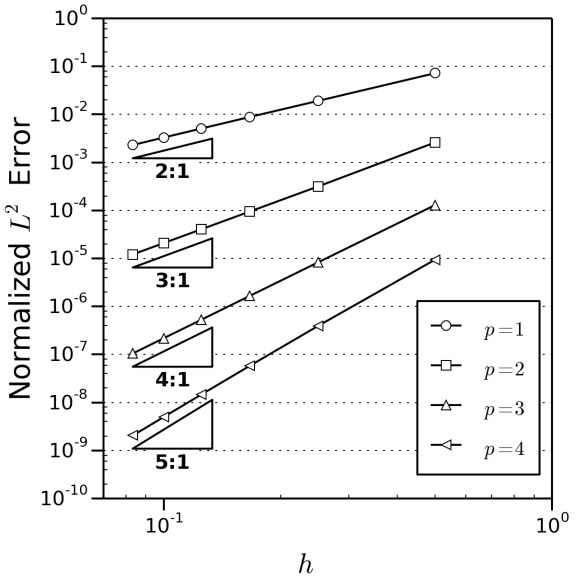


Figure 4. L^2 **error in the steady-advection solution for different mesh spacing using the SBP-**Γ **family of operators.**

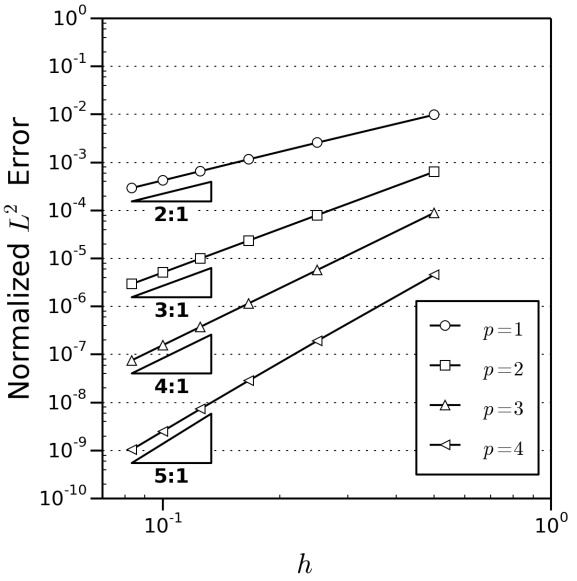


Figure 5. L^2 **error in the steady-advection solution for different mesh spacing using the SBP-**Ω **family of operators.**

2. *Functional Accuracy*

Classical and generalized SBP-SAT finite-difference methods have been shown to produce superconvergent functionals if the discretizations are dual consistent [19–21]. In this section, we verify that superconvergent functionals are also produced by multi-dimensional SBP operators using the proposed SATs.

We use the total flux over the outflow boundary as the functional of interest. For the problem at hand, the analytical value of the total outflow flux is given by

$$\mathcal{J}(\mathcal{U}) = \int_{\Gamma^+} \mathcal{U}(x,y)\ (\boldsymbol{\lambda}\cdot\boldsymbol{n})\,\mathrm{d}\Gamma = \int_{x=0}^{1} \mathcal{U}(x,1)\,\mathrm{d}\Gamma + \int_{y=0}^{1} \mathcal{U}(1,y)\,\mathrm{d}\Gamma = 2(e-1),$$

where $\Gamma^+ = \{(x,y) \in \Gamma \mid n_x + n_y > 0\}$. To ensure dual consistency, the discretized functional is evaluated using the volume-to-face interpolation/extrapolation operators, R, and face cubature points. Thus, the discrete function is defined by

$$J_h(\boldsymbol{u}) = \sum_{\nu\in\Gamma_h^+} \mathbf{1}^T \mathsf{B}_\nu \mathsf{R}_\nu \boldsymbol{u}_\nu,$$

where Γ_h^+ is the set of element edges lying on the outflow, and $\boldsymbol{u}_\nu$ is the discrete solution at the nodes of the element associated with boundary face ν.

Figures 6 and 7 plot the relative functional error, $|\mathcal{J} - J_h|/\mathcal{J}$, for the SBP-$\Gamma$ and SBP-Ω families, respectively. All schemes exhibit superconvergent functional estimates with rates of at least $2p$. Note that some of the higher-order schemes experience round-off errors as the mesh is refined, limiting their error to $\mathrm{O}(10^{-14})$.

