Computers & Fluids 95 (2014) 171–196

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations

David C. Del Rey Fernández [a,*], Jason E. Hicken [b], David W. Zingg [a,1]

[a] Institute for Aerospace Studies, University of Toronto, Toronto, Ontario M3H 5T6, Canada
[b] Department of Mechanical, Aerospace, and Nuclear Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

ARTICLE INFO

Article history:
Received 11 February 2014
Accepted 20 February 2014
Available online 12 March 2014

Keywords:
Summation-by-parts operators
Finite-difference methods
Simultaneous approximation terms
Boundary closures
Numerical boundary schemes
Higher-order methods
Stability
Energy-method

ABSTRACT

Summation-by-parts (SBP) operators have a number of properties that make them an attractive option for higher-order spatial discretizations of partial differential equations. In particular, they enable the derivation of higher-order boundary closures leading to provable time stability. When implemented on multi-block structured meshes in conjunction with simultaneous approximation terms (SATs)—penalty terms that impose boundary and interblock-coupling conditions in a weak sense—they offer additional properties of value, even for second-order accurate schemes and steady problems. For example, they involve low communication overhead for efficient parallel algorithms and relax the continuity requirements of both the mesh and the solution across block interfaces. This paper provides a brief history of seminal contributions to, and applications of, SBP-SAT methods followed by a description of their properties and a methodology for deriving SBP operators for first derivatives and second derivatives with variable coefficients. A procedure for deriving SATs is also provided. Practical aspects are discussed, including artificial dissipation, transformation to curvilinear coordinates, and application to the Navier–Stokes equations. Recent developments are reviewed, including a variational interpretation, the connection to quadrature rules, functional superconvergence, error estimates, and dual consistency. Finally, the connection to quadrature rules is exploited to provide a generalization of the SBP concept to a broader class of operators, enabling a unification and rigorous development of SATs for operators such as nodal-based pseudo-spectral and some discontinuous Galerkin operators.

1. Introduction

In the numerical solution of partial differential equations (PDEs), the potential improvements in efficiency from higher-order methods have long been recognized [1,2]. Their advantage increases as the error tolerance of a computation becomes more stringent. In computational fluid dynamics, higher-order methods are often applied to time-dependent problems requiring high resolution. Despite evidence that they can also be more efficient than second-order methods for steady turbulent flows represented by the Reynolds-averaged Navier–Stokes equations [3], their use in this context is less common. This arises in part because the numerical accuracy requirement in such problems is often less stringent due to the fact that there can be a significant physical-model error associated with, for example, the turbulence model. A second issue impacting the effectiveness of higher-order methods when applied to practical problems is that such problems are often characterized by singularities and discontinuities of various types, such as shock waves. This can impair the ability of the higher-order method to achieve its design order, as the theory typically assumes a sufficiently smooth solution. Further study is needed to determine how this can be addressed. Nevertheless, there is growing interest in solving practical aerodynamic problems to a high degree of accuracy, and higher-order methods are being increasingly applied to such problems [4,5].

As a result of several advantageous properties, summation-by-parts (SBP) finite-difference operators with simultaneous approximation terms (SATs) for enforcing boundary and mesh interface conditions have emerged as one of several viable higher-order spatial discretizations for PDEs, for example the Navier–Stokes (NS) equations governing the flow of a continuum fluid. The use of finite-difference methods typically involves the use of multi-block structured meshes as opposed to fully unstructured

[*] Corresponding author. Tel.: +1 4166677709; fax: +1 416 667 7799.
E-mail address: dcdelrey@gmail.com (D.C. Del Rey Fernández).

[1] Tier 1 Canada Research Chair in Computational Aerodynamics, J. Armand Bombardier Foundation Chair in Aerospace Flight.

http://dx.doi.org/10.1016/j.compfluid.2014.02.016
0045-7930/© 2014 Elsevier Ltd. All rights reserved.

meshes, which can be easier to generate about complex geometries. Nevertheless, for geometries where a multi-block structured mesh can be readily generated, including many complex geometries of practical interest [6], finite-difference discretizations can be the most efficient option [7]. In particular, higher-order finite-difference methods can be very efficiently implemented on structured meshes [8,3]. Moreover, the SBP-SAT approach on multi-block meshes has proven to be advantageous even in a second-order implementation. For example, Hicken and Zingg [9] and Osusky and Zingg [6] have developed efficient flow solution methodologies combining the SBP-SAT approach with a Newton–Krylov–Schur parallel implicit algorithm for the Euler and Reynolds-averaged Navier–Stokes equations, respectively, that have been applied to various complex geometries, including full aircraft configurations [10].

We begin this review with an introduction to the basic concepts underpinning SBP-SAT schemes in the context of the linear convection equation with a unit wave speed in one dimension discretized on a uniform mesh. This is intended for the reader who is new to SBP-SAT schemes. The reader who is already familiar with the basic concepts can skip past (26).

With a positive unit wave speed, the linear advection equation is given by

$$\frac{\partial \mathcal{U}}{\partial t}+\frac{\partial \mathcal{U}}{\partial x}=0 \tag{1}$$

on the domain $x_L \leqslant x \leqslant x_R$ with the boundary condition $\mathcal{U}(0,t)=\mathcal{U}_L(t)$.

Time stability of the partial differential equation with $\mathcal{U}_L=0$ is readily shown. The time rate of change of the energy in the domain is

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_L}^{x_R}\mathcal{U}^2\mathrm{d}x &= \int_{x_L}^{x_R}\frac{\partial \mathcal{U}^2}{\partial t}\mathrm{d}x\\ &= \int_{x_L}^{x_R}2\mathcal{U}\frac{\partial \mathcal{U}}{\partial t}\mathrm{d}x\\ &= -2\int_{x_L}^{x_R}\mathcal{U}\frac{\partial \mathcal{U}}{\partial x}\mathrm{d}x.\end{aligned} \tag{2}$$

Applying integration by parts we find

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_L}^{x_R}\mathcal{U}^2\mathrm{d}x=-(\mathcal{U}_R^2-\mathcal{U}_L^2), \tag{3}$$

which is nonpositive when $\mathcal{U}_L=0$.

Next consider a discretization in space on a mesh with $N+1$ equally spaced nodes indexed from 0 to N such that $\mathbf{u}=[u_0,u_1,\ldots,u_N]^T$. We define an SBP finite-difference operator for a first derivative D_1 as

$$\mathsf{H}\mathsf{D}_1\mathbf{u}=\mathsf{Q}\mathbf{u},\quad \text{i.e.}\ \frac{\partial \mathcal{U}}{\partial x}\approx \mathsf{H}^{-1}\mathsf{Q}\mathbf{u}, \tag{4}$$

where H is a diagonal positive definite matrix that defines an inner product, norm, and quadrature by

$$\begin{aligned}&(\mathbf{u},\mathbf{v})_{\mathsf{H}}=\mathbf{u}^T\mathsf{H}\mathbf{v},\quad \|\mathbf{u}\|_{\mathsf{H}}^2=\mathbf{u}^T\mathsf{H}\mathbf{u},\\ &\int_{x_L}^{x_R}\mathcal{U}\mathcal{V}\mathrm{d}x\approx \mathbf{u}^T\mathsf{H}\mathbf{v},\end{aligned} \tag{5}$$

and

$$\begin{aligned}\mathsf{Q}+\mathsf{Q}^T &= \mathsf{E}_N-\mathsf{E}_0=\mathrm{diag}[0,\ldots,0,1]-\mathrm{diag}[1,0,\ldots,0]\\ &= \begin{bmatrix}-1 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & 1\end{bmatrix}.\end{aligned} \tag{6}$$

We will see later that the restriction to a diagonal H is not necessary, but it simplifies our initial introduction to SBP schemes.

With these definitions the discrete SBP operator mimics the integration by parts result obtained in the continuous case:

$$(\mathbf{u},\mathsf{H}^{-1}\mathsf{Q}\mathbf{v})_{\mathsf{H}}=\mathbf{u}^T(\mathsf{E}_N-\mathsf{E}_0)\mathbf{v}-(\mathsf{H}^{-1}\mathsf{Q}\mathbf{u},\mathbf{v})_{\mathsf{H}}. \tag{7}$$

This enables the following energy estimate (ignoring the boundary condition for now):

$$\begin{aligned}\frac{\mathrm{d}\mathbf{u}^T\mathsf{H}\mathbf{u}}{\mathrm{d}t} &= \mathbf{u}^T\mathsf{H}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}+\frac{\mathrm{d}\mathbf{u}^T}{\mathrm{d}t}\mathsf{H}\mathbf{u}=\mathbf{u}^T\mathsf{H}(-\mathsf{H}^{-1}\mathsf{Q})\mathbf{u}+\mathbf{u}^T(-\mathsf{Q}^T\mathsf{H}^{-1})\mathsf{H}\mathbf{u}\\ &= -\mathbf{u}^T\mathsf{Q}\mathbf{u}-\mathbf{u}^T\mathsf{Q}^T\mathbf{u}=-\mathbf{u}^T(\mathsf{Q}+\mathsf{Q}^T)\mathbf{u}=-\mathbf{u}^T(\mathsf{E}_N-\mathsf{E}_0)\mathbf{u}\\ &= -(u_N^2-u_0^2),\end{aligned}$$

which mimics the continuous case (3).

Before moving on to SATs, we consider an example of an SBP operator. The parameter p defines the scheme's order of accuracy. For a first derivative, with diagonal H, the scheme has interior order $2p$, boundary order p, and global order $p+1$. For example, with $p=2$ we have interior order 4 and global order 3. With $p=2$ the matrices H and Q have the following form:

$$\mathsf{H}=\Delta x\begin{bmatrix}h_{11} & & & & & \\ & h_{22} & & & & \\ & & h_{33} & & & \\ & & & h_{44} & & \\ & & & & 1 & \\ & & & & & \ddots\end{bmatrix}, \tag{8}$$

$$\mathsf{Q}=\begin{bmatrix}-\frac{1}{2} & \theta_{12} & \theta_{13} & \theta_{14} & & & & \\ -\theta_{12} & 0 & \theta_{23} & \theta_{24} & & & & \\ -\theta_{13} & -\theta_{23} & 0 & \theta_{34} & -\frac{1}{12} & & & \\ -\theta_{14} & -\theta_{24} & -\theta_{34} & 0 & \frac{8}{12} & -\frac{1}{12} & & \\ 0 & 0 & \frac{1}{12} & -\frac{8}{12} & 0 & \frac{8}{12} & -\frac{1}{12} & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots\end{bmatrix}, \tag{9}$$

where both matrices have corresponding entries in the lower right-hand corner. In the interior of the domain, the operator is the standard fourth-order centered difference operator. The various unspecified entries in H and Q must be determined to satisfy the order of accuracy requirements; this is further discussed in Section 4. The SBP property is obtained by construction (the entries in H must be positive).

SATs are penalty terms that impose boundary conditions in a weak sense. For our simple example, we have

$$\mathsf{H}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}=-\mathsf{Q}\mathbf{u}-\sigma(u_0-\mathcal{U}_L)\mathbf{e}_0, \tag{10}$$

where σ is a parameter, and $\mathbf{e}_0=[1,0,\ldots,0]^T$. With the SAT term included we get the following energy estimate (with $\mathcal{U}_L=0$):

$$\begin{aligned}\frac{\mathrm{d}\mathbf{u}^T\mathsf{H}\mathbf{u}}{\mathrm{d}t} &= \mathbf{u}^T\mathsf{H}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}+\frac{\mathrm{d}\mathbf{u}^T}{\mathrm{d}t}\mathsf{H}\mathbf{u}=-\mathbf{u}^T(\mathsf{Q}+\mathsf{Q}^T)\mathbf{u}-2\mathbf{u}^T\sigma u_0\mathbf{e}_0\\ &= -\mathbf{u}^T(\mathsf{E}_N-\mathsf{E}_0)\mathbf{u}-2\mathbf{u}^T\sigma u_0\mathbf{e}_0=-u_N^2+u_0^2-2\sigma u_0^2,\end{aligned} \tag{11}$$

which is nonnegative for $\sigma \geqslant 1/2$.

In order to demonstrate that an SBP operator is conservative, we must show that Gauss's theorem

$$\oint_S \mathbf{n}\cdot\mathbf{F}\mathrm{d}S=\int_V \nabla\cdot\mathbf{F}\mathrm{d}V, \tag{12}$$

is satisfied discretely. For the one-dimensional linear advection equation we require:

$$\mathcal{U}(x_R)-\mathcal{U}(x_L)=\int_{x_L}^{x_R}\frac{\partial \mathcal{U}}{\partial x}\mathrm{d}x. \tag{13}$$

The discrete expression for the integral on the right-hand side is:

$$\begin{aligned}\mathbf{1}^T\mathsf{H}\mathsf{D}_1\mathbf{u} &= \mathbf{1}^T\mathsf{H}\mathsf{H}^{-1}\mathsf{Q}\mathbf{u} = \mathbf{1}^T\mathsf{Q}\mathbf{u} = \mathbf{1}^T[(\mathsf{E}_N - \mathsf{E}_0) - \mathsf{Q}^T]\mathbf{u} \\ &= \mathbf{1}^T[(\mathsf{E}_N - \mathsf{E}_0)]\mathbf{u} = u_N - u_0, \end{aligned} \tag{14}$$

where we have used $\mathbf{1}^T\mathsf{Q}^T = (\mathsf{D}_1\mathbf{1})^T\mathsf{H} = \mathbf{0}$. Adding in the SAT penalty term gives:

$$\mathbf{1}^T\mathsf{H}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -\mathbf{1}^T\mathsf{Q}\mathbf{u} - \sigma(u_0 - \mathcal{U}_L) \tag{15}$$

$$= -(u_N - u_0) - \sigma(u_0 - \mathcal{U}_L) \tag{16}$$

$$= -(u_N - \mathcal{U}_L), \quad \text{if } \sigma = 1. \tag{17}$$

The choice $\sigma = 1$ therefore satisfies both conservation and time stability.

The same approach to the selection of the penalty parameter enables conservation at block interfaces even though the solution is multi-valued there as a result of the weak enforcement of the interface condition. Abutting or patched multi-block grids are often used for geometries where a single simply-connected structured grid cannot be generated. As a simple example, consider a block interface in the middle of the above domain. Discretizing in space with N_L cells to the left of the interface and N_R cells to the right of the interface, we have two solution vectors as follows:

$$\begin{aligned}\mathbf{u}_L &= \left[u_0^L, u_1^L, \ldots, u_{N_L}^L\right]^T, \\ \mathbf{u}_R &= \left[u_0^R, u_1^R, \ldots, u_{N_R}^R\right]^T. \end{aligned} \tag{18}$$

Note that the last node in the left block is coincident with the first node of the right block, yet $u_{N_L}^L \neq u_0^R$, i.e. the numerical solution is multi-valued at the interface. Introducing a SAT term into the semi-discrete form for the right domain, we have:

$$\begin{aligned}\mathsf{H}_L\frac{\mathrm{d}\mathbf{u}_L}{\mathrm{d}t} &= -\mathsf{Q}_L\mathbf{u}_L - \sigma_L(u_0^L - \mathcal{U}_L)\mathbf{e}_0, \\ \mathsf{H}_R\frac{\mathrm{d}\mathbf{u}_R}{\mathrm{d}t} &= -\mathsf{Q}_R\mathbf{u}_R - \sigma_R\left(u_0^R - u_{N_L}^L\right)\mathbf{e}_0. \end{aligned} \tag{19}$$

Note that we have taken advantage of the fact that our wave speed is positive; hence a SAT is needed only at the left side of each block. Additional SATs are required if the sign of the wave speed is not known or for a system where waves propagate in both directions. The penalty parameters σ_L and σ_R must be selected to maintain both stability and conservation.

Considering conservation first, we see immediately from (15) that the choice $\sigma_L = 1$ ensures that

$$\mathbf{1}^T\mathsf{H}_L\frac{\mathrm{d}\mathbf{u}_L}{\mathrm{d}t} = -\left(u_{N_L}^L - \mathcal{U}_L\right). \tag{20}$$

For the right-hand block we have

$$\begin{aligned}\mathbf{1}^T\mathsf{H}_R\frac{\mathrm{d}\mathbf{u}_R}{\mathrm{d}t} &= -\mathbf{1}^T\mathsf{Q}_R\mathbf{u}_R - \sigma_R\left(u_0^R - u_{N_L}^L\right) \\ &= -\left(u_{N_R}^R - u_0^R\right) - \sigma_R\left(u_0^R - u_{N_L}^L\right) \\ &= -\left(u_{N_R}^R - u_{N_L}^L\right), \quad \text{if } \sigma_R = 1. \end{aligned} \tag{21}$$

Hence with $\sigma_L = \sigma_R = 1$ we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\mathbf{1}^T\mathsf{H}_L\mathbf{u}_L + \mathbf{1}^T\mathsf{H}_R\mathbf{u}_R\right] = -\left(u_{N_R}^R - \mathcal{U}_L\right), \tag{22}$$

and the block interface is conservative.

For time stability we must have

$$\frac{\mathrm{d}}{\mathrm{d}t}\left((\mathbf{u}_L)^T\mathsf{H}_L\mathbf{u}_L + (\mathbf{u}_R)^T\mathsf{H}_R\mathbf{u}_R\right) \leqslant 0, \tag{23}$$

when $\mathcal{U}_L = 0$. For the left-hand block we have from (11) with $\sigma_L = 1$:

$$\frac{\mathrm{d}(\mathbf{u}_L)^T\mathsf{H}_L\mathbf{u}_L}{\mathrm{d}t} = -\left(u_{N_L}^L\right)^2 - \left(u_0^L\right)^2. \tag{24}$$

For the right-hand block, we obtain with $\sigma_R = 1$:

$$\frac{\mathrm{d}(\mathbf{u}_R)^T\mathsf{H}_R\mathbf{u}_R}{\mathrm{d}t} = -\left(u_{N_R}^R\right)^2 - \left(u_0^R\right)^2 + 2u_0^R u_{N_L}^L. \tag{25}$$

Summing (24) and (25) we find

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left((\mathbf{u}_L)^T\mathsf{H}_L\mathbf{u}_L + (\mathbf{u}_R)^T\mathsf{H}_R\mathbf{u}_R\right) &= -\left(u_0^L\right)^2 - \left(u_{N_R}^R\right)^2 - \left(u_{N_L}^L\right)^2 - \left(u_0^R\right)^2 \\ &\quad + 2u_0^R u_{N_L}^L = -\left(u_0^L\right)^2 - \left(u_{N_R}^R\right)^2 \\ &\quad - \left(u_{N_L}^L - u_0^R\right)^2, \end{aligned} \tag{26}$$

which is nonpositive, as required for time stability.

With this background we can now introduce some of the useful properties of the SBP-SAT approach. To date much of the use of SBP-SAT schemes has been in a finite-difference context, generally requiring rectilinear grids or curvilinear grids with a coordinate transformation. However, many of the associated concepts apply more generally, as will be discussed below. One of the main purposes of the SBP-SAT formulation is to facilitate the derivation of higher-order spatial discretizations that are provably time stable based on the energy method. Development of numerical boundary schemes for higher-order methods that are provably stable and of a suitable order of accuracy can be a challenging task; applying the energy method provides a relatively straightforward approach. Another useful property of SBP-SAT schemes is in the handling of interfaces between blocks. The approximation of the derivative in each of two neighboring blocks is completely determined by the solution in that block. This means that the amount of information passed between blocks is minimized, which is helpful for parallel algorithms, and independent of the order of the scheme. Moreover, neither the grid nor the solution need be continuous across the interface. This is particularly useful for problems where there can be different material properties on either side of the interface, such as conjugate heat transfer or electromagnetics. However, this can be advantageous in CFD as well, by relaxing the requirement of grid smoothness across the block interface. Furthermore, a block interface can intentionally be placed at a known discontinuity such as a geometric discontinuity.

The objective of this paper is to provide the reader with a thorough understanding of the SBP-SAT methodology, with a particular emphasis on the derivation of operators; several examples of operators are given. For a detailed review of theoretical aspects of SBP schemes the reader is referred to the excellent paper by Svärd and Nordström [11]. A brief history of key contributions is provided in the next section along with a description of some applications. This is followed by a detailed treatment of SBP operators for first derivatives and second derivatives with variable coefficients, including a methodology for deriving them. Simultaneous approximation terms are then described in the context of the linear convection–diffusion equation. This is followed by a section discussing application to the Navier–Stokes equations, including SATs, numerical dissipation and the transformation to curvilinear coordinates. Next a variational interpretation of SBP schemes is provided, showing the connection to, for example, discontinuous Galerkin finite-element methods. The variational interpretation and dual consistency also enable proofs of functional superconvergence and error estimates. We then describe a further generalization of the SBP concept that further reinforces the link to quadrature rules and enables unification of many operators under the SBP umbrella. This is followed by a few additional topics and conclusions.

2. A brief history

2.1. Theory

This section highlights, in roughly chronological order, some of the major advances in the theory of SBP-SAT methods. Nearly forty years ago, Kreiss and Scherer [12] laid out the basic theory of first derivative SBP operators. Their goal was to bring to higher-order FD methods a systematic means of proving stability through the energy method. SBP operators are constructed from centered difference interior point operators of order $2p$. Centered schemes are naturally SBP on periodic domains, but to retain the SBP property on finite domains, specific non-centered near-boundary point operators at nodes close to and including the boundary need to be constructed. Near the boundaries, the discretization error jumps to order p for diagonal norms or $2p-1$ for block norms. Consequently, the global order of accuracy is $p+1$ or $2p$ for diagonal and block norms, respectively [13].

During the subsequent twenty years (1974–1994), the SBP method was predominantly developed by a small group of researchers at Uppsala University (see for example [14–16]). Strand [17] summarized much of the accumulated theory for SBP operators as of 1994. He proved the existence of restricted block-norm operators with near-boundary point operators of order $2p-1$, thereby resulting in a globally $2p$ order method [13]. Moreover, he analytically derived general solutions for diagonal-norm operators with $p \in [2,4]$, and block- and restricted-block norm operators with $p = 3$ (also see Carpenter and Gottlieb [18] for construction of Padé-type SBP operators).

For one-dimensional, constant coefficient, Cauchy or periodic problems in Cartesian coordinates, the SBP property is sufficient to prove stability. However, for problems where boundaries or block interfaces are present, the traditional method of using injection or strong enforcement of boundary/interface conditions destroys the SBP property. In working on spectral methods, first Funaro [19] and then Funaro and Gottlieb [20] considered the idea of imposing boundary conditions weakly using penalty methods. In these methods, both the PDE and the boundary condition are combined at the boundary nodes. In a refinement of the concept, Carpenter, Gottlieb and Abarbanel [18] proposed the SAT method for imposing boundary conditions. Around the same time, in a series of two papers, Olsson [21,22] proposed enforcing boundary conditions using the so-called projection method. In several papers, Carpenter, Nordström and Gottlieb [23], Nordström and Carpenter [24], and Nordström and Carpenter [25] extended the SAT concept to handle various types of boundary conditions, as well as block interface conditions in curvilinear coordinates for linear problems.

Mattsson [26] systematically compared the SAT and the projection methods on the linear-convection–diffusion equation and a linear hyperbolic system of equations. He found that strict stability was lost using the projection method. After this paper, most of the development in the SBP community for imposition of boundary conditions and block interfaces has been within the SATs framework. Some additional important contributions to SATs in the context of the compressible Navier–Stokes equations Include: Svärd, Carpenter and Nordström [27] who derived far-field SATs; Svärd and Nordström [28] for no-slip wall boundary SATs; Nordström et al. [29] for block interface SATs, and Berg and Nordström for Robin solid wall boundary SAT [30].

Both diagonal-norm and block-norm operators contain free parameters after satisfying the accuracy constraints and the SBP property. Diener et al. [31] performed a systematic study examining various means of constructing optimized instances of SBP operators for the first derivative for various values of p.

Carpenter, Nordström and Gottlieb [23] were the first to derive minimum-stencil SBP operators for the second derivative for block-norms, such that the interior point operators include the same number of nodes as the interior point operators for the first derivative. Subsequently, Mattsson and Nordström [32] proposed a simpler form, mimetic of integration-by-parts, and investigated constructing both block- and diagonal-norm minimum-stencil SBP operators for the second derivative. In that paper, the authors found that using minimum-stencil second-derivative SBP operators for parabolic problems resulted in a convergence rate of $p+2$ rather than the anticipated $p+1$. This superconvergence was then theoretically proven by Svärd and Nordström [33]. In a followup paper to [32], Mattsson, Svärd and Shoeybi [34] outlined a systematic means of constructing SBP operators for the second derivative as the application of the first derivative twice plus a corrective term. Moreover, they proved that for PDEs with cross derivative terms, such as $\frac{\partial^2}{\partial x \partial y}$, for stability, the first derivative operators used to construct the second derivative and the cross derivative terms need to be the same. Kamakoti and Pantano [35] investigated the construction of minimum-stencil interior point operators for the second derivative with variable coefficients. Recently, Mattsson [36] extended the theory presented in [34] to SBP approximations of the second derivative with variable coefficients.

For nonlinear problems, some means of dissipating under-resolved high frequency modes can be required. Mattson, Svärd and Nordström [37] and then Nordström [38] developed a method for constructing dissipation models for SBP schemes that do not destroy the stability properties, while maintaining the accuracy properties of the underlying scheme. Also of interest is the use of the skew symmetric form and entropy splitting for nonlinear PDEs [39–44]).

Though block-norm operators present an improvement in the order of accuracy compared to diagonal-norm operators, Svärd proved that the SBP property is lost for curvilinear coordinates [45], consequently, much of the subsequent research has concentrated on diagonal-norm SBP operators. One of the drawbacks of diagonal-norm SBP operators is that although the interior-point operators are of order $2p$, the near-boundary operators are of order p resulting in a $p+1$ accurate scheme. This loss of accuracy can be mitigated for functionals if the SBP-SAT discretization is dual consistent: Hicken and Zingg [46] proved that when a discretization is dual consistent, functionals computed with the H-norm are superconvergent of order $2p$ (see also [47]). In part, their analysis relies on the fact that the H-norm represents a $2p$ accurate quadrature rule [48]. Berg and Nordström [49] subsequently extended the ideas in [46] to include temporal dependence.

2.2. Application

In this section, we review the application of the SBP-SAT method to complex problems. In particular, the focus is on the Euler and Navier–Stokes equations.

The majority of the mechanics for implementing SBP-SAT discretizations for the Euler and Navier–Stokes equations can be found in [24,25,32,27,28,34,29,50,36]. Mattsson, Svärd, Carpenter, and Nordström [51] examined the efficiency gains of third and fifth-order block-norm SBP operators for free-stream vortex and vortex-wing interactions for the Euler equations. They found that the efficiency gains for the higher-order methods for the simple vortex flow case translated to similar performance for the more complicated vortex-wing interaction problem.

In the last of a trilogy of papers, Nordström et al. [29] derived suitable SATs for the NS equations for multi-block schemes. They demonstrated the validity of their approach by showing physically correct temporal behavior of the shedding of von Karman vortices behind a cylinder.

In a follow-up paper to [51], Svärd, Lundberg and Nordström [52] examined a vortex-wing interaction including the effects of viscosity. They validated the proposed SBP-SAT scheme and discussed using the enforcement of the no-slip boundary condition as a metric for determining the accuracy when an analytical solution is not available.

One drawback of coupling block interfaces with SATs is that this procedure has been shown to significantly reduce the maximum stable Courant Fiedrichs Lewy number for explicit schemes [24]. To circumvent this issue Hicken and Zingg [9] proposed using a Jacobian-Free Newton–Krylov–Schur solution algorithm. This algorithm couples a two-stage Newton method for solving the nonlinear equations with a Krylov subspace method to solve the system of equations at each Newton step. They concluded that SBP-SAT schemes fit will within a parallel Newton–Krylov–Schur solution strategy; the combination provides an efficient approach to the solution of large-scale problems.

Osusky and Zingg [6] extended the algorithm in [9] to the Reynolds-averaged Navier–Stokes equations using the Spalart–Allmaras one-equation turbulence model. They further demonstrated that the Newton–Krylov–Schur algorithm is well suited to use with SBP-SAT schemes, providing an efficient parallel algorithm for the solution of steady turbulent flows over complex geometries.

The SBP-SAT method has also been successfully applied to large scale unsteady problems. Osusky, Boom, Del Rey Fernández and Zingg [53] solved the temporal evolution of the Taylor–Green vortex flow with a high-order SBP-SAT discretization using various high-order temporal discretizations. They found the combination of higher-order space and time discretization to be advantageous for efficiently capturing the evolution of vortical structures and integrated quantities. Using the same code, Boom and Zingg [54] computed implicit large-eddy simulations of the Taylor–Green vortex flow and transitional flow over an SD7003 wing. They found that for the Taylor–Green vortex, using a fourth-order method, they could reduce the computational effort, relative to a second-order method by 85%. Moreover, they found that for the grid densities under investigation, the higher-order methods were necessary in order to resolve known physical features.

There has also been extensive use of SBP schemes for direct numerical simulations (DNS) of various flow types. Sandham, Li and Yee [40] used block-norm SBP operators to compute the DNS solution to wall-bounded isothermal compressible turbulent channel flow. They used entropy splitting to condition the inviscid portion of the NS equations. The algorithm does not use any numerical dissipation and boundary conditions were imposed using the projection method. They found that it compared well with previously reported experimental and numerical results.

Sjögreen and Yee [55] compared second-order to fifth and sixth-order block-norm SBP operators for the solution of several multiscale complex unsteady viscous compressible flows including: the 2D shock tube problem with viscous effects (no-slip adiabatic wall boundary conditions), supersonic reactive flow with respect to fuel breakup, and a planar shock interacting with two hydrogen bubbles. They found that higher-order methods on coarser grids give comparable results to second-order methods on finer grids and are more efficient.

Other areas of application include: conjugate-heat transfer [56,57]; uncertainty quantification [58–60]; magnetic induction [61–63]; Wave propagation [64–68]. Other interesting examples include, DNS of the interaction between a turbulent boundary layer and a fluttering panel [69]; and, computation of acoustically-excited flow through a circular orifice backed by a hexagonal cavity [70]. SBP-SAT schemes are provably strictly time-stable, which roughly speaking means that the long term energy growth of the semi-discrete equations is less than or equal to that of the PDE (see [71,72]). This makes them particularly attractive for problems involving propagation over long distances (see for example [73]).

3. Notation

Vectors are denoted with lower-case bold fonts, while matrices are presented using upper-case sans-serif fonts, for example M. Upper-case letters with script type are used to denote continuous functions on a specified domain $x \in [\alpha, \beta]$. As an example, $\mathcal{U}(x) \in C^{\infty}[\alpha, \beta]$ denotes an infinitely differentiable function on the domain $x \in [\alpha, \beta]$. Lower-case bold fonts are used to denote the restriction of such functions onto a grid; for example the restriction of $\mathcal{U}$ onto the grid $\mathbf{x}$ is given by:

$$\mathbf{u} = [\mathcal{U}(x_0), \ldots, \mathcal{U}(x_N)]^T.$$

Vectors with a subscript h, for example $\mathbf{u}_\mathrm{h} \in \mathbb{R}^{N\times 1}$, represent the solution to a system of discrete or semi-discrete equations.

Throughout the paper monomials are used in proving the degree of various operators. These are represented by $\mathbf{x}^i = [x_0^i, \ldots, x_N^i]^T$, with the convention that $\mathbf{x}$ raised to a negative power is equal to zero. We also make frequent use of the vectors $\mathbf{e}_0 = [1, 0, \ldots, 0]^T$ and $\mathbf{e}_N = [0, 0, \ldots, 1]^T$ and matrices $\mathsf{E}_0 = \mathrm{diag}[1, 0, \ldots, 0]$, $\mathsf{E}_N = \mathrm{diag}[0, 0, \ldots, 1]$, and $\mathsf{E} = \mathsf{E}_N - \mathsf{E}_0$.

The operators for the first and second derivatives have different orders of accuracy on the interior, at the boundary, and globally. When it is necessary to distinguish between operators and the various orders of accuracy, we append a superscript to operators for the various orders of accuracy and a subscript to denote which derivative we are approximating. For example, $\mathsf{D}_{i,e}^{(a,b,c)}$, denotes the operator for the ith derivative with interior order of accuracy a, boundary closure accuracy b, and a solution with global order of accuracy c. The additional subscript e enables us to distinguish between various versions of the operator. In some cases we will not be interested in one or several of the orders of accuracy and will insert colons; as an example, $\mathsf{D}_{3}^{(2,:,:)}$ denotes an approximation to the third derivative that is second-order on the interior where we specify neither the accuracy of the operator at the boundary nor the global order of accuracy. For the second derivative with variable coefficients we will make explicit the dependence on the variable coefficients by denoting these operators as $\mathsf{D}_2^{(a,b,c)}(\mathsf{B})$, where B is a diagonal matrix with the variable coefficients along its diagonal.

4. SBP operators for first derivatives

4.1. Preliminaries

SBP operators are constructed to be mimetic of the integration by parts (IBP) property of the first derivative. There are many discrete operators that possess the SBP property; for example, Carpenter and Gottlieb [74] and Gassner [75] investigated the SBP property of pseudo-spectral collocated methods. In the continuous case, IBP has the following form:

$$\int_{x_L}^{x_R} \mathcal{V}\frac{\mathrm{d}\mathcal{U}}{\mathrm{d}x}\mathrm{d}x = \mathcal{U}\,\mathcal{V}|_{x_L}^{x_R} - \int_{x_L}^{x_R} \mathcal{U}\frac{\mathrm{d}\mathcal{V}}{\mathrm{d}x}\mathrm{d}x. \tag{27}$$

In order to construct a well-posed problem, a system of PDEs needs to be augmented by appropriate boundary conditions. In the continuous case, it is possible to use the energy method to determine boundary conditions that lead to well-posed problems. In the energy method, the PDE is multiplied by the solution and integrated in space. IBP is then used to transform volume integrals to surface integrals to allow the introduction of boundary conditions to determine the well-posedness of the PDE and the boundary conditions (for more information regarding the energy method see [76,71,72]). In the continuous case, the L_2 inner product and norm are defined as

$$(\mathcal{U},\mathcal{V}) = \int_{x_L}^{x_R} \mathcal{U}\mathcal{V}\mathrm{d}x, ||\mathcal{U}||^2 = \int_{x_L}^{x_R} \mathcal{U}^2\mathrm{d}x. \tag{28}$$

With these definitions, (27) can be interpreted as

$$\left(\mathcal{V},\frac{\partial\mathcal{U}}{\partial x}\right) = \mathcal{U}\mathcal{V}|_{x_L}^{x_R} - \left(\mathcal{U},\frac{\partial\mathcal{V}}{\partial x}\right). \tag{29}$$

SBP operators are constructed to discretely approximate (29) using a grid of $N+1$ nodes defined by $\mathbf{x} = [x_0,\ldots,x_N]^T$ for $x \in [x_L, x_R]$. The goal is to construct SBP operators of various orders. The accuracy of the operator is defined by its degree p, the maximum degree of the polynomial for which it is exact, by conditions referred to as the accuracy equations:

$$\mathsf{D}_1\mathbf{x}^j = j\mathbf{x}^{j-1}, \quad j \in [0,p]. \tag{30}$$

These can alternatively be obtained via Taylor series and represent $(N+1)\times(p+1)$ individual equations. However, the number of linearly independent equations is generally less than $(N+1)\times(p+1)$. The discrete analog of IBP is SBP. Using the discrete inner product and norm (5), where H can be diagonal or block diagonal, the SBP property has the following form:

$$(v,\mathsf{D}_1\mathbf{u})_{\mathsf{H}} = u_N v_N - u_0 v_0 - (\mathbf{u},\mathsf{D}_1\mathbf{v}), \tag{31}$$

or, equivalently

$$\mathbf{v}^T\mathsf{H}\mathsf{D}_1\mathbf{u} = \mathbf{v}^T\mathsf{E}\mathbf{u} - \mathbf{u}^T\mathsf{H}\mathsf{D}_1\mathbf{v}, \tag{32}$$

where $\mathsf{E} = \mathrm{diag}[-1,0,\ldots,0,1]$. Not all first-derivative approximations satisfy (32), and the conditions under which D_1 can satisfy (32) need to be determined. To do so, take the transpose of (32), add it to (32), and rearrange to obtain

$$\mathbf{v}^T\left[\mathsf{H}\mathsf{D}_1 + \mathsf{D}_1^T\mathsf{H}\right]\mathbf{u} + \mathbf{u}^T\left[\mathsf{D}_1^T\mathsf{H} + \mathsf{H}\mathsf{D}_1\right]\mathbf{v} = \mathbf{v}^T\mathsf{E}\mathbf{u} + \mathbf{u}^T\mathsf{E}\mathbf{v}. \tag{33}$$

Let $\mathsf{Q} = \mathsf{H}\mathsf{D}_1$ (since H is invertible, $\mathsf{D}_1 = \mathsf{H}^{-1}\mathsf{Q}$); (33) becomes

$$\mathbf{v}^T\left[\mathsf{Q} + \mathsf{Q}^T\right]\mathbf{u} + \mathbf{u}^T\left[\mathsf{Q}^T + \mathsf{Q}\right]\mathbf{v} = \mathbf{v}^T\mathsf{E}\mathbf{u} + \mathbf{u}^T\mathsf{E}\mathbf{v}, \tag{34}$$

and it is concluded that

$$\mathsf{Q} + \mathsf{Q}^T = \mathsf{E}. \tag{35}$$

To summarize, the following definition is given:

Definition 1. Summation-by-parts operator: An operator is an approximation to the first derivative of degree p with the SBP property if

(i) $\mathsf{D}_1\mathbf{x}^j = \mathsf{H}^{-1}\mathsf{Q}\mathbf{x}^j = j\mathbf{x}^{j-1}, j \in [0,p]$,
(ii) H is a positive definite (PD) symmetric matrix, and
(iii) $\mathsf{Q} + \mathsf{Q}^T = \mathsf{E}$.

SBP operators are constructed by first specifying an internal stencil and then determining what modifications are required at nodes on and near the boundary such that the SBP property is obtained and the accuracy equations satisfied. Both the diagonal and block-norm operators require a minimum of $2p$ modified stencils, at both boundaries. Beyond a certain degree of accuracy, free parameters remain after satisfying the accuracy equations and the SBP form. These free parameters can then be used to optimize the operators in some sense.

The steps for deriving diagonal or block-norm SBP operators are the same; the starting point is the accuracy Eq. (30). Multiplying both sides by H gives

$$\mathsf{Q}\mathbf{x}^j = j\mathsf{H}\mathbf{x}^{j-1}, \quad j \in [0,p]. \tag{36}$$

The matrix Q can be split into its symmetric and antisymmetric components: $\mathsf{Q} = \mathsf{Q}_\mathsf{S} + \mathsf{Q}_\mathsf{A}$. The SBP property, $\mathsf{Q} + \mathsf{Q}^T = \mathsf{E}$, leads to the conclusion that $\mathsf{Q}_\mathsf{S} = \mathrm{diag}[-\frac{1}{2},0,\ldots,0,\frac{1}{2}]$. Thus, (36) becomes

$$\mathsf{Q}_\mathsf{A}\mathbf{x}^j - j\mathsf{H}\mathbf{x}^{j-1} = -\mathsf{Q}_\mathsf{S}\mathbf{x}^j = -\frac{1}{2}\mathsf{E}\mathbf{x}^j, \quad j \in [0,p]. \tag{37}$$

4.2. Diagonal-norm SBP operators

To make the presentation transparent, we use the $p = 2$ operator as an example (SBP first derivative operators, for $p \in [1,4]$, are given in Appendix A). All SBP operators have a unique repeating interior point operator of degree $2p$. In this example, the interior point operator is of degree 4 and is given as $\frac{1}{12}S^{-2} - \frac{2}{3}S^{-1} + \frac{2}{3}S^{1} - \frac{1}{12}S^{2}$, where S is the shift operator and has the property $S^k u_j = u_{j+k}$. At the boundaries, the operators are defined by a $2p \times 2p$ block and the symmetries of the operator; for the current example, see (8) and (9).

The interior of Q is defined by the repeating interior point operator. The lower portion of the operator is the negative of the permutation of the rows and columns of the upper 4×6 block; therefore, it is nearly anti-bisymmetric. Similarly, the lower portion of H is the permutation of the rows and columns of the upper portion, i.e., $\mathsf{H}(i,i) = \mathsf{H}_{N-(i+1),N-(i+1)}$; therefore, it is bisymmetric.

Now the system of Eq. (37) is constructed for $j \in [0,2]$, which gives a total of $2p \times 3 = 12$ equations, while there are only 10 unknowns. Nevertheless, a sufficient number of equations are linearly dependent such that for $p = 2$ there is a unique solution. For $p = 3$ there is one free parameter, while for $p = 4$ there are 2 free parameters.