The difference between the two families is even more striking than it was for solution error. For example, on the finest grid, the SBP-Ω $p = 1$ and $p = 2$ schemes are almost four orders more accurate than the corresponding SBP-Γ schemes. The difference in accuracy for the higher order schemes is not as dramatic,

but it remains significant. We are currently investigating the superior functional accuracy of the SBP-Ω $p = 1$ and $p = 2$ schemes, which have convergence rates of $2p+2$, but the likely source is the $2p$ versus $2p-1$ exactness of the SBP norm H; see Table 1.

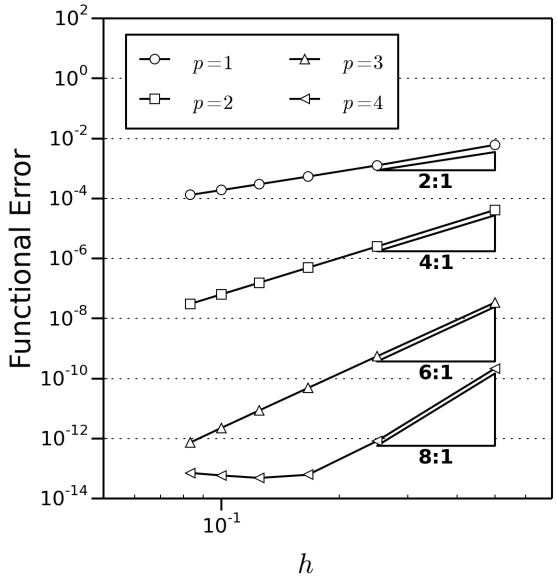


Figure 6. Normalized functional error for different mesh spacing using the SBP-Γ family of operators.

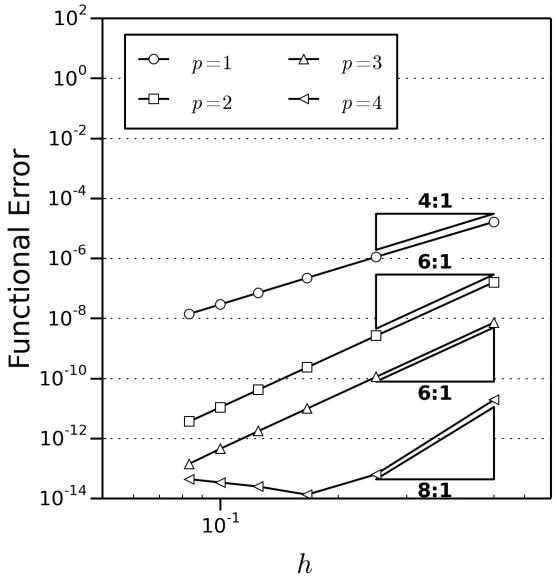


Figure 7. Normalized functional error for different mesh spacing using the SBP-Ω family of operators.

B. Linear advection with a divergence-free velocity

In this section, we verify the theory developed in Section III by investigating the SBP-SAT discretization of a scalar advection PDE with a divergence-free velocity. This experiment verifies the conservation and stability properties of SBP-SAT discretizations in the context of a spatially varying velocity field.

The continuous PDE is (2) with the advection velocity satisfying the divergence-free condition (3). The domain is the unit square, $\Omega = [0, 1]^2$, and the velocity field is

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_x \\ \lambda_y \end{bmatrix} = \begin{bmatrix} \pi \sin(\pi x) \cos(\pi y) \\ -\pi \cos(\pi x) \sin(\pi y) \end{bmatrix}. \tag{7}$$

In addition to being divergence-free, this velocity field produces zero flux over the boundary; therefore, no boundary condition is imposed on the solution. For the initial condition we take $\mathcal{U}(x, y, 0) = e^{xy}$.

1. *Discretization*

In general, the continuous velocity (7) does not satisfy the discretized divergence-free condition (5). Therefore, we seek a discrete vector field that satisfies the discrete divergence-free condition and is as close as possible, in an appropriate norm, to the continuous field. To find such a $\boldsymbol{\lambda}$, we solve a pair of convex quadratic optimization problems, which we now describe.