For operators with free parameters, $p = 3$ and $p = 4$, the question of how to optimize the diagonal-norm SBP operators for the first derivative has been investigated by several authors [31,77,34,78,79]. There are two basic criteria that are typically chosen for optimization: the truncation error or the spectral properties. However, Diener et al. [31] have shown that there is a strong relationship between the spectral radius and the truncation error; when they minimized for truncation error they also found that the spectral radius was greatly reduced.

Typically, both the solution error and functionals are computed using the H-norm. Therefore, an alternative to minimizing the sum of the point-wise boundary node truncation error is to minimize the H-norm of the truncation error. This has the advantage that the optimum minimizes the error in quantities computed using the norm. Using this idea, a simple objective function, J, based on the truncation error vector $\mathbf{e}$, is as follows:

$$\mathbf{e}_k = \mathsf{D}_1\mathbf{x}^k - k\mathbf{x}^{k-1}, \; J_k = \mathbf{e}_k^T\mathsf{H}\mathbf{e}_k, \; k \geqslant p+1. \tag{38}$$

For the operator $\mathsf{D}_1^{(6,3,:)}$ ($p = 3$) Maple's minimize function finds the minimum of J_4 as $\theta_{56} = \frac{5591070156686698065364559}{7931626489314500743872000}$.

The operator $\mathsf{D}_1^{(8,4,:)}$ ($p = 4$) has three free parameters and minimizing J_5 specifies one of the free parameters in terms of the other two. We then minimize J_6 to specify the remaining free parameters, giving to five decimal places: $\theta_{16} = 0.08314, \theta_{17} = -0.00952$, and $\theta_{47} = -0.03510$. In Appendix A, Matlab scripts are provided to construct diagonal-norm SBP operators for the first derivative for $p \in [0,4]$.

4.3. Block-norm SBP operators

The block-norm H is bisymmetric, and for $p = 2$ is given as

$$\mathsf{H} = \Delta x \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & & \\ h_{12} & h_{22} & h_{23} & h_{24} & & \\ h_{13} & h_{23} & h_{33} & h_{34} & & \\ h_{14} & h_{24} & h_{34} & h_{44} & & \\ & & & & 1 & \\ & & & & & \ddots \end{bmatrix}. \tag{39}$$

Solving the accuracy Eq. (36) results in an underdetermined system with a number of free parameters. The next step is to use the free parameters to optimize the operator in some way while maintaining a PD H. The latter presents a challenge, particularly for $p > 2$, as the block portion of H grows as $2p \times 2p$, and the number of free parameters increases as well. Nevertheless, it is possible. An alternative is to construct H such that it is PD by definition. Consider constructing the upper portion of H as

$$\mathsf{H}(1:2p, 1:2p) = \mathsf{L}^T \wedge \mathsf{L}, \tag{40}$$

where L is a lower unitriangular matrix and $\wedge$ is a diagonal matrix. In this form, H is guaranteed to be PD if $\Lambda > 0$; the downside is that (36) becomes a system of nonlinear equations.

To avoid solving nonlinear systems we take a different approach. We solve the accuracy equations and set $\mathsf{H}_{1,j} = 0$, starting with $j = 2p$, until a unique solution is obtained. Since, H has not been constructed using form (40) there is no guarantee that the resultant H is PD. However, for $p \in [1, 4]$, this strategy results in PD H and for $p = 2$ gives

$$\mathsf{H}^{(4,3,4)} = \Delta x \begin{bmatrix} \frac{173}{648} & \frac{41}{1296} & 0 & 0 & & \\ \frac{41}{1296} & \frac{1135}{648} & -\frac{353}{648} & \frac{17}{108} & & \\ 0 & -\frac{353}{648} & \frac{901}{648} & -\frac{151}{1296} & & \\ 0 & \frac{17}{108} & -\frac{151}{1296} & \frac{671}{648} & & \\ & & & & 1 & \\ & & & & & \ddots \end{bmatrix}, \tag{41}$$

and a Q matrix

$$\mathsf{Q}^{(4,3,4)} = \begin{bmatrix} -\frac{1}{2} & \frac{2035}{2592} & -\frac{239}{648} & \frac{217}{2592} & 0 & 0 & & \\ -\frac{2035}{2592} & 0 & \frac{829}{864} & -\frac{113}{648} & 0 & 0 & & \\ \frac{239}{648} & -\frac{829}{864} & 0 & \frac{1747}{2592} & -\frac{1}{12} & 0 & & \\ -\frac{217}{2592} & \frac{113}{648} & -\frac{1747}{2592} & 0 & \frac{2}{3} & -\frac{1}{12} & & \\ & & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}. \tag{42}$$

5. SBP operators for second derivatives with variable coefficients

Many equations in the physical sciences contain second-order terms. As an example, consider the compressible NS equations, which require second-order derivatives of the form $\frac{\partial}{\partial x}\left(\mathcal{B}\frac{\partial \mathcal{Q}}{\partial x}\right)$, where $\mathcal{Q}$ is a conserved variable and $\mathcal{B}$ is a variable coefficient. Equations with second-order terms can be decomposed into a new system of first order terms; however, this results in an expanded system of equations that must be solved. To circumvent this, the second derivative can be approximated directly. For SBP operators, the simplest means of doing so is application of the first derivative twice, which have interior point operators that use $4p + 1$ nodes, referred to as wide-stencil operators. Alternatively, minimum-stencil SBP operators can be derived that have interior point operators that use the same number of nodes as the interior point operators for the first derivative $(2p + 1)$. Minimum-stencil operators have several numerical advantages over wide-stencil operators [34]; they have lower global error and are more dissipative of high wavenumber modes. Moreover, they have a smaller bandwidth and thus require less computational resources, particularly if one is interested applications for which the Jacobian is constructed (e.g. adjoint-based sensitivity analysis). Finally, using wide-stencil SBP operators results in the loss of one order of accuracy in the context of parabolic PDEs [32].

Mattsson [36] extended the constant coefficient algorithm presented in [34] to the variable coefficient case and derived diagonal-norm minimum-stencil SBP operators for the second derivative with variable coefficients.

We present a procedure for deriving minimum-stencil SBP operators for the second derivative with constant or variable coefficients based on the ideas [32,34,36], but with formalism as in [80,81]. First, we define the SBP operators for the second derivative with constant or variable coefficients and give a generic structure to construct them. We define the discrete SBP operator for the second derivative with constant coefficients, $\mathbf{B} = \mathrm{diag}(1, \ldots, 1)$, or variable coefficients as follows:

Definition 2. SBP Second Derivative: The matrix $\mathsf{D}_2^{(2p,p,\hat{p})}(\mathsf{B}) \in \mathbb{R}^{(N+1)\times(N+1)}$ is an SBP operator for the differential operator

$$\frac{\partial}{\partial x}\left(\beta \frac{\partial u}{\partial x}\right),$$

if it is of of the form,

$$\mathsf{D}_2^{(2p,p,\hat{p})}(\mathsf{B}) = \mathsf{H}^{-1}\{-\mathsf{M} + \mathsf{E}\mathsf{B}\mathsf{D}_b\},$$

where $\mathsf{B} = \mathrm{diag}(\beta_0, \ldots, \beta_N), \mathsf{D}_b^{(:,\geqslant p+1,:)}$ is an approximation to the first derivative at the boundaries, and

$$\mathsf{M} = \left(\mathsf{D}_1^{(2p,p,p+1)}\right)^T \mathsf{H}\mathsf{B}\mathsf{D}_1^{(2p,p,p+1)} + \mathsf{R}.$$

The matrices B, M and R are positive-semi-definite (PSD) and symmetric.

In order to construct stable SBP-SAT discretizations, the first derivative and second derivative must have the same norm [34]. Furthermore, if cross-derivative terms are present the SBP operators for the first derivative in Definition 2 must be the same as the SBP operators used for the cross derivative terms (if in addition, as in Definition 2, R is PSD then D_1 and $\mathsf{D}_2^{(2p,p,\hat{p})}(\mathsf{B})$ are said to be compatible [34]).

Before proceeding, let us make clear how the given definition leads to an SBP operator and show the relationship between the minimum- and wide-stencil operators. Consider the variable-coefficient parabolic equation on the interval $[0, 1]$:

$$\frac{\partial \mathcal{U}}{\partial t} = \frac{\partial}{\partial x}\left(\beta \frac{\partial \mathcal{U}}{\partial x}\right). \tag{43}$$

Applying the energy method to (43) gives

$$\frac{\mathrm{d}||\mathcal{U}||^2}{\mathrm{d}t} = 2\left(\beta \mathcal{U}\frac{\partial \mathcal{U}}{\partial x}\right)\Big|_0^1 - 2\int_0^1 \beta \left(\frac{\partial \mathcal{U}}{\partial x}\right)^2 \mathrm{d}x. \tag{44}$$

Ignoring boundary conditions, the semi-discrete equations are

$$\frac{\mathrm{d}\mathbf{u}_\mathrm{h}}{\mathrm{d}t} = \mathsf{H}^{-1}\{-\mathsf{M} + \mathsf{E}\mathsf{B}\mathsf{D}_b\}\mathbf{u}_\mathrm{h}.$$

Multiplying by $\mathbf{u}_\mathrm{h}^T\mathsf{H}$ and adding the transpose of the product gives

$$\begin{aligned} \frac{\mathrm{d}||\mathbf{u}_\mathrm{h}||_\mathsf{H}^2}{\mathrm{d}t} &= 2\mathbf{u}_\mathrm{h}^T\mathsf{E}\mathsf{B}\mathsf{D}_b\mathbf{u}_\mathrm{h} - 2\mathbf{u}_\mathrm{h}^T\mathsf{M}\mathbf{u}_\mathrm{h} \\ &= 2\mathbf{u}_\mathrm{h}^T\mathsf{E}\mathsf{B}\mathsf{D}_b\mathbf{u}_\mathrm{h} - 2(\mathsf{D}\mathbf{u}_\mathrm{h})^T\mathsf{H}\mathsf{B}\mathsf{D}\mathbf{u}_\mathrm{h} - 2\mathbf{u}_\mathrm{h}^T\mathsf{R}\mathbf{u}_\mathrm{h}. \end{aligned} \tag{45}$$

Let us first consider the application of the first derivative twice; in this case, $\mathsf{R} = 0$ and $\mathsf{D}_b = \mathsf{D}_1$, and (45) reduces to

$$\frac{\mathrm{d}||\mathbf{u}_{\mathrm{h}}||_{\mathrm{H}}^{2}}{\mathbf{d}t} = 2\mathbf{u}_{\mathrm{h}}^{T}\mathsf{EBD}\mathbf{u}_{\mathrm{h}} - 2(\mathsf{D}\mathbf{u}_{\mathrm{h}})^{T}\mathsf{HBD}\mathbf{u}_{\mathrm{h}}.$$

Since the discrete norm represents a $2p$-order quadrature, we see that $(\mathsf{D}\mathbf{u}_{\mathrm{h}})^{T}\mathsf{HBD}\mathbf{u}_{\mathrm{h}} \approx \int_0^1 \frac{\partial\mathcal{U}}{\partial x}\mathcal{B}\frac{\partial\mathcal{U}}{\partial x}\mathrm{d}x$, while $\mathbf{u}_{\mathrm{h}}^{T}\mathsf{EBD}\mathbf{u}_{\mathrm{h}} \approx \mathcal{B}\frac{\partial\mathcal{U}}{\partial x}\Big|_0^1$. The difference in the minimum-stencil operator is the addition of the term $\mathbf{u}_{\mathrm{h}}^{T}\mathsf{R}\mathbf{u}_{\mathrm{h}}$. From (45) we see that for stability, R must be PSD.

To construct these operators, we need R. For an operator with $2p$ accuracy on the interior, we posit the general form [80,81]:

$$\mathsf{R}_p = \frac{1}{h}\sum_{i=p+1}^{2p}\alpha_i\left(\widetilde{\mathsf{D}}_{i,p}^{(2,1,:)}\right)^T\mathsf{C}_i^{(p)}\mathsf{B}\widetilde{\mathsf{D}}_{i,p}^{(2,1,:)},$$

where h is the mesh spacing. The $\widetilde{\mathsf{D}}_{i,p}^{(2,1,:)}$ operators have an interior point operator that spans $2p+1$ nodes, while near-boundary point operators spanning the first or last $3p$ nodes are used at the first $2p$ and last $2p$ nodes. The interior point operator is a second-order centered difference approximation to the ith derivative, while the near-boundary point operators are first-order accurate. The tilde notation denotes an undivided difference approximation. Constructed as such, the operator is guaranteed to be PSD, as long as the $\mathsf{C}_i^{(p)}$, which are diagonal matrices of the form $\mathsf{C}_i^{(p)} = \mathrm{diag}\left(c_{i,1}^{(p)},\ldots,c_{i,2p}^{(p)},1,\ldots,1,c_{i,2p}^{(p)},\ldots,c_{i,1}^{(p)}\right)$, are PSD.

The interior point operator that is compatible with the proposed construction of the second derivative is given as [80,81]

$$D_{2,\mathrm{int}}^{(2p,:,:)} = -\left(D_{1,\mathrm{int}}^{(2p,:,:)}\right)^T b_j D_{1,\mathrm{int}}^{(2p,:,:)} - \frac{1}{h^2}\sum_{i=p+1}^{2p}\alpha_i\left(\widetilde{D}_{i,p,\mathrm{int}}^{(2,:,:)}\right)^T b_j\widetilde{D}_{i,p,\mathrm{int}}^{(2,:,:)}, \tag{46}$$

where $D_{1,\mathrm{int}}^{(2p,:,:)}$ and $\widetilde{D}_{i,\mathrm{int}}^{(:,:)}$ are the interior point operators of the respective matrix operators. The interior point operators are constructed using the shift operator; in this context, the transpose operation means that the exponents of the shift operators are multiplied by -1. For example, consider the interior point operator

$$\widetilde{D} = \alpha_1 S^{-1} + \alpha_2 S^0 + \alpha_3 S^1 + \alpha_4 S^3. \tag{47}$$

The transpose operation applied to (47) results in

$$\widetilde{D}^T = \alpha_1 S^1 + \alpha_2 S^0 + \alpha_3 S^{-1} + \alpha_4 S^{-3}. \tag{48}$$

Taking $b_j = 1$, (46) is the classical decomposition of minimum-stencil $2p$ accurate centered difference approximations to the second derivative into the application of the first derivative twice plus a corrective term. The global order of accuracy, $\hat{p}$, will depend on the particular problem being solved and can be in the range $\hat{p} \in [p+1, p+2]$ [33]. The general form is

$$\mathsf{D}_2^{(2p,p,:)}(\mathsf{B}) = \mathsf{H}^{-1}\left[-\left(\mathsf{D}_1^{(2p,p,:)}\right)^T\mathsf{HBD}_1^{(2p,p,:)} + \mathsf{EBD}_b^{(:,\geqslant p+1,:)}\right] - \frac{1}{h}\mathsf{H}^{-1}\sum_{i=p+1}^{2p}\alpha_i^{(p)}\left(\widetilde{\mathsf{D}}_{i,p}^{(2,1,:)}\right)^T\mathsf{C}_i^{(p)}\mathsf{B}\widetilde{\mathsf{D}}_{i,p}^{(2,1,:)}, \tag{49}$$

where the $\alpha_i^{(p)}$ coefficients are

- $p = 1 : \alpha_2^{(1)} = \frac{1}{4}$;
- $p = 2 : \alpha_3^{(2)} = \frac{1}{18}, \alpha_4^{(2)} = \frac{1}{48}$;
- $p = 3 : \alpha_4^{(3)} = \frac{1}{80}, \alpha_5^{(3)} = \frac{1}{100}, \alpha_6^{(3)} = \frac{1}{720}$; and
- $p = 4 : \alpha_5^{(4)} = \frac{1}{350}, \alpha_6^{(4)} = \frac{1}{252}, \alpha_7^{(4)} = \frac{1}{980}, \alpha_8^{(4)} = \frac{1}{11200}$.

The above formulation leads to multiple solutions with many free parameters for which values must be chosen.

The SBP operator for the second derivative is a function of the matrix B, where for constant coefficients, B is the identity matrix. As with the first derivative, monomials are used to construct the equations that the operator needs to satisfy. In contrast to the first derivative, the degree and the order of the operator are not the same; for the second derivative, the degree is the order plus one. Thus for constant coefficients, the accuracy equations have the following form:

$$\mathsf{D}_2^{(2p,p,:)}\mathbf{x}^j = j(j-1)\mathbf{x}^{j-2},\ j \in [0, p+1]. \tag{50}$$

Therefore, for each j (50) results in a system of $N+1$ equations. However, by construction the operator satisfies these equations on the interior. What remains are equations from the first and last $3p$ nodes. This results from the application of the first derivative twice, which reduces the order of accuracy at the first and last $3p$ nodes. However, the operator is constructed so that it is invariant under the transformation $x \to -x$, which means that only the first $3p$ equations need to be solved, giving a total of $3p \times (p+2)$ equations. The general form of the constituent components of SBP operators for the second derivative with variable coefficients is given in Appendix B.

For variable coefficients, it is not as clear how to pose the accuracy equations. The SBP operator must approximate $\frac{\partial}{\partial x}\left(\mathcal{B}\frac{\partial\mathcal{U}}{\partial x}\right)$ and be of degree $p+1$. We want to know what degree monomials to insert for $\mathcal{B}$ and $\mathcal{U}$ in constructing the accuracy equations. Taking $\mathcal{B} = x^q$ and $\mathcal{U} = x^r$ and inserting into the second derivative gives

$$\frac{\partial}{\partial x}\left(x^q\frac{\partial x^r}{\partial x}\right) = x^{q+r-2}. \tag{51}$$

This means that all combinations $q + r \leqslant p+1$ must be satisfied; thus, for variable coefficients, the accuracy equations have the following form:

$$\mathsf{D}_2^{(2p,p,:)}(\mathrm{diag}(\mathbf{x}^q))\mathbf{x}^r = r(q+r-1)\mathbf{x}^{q+r-2},\ q+r \leqslant p+1. \tag{52}$$

Given that the Eq. (50) are a subset of (52), it is sufficient to solve (52).

We demonstrate each step with respect to the operator $\mathsf{D}_2^{(4,2,:)}(\mathsf{B})$, while a Matlab script to construct the operator is provided in Appendix B. In expanded form, the operator is given as

$$\mathsf{D}_2^{(4,2,:)}(\mathsf{B}) = \mathsf{H}^{-1}\left[-\left(\mathsf{D}_1^{(4,2,:)}\right)^T\mathsf{HBD}_1^{(4,2,:)} - \frac{1}{18h}\left(\widetilde{\mathsf{D}}_{3,2}^{(2,1,:)}\right)^T\mathsf{C}_3^{(2)}\mathsf{B}\widetilde{\mathsf{D}}_{3,2}^{(2,1,:)} - \frac{1}{48h}\left(\widetilde{\mathsf{D}}_{4,2}^{(2,1,:)}\right)^T\mathsf{C}_4^{(2)}\mathsf{B}\widetilde{\mathsf{D}}_{4,2}^{(2,1,:)} + \mathsf{EBD}_b^{(:,\geqslant 3,:)}\right]. \tag{53}$$

Construction of $\mathsf{D}_2^{(4,2,:)}(\mathsf{B})$ requires the matrices

- $\mathsf{D}_1^{(4,2,:)}$ and H;
- $\widetilde{\mathsf{D}}_{3,2}^{(2,1,:)}, \mathsf{C}_3^{(2)}, \widetilde{\mathsf{D}}_{4,2}^{(2,1,:)}, \mathsf{C}_4^{(2)}$; and
- $\mathsf{D}_b^{(:,\geqslant 3,:)}$.

The matrices for the first derivative, $\mathsf{D}_1^{(4,2,:)}$ and H, are separately derived (see Section 4.2). For $p = 2$ and near-boundary point operators for the first and last four nodes, the matrices are unique and are given in the Matlab script in Appendix A. The matrices $\widetilde{\mathsf{D}}_{j,p}^{(2,1,:)}$ have centered interior point operators that are undivided second-order approximations to the jth derivative that include $2p+1$ nodes. However, they are first order at the first $2p$ and last $2p$ boundary nodes and include $3p$ nodes (see Appendix B).The matrices used to construct $\mathsf{D}_2(\mathsf{B})$ differ from those in [36]; however they result in operators with identical interior point operators.

The matrix $\mathsf{C}_3^{(2)}$ is bisymmetric and defined by $\mathsf{C}_3^{(2)} = \mathrm{diag}\left[c_{3,1}^{(2)}, c_{3,2}^{(2)}, c_{3,3}^{(2)}, c_{3,4}^{(2)}, 1, \ldots\right]$; similarly, $C_4^{(2)}$ is bisymmetric and is defined by $\mathsf{C}_4^{(2)} = \mathrm{diag}\left[c_{4,1}^{(2)}, c_{4,2}^{(2)}, c_{4,3}^{(2)}, c_{4,4}^{(2)}, 1, \ldots\right]$. Finally, the operator $\widetilde{\mathsf{D}}_b^{(:,\geqslant p+1,:)}$ is an approximation to the first derivative at the boundary nodes of at least degree $p+1$. In the current example, the first row is given as $\left[\widetilde{D}_{b,1}, \widetilde{D}_{b,2}, \widetilde{D}_{b,3}, \widetilde{D}_{b,4}, \widetilde{D}_{b,5}, \widetilde{D}_{b,6}, 0, \ldots\right]$, while the last row is the negative of the permutation of the columns, with the remaining entries of the operator being zero.

The resultant nonlinear system of equations is underdetermined and gives four families of solutions, each with free parameters. In order to choose a family of solutions and the optimal member of that family requires a criterion for optimization. Here, we follow the same idea proposed for the first derivative and construct an objective function, J, based on the truncation error vector $\mathbf{e}$ as follows:

$$\begin{aligned}
\mathbf{e}_{q,r} &= \mathsf{D}_2(\mathrm{diag}(\mathbf{x}^q))\mathbf{x}^r - q(q+r-1)\mathbf{x}^{q+r-2},\\
J_{q,r} &= \mathbf{e}_{q,r}^T \mathsf{H}\mathbf{e}_{q,r},\\
J &= \sum_{i=1}^{p+3} J_{(p+3-i),(i-1)}.
\end{aligned} \tag{54}$$

Besides the objective function (54), the entries of the C matrices are constrained to be $\geqslant 0$, which is necessary for stability. An example optimized solution is given in Appendix B, where Maple's numerical minimization function, Minimize, was used to perform the optimization. The optimization was carried out until the objective function value did not vary in the second decimal place.

6. Simultaneous approximation terms

The theory of SATs depends on the concept of well-posednes and stability. In this section we introduce this theory. First, using the linear convection–diffusion (LCD) equation as an example, we develop the basic approach for constructing SATs. One starts with the continuous PDE and uses the energy method to determine data (boundary and initial conditions and forcing function) that lead to a well-posed problem. This analysis helps determine the SATs that are constructed for the semi-discrete equations. The energy method is then applied to determine the additional conditions on the SATs so that the semi-discrete form is stable. Second, the form of the SATs that are constructed for the LCD equation motivate the form of the SATs we will present for the NS equations in Section 7. A good review of the details of the energy method, as applied to the continuous and the semi-discrete problems, is given by Svärd and Nordström [11]. For a detailed understanding, the following three books are indispensable [76,71,72].

A mathematical model is well-posed if a there exists a solution that is unique, and small perturbations to the data result in small perturbations to the solution, or equivalently the solution is bounded by the data [82]. The energy method is a means of answering the question of boundedness, which can then be used to answer the question of uniqueness. We limit the presentation to PDEs with homogeneous boundary conditions and forcing function; however, the analysis can be extended to the nonhomogeneous case [72]. We need the following definition [72]:

Definition 3. A PDE, on $x \in [0,1]$ with homogeneous data, is called stable if the solution satisfies the estimate[2]

$$\|u(\cdot,t)\| \leqslant Ke^{\alpha t}\|f(\cdot)\|, \tag{55}$$

where f is the initial condition, and K and α do not depend on f. A PDE with homogeneous data is called well-posed if it has a unique smooth solution and is stable [72].

Now we use the energy method in conjunction with the above definition to determine boundary conditions that lead to a well-posed problem for the LCD equation given as

$$\begin{aligned}
&\frac{\partial\mathcal{U}}{\partial t} = -\frac{\partial\mathcal{U}}{\partial x} + \frac{\partial}{\partial x}\left(\mathcal{B}(x)\frac{\partial\mathcal{U}}{\partial x}\right), \quad 0 \leqslant x \leqslant 1,\ t \geqslant 0,\\
&\mathcal{U}(x,0) = f(x),\\
&\alpha_0\mathcal{U}(0,t) - \mathcal{B}(0)\frac{\partial\mathcal{U}}{\partial x}(0,t) = \mathcal{G}_0(t),\\
&\alpha_1\mathcal{U}(1,t) + \mathcal{B}(1)\frac{\partial\mathcal{U}}{\partial x}(1,t) = \mathcal{G}_1(t),
\end{aligned} \tag{56}$$

where $\mathcal{B}(x) \geqslant 0$ and for homogeneous boundary conditions $\mathcal{G}_0(t) = \mathcal{G}_1(t) = 0$. Applying the energy method to (56) gives

$$\frac{\mathrm{d}\|\mathcal{U}(\cdot,t)\|^2}{\mathrm{d}t} = -\mathcal{U}^2\Big|_0^1 + 2\mathcal{B}\mathcal{U}\frac{\partial\mathcal{U}}{\partial x}\Big|_0^1 - 2\left\|\frac{\partial\mathcal{U}}{\partial x}\right\|_{\mathcal{B}}^2, \tag{57}$$

where $\|\cdot\|_{\mathcal{B}}$ is a weighted L_2 norm. Now we use the boundary conditions in (56) to replace the derivative term in (57), which results in

$$\frac{\mathrm{d}\|\mathcal{U}(\cdot,t)\|^2}{\mathrm{d}t} = (-1-2\alpha_1)\mathcal{U}(1,t)^2 + (1-2\alpha_0)\mathcal{U}(0,t)^2 - 2\left\|\frac{\partial\mathcal{U}}{\partial x}\right\|_{\mathcal{B}}^2. \tag{58}$$

Integrating in time, using Leibniz's rule, and applying the initial condition gives

$$\begin{aligned}
\|\mathcal{U}(\cdot,t)\|^2 = \|f(\cdot)\|^2 + \int_0^t &(-1-2\alpha_1)\mathcal{U}(1,\tau)^2\\
&+ (1-2\alpha_0)\mathcal{U}(0,\tau)^2\mathrm{d}\tau - 2\int_0^t \left\|\frac{\partial\mathcal{U}}{\partial x}\right\|_{\mathcal{B}}^2\mathrm{d}\tau.
\end{aligned} \tag{59}$$

For $(-1-2\alpha_1) \leqslant 0$ and $(1-2\alpha_0 \leqslant 0)$ we can replace (59) with

$$\|\mathcal{U}(\cdot,t)\|^2 \leqslant \|f(\cdot)\|^2, \tag{60}$$

which is of the form (55), and so with the restrictions $(-1-2\alpha_1) \leqslant 0$ and $(1-2\alpha_0 \leqslant 0)$ the problem is well-posed.

For the semi-discrete equations we proceed as in the continuous case. First we require the following definition of stability for the semi-discrete form [72]:

Definition 4. A semi-discrete equation, with homogeneous data,[3] is called stable if it has a unique solution, and for all $h \leqslant h_0$, where h is the mesh spacing, there are constants K and α such that, for all $\mathbf{f}$

$$\|\mathbf{u}_h(t)\|_{\mathsf{H}} \leqslant Ke^{\alpha t}\|\mathbf{f}\|_{\mathsf{H}}. \tag{61}$$

With this definition we can now try and determine appropriate SATs, mimicking of the continuous case, such that the semi-discrete LCD equations are stable. Application of the SBP-SAT method to (56) results in

$$\frac{\mathrm{d}\mathbf{u}_h}{\mathrm{d}t} = -\mathsf{D}\mathbf{u}_h + \mathsf{D}_2(\mathsf{B})\mathbf{u}_h + \mathbf{SAT}_0 + \mathbf{SAT}_N, \tag{62}$$

where the additional terms, $\mathbf{SAT}_0$ and $\mathbf{SAT}_1$, are added to weakly impose the boundary conditions at $x = 0$ and $x = 1$, respectively. The continuous case analysis leads us to propose the following type of SATs, where we add the additional parameters $\tau_{0/N}$ so that we have sufficient degrees of freedom such that the SATs lead to a stable semi-discrete form:

$$\begin{aligned}
\mathbf{SAT}_0 &= \tau_0\mathsf{H}^{-1}\mathsf{E}_0(\alpha_0\mathbf{u}_h - \mathsf{D}_b\mathbf{u}_h - g_0\mathbf{e}_0),\\
\mathbf{SAT}_N &= \tau_N\mathsf{H}^{-1}\mathsf{E}_N(\alpha_N\mathbf{u}_h + \mathsf{D}_b\mathbf{u}_h - g_N\mathbf{e}_N),
\end{aligned} \tag{63}$$

where for homogeneous boundary conditions $g_0 = g_N = 0$. Here the general form of the second derivative is used, which can either be

[2] It is assumed that the PDE under consideration has a differential operator and boundary condition operators that are not time dependent, and that the initial condition is compatible with the boundary conditions [72].

[3] Analogous assumptions are made as in the continuous Definition 3.

the application of the first derivative twice, with $\mathsf{R} = 0$ and $\mathsf{D}_b = \mathsf{D}_1$, or the minimum-stencil operator.

Applying the discrete energy method, multiplying (62) by $\mathbf{u}^T\mathsf{H}$ and adding the transpose of this product, results in

$$\begin{aligned}\frac{\mathrm{d}\|\mathbf{u}\|_{\mathsf{H}}^2}{\mathrm{d}t} &= -\mathbf{u}_h^T[\mathsf{Q}+\mathsf{Q}^T]\mathbf{u}_h \\ &- 2\mathbf{u}_h^T\mathsf{D}^T\mathsf{H}\mathsf{D}\mathbf{u}_h - 2\mathbf{u}_h^T\mathsf{R}\mathbf{u}_h + 2\mathbf{u}_h^T\mathsf{E}\mathsf{D}_b\mathbf{u}_h \\ &+ 2\tau_0\mathbf{u}_h^T\mathsf{E}_0(\alpha_0\mathbf{u}_h - \mathsf{D}_b\mathbf{u}_h - g_0\mathbf{e}_0) \\ &+ 2\tau_N\mathbf{u}_h^T\mathsf{E}_N(\alpha_N\mathbf{u}_h + \mathsf{D}_b\mathbf{u}_h - g_N\mathbf{e}_N).\end{aligned} \tag{64}$$

Expanding and rearranging results in

$$\begin{aligned}\frac{\mathrm{d}\|\mathbf{u}\|_{\mathsf{H}}^2}{\mathrm{d}t} = &+2\mathbf{u}_h^T\mathsf{D}^T\mathsf{H}\mathsf{D}\mathbf{u}_h - 2\mathbf{u}_h^T\mathsf{R}\mathbf{u}_h + 2\mathbf{u}_h^T\mathsf{E}\mathsf{D}_b\mathbf{u}_h - 2\tau_0\mathbf{u}_h^T\mathsf{E}_0\mathsf{D}_b\mathbf{u}_h \\ &+ 2\tau_N\mathbf{u}_h^T\mathsf{E}_N\mathsf{D}_b\mathbf{u}_h\left(u_0^2 - 2\alpha_0\tau_0u_0^2 - 2\tau_0u_0g_0\right) \\ &+ \left(-u_N^2 + 2\alpha_N\tau_Nu_N^2 - 2\tau_Nu_Ng_N\right).\end{aligned} \tag{65}$$

We see that if $\tau_0 = 1$ and $\tau_N = 1$ the term $2\mathbf{u}_h^T\mathsf{E}\mathsf{D}_b\mathbf{u}_h$ cancels, giving

$$\begin{aligned}&\frac{\mathrm{d}\|\mathbf{u}\|_{\mathsf{H}}^2}{\mathrm{d}t} = -2\mathbf{u}_h^T\mathsf{D}^T\mathsf{H}\mathsf{D}\mathbf{u}_h - 2\mathbf{u}_h^T\mathsf{R}\mathbf{u}_h \\ &\left(u_0^2 - 2\alpha_0u_0^2 - 2u_0g_0\right) + \left(-u_1^2 - 2\alpha_1u_1^2 + 2u_1g_1\right).\end{aligned} \tag{66}$$

Before proceeding, notice that the term $-2\mathbf{u}_h^T\mathsf{D}^T\mathsf{H}\mathsf{D}\mathbf{u}_h \approx \|\frac{\partial\mathcal{U}}{\partial x}\|_{\mathsf{B}}^2$ and that the term $-2\mathbf{u}_h^T\mathsf{R}\mathbf{u}_h$ is negative semidefinite. Now we choose α_0 and α_N such that we can construct an energy estimate and demonstrate that these parameter values coincide with the continuous case. Integrating (66) in time, and assuming homogeneous boundary conditions results in

$$\begin{aligned}\|\mathbf{u}_h(t)\|_{\mathsf{H}}^2 = \|\mathbf{f}\|_{\mathsf{H}}^2 - 2\int_0^t \mathbf{u}_h^T\mathsf{D}^T\mathsf{H}\mathsf{D}\mathbf{u}_h + \mathbf{u}_h^T\mathsf{R}\mathbf{u}_h\mathrm{d}\tau \\ + \int_0^t (-1-2\alpha_N)u_N^2(\tau) + (1-2\alpha_0)u_0^2(\tau)\mathrm{d}\tau,\end{aligned} \tag{67}$$

just as in the continuous case, for $(-1-2\alpha_1) \leqslant 0$ and $(1-2\alpha_0) \leqslant 0$ we construct the estimate

$$\|\mathbf{u}_h(t)\|_{\mathsf{H}}^2 \leqslant \|\mathbf{f}\|_{\mathsf{H}}^2, \tag{68}$$

and we conclude that the problem (62) is stable. For simplicity, we have chosen to deal with definitions of well-posedness and stability applicable to PDEs with homogeneous boundary conditions and with differential operators that do not vary in time. Moreover, we have assumed that the initial condition is compatible with the homogeneous boundary conditions. The analysis can be extended to problems without these simplifications, such as non-homogeneous boundary conditions; this brings in various definitions of stability [76,71,72].

7. Practical aspects

7.1. Application to the Navier–Stokes equations

In this section we describe the details necessary to implement SBP-SAT schemes for the Euler and NS equations. In curvilinear coordinates, (ξ,η,ζ), the NS equations are given by

$$\frac{\partial\widehat{\mathbf{Q}}}{\partial\tau} + \frac{\partial\widehat{\mathbf{E}}}{\partial\xi} + \frac{\partial\widehat{\mathbf{F}}}{\partial\eta} + \frac{\partial\widehat{\mathbf{G}}}{\partial\zeta} = 0, \tag{69}$$

where

$$\begin{aligned}\widehat{\mathbf{E}} &= \widehat{\mathbf{E}}_I - \frac{1}{Re}\widehat{\mathbf{E}}_V, \\ \widehat{\mathbf{F}} &= \widehat{\mathbf{F}}_I - \frac{1}{Re}\widehat{\mathbf{F}}_V, \\ \widehat{\mathbf{G}} &= \widehat{\mathbf{G}}_I - \frac{1}{Re}\widehat{\mathbf{G}}_V,\end{aligned} \tag{70}$$

where the subscripts I and V refer to inviscid and viscous, $\widehat{\mathbf{Q}}$ is the vector of conserved variables scaled by the metric Jacobian, the vectors $\widehat{\mathbf{E}}_i, \widehat{\mathbf{F}}_i, \widehat{\mathbf{G}}_i$, are the inviscid flux vectors in curvilinear coordinates, and the vectors $\widehat{\mathbf{E}}_V, \widehat{\mathbf{F}}_V$, and $\widehat{\mathbf{G}}_V$ are the viscous flux vectors in curvilinear coordinates (for the form of the various vectors see [83]). Application of the one-dimensional SBP operators to systems of equations and or multiple dimensions necessitates the use of Kronecker products. Given a $m \times n$ matrix C and a $p \times q$ matrix B the Kronecker product $\mathsf{C} \otimes \mathsf{B}$ is the $mp \times nq$ matrix defined by

$$\mathsf{C} \otimes \mathsf{B} = \begin{bmatrix} \mathsf{C}(1,1)\mathsf{B} & \dots & \mathsf{C}(1,n)\mathsf{B} \\ \vdots & & \\ \mathsf{C}(m,1)\mathsf{B} & \dots & \mathsf{C}(m,n)\mathsf{B} \end{bmatrix}. \tag{71}$$

The various necessary operators are given by

$$\begin{aligned}&\widehat{\mathsf{H}} = \mathsf{H}_\zeta \otimes \mathsf{H}_\eta \otimes \mathsf{H}_\xi \otimes \mathsf{I}_5, \\ &\widehat{\mathsf{H}}_\xi = \mathsf{I}_\zeta \otimes \mathsf{I}_\eta \otimes \mathsf{H}_\xi \otimes \mathsf{I}_5, \quad \widehat{\mathsf{D}}_\xi = \mathsf{I}_\zeta \otimes \mathsf{I}_\eta \otimes \mathsf{D}_\xi \otimes \mathsf{I}_5, \\ &\widehat{\mathsf{H}}_\eta = \mathsf{I}_\zeta \otimes \mathsf{H}_\eta \otimes \mathsf{I}_\xi \otimes \mathsf{I}_5, \quad \widehat{\mathsf{D}}_\eta = \mathsf{I}_\zeta \otimes \mathsf{D}_\eta \otimes \mathsf{I}_\xi \otimes \mathsf{I}_5, \\ &\widehat{\mathsf{H}}_\zeta = \mathsf{H}_\zeta \otimes \mathsf{I}_\eta \otimes \mathsf{I}_\xi \otimes \mathsf{I}_5, \quad \widehat{\mathsf{D}}_\zeta = \mathsf{D}_\zeta \otimes \mathsf{I}_\eta \otimes \mathsf{I}_\xi \otimes \mathsf{I}_5,\end{aligned} \tag{72}$$

where H_ξ is the H-norm matrix in the ξ direction, $\mathsf{D}_\xi = \mathsf{H}_\xi^{-1}\mathsf{Q}_\xi$ is the SBP operator for the first derivative in the ξ direction and similarly for the other operators. Furthermore, the matrices $\mathsf{I}_\xi, \mathsf{I}_\eta$, and I_ζ are identity matrices whose dimensions match the number of nodes in the corresponding coordinate direction. The second derivative with variable coefficients is represented in a generic fashion:

$$\begin{aligned}&\widehat{\mathsf{D}}_{2\xi}(\mathsf{B}) = \mathsf{I}_\zeta \otimes \mathsf{I}_\eta \otimes \mathsf{D}_{2\xi}(\mathsf{B}) \otimes \mathsf{I}_5, \\ &\widehat{\mathsf{D}}_{2\eta}(\mathsf{B}) = \mathsf{I}_\zeta \otimes \mathsf{D}_{2\eta}(\mathsf{B}) \otimes \mathsf{I}_\xi \otimes \mathsf{I}_5, \\ &\widehat{\mathsf{D}}_{2\zeta}(\mathsf{B}) = \mathsf{D}_{2\zeta}(\mathsf{B}) \otimes \mathsf{I}_\eta \otimes \mathsf{I}_\xi \otimes \mathsf{I}_5.\end{aligned} \tag{73}$$

The various cross-derivative terms, for example in the ξ-direction, are given by

$$\begin{aligned}&\widehat{\mathsf{D}}_{\xi,\eta}(\mathsf{B}) = \widehat{\mathsf{D}}_\xi\mathsf{B}\widehat{\mathsf{D}}_\eta, \\ &\widehat{\mathsf{D}}_{\xi,\zeta}(\mathsf{B}) = \widehat{\mathsf{D}}_\xi\mathsf{B}\widehat{\mathsf{D}}_\zeta.\end{aligned} \tag{74}$$

Finally, numerical dissipation is applied in the three directions by the following operators (see Section 7.2):

$$\begin{aligned}&\widehat{\mathsf{A}}_{\mathsf{D},\xi} = \mathsf{H}_\zeta^{-1} \otimes \mathsf{H}_\eta^{-1} \otimes \mathsf{A}_{\mathsf{D},\xi} \otimes \mathsf{I}_5, \\ &\widehat{\mathsf{A}}_{\mathsf{D},\eta} = \mathsf{H}_\zeta^{-1} \otimes \mathsf{A}_{\mathsf{D},\eta} \otimes \mathsf{H}_\xi^{-1} \otimes \mathsf{I}_5, \\ &\widehat{\mathsf{A}}_{\mathsf{D},\zeta} = \mathsf{A}_{\mathsf{D},\zeta} \otimes \mathsf{H}_\eta^{-1} \otimes \mathsf{H}_\xi^{-1} \otimes \mathsf{I}_5.\end{aligned} \tag{75}$$

For multi-block domain decompositions, the discretization is applied block by block, while the boundary and interface conditions are enforced using SATs, the form of which will be discussed shortly. With the previously defined operators the semi-discrete equations, with boundary and interface conditions enforced with SATs, on a single block are given as follows:

$$\begin{aligned}\frac{\mathrm{d}\widehat{\mathbf{Q}}_h}{\mathrm{d}t} = &-\widehat{\mathsf{D}}_\xi\widehat{\mathbf{E}}_{I,h} - \widehat{\mathsf{D}}_\eta\widehat{\mathbf{F}}_{I,h} - \widehat{\mathsf{D}}_\zeta\widehat{\mathbf{G}}_{I,h} \\ &+ \frac{1}{Re}\left(\widehat{\mathsf{D}}_{2\xi}\left(\mathsf{B}_{\widehat{\mathbf{E}}_{V,h}}\right) + \widehat{\mathsf{D}}_{\xi,\eta}\left(\mathsf{B}_{\widehat{\mathbf{E}}_{V,h}}\right) + \widehat{\mathsf{D}}_{\xi,\zeta}\left(\mathsf{B}_{\widehat{\mathbf{E}}_{V,h}}\right)\right)\mathbf{v}_h \\ &+ \frac{1}{Re}\left(\widehat{\mathsf{D}}_{\eta,\xi}\left(\mathsf{B}_{\widehat{\mathbf{F}}_{V,h}}\right) + \widehat{\mathsf{D}}_{2\eta}\left(\mathsf{B}_{\widehat{\mathbf{F}}_{V,h}}\right) + \widehat{\mathsf{D}}_{\eta,\zeta}\left(\mathsf{B}_{\widehat{\mathbf{F}}_{V,h}}\right)\right)\mathbf{v}_h \\ &+ \frac{1}{Re}\left(\widehat{\mathsf{D}}_{\zeta,\xi}\left(\mathsf{B}_{\widehat{\mathbf{G}}_{V,h}}\right) + \widehat{\mathsf{D}}_{\zeta,\eta}\left(\mathsf{B}_{\widehat{\mathbf{G}}_{V,h}}\right) + \widehat{\mathsf{D}}_{2\zeta}\left(\mathsf{B}_{\widehat{\mathbf{G}}_{V,h}}\right)\right)\mathbf{v}_h \\ &+ \widehat{\mathsf{A}}_{\mathsf{D},\xi}\widehat{\mathbf{Q}}_h + \widehat{\mathsf{A}}_{\mathsf{D},\eta}\widehat{\mathbf{Q}}_h + \widehat{\mathsf{A}}_{\mathsf{D},\zeta}\widehat{\mathbf{Q}}_h + \mathbf{SAT}_{\xi,0} + \mathbf{SAT}_{\xi,N} \\ &+ \mathbf{SAT}_{\eta,0} + \mathbf{SAT}_{\eta,N} + \mathbf{SAT}_{\zeta,0} + \mathbf{SAT}_{\zeta,N},\end{aligned} \tag{76}$$

where $\mathbf{v}_h$, at a node in the grid, has components $[0, u, v, w, a^2]$, and, for example $\mathsf{B}_{\widehat{\mathbf{E}}_{V,h}}$ is a diagonal matrix whose components are determined from $\widehat{\mathbf{E}}_{V,h}$.