First we solve for the face-normal velocities $\lambda_n = \boldsymbol{\lambda} \cdot \boldsymbol{n}$, i.e. the elements in the B_λ matrices. A constraint on the λ_n for each element is obtained by substituting $\boldsymbol{v}_L = \mathbf{1}$ in the first equation of (6) (or $\boldsymbol{v}_R = \mathbf{1}$ in the second equation):

$$\sum_\nu \mathbf{1}^{\mathrm{T}} \mathsf{R}_L^{\mathrm{T}} \mathsf{B}_\lambda \mathsf{R}_L \mathbf{1} = \sum_\nu \sum_{i=1}^{n_\nu} b_1 (\boldsymbol{\lambda} \cdot \boldsymbol{n})_i = 0,$$

where ν indexes the faces of the element. This constraint is simply a discretization of $\int_\Gamma \boldsymbol{\lambda} \cdot \boldsymbol{n} \, \mathrm{d}\hat{\Gamma} = 0$ on each element. There are fewer elements than face-normal velocities, so we solve a quadratic optimization problem that minimizes the Cartesian norm between the discrete and continuous values of λ_n, subject to the above

Table 2. Conservation metric at time $t_f = 10$ for the SBP-Γ and SBP-Ω discretizations applied to the divergence-free advection problem with 288 elements.

	degree			
family	**p = 1**	**p = 2**	**p = 3**	**p = 4**
SBP-Γ	2.22×10^{-16}	2.00×10^{-15}	3.55×10^{-15}	2.00×10^{-15}
SBP-Ω	3.33×10^{-15}	4.44×10^{-16}	7.99×10^{-15}	1.11×10^{-14}

identity on the B_λ matrices. In practice, we include scaling due to the coordinate transformations in the λ_n values.

Once the B_λ matrices are determined, we solve for the $\boldsymbol{\lambda}_x$ and $\boldsymbol{\lambda}_y$. We follow a procedure analogous to the one used for B_λ, except that (5) becomes the constraint, and the optimization problems on each element are decoupled.

With the discrete velocity field defined, the semi-discretization given by (4) is fully defined for a given mesh. We adopt the mesh with $N = 12$ that was used in the constant-coefficient advection case; see Section V.A for the details. Finally, the time derivative is discretized using the standard fourth-order Runge-Kutta method. For the following results, the time-step was kept at 50% its maximally stable value for linear advection.

2. *Conservation and Stability*

For the given problem, the integral of the analytical solution of (2) is constant in time, because the PDE is conservative and there is no boundary flux. Based on the analysis in Section III, the SBP-SAT discretization should mimic this property, and the conservation error should be zero to machine precision.

Table 2 lists the conservation error for the SBP-Γ and SBP-Ω discretizations using $N = 12$ faces along each edge of the domain (288 triangular elements). The conservation metric is defined by

$$\text{Conservation Metric} \equiv |\mathbf{1}^T \mathsf{H} \boldsymbol{u}_0 - \mathbf{1}^T \mathsf{H} \boldsymbol{u}_f|,$$

where u_f is the discrete solution at $t = 10$ and u_0 is the initial condition evaluated at the nodes. The results in Table 2 provide strong evidence that the SBP-SAT discretizations are conservative.

To demonstrate that the SBP-SAT discretizations are energy stable, Figure 8 shows the change in normalized energy as the discrete solutions evolve from $t = 0$ to $t = 10$. The normalized change in energy is given by