It is worth mentioning that the formalism presented here, with Kronecker products, is a theoretical tool used for the purpose of analysis. In practice, these products are never formed and the one-dimensional operators are applied directly at nodes in the grid.

7.1.1. SATs for the Navier–Stokes equations

In Section 6 we demonstrated how to construct SATs by first using the energy method to determine well-posed data for a given PDE. The SATs were then constructed to mimic the well-posed boundary conditions from the continuous case. The discrete energy method was then applied to the semi-discrete form to specify penalty parameters such that the resultant semi-discrete form is stable. The energy method is typically applied to constant-coefficient problems that can be symmetrized. In this way, the difficulty of the analysis is greatly reduced. However, the compressible Euler and NS equations are nonlinear PDEs. Nevertheless, for certain classes of PDEs, which the Euler and NS equations belong to, and with some restrictions on the solution, it can be shown that the constant-coefficient analysis is sufficient [76].

Here we present some example SATs for the compressible Euler and NS equations. For an excellent summary and discussion of a large number of SATs for the NS equations see [84]. The SATs are divided into inviscid SATs and viscous SATs.

We consider the SATs in the ξ direction. The inviscid SATs have generic form [27]

$$\begin{aligned}\mathbf{SAT}_0 &= -\widehat{\mathsf{H}}_{\xi}^{-1}\widehat{\mathsf{E}}_{0,\xi}\left(\widehat{\mathsf{A}}_{\xi}^{+}\right)\left(\widehat{\mathbf{Q}}-\widehat{\mathbf{Q}}_{\text{target}}\right),\\ \mathbf{SAT}_N &= \widehat{\mathsf{H}}_{\xi}^{-1}\widehat{\mathsf{E}}_{N,\xi}\left(\widehat{\mathsf{A}}_{\xi}^{-}\right)\left(\widehat{\mathbf{Q}}-\widehat{\mathbf{Q}}_{\text{target}}\right),\end{aligned} \tag{77}$$

where $\widehat{\mathsf{E}}_{0,\xi} = \mathsf{I}_{\zeta}\otimes\mathsf{I}_{\eta}\otimes\mathsf{E}_{0,\xi}\otimes\mathsf{I}_5$, $\widehat{\mathsf{E}}_{N,\xi} = \mathsf{I}_{\zeta}\otimes\mathsf{I}_{\eta}\otimes\mathsf{E}_{N,\xi}\otimes\mathsf{I}_5$, and $\widehat{\mathsf{A}}_{\xi}^{\pm} = \text{diag}\left[\mathsf{A}_{\xi,(0,0,0)}^{\pm},\ldots,\mathsf{A}_{\xi,(N,N,N)}^{\pm}\right]$ is a block diagonal matrix constructed from the flux Jacobian, in the ξ direction, at node (j,k,m), as

$$\mathsf{A}_{\xi_{(i,j,k)}}^{\pm} = \frac{\mathsf{A}_{\xi}\pm|\mathsf{A}|_{\xi}}{2}, \quad \mathsf{A}_{\xi} = \frac{\partial\widehat{\mathbf{E}}_I}{\partial\widehat{\mathbf{Q}}}, \tag{78}$$

while $\widehat{\mathbf{Q}}_{\text{target}}$ depends on the type of SAT. For example, at block interfaces $\widehat{\mathbf{Q}}_{\text{target}}$ takes the value of the solution in the adjoining block. For the far-field $\widehat{\mathbf{Q}}_{\text{target}} = \widehat{\mathbf{Q}}_{\infty}$, where $\widehat{\mathbf{Q}}_{\infty}$ is the far-field state. At a wall boundary, the inviscid SAT has $\widehat{\mathbf{Q}}_{\text{target}}$ that enforces flow tangency.

In analogy with the derivation for the LCD equation, two types of SATs are typically implemented for the viscous portion of the NS equations. The first weakly enforces continuity of the viscous fluxes and has the general form [29]:

$$\begin{aligned}\mathbf{SAT}_0 &= \frac{1}{Re}\widehat{\mathsf{H}}_{\xi}^{-1}\widehat{\mathsf{E}}_{0,\xi}\left(\widehat{\mathbf{E}}_{v}-\widehat{\mathbf{E}}_{\text{v,target}}\right),\\ \mathbf{SAT}_N &= \frac{-1}{Re}\widehat{\mathsf{H}}_{\xi}^{-1}\widehat{\mathsf{E}}_{N,\xi}\left(\widehat{\mathbf{E}}_{v}-\widehat{\mathbf{E}}_{\text{v,target}}\right).\end{aligned} \tag{79}$$

This form of viscous SAT is used for the far-field, where $\widehat{\mathbf{E}}_{\text{v,target}} = 0$, and at mesh block interfaces, where $\widehat{\mathbf{E}}_{\text{v,target}}$ is the value of $\widehat{\mathbf{E}}_{v}$ from the neighboring block.

The second viscous SAT weakly enforces continuity of the conserved variables. The mesh block interface SAT for the viscous portion of the NS equations, besides requiring (79), necessitates a SAT of the form (see Nordström et al. [29] for the derivation):

$$\mathbf{SAT}_{v,\text{int}} = \frac{\sigma_{v2}}{JRe}\widehat{\mathsf{H}}_{\xi}^{-1}\widehat{\mathsf{E}}_{0,\xi}\widehat{\mathsf{B}}_{\xi,\text{int}}\left(\widehat{\mathbf{Q}}_{v}-\widehat{\mathbf{Q}}_{\text{target}}\right), \tag{80}$$

where, for stability $\sigma_{v2}\leqslant\frac{1}{2}$, and $\widehat{\mathsf{B}}_{\xi,\text{int}} = \text{diag}\left[\mathsf{B}_{\xi,\text{int},(0,0,0)},\ldots,\mathsf{B}_{\xi,\text{int},(N,N,N)}\right]$ is a block-diagonal matrix; the form of $\mathsf{B}_{\xi,\text{int},(j,k,m)}$ and the full interface SAT term are given in Appendix C.

As a final example, to enforce a no-slip adiabatic wall boundary condition, in addition to the Euler SAT, the following is added [6] (see Svärd and Nordström for the derivation [28]):

$$\mathbf{SAT}_{v,\text{wall1}} = \frac{\sigma_W}{Re}\widehat{\mathsf{H}}_{\xi}^{-1}\widehat{\mathsf{E}}_{0,\xi}\left(\widehat{\mathbf{Q}}_{v}-\widehat{\mathbf{Q}}_{\text{target}}\right), \tag{81}$$

where

$$\sigma_W \leqslant -\frac{\mu\left(\xi_x^2+\xi_y^2+\xi_z^2\right)}{2J\rho}\max\left(\frac{\gamma}{Pr},\frac{5}{3}\right), \tag{82}$$

and

$$\widehat{\mathbf{Q}}_{\text{target}} = \left[\rho, 0, 0, 0, \frac{\rho T_2}{\gamma(\gamma-1)}\right]^T. \tag{83}$$

In the above approach to the wall-boundary SAT, it is assumed that the grid-lines are perpendicular to the surface. Many times this is not the case; an alternative approach is discussed by Osusky and Zingg [6].

For outflow SATs, Osusky [84] provides a procedure based on projecting a pyramid structure behind the aerodynamic object, to identify points in the outflow (for an alternative treatment see Svärd and Nordström [28]). Finally, SAT terms have been constructed for the Spalart–Allmaras one-equation turbulence model [6].

7.2. Numerical dissipation

Nonlinear terms in a PDE can result in the generation of high-frequency modes. Centred-difference FD methods have no mechanism to remove such modes. Thus, it is necessary to augment SBP methods with some mechanism to damp these under-resolved modes. There are various means of doing so, such as modifying the SBP operators so that they are upwind operators [85] or using high-order filters (see for example [42]). Typically though, SBP schemes are augmented with artificial dissipation. The added dissipation model must be carefully constructed such that the stability properties of the scheme are not lost. Consider a dissipation model constructed as $\mathsf{A}_\mathsf{D} = \mathsf{H}^{-1}\widetilde{\mathsf{A}}_\mathsf{D}$, where $\widetilde{\mathsf{A}}_\mathsf{D}$ is symmetric, added to a stable semi-discrete form:

$$\frac{\mathrm{d}\mathbf{u}_\mathrm{h}}{\mathrm{d}t} = \mathsf{H}^{-1}\mathsf{P}\mathbf{u}_\mathrm{h} + \mathsf{H}^{-1}\widetilde{\mathsf{A}}_\mathsf{D}\mathbf{u}_\mathrm{h}, \tag{84}$$

where $\mathsf{H}^{-1}\mathsf{P}\mathbf{u}_h$ represents the SBP discretization of a PDE. Assuming a stable semi-discrete system of equations, it is sufficient to examine the stability of the dissipation model itself. Applying the discrete energy method to (84) and only considering the contribution of the dissipation model results in

$$\frac{\mathrm{d}\|\mathbf{u}_\mathrm{h}\|_\mathsf{H}^2}{\mathrm{d}t} = 2\mathbf{u}_\mathrm{h}^T\widetilde{\mathsf{A}}_\mathsf{D}\mathbf{u}_\mathrm{h}, \tag{85}$$

and if $\widetilde{\mathsf{A}}_\mathsf{D}$ is negative semi-definite, then an energy estimate exists. This means that, besides being of the order of the discretization error or smaller, it is sufficient to consider dissipation models constructed as $\mathsf{H}^{-1}\widetilde{\mathsf{A}}_\mathsf{D}$, where $\widetilde{\mathsf{A}}_\mathsf{D}$ is negative semi-definite.

Mattsson, Svärd and Nordström [37] were the first to construct dissipation models specifically for SBP schemes. For operators of order $2p$ on the interior, their dissipation model has the following form:

$$\mathsf{A}_\mathsf{D} = -\widetilde{\mathsf{H}}^{-1}\widetilde{\mathsf{D}}_p^T\mathsf{M}\widetilde{\mathsf{D}}_p. \tag{86}$$

The tilde denotes that the matrices are undivided difference approximations, and $\widetilde{\mathsf{H}} = \Delta x \mathsf{H}$. Furthermore, $\mathsf{D}_p = h^{-p}\widetilde{\mathsf{D}}_p$ and is an approximation of $\frac{d^p}{dx^p}$ of minimum width. Taking M as the identity matrix results in a dissipation model that adds components of order p near the boundary, which is sufficient for diagonal-norm operators. However, block-norm operators have near-boundary point operators of order $2p-1$ and with M an identity matrix (86) destroys the additional accuracy accrued by using block-norm operators. To remedy this issue, Mattson, Svärd and Nordström [37] proposed constructing the diagonal M with entries as the restriction of a positive, sufficiently smooth function varying from order of $\mathcal{O}(h^{\hat{p}})$ near the boundaries to unity on the interior, where $\hat{p} \geqslant 1$. In a subsequent paper, Nordström [38] applied similar ideas to construct dissipation models tailored to linear hyperbolic problems with variable coefficients. Also, while deriving optimized first derivative operators, Diener et al. [31] constructed dissipation operators up to $p = 5$, based on the work of [37].

Hicken and Zingg [9] and Osusky and Zingg [6] use a generalized version of the classical scalar [86] and matrix [87] dissipation models to stabilize their SBP-SAT discretization of the Euler and NS equations.

7.3. Curvilinear coordinates

SBP approximations to the first derivative are one-dimensional operators on evenly spaced nodal distributions. In order to apply the SBP method to more general problems, a semi-structured, or multi-block, approach is used, in which the 2-dimensional or 3-dimensional domain is subdivided into quadrilateral or hexahedral subdomains, respectively. Within each subdomain, the PDE is transformed to a curvilinear coordinate system. In this way, an SBP scheme is applied to an orthogonal, evenly spaced grid with respect to the curvilinear coordinate system. Moreover, multidimensional problems are discretized by application of the one-dimensional SBP operators using Kronecker products. In curvilinear coordinates, only diagonal-normed SBP operators are still provably stable (see [45]) and the present discussion is thus limited to diagonal-norm SBP operators.

Transformation of a hyperbolic system of PDEs, for example the Euler equations, results in terms called the transformation invariants, which are analytically satisfied by virtue of the commutative property of the derivative [83]. These are also known as the geometric conservation laws (GCL), i.e. the surface conservation law and the volume conservation law. In order to remain conservative, the computed grid metrics must satisfy these laws within roundoff error.

Vinokur and Yee [88] derived a coordinate invariant discrete representation of the grid metrics, which satisfies the surface conservation law to within roundoff error regardless of the mapping. Deng et al. [89] showed that the coordinate invariant form of the grid metrics has smaller discretization error versus alternatives. The scaled form of the coordinate invariant grid metrics proposed by Vinokur and Yee [88] and the Jacobian are:

$$\begin{aligned}
\hat{\xi}_{\mathbf{x}} &= J^{-1}[\xi_x, \xi_y, \xi_z]^T = \frac{1}{2}[(\mathbf{r}_\eta \times \mathbf{r})_\zeta - (\mathbf{r}_\zeta \times \mathbf{r})_\eta],\\
\hat{\boldsymbol{\eta}}_{\mathbf{x}} &= J^{-1}[\eta_x, \eta_y, \eta_z]^T = \frac{1}{2}[(\mathbf{r}_\zeta \times \mathbf{r})_\xi - (\mathbf{r}_\xi \times \mathbf{r})_\zeta],\\
\hat{\zeta}_{\mathbf{x}} &= J^{-1}[\zeta_x, \zeta_y, \zeta_z]^T = \frac{1}{2}[(\mathbf{r}_\xi \times \mathbf{r})_\eta - (\mathbf{r}_\eta \times \mathbf{r})_\xi],\\
J^{-1} &= \mathbf{r}_\xi \cdot (\mathbf{r}_\eta \times \mathbf{r}_\zeta) = \mathbf{r}_\eta \cdot (\mathbf{r}_\zeta \times \mathbf{r}_\xi) = \mathbf{r}_\zeta \cdot (\mathbf{r}_\xi \times \mathbf{r}_\eta),
\end{aligned} \tag{87}$$

where $\mathbf{r}_m = x_m\mathbf{i} + y_m\mathbf{j} + z_m\mathbf{k}$, with $m \in [\xi, \eta, \zeta]$. The derivatives x_m and terms such as $(\mathbf{r}_\eta \times \mathbf{r})_\zeta$ are computed using the same SBP operator for the first derivative as for the discretization of the PDE. Instead of the above formulation for the Jacobian, Deng et al. [89] derived a means of computing the Jacobian that automatically satisfies the volume conservation law and is given by

$$J^{-1} = \frac{1}{3}\left[\left(\mathbf{r} \cdot \hat{\xi}_{\mathbf{x}}\right)_\xi + (\mathbf{r} \cdot \hat{\boldsymbol{\eta}}_{\mathbf{x}})_\eta + \left(\mathbf{r} \cdot \hat{\zeta}_{\mathbf{x}}\right)_\zeta\right]. \tag{88}$$

8. Variational interpretation of SBP-SAT schemes

When Kreiss and Scherer introduced SBP finite-difference operators, their objective was to mimic the stability properties of Galerkin finite-element methods (FEMs); however, by mimicking this one property, SBP discretizations inherit other useful properties from the Galerkin FEM, including superconvergent functionals and error estimates. At the heart of these properties is the variational interpretation of SBP finite-difference schemes and the concept of dual consistency, both of which we discuss in this section.

8.1. SBP quadrature and the variational interpretation

To illustrate the variational interpretation of SBP discretizations, we consider the one-dimensional, steady, scalar linear convection equation. Let $\Omega = [x_L, x_R]$ denote the domain. Then our model PDE is

$$\begin{aligned}
&\frac{d}{dx}(\mathcal{A}\mathcal{U}) - \mathcal{F} = 0, \qquad \forall x \in \Omega,\\
&\mathcal{U} - \mathcal{U}_L = 0, \qquad x = 0.
\end{aligned} \tag{89}$$

where $\mathcal{A}(x) > 0$ is the velocity field, $\mathcal{U}$ is the solution, and the function $\mathcal{F}(x) \in \mathsf{L}^2(\Omega)$ is the source. As in earlier sections, the boundary value at the inlet $x = x_L$ is denoted $\mathcal{U}_L$.

The variational, or weak, form of (89) is obtained by forming the integral inner product of the PDE with arbitrary test functions and insisting that all such products vanish. Formally, we seek a solution $\mathcal{U} \in \mathsf{W}^1$ such that

$$\int_\Omega \mathcal{V}\frac{d}{dx}(\mathcal{A}\mathcal{U})\,dx - \int_\Omega \mathcal{V}\mathcal{F}\,dx = 0, \qquad \forall \mathcal{V} \in \mathsf{W}^1.$$

Applying integration by parts to the above, the variational statement can be written succinctly as

$$\mathcal{R}(\mathcal{V}, \mathcal{U}) = 0, \qquad \forall \mathcal{V} \in \mathsf{W}^1, \tag{90}$$

where we have introduced the bilinear form

$$\mathcal{R}(\mathcal{V}, \mathcal{U}) = -\int_\Omega \left(\mathcal{A}\frac{d\mathcal{V}}{dx}\right)\mathcal{U}\,dx + (\mathcal{V}\mathcal{A}\mathcal{U})|_{x=1} - (\mathcal{V}\mathcal{A}\mathcal{U}_L)|_{x=0} - \int_\Omega \mathcal{V}\mathcal{F}\,dx.$$

We have replaced $\mathcal{U}(0)$ with its boundary value $\mathcal{U}_L$, so the boundary condition is imposed weakly in the variational statement.

The space W^1 denotes the Sobelev space of functions whose derivatives are square integrable on $\Omega = [0, 1]$. In general, the test functions $\mathcal{V}$ and solution $\mathcal{U}$ will come from distinct spaces; for example, the space for $\mathcal{U}$ will satisfy the boundary conditions, while the space for $\mathcal{V}$ will satisfy the corresponding homogeneous boundary conditions. Here, since weakly imposed boundary conditions are used, the spaces can be equivalent.

To connect the weak form of the PDE to its SBP-SAT discretization, we make use of the fact that the SBP norm, H, defines a quadrature.

Theorem 1. *Let H be a full, restricted-full, or diagonal weight matrix from an SBP first-derivative operator $\mathsf{D}_1 = (\mathsf{H}^{-1}\mathsf{Q})$, which is a $2p$-order-accurate approximation to d/dx in the interior. Then the H matrix constitutes a $2p$-order-accurate quadrature for integrands $\mathcal{U} \in C^{2p}[0, 1]$. In other words*

$$1^T \mathsf{H}\mathbf{u} = \int_0^1 \mathcal{U}\,dx + \mathrm{O}(h^{2p}),$$

where $\mathbf{u}$ *is the restriction of* $\mathcal{U}$ *to the grid. Moreover, if* $\mathcal{V}\frac{d\mathcal{U}}{dx} \in C^{2p}[0,1]$, *then*

$$\mathbf{v}^T \mathsf{H}(\mathsf{D}_1\mathbf{u}) = \int_0^1 \mathcal{V}\frac{d\mathcal{U}}{dx}\,dx + \mathrm{O}(h^{2p}),$$

where $\mathbf{v}$ *is the restriction of* $\mathcal{V}$ *to the grid.*

See [48] for the proof. This result is somewhat surprising, because the accuracy of H as a quadrature rule is not explicitly part of the definition of the SBP operators. While reference [48] was, to the best of our knowledge, the first documented evidence and proof of this property, we have since come to learn that practitioners at Uppsala were also aware that H defines a quadrature rule [90].

The quadrature defined by H allows us to approximate the integral inner product using the H-inner product:

$$\int_0^1 \mathcal{V}\mathcal{U}\,dx = \mathbf{v}^T \mathsf{H}\mathbf{u} + \mathrm{O}(h^{2p}).$$

Consequently, we can discretize the bilinear form using the SBP operator and its norm as follows:

$$\begin{aligned}\mathcal{R}(\mathcal{V},\mathcal{U}) = &-(\mathsf{A}\mathsf{D}_1\mathbf{v})^T\mathsf{H}\mathbf{u} + \mathbf{v}^T\mathbf{e}_N\mathbf{e}_N^T\mathsf{A}\mathbf{u} - \mathbf{v}^T\mathbf{e}_0\mathbf{e}_0^T\mathsf{A}(\mathbf{e}_0\mathcal{U}_L)\\ &- \mathbf{v}^T\mathsf{H}\mathbf{f} + \mathrm{O}(h^{2p}),\end{aligned} \tag{91}$$

where

$$\begin{aligned}&\mathsf{A} = \mathrm{diag}(\mathcal{A}(x_0), \mathcal{A}(x_1), \ldots, \mathcal{A}(x_N)),\\ &\mathbf{f} = [\,\mathcal{F}(x_0), \quad \mathcal{F}(x_1), \quad \cdots, \quad \mathcal{F}(x_N)\,]^T,\end{aligned} \tag{92}$$

and the truncation error follows for sufficiently smooth functions. The $\mathbf{u}$ and $\mathbf{v}$ vectors appearing in the discretized bilinear form—the right-hand side of (91) — are restrictions of the (infinite-dimensional) solution and test function to the grid. The final discrete problem is obtained by replacing $\mathbf{u}$ and $\mathbf{v}$ by $\mathbf{u}_h$ and $\mathbf{v}_h$, respectively, and requiring the discretized bilinear form to vanish for all $\mathbf{v}_h \in \mathbb{R}^{N+1}$. That is, we seek $\mathbf{u}_h \in \mathbb{R}^{N+1}$ such that

$$R_h(\mathbf{v}_h, \mathbf{u}_h) = 0, \qquad \forall \mathbf{v}_h \in \mathbb{R}^{N+1},$$

where

$$\begin{aligned}R_h(\mathbf{v}_h, \mathbf{u}_h) \ \equiv &-(\mathsf{A}\mathsf{D}_1\mathbf{v}_h)^T\mathsf{H}\mathbf{u}_h + \mathbf{v}_h^T\mathbf{e}_N\mathbf{e}_N^T\mathsf{A}\mathbf{u}_h\\ &- \mathbf{v}_h^T\mathbf{e}_0\mathbf{e}_0^T\mathsf{A}(\mathcal{U}_L\mathbf{e}_0) - \mathbf{v}_h^T\mathsf{H}\mathbf{f}.\end{aligned} \tag{93}$$

To make the final connection to SBP-SAT discretizations, we rearrange the discrete bilinear form making use of the properties of the SBP operator: we essential undo the integration by parts in discrete space.

$$R_h(\mathbf{v}_h, \mathbf{u}_h) = \mathbf{v}_h^T\mathsf{H}\left[\mathsf{D}_1(\mathsf{A}\mathbf{u}_h) - f + \mathsf{H}^{-1}\mathbf{e}_0\mathbf{e}_0^T\mathsf{A}(\mathbf{u}_h - \mathcal{U}_L\mathbf{e}_0)\right].$$

Since the discrete residual must vanish for all $\mathbf{v}_h \in \mathbb{R}^{N+1}$, it follows that the quantity in the brackets must vanish; in other words

$$\mathsf{D}_1(\mathsf{A}\mathbf{u}_h) = f - \mathsf{H}^{-1}\mathbf{e}_0\mathbf{e}_0^T\mathsf{A}(\mathbf{u}_h - \mathcal{U}_L\mathbf{e}_0), \tag{94}$$

which we recognize as the SBP-SAT discretization of the PDE (89).

Thus, SBP-SAT discretizations can be regarded as a discretized variational statement. For diagonal-norm operators, this interpretation can be easily extended to SBP-SAT discretizations of nonlinear systems of PDEs on multi-block curvilinear domains in 2 and 3-dimensions; see, for example, [47].

This intimate connection with the weak form of the PDE allows several results from the theory of Galerkin FEMs to be generalized to SBP-SAT discretizations, which we highlight below.

8.2. Functional superconvergence

One consequence of the variational interpretation of SBP-SAT schemes is its impact on integral functionals that depend on the numerical solution. Let $\mathcal{G} \in C^{2p}$ and consider the functional

$$\mathcal{J}(\mathcal{U}) = \int_0^1 \mathcal{G}\mathcal{U}\,dx + \alpha(\mathcal{A}\mathcal{U})|_{x=1}.$$

The integral in this functional can be approximated by the SBP norm:

$$J_h(\mathbf{u}) = \mathbf{g}^T\mathsf{H}\mathbf{u} + \alpha\mathbf{e}_N^T\mathsf{A}\mathbf{u}, \tag{95}$$

where

$$\mathbf{g} = [\,\mathcal{G}(x_0), \quad \mathcal{G}(x_1), \quad \ldots, \quad \mathcal{G}(x_N)\,]^T.$$

In light of Theorem 1, we have

$$\mathcal{J}(\mathcal{U}) - J_h(\mathbf{u}) = \mathrm{O}(h^{2p}).$$

This result is not useful in practice, because we do not have the exact solution $\mathcal{U}$ or its restriction $\mathbf{u}$. What is useful, however, is that the accuracy of the functional remains $\mathrm{O}(h^{2p})$ when $\mathbf{u}$ is replaced with the discrete solution, $\mathbf{u}_h$.

Theorem 2. *Let* $\mathcal{U}$ *denote the solution of the PDE (89), and assume that the source* $\mathcal{F}$ *and velocity* $\mathcal{A}$ *are such that* $\mathcal{U} \in C^{2p}[0,1]$. *Further, let* $\mathbf{u}_h$ *denote the solution to the SBP-SAT discretization (94) using a diagonal-norm SBP operator. If* $\mathcal{G} \in C^{2p}$, *then the discrete functional* $J_h(\mathbf{u}_h)$, *defined above, is a* $\mathrm{O}(h^{2p})$ *accurate approximation of the functional* $\mathcal{J}(\mathcal{U})$:

$$\mathcal{J}(\mathcal{U}) - J_h(\mathbf{u}_h) = \mathrm{O}(h^{2p}).$$

See [46] for the proof. This theorem is remarkable because $\mathbf{u}_h$ is typically $\mathrm{O}(h^{p+1})$ at best, so the functional has an asymptotically superconvergent rate relative to the solution error. SBP superconvergence is not limited to steady problems: it was recently extended to time-dependent functionals by Berg and Nordström [49].

Superconvergent functionals can be valuable in scientific and engineering applications where a functional is the ultimate quantity of interest, rather than the solution itself. A quintessential example is aerodynamic drag. For a given mesh resolution, a superconvergent functional can be significantly more accurate than a functional whose order of accuracy is the same as the solution's. Consequently, overall efficiency is improved.

The proof of Theorem 2 relies on the dual consistency[4] of the discretization (94). A discretization is dual consistent if the discrete adjoint equation is a consistent discretization of the continuous adjoint equation [91,92]. For the linear advection equation, dual consistency is easily achieved. Superconvergence has also been proven for elliptic problems and has been observed for more complex PDEs such as the Euler equations [47]. Ensuring dual consistency for nonlinear systems of PDEs, like the Euler and Navier–Stokes equations, is possible, but more involved; see [93,47]. We emphasize that dual consistency is a property of the PDE and functional discretizations, and superconvergent functionals do not require the solution of an adjoint.

Functional superconvergence was originally observed and explained in the finite-element community; examples from the DG literature include [92,94]. In hindsight, the extension of such superconvergence to SBP-SAT schemes now seems obvious, given the variational interpretation of SBP-SAT discretizations and the accuracy of SBP quadrature.

[4] Sometimes called adjoint consistency in the finite-element literature.

8.3. Functional error estimates

The variational interpretation of SBP-SAT schemes also has implications for functional error estimates. One of the most effective methods for estimating functional errors on a given mesh is the dual-weighted residual method (or adjoint-weighted residual method). For SBP-SAT schemes, the dual-weighted residual method is particularly attractive.

Theorem 3. *Let $\mathsf{D}_p = \mathsf{D}_1^{(2p,p,p+1)}$ and $\mathsf{D}_q = \mathsf{D}_1^{(2q,q,q+1)}$ be diagonal-norm SBP operators with $p < q$. Let $\mathbf{u}_h \in \mathbb{R}^{N+1}$ be the solution to the SBP-SAT discretization (94) based on D_p. Let $\mathbf{v}_h \in \mathbb{R}^{N+1}$ be the solution to the discrete adjoint equation corresponding to the primal discretization (94) and the discrete functional (95): that is, $\mathbf{v}_h$ satisfies*

$$-\mathsf{A}\mathsf{D}_p\mathbf{v}_h = -\mathbf{g} - \mathsf{H}_p^{-1}\mathbf{e}_N\mathbf{e}_N^T\mathsf{A}(\mathbf{v}_h + \alpha\mathbf{e}_N).$$

Finally, let $R_{h,q}$ denote the discrete residual (93) evaluated using D_q. Then the functional error estimate

$$\delta J_{h,p} \equiv \mathbf{g}^T(H_p - \mathsf{H}_q)\mathbf{u}_h - R_{h,q}(\mathbf{u}_h, \mathbf{v}_h) \qquad (96)$$

is a h^{2p+2}-order accurate approximation to the true functional error, $\delta\mathcal{J}_{h,p} \equiv J_{h,p}(\mathbf{u}_h) - \mathcal{J}(\mathcal{U})$.

As with functional superconvergence, dual consistency is critical to the effectiveness of the error estimate $\delta J_{h,p}$. If the SBP-SAT discretization is not carefully constructed to be dual consistent, the ideal convergence rate is lost.

Unlike functional superconvergence, the error estimate (96) requires the adjoint $\mathbf{v}_h$. However, this adjoint variable is computed on the same mesh as the primal solution $\mathbf{u}_h$. In contrast, most implementations of the dual-weighted residual method require an approximation to the adjoint on a refined space [95–99]. In the case of (96), this approximation is obtained through the implicit interpolation defined by the higher-order operator D_q used in $R_{h,q}$.

The proof of Theorem 3 can be found in [100], where extensions of the result to more general PDEs are discussed and illustrated. In particular, we note that the error estimate drops to h^{2p+1} for PDEs with second-order derivative operators.

9. Generalization of SBP operators

Recently, Del Rey Fernández, Boom and Zingg [101] generalized the classical SBP-SAT theory to encompass a larger class of SBP operators, where the main extensions are (i) non-repeating interior point operators, (ii) nonuniform nodal distribution in the computational domain, and (iii) operators that do not include one or both boundary nodes. In this section, we give a brief review of the theory and discuss some of the implications of this generalization. In their work, the authors consider SBP operators approximating the first derivative on the interval $x \in [\alpha, \beta]$, where the nodal distribution $\mathbf{x}$ does not need to be uniform and does not need to include one or both boundary nodes. They present the following generalized definition of an SBP operator:

Definition 5. Summation-by-parts operator: An operator D_1 is an approximation to the first derivative of degree q with the SBP property if

(i) $\mathsf{D}_1\mathbf{x}^j = \mathsf{H}^{-1}\mathsf{Q}\mathbf{x}^j = j\mathbf{x}^{j-1}, j \in [0, q]$,
(ii) H is a PD symmetric matrix, and
(iii) $\mathsf{Q} + \mathsf{Q}^T = \widetilde{\mathsf{E}}$, where $(\mathbf{x}^i)^T\widetilde{\mathsf{E}}\mathbf{x}^j = \beta^{i+j} - \alpha^{i+j}, i, j \in [0, r], r \geqslant q$.

This definition encompasses the case where both boundary nodes are included, $\widetilde{\mathsf{E}} = \mathsf{E}$ for $r = \infty$, and uniform nodal distributions.

To see the meaning of this new definition, consider that IBP can be cast as

$$\int_\alpha^\beta \mathcal{V}\frac{\partial\mathcal{U}}{\partial x}\mathrm{d}x + \int_\alpha^\beta \mathcal{U}\frac{\partial\mathcal{V}}{\partial x}\mathrm{d}x = \mathcal{V}\mathcal{U}|_\alpha^\beta. \qquad (97)$$

The discrete version of the left-hand side of (97), used in the discrete energy method to prove stability, is

$$\mathbf{v}^T\mathsf{H}\mathsf{D}\mathbf{u} + \mathbf{u}^T\mathsf{H}\mathsf{D}\mathbf{v}. \qquad (98)$$

If (98) is an approximation to $\mathcal{V}\mathcal{U}|_\alpha^\beta$ then it is an approximation to (97). Using Definition 5, $\mathsf{H}\mathsf{D} = \mathsf{Q}$ and $\mathsf{Q} = \widetilde{\mathsf{E}} - \mathsf{Q}^T$, (98) becomes

$$\mathbf{v}^T\mathsf{H}\mathsf{D}\mathbf{u} + \mathbf{u}^T\mathsf{H}\mathsf{D}\mathbf{v} = \mathbf{v}^T\left[\widetilde{\mathsf{E}} - \mathsf{Q}^T\right]\mathbf{u} + \mathbf{u}^T\mathsf{Q}\mathbf{v}. \qquad (99)$$

Since $\mathbf{u}^T\mathsf{Q}\mathbf{v}$ is a scalar, $\mathbf{u}^T\mathsf{Q}\mathbf{v} = \mathbf{v}^T\mathsf{Q}^T\mathbf{u}$, which gives

$$\mathbf{v}^T\mathsf{H}\mathsf{D}\mathbf{u} + \mathbf{u}^T\mathsf{H}\mathsf{D}\mathbf{v} = \mathbf{v}^T\widetilde{\mathsf{E}}\mathbf{u}, \qquad (100)$$

but by Definition 5, $\mathbf{v}^T\widetilde{\mathsf{E}}\mathbf{u} \approx \mathcal{V}\mathcal{U}|_\alpha^\beta$, which is the sought-after relationship.

Nodal SBP operators, through their norm matrix H, are inextricably linked to quadrature rules and their degree depends on the degree of the associated quadrature rule. Thus, a necessary condition for an SBP operator of degree q is that its norm matrix be associated with a quadrature rule of at least degree $q - 1$, summarized in the following theorem [101]:

Theorem 4. *The norm matrix, H, of an SBP operator of degree q that satisfies Definition 5 must be associated with a quadrature rule of at least degree $q - 1$.*

For diagonal-norm SBP operators, necessary and sufficient conditions are given in the following theorem [101]:

Theorem 5. *A quadrature rule of degree τ with positive weights for a nodal distribution $\mathbf{x}$, with $N + 1$ nodes, is necessary and sufficient for the existence of a diagonal-norm SBP approximation to the first derivative, $\mathsf{D}_1 = \mathsf{H}^{-1}\mathsf{Q}$, that is exact for polynomials of degree $q \leqslant \min\left(\lceil\frac{\tau}{2}\rceil, N\right)$, where $N + 1 \geqslant 2$ is the size of D_1, where $\lceil\rceil$ is the ceiling operator.*

For dense-norm SBP operators, meaning non-diagonal norm SBP operators, the following two theorems link the existence of dense-norm SBP operators and their degree to the existence and degree of a quadrature rule for a given nodal distribution [101]:

Theorem 6. *Given a nodal distribution $\mathbf{x}$, with $N + 1$ nodes, there exists an SBP operator $\mathsf{D}_1 = \mathsf{H}^{-1}\mathsf{Q}$ of degree $q \leqslant N$, with a dense-norm H and an associated quadrature rule $\mathbf{w} = [w_1, \ldots, w_n]$ of degree $\tau \geqslant q - 1$ such that $\int_{x_1}^{x_n}\mathcal{F}\mathrm{d}x \approx \sum_{k=1}^n w_k f_k$, for $N + 1 \geqslant 2$.*

Theorem 7. *A quadrature rule $\mathbf{w} = [w_1, \ldots, w_n]$ of degree τ on a nodal distribution $\mathbf{x}$, such that $\int_{x_1}^{x_n}\mathcal{F}\mathrm{d}x \approx \sum_{k=1}^n w_k f_k$, is necessary and sufficient for the existence of a dense PD norm H that satisfies $\mathbf{1}^T\mathsf{H}\mathbf{f} = \mathbf{w}\mathbf{f}$ and an associated SBP operator, $\mathsf{D}_1 = \mathsf{H}^{-1}\mathsf{Q}$ of degree $q = \min(\tau + 1, N)$.*

As a simple example, consider the Newton–Cotes quadrature rule on four equally spaced nodes which has positive weights and is of degree 3. Theorem 5 guarantees the existence of an SBP operator with maximal degree 2 and a PD norm:

$$\mathsf{H} = \Delta x \begin{bmatrix} \frac{3}{8} & 0 & 0 & 0 \\ 0 & \frac{9}{8} & 0 & 0 \\ 0 & 0 & \frac{9}{8} & 0 \\ 0 & 0 & 0 & \frac{3}{8} \end{bmatrix}, \tag{101}$$

where h is the spacing between nodes. Setting up the accuracy Eq. (37) and solving gives an SBP operator of degree 2:

$$\mathsf{D}_1 = \frac{1}{\Delta x} \begin{bmatrix} -\frac{4}{3} & \frac{3}{2} & 0 & -\frac{1}{6} \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{6} & 0 & -\frac{3}{2} & \frac{4}{3} \end{bmatrix}, \tag{102}$$

where

$$\mathsf{Q} = \begin{bmatrix} -\frac{1}{2} & \frac{9}{16} & 0 & -\frac{1}{16} \\ -\frac{9}{16} & 0 & \frac{9}{16} & 0 \\ 0 & -\frac{9}{16} & 0 & \frac{9}{16} \\ \frac{1}{16} & 0 & -\frac{9}{16} & \frac{1}{2} \end{bmatrix}. \tag{103}$$

Using the classical FD-SBP approach on four nodes it is only possible to obtain a degree one operator, which is

$$\mathsf{D}_1 = \frac{1}{\Delta x} \begin{bmatrix} -1 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & -1 & 1 \end{bmatrix}. \tag{104}$$

The quadrature rule associated with (104) is the composite trapezoidal rule, which is of degree one (order two) and leads to

$$\mathsf{H} = \Delta x \begin{bmatrix} \frac{1}{2} & & & \\ & 1 & & \\ & & 1 & \\ & & & \frac{1}{2} \end{bmatrix}. \tag{105}$$

Having constructed a generalized SBP operator, the question becomes how to construct SATs for the imposition of boundary and interface conditions. If the nodal distribution contains both boundary nodes, then one can take $\widetilde{\mathsf{E}} = \mathsf{E}$, and the resultant SATs are identical in form to those used for SBP operators. To construct SATs for SBP operators that do not include one or both boundary nodes, it is sufficient to consider the case where neither boundary node is present. Suppose we have a nodal distribution $\mathbf{x}$, where $\alpha < x_0 <, \ldots < x_N < \beta$; for $N+1$ distinct nodes, it is possible to construct a one-dimensional interpolant of degree N. Evaluating the interpolant of $\mathcal{U}$ at the boundaries yields

$$\mathbf{t}_\alpha^T \mathbf{u} \approx \mathcal{U}(\alpha), \quad \mathbf{t}_\beta^T \mathbf{u} \approx \mathcal{U}(\beta), \tag{106}$$

where $\mathbf{t}_\alpha$ and $\mathbf{t}_\beta$ have properties

$$\mathbf{t}_\alpha^T \mathbf{x}^j = \alpha^j, \quad \mathbf{t}_\beta^T \mathbf{x}^j = \beta^j, \quad j \in [0, N]. \tag{107}$$

These operators can be combined to form the matrix operator

$$\mathsf{T} = \mathbf{e}_0 \mathbf{t}_\alpha^T + \mathbf{e}_N \mathbf{t}_\beta^T. \tag{108}$$

Taking $\widetilde{\mathsf{E}} = \mathsf{T}^T \mathsf{E} \mathsf{T}$, gives the required property for $\widetilde{\mathsf{E}}$. With this definition of $\widetilde{\mathsf{E}}$ and the interpolant operators $\mathbf{t}_{\alpha/\beta}$ it is possible to construct SATs that lead to consistent, conservative, stable discretizations [101].