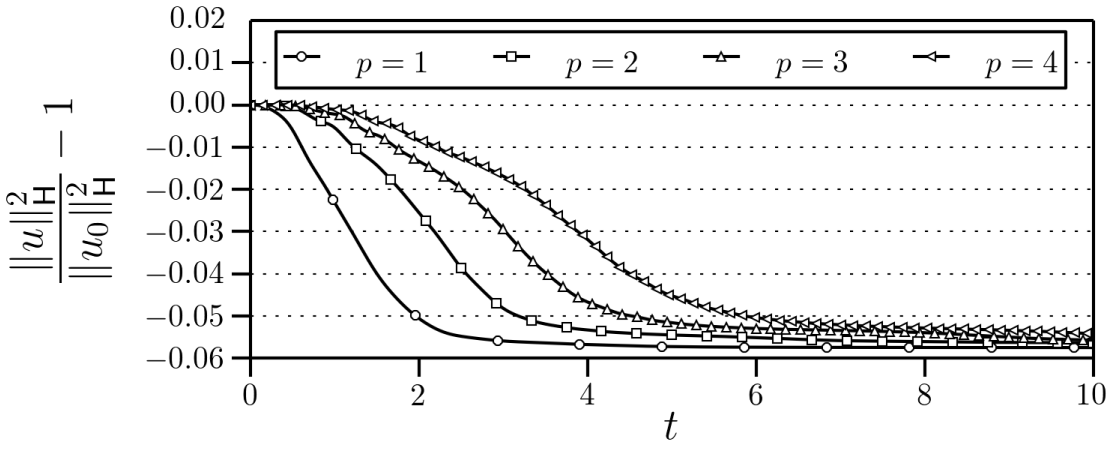
$$\frac{\boldsymbol{u}^T \mathsf{H} \boldsymbol{u} - \boldsymbol{u}_0^T \mathsf{H} \boldsymbol{u}_0}{\boldsymbol{u}_0^T \mathsf{H} \boldsymbol{u}_0} = \frac{\|\boldsymbol{u}\|_{\mathsf{H}}^2}{\|\boldsymbol{u}_0\|_{\mathsf{H}}^2} - 1$$

where $\boldsymbol{u}$ denotes the discrete solution. As with the conservation metric, we consider a uniform triangulation with $N = 12$ edges in each direction and 288 elements total. The plots show that the SBP-SAT discretizations have nonincreasing energies, as expected from the analysis.

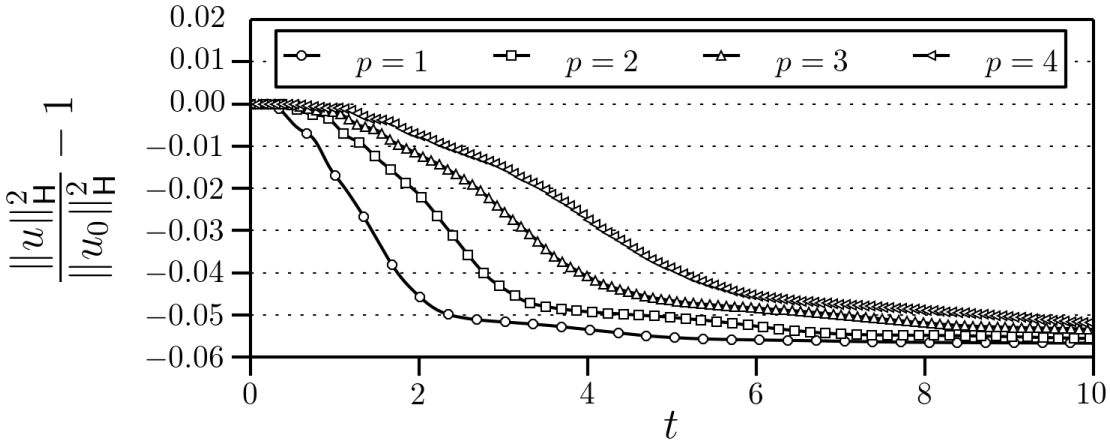
VI. Conclusions

We have shown how simultaneous approximation terms (SATs) can be applied to discretizations based on multi-dimensional summation-by-parts (SBP) operators. Our key observation is that the symmetric part of SBP operators, which corresponds to integration over the element boundary, can be written in terms of a face cubature and an interpolation/extrapolation operator. Using this decomposition, it is straightforward to construct conservative and stable SATs that couple arbitrary SBP elements.

The SAT methodology was illustrated using SBP operators on triangular elements. We considered two SBP families; the SBP-Γ family with $p + 1$ nodes on each face and the SBP-Ω family with strictly interior nodes. The accuracy of the SBP-SAT discretizations was verified using solution and functional errors for a constant-coefficient advection equation. Both families were shown to produce design-accuracy solutions and superconvergent functionals, but the SBP-Ω family was found to outperform the SBP-Γ family in terms of accuracy. Conservation and stability of the SBP-SAT discretizations were verified using linear advection with a nontrivial, divergence-free advection velocity.



(a) SBP-Γ family



(b) SBP-Ω family

Figure 8. Normalized change in energy versus time. Every 100th time-step is marked with a symbol.

Acknowledgments

The first author would like to acknowledge the support of the Scientific Computation Research Center and Rensselaer Polytechnic Institute.