The generalization of the SBP concept proposed by Del Rey Fernańdez, Boom and Zingg [101] allows for a large class of operators to be considered SBP operators, for example nodal-based pseudo-spectral operators on Gauss, Gauss–Lobatto and Gauss-Radau quadrature points used in some discontinuous Galerkin schemes, and enables the rigorous development of SATs for such operators. Moreover, the generalization provides a straightforward means of deriving novel SBP operators on nearly arbitrary nodal distributions, by first constructing a quadrature rule.

10. Additional topics

10.1. Nonlinear conservation laws

Like any finite-difference method, SBP-SAT schemes can be used to discretize nonlinear conservation laws; however, additional considerations are necessary to maintain their stability properties. We have already discussed one aspect of these considerations, namely numerical dissipation. In this section we expand on the use of SBP operators for nonlinear PDEs, addressing the stable discretization of nonlinear convective terms.

The issues presented by nonlinear conservation laws are not unique to SBP operators, and there is a vast literature on this subject. Our aim here is to highlight how SBP operators have been adapted to handle these issues.

To begin, we use the inviscid Burgers equation to illustrate the pitfalls of using SBP operators directly, i.e. without changing the form of the convective term. Consider the time-dependent inviscid Burgers equation

$$\begin{aligned} &\frac{\partial \mathcal{U}}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}\mathcal{U}^2\right) = 0, \qquad \forall x \in \Omega, \\ &\mathcal{U}(x, 0) > 0, \qquad \forall x \in \Omega, \\ &\mathcal{U}(x_L, t) - \mathcal{U}_L(t) = 0, \qquad \forall t \in [0, T], \end{aligned} \tag{109}$$

where $\Omega = [x_L, x_R]$ and $T > 0$. To simplify the boundary conditions, we have assumed that the initial condition is strictly positive; this is not necessary in general.

Using integration by parts, it is straightforward to show that a solution to (109) satisfies

$$\frac{d}{dt}\int_\Omega \frac{1}{2}\mathcal{U}^2\, dx = -\frac{1}{3}\mathcal{U}^3(x_R, t) + \frac{1}{3}\mathcal{U}_L^3.$$

As in the linear case, we observe that energy, $\mathcal{U}^2/2$, grows only due to transport through the boundaries. In particular, if $U_L = 0$ and the initial condition is strictly positive, then the energy decreases with time.

Now consider a naive SBP-SAT semi-discretization of the Burgers equation given by

$$\frac{d\mathbf{u}_h}{dt} + \frac{1}{2}\mathsf{D}_1(\mathsf{A}\mathbf{u}_h) = -\frac{1}{2}\mathsf{H}^{-1}\mathbf{e}_0\mathbf{e}_0^T\mathsf{A}(\mathbf{u}_h - \mathcal{U}_L\mathbf{e}_0),$$

where

$$\mathsf{A} = \mathrm{diag}(\mathbf{u}_h) = \mathrm{diag}(u_0, u_1, \ldots, u_N).$$

If we apply the standard energy method to this semi-discretization, we find

$$\frac{d}{dt}\left(\frac{1}{2}\|\mathbf{u}_h\|_{\mathsf{H}}^2\right) + \frac{1}{4}\mathbf{u}_h^T(\mathsf{QA} + \mathsf{AQ}^T)\mathbf{u}_h = -\frac{1}{2}u_0^2(u_0 - \mathcal{U}_L).$$

In the constant linear-convection case, where $\mathsf{A} = \mathrm{diag}(a)$, we can replace $\mathsf{QA} + \mathsf{AQ}^T$ with $\mathsf{A}(\mathsf{E}_N - \mathsf{E}_0)$. For the nonlinear advection velocity above—and spatially varying coefficients in general—we cannot proceed, because Q and A do not commute.

The solution to this problem is to discretize the convective terms using the skew-symmetric form, or canonical splitting, proposed by Olsson and Oliger [39]. The skew-symmetric operator is a particular linear combination of the divergence form of the convective terms,

$$\frac{\partial}{\partial x}\left(\frac{1}{2}\mathcal{U}^2\right) \approx \frac{1}{2}\mathsf{D}_1(\mathsf{A}\mathbf{u}_h),$$

and the advective, or primitive, form of the convective terms,

$$\mathcal{U}\frac{\partial \mathcal{U}}{\partial x} \approx \mathsf{A}\mathsf{D}_1\mathbf{u}_h.$$

In the continuous case, with a smooth solution, the divergence and advective forms are identical; however, they are not equivalent in the discrete case: $\frac{1}{2}\mathsf{D}_1(\mathsf{A}\mathbf{u}_h) \neq \mathsf{A}\mathsf{D}_1\mathbf{u}_h$, in general.

A general convex combination of the divergence and advective convective forms is given by

$$\alpha\frac{1}{2}\mathsf{D}_1(\mathsf{A}\mathbf{u}_h) + (1-\alpha)\mathsf{A}\mathsf{D}_1\mathbf{u}_h,$$

and the skew-symmetric convective operator is obtained by setting $\alpha = 2/3$. Thus, the skew-symmetric discretization of the inviscid Burger's equation is given by

$$\frac{du_h}{dt} + \frac{1}{3}\mathsf{A}\mathsf{D}_1\mathbf{u}_h + \frac{1}{3}\mathsf{D}_1(\mathsf{A}\mathbf{u}_h) = -\frac{2}{3}\mathsf{H}^{-1}\mathbf{e}_0\mathbf{e}_0^T\mathsf{A}(\mathbf{u}_h - \mathcal{U}_L\mathbf{e}_0).$$

Notice that we have also updated the SAT penalty parameter to be consistent with the skew-symmetric form. Applying the energy method here we find

$$\frac{d}{dt}\left(\frac{1}{2}\|\mathbf{u}_h\|_{\mathsf{H}}^2\right) + \frac{1}{6}\mathbf{u}_h^T(\mathsf{Q}^T+\mathsf{Q})\mathsf{A}\mathbf{u}_h + \frac{1}{6}\mathbf{u}_h^T\mathsf{A}(\mathsf{Q}^T+\mathsf{Q})\mathbf{u}_h$$
$$= -\frac{2}{3}u_0^2(u_0 - \mathcal{U}_L).$$

To obtain this intermediate equation, we tacitly assumed that H commutes with A. This is true for diagonal-norm SBP operators, but not for more general H. Using the additional property that A commutes with E_0 and E_N, the above simplifies to

$$\frac{d}{dt}\left(\frac{1}{2}\|\mathbf{u}_h\|_{\mathsf{H}}^2\right) + \frac{1}{3}\mathbf{u}_h^T\mathsf{A}(\mathsf{E}_N - \mathsf{E}_0)\mathbf{u}_h = -\frac{2}{3}u_0^2(u_0 - \mathcal{U}_L),$$

and so

$$\frac{d}{dt}\left(\frac{1}{2}\|\mathbf{u}_h\|_{\mathsf{H}}^2\right) = -\frac{1}{3}u_N^3 + \frac{1}{3}u_0\mathcal{U}_L^2 - \frac{1}{3}u_0(u_0 - \mathcal{U}_L)^2.$$

The first two terms on the right-hand-side mimic the continuous case, while the third term, $-\frac{1}{3}u_0(u_0 - U_L)^2$, is a small damping term on the order of the solution error. Thus, using the skew-symmetric splitting, the method is stable.

The concept of a skew-symmetric operator can be extended to nonlinear systems if the system admits an entropy variable that permits symmetrization of the Jacobians; see, for example, [102,39]. Yee et al. [41] and Sandham et al. [40] were among the first to use this entropy-splitting concept in the context of high-order SBP operators. See also the paper by Kitson et al. [103]. However, suitable entropy-stable SATs for the skew-symmetric approach remains an active area of research.

The skew-symmetric form solves one problem, but it has to the potential to introduce another: lack of discrete conservation. Discrete conservation is required by the Lax-Wendroff theorem to ensure that discontinuities are accurately captured: if an SBP-SAT discretization is not discretely conservative, then shocks will not be predicted correctly. Fortunately, Fisher et al. [43] recently proved the remarkable result that the skew-symmetric form discretized using SBP operators yields a discretely conservative scheme.

10.2. Other extensions

The difficulty of generating multi-block grids around arbitrary geometries has motivated research into so-called meshless methods. A natural question in this context is whether or not SBP operators can be defined on point clouds rather than using tensor-product definitions. Kitson et al. [103] initiated such an inquiry in an important and often overlooked article. For 1-dimensional periodic grids, they showed that it is indeed possible to construct diagonal-norm SBP operators on non-smooth nodal distributions, but there are some significant caveats:

1. The coefficients in the SBP operator become globally coupled;
2. The bandwidth is increased (e.g. a second-order operator has a bandwidth of 5 rather than 3, in general), and;
3. For a given bandwidth and target accuracy, there exist grids for which no positive-definite H exists.

Recently, Chiu et al. [104] presented two methods to construct SBP operators on point clouds. They considered second-order operators, and their results confirm that larger stencils are necessary (9 points on average rather than 5 in 2-dimensions). It remains to be seen if high-order diagonal-norm SBP operators on arbitrary point distributions 1) exist for practical, high-Reynolds number grids; 2) are worth the additional cost of solving the coupled problem to find the coefficients, and; 3) are efficient, despite the larger stencil size.

Reichert et al. [105] used an expanded definition of SBP operators proposed and developed in [106–108], with $\mathsf{Q} + \mathsf{Q}^T = \mathrm{diag}[\mathsf{Q}_L, 0, \ldots, 0, \mathsf{Q}_R]$, where $\mathsf{Q}_{L/R}$ are symmetric matrices. With this definition and specially constructed interpolants, they derived SBP operators that allow for overlapping domains.

Nordström and Lundquist [109] extended the use of SBP operators to the discretization of time. They proved that with initial and block-interface conditions applied using SATs, the resultant time-marching method is unconditionally stable. Other notable contributions include stable and accurate interpolation operators [110] and the method for locally adapting the order of accuracy in [111].

One major difficulty in computational fluid dynamics is the solution of problems with strong discontinuities. A viable strategy that has developed for dealing with such flows is high-order weighted essentially non-oscillatory (WENO) schemes. These schemes use a weighted combination of point operators to construct a discrete approximation to the derivative. Thus, away from discontinuities, WENO schemes benefit from the resolving power of higher-order point operators while being essentially non-oscillatory near discontinuities. In a first of several papers, Yamaleev and Carpenter [112] extended the SBP concept to WENO schemes for scalar and linear hyperbolic systems on periodic domains. They applied their energy-stable WENO (ESWENO) schemes to several one-dimensional hyperbolic problems, including the quasi-one-dimensional nozzle problem, finding that the proposed ESWENO to be more accurate than available alternatives (see also [113]). This method was then later extended to non-periodic problems [114,115]. Furthermore, Fisher and Carpenter [44] combined their previous work on ESWENO schemes and split-form conservative SBP operators for split-form conservation laws [43] to construct entropy stable WENO schemes.

11. Conclusions

SBP operators have a number of advantageous properties that make them an important option for higher-order spatial discretization of partial differential equations. This paper provides a thorough review of SBP-SAT methods with an emphasis on the methodology for deriving operators and their application to practical problems in computational fluid dynamics. In addition, generalizations of the SBP-SAT approach are reviewed, based in a variational interpretation and the close connection to quadrature rules. These generalizations provide a unification with several other methods.

SBP-SAT schemes have reached a level of maturity where they are regularly applied to practical problems in computational fluid dynamics, both steady and unsteady. The combination of SBP-SAT schemes with a parallel Newton–Krylov–Schur algorithm has proven to be a particularly powerful approach. Nevertheless, there remain numerous avenues for future research related to SBP-SAT schemes. These include the use of the skew-symmetric form and entropy splitting, solution-adaptive meshing, including *p* and *h* refinement, and SBP schemes in time, just to name a few. In particular, the generalizations reviewed provide numerous opportunities for important extensions of the SBP concept, for example to unstructured meshes.

Appendix A. Diagonal-norm SBP operators for the first derivative

In this appendix we provide Matlab scripts for computing the undivided difference diagonal norm SBP operators for the first derivative, for $p \in [2,4]$. The inputs are the number of nodes, *n*, and the various free parameters. Note that optimized values are used by default and must be commented out to use other values for the free parameters.

Script for $D_1^{(4,2,:)}$ operator

```
function [H,D1]=D1_4(n)
h11 = 17.d0/48.d0; h22 = 59.d0/48.d0; h33 = 43.d0/
  48.d0; h44 = 49.d0/48.d0;
q11 = −1.d0/2.d0; q12 = 59.d0/96.d0; q13 = −1.d0/
  12.d0; q14 = −1.d0/32.d0;
q23 = 59.d0/96.d0; q24 = 0.d0;q34 = 59.d0/96.d0;
%setting up H and D1
H = eye(n,n);
H(1,1) = h11;H(n,n) = h11;H(2,2) = h22;H(n-1,n-1) =
  h22;
H(3,3) = h33;H(n-2,n-2) = h33;H(4,4) = h44;H(n-3,n-3)
  = h44;
Q = zeros(n,n);
Q(1,1) = q11;Q(1,2) = q12;Q(1,3) = q13; Q(1,4) = q14;
Q(2,1) = −q12; Q(2,3) = q23; Q(2,4) = q24;Q(3,1) =
  −q13;Q(3,2) = −q23;
Q(3,4) = q34; Q(3,5) = −1.d0/12.d0;Q(4,1) =
  −q14;Q(4,2) = −q24;
Q(4,3) = −q34; Q(4,5) = 2.d0/3.d0; Q(4,6) = −1.d0/
  12.d0;
%internal nodes
for i =5:n-4
  Q(i,i-2:i+2) = ...
    [1.d0/12.d0,−2.d0/3.d0,0.d0,2.d0/3.d0,−1.d0/
    12.d0];
end

%bottom portion of the matrix
for i = 1:6
  for j = 1:6
    Q(n-(i-1),n-(j-1)) = -Q(i,j);
  end
end
D1 = inv(H)*Q;
```

Script for $D_1^{(6,3,:)}$ operator

```
function [H,D1]=D1_6(n,q56)
% optimized value comment out to use other values
q56 =5591070156686698065364559.d0/
  7931626489314500743872000.d0;
h11 = 0.13649d5 / 0.43200d5;
  h22 = 0.12013d5 / 0.8640d4;
h33 = 0.2711d4 / 0.4320d4; h44 = 0.5359d4 / 0.4320d4;
h55 = 0.7877d4 / 0.8640d4; h66 = 0.43801d5 /
  0.43200d5;
q11 = −1.d0/2.d0;q12 = −0.953d3 / 0.16200d5 + q56;
q13 = 0.715489d6 / 0.259200d6 - (4.d0 * q56);
q14 = −0.62639d5 / 0.14400d5 + (6.d0 * q56);
q15 = 0.147127d6 / 0.51840d5 - (4.d0 * q56);
q16 = −0.89387d5 / 0.129600d6 + q56;
q23 = −0.57139d5 / 0.8640d4 + (10.d0 * q56);
q24 = 0.745733d6 / 0.51840d5 - (20.d0 * q56);
q25 = −0.18343d5 / 0.1728d4 + (15.d0 * q56);
q26 = 0.240569d6 / 0.86400d5 - (4.d0 * q56);
q34 = −0.176839d6 / 0.12960d5 + (20.d0 * q56);
q35 = 0.242111d6 / 0.17280d5 - (20.d0 * q56);
q36 = −0.182261d6 / 0.43200d5 + (6.d0 * q56);
q45 = −0.165041d6 / 0.25920d5 + (10.d0 * q56);
q46 = 0.710473d6 / 0.259200d6 - (4.d0 * q56);
q47 = 1.d0/6.d1;q57 = −3.D0/2.d1; q58 = 1.d0/6.d1;
%setting up H and D1
H = eye(n,n);
H(1,1) = h11;H(n,n) = h11;H(2,2) = h22;H(n-1,n-1) =
  h22;
H(3,3) = h33;H(n-2,n-2) = h33;H(4,4) = h44;H(n-3,n-3)
  = h44;
H(5,5) = h55;H(n-4,n-4) = h55;H(6,6) = h66;H(n-5,n-5)
  = h66;
Q = zeros(n,n);
Q(1,1) = q11;Q(1,2) = q12;Q(1,3) = q13; Q(1,4) = q14;
Q(1,5) = q15; Q(1,6) = q16;Q(2,1) = −q12; Q(2,3) = q23;
Q(2,4) = q24; Q(2,5) = q25;Q(2,6) = q26;Q(3,1) = −q13;
  Q(3,2) = −q23;
Q(3,4) = q34; Q(3,5) = q35;Q(3,6) = q36;
Q(4,1) = −q14;Q(4,2) = −q24;Q(4,3) = −q34; Q(4,5) =
  q45;
Q(4,6) = q46; Q(4,7) = q47;Q(5,1) = −q15; Q(5,2) =
  −q25;
Q(5,3) = −q35; Q(5,4) = −q45;Q(5,6) = q56; Q(5,7) =
  q57;
Q(5,8) = q58;Q(6,1) = −q16; Q(6,2) = −q26; Q(6,3) =
  −q36; Q(6,4) = −q46;
Q(6,5) = −q56; Q(6,7) = 3.d0/4.d0;Q(6,8) = −3.d0/2.d1;
  Q(6,9) = 1.d0/6.d1;
%internal nodes
for i =7:n-6
  Q(i,i-3:i+3) = ...
[−1.d0/6.d1, 3.d0/2.d1, −3.d0/4.d0, 0.d0, 3.d0/4.d0,
  −3.d0/2.d1, 1.d0/6.d1]; end

%bottom portion of the matrix
for i = 1:9
  for j = 1:9
    Q(n-(i-1),n-(j-1)) = -Q(i,j);
  end
end
D1 = inv(H)*Q;
```