References

[1]Kreiss, H.-O. and Scherer, G., "Finite element and finite difference methods for hyperbolic partial differential equations," *Mathematical aspects of finite elements in partial differential equations*, Academic Press, New York/London, 1974, pp. 195–212.

[2]Strand, B., "Summation by parts for finite difference approximations for d/dx," *Journal of Computational Physics*, Vol. 110, No. 1, 1994, pp. 47–67.

[3]Nordström, J. and Carpenter, M. H., "Boundary and interface conditions for high-order finite-difference methods applied to the Euler and Navier-Stokes equations," *Journal of Computational Physics*, Vol. 148, No. 2, 1999, pp. 621–645.

[4]Mattsson, K., Svärd, M., and Nordström, J., "Stable and accurate artificial dissipation," *Journal of Scientific Computing*, Vol. 21, No. 1, 2004, pp. 57–79.

[5]Mattsson, K., Svärd, M., and Shoeybi, M., "Stable and accurate schemes for the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 227, No. 4, 2008, pp. 2293–2316.

[6]Svärd, M. and Nordström, J., "A stable high-order finite difference scheme for the compressible NavierStokes equations," *Journal of Computational Physics*, Vol. 227, No. 10, May 2008, pp. 4805–4824.

[7]Del Rey Fernández, D. C. and Zingg, D. W., "Generalized summation-by-parts operators for the second derivative with a variable coefficient," *SIAM Journal on Scientific Computing*, Vol. 37, No. 6, 2015, pp. A2840–A2864.

[8]Svärd, M. and Nordström, J., "Review of summation-by-parts schemes for initial-boundary-value-problems," *Journal of Computational Physics*, Vol. 268, No. 1, 2014, pp. 17–38.

[9]Del Rey Fernández, D. C., Hicken, J. E., and Zingg, D. W., "Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations," *Computers & Fluids*, Vol. 95, No. 22, 2014, pp. 171–196.

[10]Del Rey Fernández, D. C., Boom, P. D., and Zingg, D. W., "A generalized framework for nodal first derivative summation-by-parts operators," *Journal of Computational Physics*, Vol. 266, No. 1, 2014, pp. 214–239.

[11]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. (in press), 2016, (see also http://arxiv.org/abs/1505.03125).

[12]Hicken, J. E., Del Rey Fernández, D. C., and Zingg, D. W., "Opportunities for efficient high-order methods based on the summation-by-parts property," *22nd AIAA Computational Fluid Dynamics Conference*, No. AIAA–2015–3198, Dallas, Texas, June 2015.

[13]Cools, R., "Monomial cubature rules since "Stroud": a compilation — part 2," *Journal of Computational and Applied Mathematics*, Vol. 112, No. 12, 1999, pp. 21 – 27.

[14]Funaro, D. and Gottlieb, D., "A new method of imposing boundary conditions in pseudospectral approximations of hyperbolic equations," *Mathematics of Computation*, Vol. 51, No. 184, Oct. 1988, pp. 599–613.

[15]Carpenter, M. H., Gottlieb, D., and Abarbanel, S., "Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes," *Journal of Computational Physics*, Vol. 111, No. 2, 1994, pp. 220–236.

[16]Hesthaven, J. S. and Warburton, T., *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, Springer-Verlag, New York, 2008.

[17]Svärd, M., "On coordinate transformations for summation-by-parts operators," *Journal of Scientific Computing*, Vol. 20, No. 1, Feb. 2004, pp. 29–42.

[18]Del Rey Fernández, D. C., Hicken, J. E., and Zingg, D. W., "A framework for simultaneous approximation terms for multi-dimensional summation-by-parts operators," May 2016, arXiv:submit/1557005.

[19]Hicken, J. E. and Zingg, D. W., "Superconvergent functional estimates from summation-by-parts finite-difference discretizations," *SIAM Journal on Scientific Computing*, Vol. 33, No. 2, 2011, pp. 893–922.

[20]Hicken, J. E. and Zingg, D. W., "Dual consistency and functional accuracy: a finite-difference perspective," *Journal of Computational Physics*, Vol. 256, Jan. 2014, pp. 161–182.

[21]Boom, P. D. and Zingg, D. W., "High-order implicit time-marching methods based on generalized summation-by-parts operators," *SIAM Journal on Scientific Computing*, Vol. 37, No. 6, 2015, pp. A2682–A2709.