Script for $D_1^{(8,4,:)}$ operator

```
function [H,D1]=D1_8(n,q16,q17,q47)
% optimized value comment out to use other values
q16 = 0.0831482994912206046230504790790872066335d0;
q17 = −0.9521334029619388274601963790830716099197d-2;
q47 = −0.3510216710115618609017136924794334791187d-1;
h11 = 0.1498139d7 / 0.5080320d7; h22 = 0.1107307d7 / 0.725760d6;
h33 = 0.20761d5 / 0.80640d5; h44 = 0.1304999d7 / 0.725760d6;
h55 = 0.299527d6 / 0.725760d6; h66 = 0.103097d6 / 0.80640d5;
h77 = 0.670091d6 / 0.725760d6; h88 = 0.5127739d7 / 0.5080320d7;
q11 = −1.d0/2.d0;
q12 = 0.59065123d8 / 0.91445760d8 + q16 / 0.3d1 + 0.2d1 / 0.3d1 ∗ q17;
q13 = 0.771343d6 / 0.10160640d8 - 0.8d1 / 0.5d1 ∗ q16 - 0.3d1 ∗ q17;
q14 = −0.8276887d7 / 0.20321280d8 + (3.d0 ∗ q16) + (5.d0 ∗ q17);
q15 = 0.17658817d8 / 0.91445760d8 - 0.8d1 / 0.3d1 ∗ q16 - 0.10d2 / 0.3d1 ∗ q17;
q18 = −0.1394311d7 / 0.182891520d9 - q16 / 0.15d2 - q17 / 0.3d1;
q23 = q47 / 0.45d2 + 0.14d2 / 0.3d1 ∗ q16 + 0.77d2 / 0.9d1 ∗ q17 - 0.14866699d8 /
    .130636800d9;
q24 = 0.18734719d8 / 0.13063680d8 - 0.35d2 / 0.3d1 ∗ q16 - 0.175d3 / 0.9d1 ∗ q17 -
    q47 / 0.9d1;
q25 = −0.2642179d7 / 0.3265920d7 + 0.35d2 / 0.3d1 ∗ q16 + 0.140d3 / 0.9d1 ∗ q17 +
    0.2d1 / 0.9d1 ∗ q47;
q26 = 0.1736509d7 / 0.13063680d8 - 0.14d2 / 0.3d1 ∗ q16 - 0.14d2 / 0.9d1 ∗ q17 -
    0.2d1 / 0.9d1 ∗ q47;
q27 = −0.13219d5 / 0.1244160d7 - 0.35d2 / 0.9d1 ∗ q17 + q47 / 0.9d1;
q28 = 0.1407281d7 / 0.114307200d9 + q16 / 0.3d1 + 0.13d2 / 0.9d1 ∗ q17 - q47 /
    0.45d2;
q34 = −0.3056891d7 / 0.4354560d7 + (14.d0 ∗ q16) + 0.70d2 / 0.3d1 ∗ q17 + q47 /
    0.3d1;
q35 = 0.765701d6 / 0.653184d6 - 0.56d2 / 0.3d1 ∗ q16 - 0.245d3 / 0.9d1 ∗ q17 −
    0.8d1 / 0.9d1 ∗ q47;
q36 = −0.238939d6 / 0.414720d6 + 0.42d2 / 0.5d1 ∗ q16 + 0.7d1 ∗ q17 + q47;
q37 = 0.754291d6 / 0.21772800d8 + 0.14d2 / 0.3d1 ∗ q17 - 0.8d1 / 0.15d2 ∗ q47;
q38 = 0.762499d6 / 0.22861440d8 - 0.2d1 / 0.3d1 ∗ q16 - 0.20d2 / 0.9d1 ∗ q17 + q47 /
    0.9d1;
q45 = −0.10064459d8 / 0.26127360d8 + 0.35d2 / 0.3d1 ∗ q16 + 0.175d3 / 0.9d1 ∗ q17 +
    0.10d2 / 0.9d1 ∗ q47;
q46 = 0.62249d5 / 0.77760d5 - (7.d0 ∗ q16) - 0.35d2 / 0.3d1 ∗ q17 - 0.5d1 / 0.3d1
    ∗ q47;
q48 = −0.8276887d7 / 0.91445760d8 + 0.2d1 / 0.3d1 ∗ q16 + 0.10d2 / 0.9d1 ∗ q17 −
    0.2d1 / 0.9d1 ∗ q47;
q56 = 0.792095d6 / 0.2612736d7 + 0.7d1 / 0.3e1 ∗ q16 + 0.70d2 / 0.9d1 ∗ q17 +
    0.10d2 / 0.9d1 ∗ q47;
q57 = −0.42403d5 / 0.207360d6 - 0.35d2 / 0.9d1 ∗ q17 - 0.8d1 / 0.9d1 ∗ q47;
q58 = 0.13906657d8 / 0.182891520d9 - q16 / 0.3d1 + 0.5d1 / 0.9d1 ∗ q17 + 0.2d1 /
    0.9d1 ∗ q47;
q59 = −1.d0/280.d0;
q67 = 0.1360207d7 / 0.1741824d7 + 0.7d1 / 0.3d1 ∗ q17 + q47 / 0.3d1;
q68 = −0.289189d6 / 0.1866240d7 + q16 / 0.15d2 - 0.7d1 / 0.9d1 ∗ q17 - q47 / 0.9d1;
q69 = 4.d0/105.d0;q610 = −1.d0/280.d0;
q78 = 0.16676111d8 / 0.21772800d8 + 0.2d1 / 0.9d1 ∗ q17 + q47 / 0.45d2;
q79 = −1.d0/5.d0;q710 = 4.d0/105.d0;q711 = −1.d0/280.d0; q89 = 4.d0/5.d0;
q810 = −1.d0/5.d0; q811 = 4.d0/105.d0; q812 = −1.d0/280.d0;
%setting up H and D1
H = eye(n,n);
H(1,1) = h11;H(n,n) = h11;H(2,2) = h22;H(n-1,n-1) = h22;H(3,3) = h33;H(n-2,n-2) =
    h33;
H(4,4) = h44;H(n-3,n-3) = h44;H(5,5) = h55;H(n-4,n-4) = h55;H(6,6) =
    h66;H(n-5,n-5) = h66;
H(7,7) = h77;H(n-6,n-6) = h77;H(8,8) = h88;H(n-7,n-7) = h88;
Q = zeros(n,n);
Q(1,1) = q11;Q(1,2) = q12;Q(1,3) = q13; Q(1,4) = q14;Q(1,5) = q15; Q(1,6) = q16;
Q(1,7) = q17; Q(1,8) = q18;Q(2,1) = −q12; Q(2,3) = q23; Q(2,4) = q24; Q(2,5) = q25;
```

```
Q(2,6) = q26; Q(2,7) = q27;Q(2,8) = q28;Q(3,1) = −q13;Q(3,2) = −q23;
Q(3,4) = q34; Q(3,5) = q35;Q(3,6) = q36; Q(3,7) = q37;Q(3,8) = q38;
Q(4,1) = −q14;Q(4,2) = −q24;Q(4,3) = −q34; Q(4,5) = q45;Q(4,6) = q46; Q(4,7) = q47;
Q(4,8) = q48;Q(5,1) = −q15; Q(5,2) = −q25;Q(5,3) = −q35; Q(5,4) = −q45;
Q(5,6) = q56; Q(5,7) = q57;Q(5,8) = q58; Q(5,9) = q59;Q(6,1) = −q16; Q(6,2) = −q26;
Q(6,3) = −q36; Q(6,4) = −q46;Q(6,5) = −q56; Q(6,7) = q67;Q(6,8) = q68; Q(6,9) =
   q69;
Q(6,10) = q610;Q(7,1) = −q17; Q(7,2) = −q27;Q(7,3) = −q37; Q(7,4) = −q47;
Q(7,5) = −q57; Q(7,6) = −q67;Q(7,8) = q78; Q(7,9) = q79;Q(7,10) = q710; Q(7,11) =
   q711;
Q(8,1) = −q18; Q(8,2) = −q28;Q(8,3) = −q38; Q(8,4) = −q48;Q(8,5) = −q58; Q(8,6) =
   −q68;
Q(8,7) = −q78; Q(8,9) = q89;Q(8,10) = q810; Q(8,11) = q811;Q(8,12) = q812;

%internal nodes
for i =9:n-8
   Q(i,i-4:i+4) = ...
[1.d0/280.d0, −4.d0/105.d0, 1.d0/5.d0, −4.d0/5.d0, 0.d0, 4.d0/5.d0, −1.d0/5.d0,
   4.d0/105.d0, −1.d0/280.d0];
end

%bottom portion of the matrix
for i = 1:12
  for j = 1:12
   Q(n-(i-1),n-(j-1)) = -Q(i,j);
  end
end
D1 = inv(H)*Q;
```

Appendix B. SBP operators for the second derivative with variable coefficients

In this Appendix we provide the general form for the SBP operator for the second derivative with variable coefficients for $p \in [3, 4]$, while a Matlab script is provided for the operator with $p = 2$. The script has as input the number of nodes, n, and a vector with the variable coefficients, $\mathbf{b}$. The script returns an undivided difference approximation. The free parameters are by defualt set to optimized values and must be commented out to use other values.

The operator $\widetilde{\mathrm{D}}_2^{(6,3,:)}(\mathrm{B})$ has form

$$\begin{aligned}\widetilde{\mathrm{D}}_2^{(6,3,:)}(\mathrm{B}) &= -\mathrm{H}^{-1}\left(\mathrm{D}_1^{(6,3,:)}\right)^T\mathrm{HBD}_1^{(6,3,:)}\\ &-\frac{\mathrm{H}^{-1}}{80h}\left(\widetilde{\mathrm{D}}_{4,3}^{(2,1,:)}\right)^T\mathrm{C}_4^{(3)}\mathrm{B}\widetilde{\mathrm{D}}_{4,3}^{(2,1,:)}-\frac{\mathrm{H}^{-1}}{100h}\left(\widetilde{\mathrm{D}}_{5,3}^{(2,1,:)}\right)^T\mathrm{C}_5^{(3)}\mathrm{B}\widetilde{\mathrm{D}}_{5,3}^{(2,1,:)}\\ &-\frac{\mathrm{H}^{-1}}{720h}\left(\widetilde{\mathrm{D}}_{6,3}^{(2,1,:)}\right)^T\mathrm{C}_6^{(3)}\mathrm{B}\widetilde{\mathrm{D}}_{6,3}^{(2,1,:)}+\mathrm{H}^{-1}\mathrm{EBD}_b^{(:,\geqslant 4,:)}\end{aligned} \tag{B.1}$$

The matrix $\widetilde{\mathrm{D}}_{4,3}^{(2,1,:)}$ is defined by a 6×9 block, the entries of which are denoted by $d_{4,i,j}$, and an interior repeating point operator given as

- $d_{4,j,1} = -d_{4,j,6} - 5d_{4,j,7} - 15d_{4,j,8} - 35d_{4,j,9} + 1$,
- $d_{4,j,2} = -4 + 5d_{4,j,6} + 24d_{4,j,7} + 70d_{4,j,8} + 160d_{4,j,9}$,
- $d_{4,j,3} = 6 - 10d_{4,j,6} - 45d_{4,j,7} - 126d_{4,j,8} - 280d_{4,j,9}$,
- $d_{4,j,4} = -4 + 10d_{4,j,6} + 40d_{4,j,7} + 105d_{4,j,8} + 224d_{4,j,9}$,
- $D_{4,j,5} = 1 - 5d_{4,j,6} - 15d_{4,j,7} - 35d_{4,j,8} - 70d_{4,j,9}$, for $j \in [1, 6]$.
- Internal point operator: $\left[\frac{1}{3}, -1, 1, -\frac{2}{3}, 1, -1, \frac{1}{3}\right]$.

The matrix $\widetilde{\mathrm{D}}_{5,3}^{(2,1,:)}$ is defined by a 6×9 block, the entries of which are denoted by $d_{5,i,j}$, and an interior repeating point operator given as

- $d_{5,j,1} = -1 + d_{5,j,7} + 6d_{5,j,8} + 21d_{5,j,9}$,
- $d_{5,j,2} = 5 - 6d_{5,j,7} - 35d_{5,j,8} - 120d_{5,j,9}$,
- $d_{5,j,3} = -10 + 15d_{5,j,7} + 84d_{5,j,8} + 280d_{5,j,9}$,
- $d_{5,j,4} = 10 - 20d_{5,j,7} - 105d_{5,j,8} - 336d_{5,j,9}$,
- $d_{5,j,5} = -5 + 15d_{5,j,7} + 70d_{5,j,8} + 210d_{5,j,9}$,
- $d_{5,j,6} = 1 - 6d_{5,j,7} - 21D_{5,j,8} - 56d_{5,j,9}$, for $j \in [1, 6]$.
- Internal point operator: $\left[-\frac{1}{2}, 2, -\frac{5}{2}, 0, \frac{5}{2}, -2, \frac{1}{2}\right]$.

The matrix $\widetilde{\mathrm{D}}_{6,3}^{(2,1,:)}$ is defined by a 6×9 block, the entries of which are denoted by $d_{6,i,j}$, and an interior repeating point operator given as

- $d_{6,j,1} = 1 - d_{6,j,8} - 7d_{6,j,9}$,
- $d_{6,j,2} = -6 + 7d_{6,j,8} + 48d_{6,j,9}$,
- $d_{6,j,3} = 15 - 21d_{6,j,8} - 140d_{6,j,9}$,
- $d_{6,j,4} = -20 + 35d_{6,j,8} + 224d_{6,j,9}$,
- $d_{6,j,5} = 15 - 35d_{6,j,8} - 210d_{6,j,9}$,
- $d_{6,j,6} = -6 + 21d_{6,j,8} + 112d_{6,j,9}$,
- $d_{6,j,7} = 1 - 7d_{6,j,8} - 28d_{6,j,9}$, for $j \in [1, 6]$.
- Internal point operator: $[1, -6, 15, -20, 15, -6, 1]$.

The operator $\widetilde{\mathrm{D}}_2^{(8,4,:)}(\mathrm{B})$ has form

$$\begin{aligned}\widetilde{\mathrm{D}}_2^{(6,3,:)}(\mathrm{B}) &= -\mathrm{H}^{-1}\left(\mathrm{D}_1^{(8,4,:)}\right)^T\mathrm{HBD}_1^{(8,4,:)}\\ &-\frac{\mathrm{H}^{-1}}{350h}\left(\widetilde{\mathrm{D}}_{5,4}^{(2,1,:)}\right)^T\mathrm{C}_5^{(4)}\mathrm{B}\widetilde{\mathrm{D}}_{5,4}^{(2,1,:)}-\frac{\mathrm{H}^{-1}}{252h}\left(\widetilde{\mathrm{D}}_{6,4}^{(2,1,:)}\right)^T\mathrm{C}_6^{(4)}\mathrm{B}\widetilde{\mathrm{D}}_{6,4}^{(2,1,:)}\\ &-\frac{\mathrm{H}^{-1}}{980h}\left(\widetilde{\mathrm{D}}_{7,4}^{(2,1,:)}\right)^T\mathrm{C}_7^{(4)}\mathrm{B}\widetilde{\mathrm{D}}_{7,4}^{(2,1,:)}-\frac{\mathrm{H}^{-1}}{11200h}\left(\widetilde{\mathrm{D}}_{8,4}^{(2,1,:)}\right)^T\mathrm{C}_8^{(4)}\mathrm{B}\widetilde{\mathrm{D}}_{8,4}^{(2,1,:)}\\ &+\mathrm{H}^{-1}\mathrm{EBD}_b^{(:,\geqslant 5,:)}\end{aligned} \tag{B.2}$$

The matrix $\widetilde{\mathrm{D}}_{5,4}^{(2,1,:)}$ is defined by a 8×12 block, the entries of which are denoted by $d_{5,i,j}$, and an interior repeating point operator given as

- $d_{5,j,1} = -\frac{1}{56} + \frac{1}{56}d_{5,j,10} + \frac{3}{28}d_{5,j,11} + \frac{3}{8}d_{5,j,12} - \frac{3}{8}d_{5,j2} - \frac{3}{28}d_{5,j,3} - \frac{1}{56}d_{5,j,4}$,
- $d_{5,j,5} = \frac{5}{4} - \frac{9}{4}d_{5,j,10} - \frac{25}{2}d_{5,j,11} - \frac{165}{4}d_{5,j,12} - \frac{35}{4}d_{5,j,2} - \frac{15}{2}d_{5,j,3} - \frac{15}{4}d_{5,j,4}$,

- $d_{5,j,6} = -4 + 9d$
  $5, j, 10 + 48d_{5,j,11} + 154d_{5,j,12} + 21d_{5,j,2} + 16d_{5,j,3} + 6d_{5,j,4}$,
- $d_{5,j,7} = 5 - 15d_{5,j,10}$
  $-75d_{5,j,11} - 231d_{5,j,12} - 21d_{5,j,2} - 15d_{5,j,3} - 5d_{5,j,4}$,
- $d_{5,j,8} = -\frac{20}{7} + \frac{90}{7}d_{5,j,10} + \frac{400}{7}d_{5,j,11} + 165$
  $d_{5,j,12} + 10d_{5,j,2} + \frac{48}{7}d_{5,j,3} + \frac{15}{7}d_{5,j,4}$,
- $d_{5,j,9} = \frac{5}{8} - \frac{45}{8}D_{5,j,10} - \frac{75}{4}d_{5,j,11} - \frac{385}{8}d_{5,j,12} - \frac{15}{8}d_{5,j,2} - \frac{5}{4}d_{5,j,3} - \frac{3}{8}d_{5,j,4}$ for $j \in [1, 8]$.
- Internal point operator: $[-\frac{1}{4}, 1, -\frac{3}{2}, 1, 0, -1, \frac{3}{2}, -1, \frac{1}{4}]$.

The matrix $\widetilde{\mathrm{D}}_{6,4}^{(2,1,:)}$ is defined by a 8 × 12 block, the entries of which are denoted by $d_{6,i,j}$, and an interior repeating point operator given as

- $d_{6,j,1} = \frac{7}{33}d_{6,j,5} + d_{16}33d_{6,j,6} + \frac{20}{33}d_{6,j,7} + \frac{16}{33}d_{6,j,8} + \frac{7}{33}d_{6,j,9} + \frac{4}{33}$,
- $d_{6,j,10} = -\frac{1}{6}d_{6,j,5} - \frac{25}{42}d_{6,j,6} - \frac{25}{21}d_{6,j,7} - \frac{5}{3}d_{6,j,8} - \frac{5}{3}d_{6,j,9} + \frac{5}{42}$,
- $d_{6,j,11} = \frac{1}{6}d_{6,j,5} + \frac{4}{7}d_{6,j,6} + \frac{15}{14}d_{6,j,7} + \frac{4}{3}d_{6,j,8} + d_{6,j,9} - \frac{1}{7}$,
- $d_{6,j,12} = \frac{1}{22} - \frac{1}{22}d_{6,j,5} - \frac{5}{33}d_{6,j,6} - \frac{3}{11}d_{6,j,7} - \frac{7}{22}d_{6,j,j8} - \frac{7}{33}d_{6,j,9}$,
- $d_{6,j,2} = -\frac{1}{2} - \frac{7}{6}d_{6,j,5} - \frac{5}{2}d_{6,j,6} - 3d_{6,j,7} - \frac{7}{3}d_{6,j,8} - d_{6,j,9}$,
- $d_{6,j,3} = \frac{5}{7} + \frac{5}{2}d_{6,j,5} + \frac{100}{21}d_{6,j,6} + \frac{75}{14}d_{6,j,7} + 4d_{6,j,8} + \frac{5}{3}d_{6,j,9}$,
- $d_{6,j,4} = -\frac{5}{14} - \frac{5}{2}d_{6,j,5} - \frac{25}{7}d_{6,j,6} - \frac{25}{7}d_{6,j,7} - \frac{5}{2}d_{6,j,8} - d_{6,j,9}$, for $j \in [1, 8]$.
- Internal point operator: $[\frac{3}{10}, -\frac{7}{5}, \frac{12}{5}, -\frac{9}{5}, 1, -\frac{9}{5}, \frac{12}{5}, -\frac{7}{5}, \frac{3}{10}]$.

The matrix $\widetilde{\mathrm{D}}_{7,4}^{(2,1,:)}$ is defined by a 8 × 12 block, the entries of which are denoted by $d_{7,i,j}$, and an interior repeating point operator given as

- $d_{7,j,1} = \frac{1}{8}d_{7,j,10} + d_{7,j,11} + \frac{9}{2}d_{7,j,12} - \frac{1}{8}d_{7,j,2} - \frac{1}{8}$,
- $d_{7,j,3} = \frac{7}{2} - \frac{9}{2}d_{7,j,10} - 35d_{7,j,11} - 154d_{7,j,12} - \frac{7}{2}d_{7,j,2}$,
- $d_{7,j,4} = -14 + 21d_{7,j,10} + 160d_{7,j,11} + 693d_{7,j,12} + 7d_{7,j,2}$,
- $d_{7,j,5} = -\frac{189}{4}d_{7,j,10} - 350d_{7,j,11} - 1485d_{7,j,12} - \frac{35}{4}d_{7,j,2} + \frac{105}{4}$,
- $d_{7,j,6} = -28 + 63d_{7,j,10} + 448d_{7,j,11} + 1848d_{7,j,12} + 7d_{7,j,2}$,
- $d_{7,j,7} = \frac{35}{2} - \frac{105}{2}d_{7,j,10} - 350d_{7,j,11} - 1386d_{7,j,12} - \frac{7}{2}d_{7,j,2}$,
- $d_{7,j,8} = -6 + 27d_{7,j,10} + 160d_{7,j,11} + 594d_{7,j,12} + d_{7,j,2}$,
- $d_{7,j,9} = \frac{7}{8} - \frac{63}{8}d_{7,j,10} - 35d_{7,j,11} - \frac{231}{2}d_{7,j,12} - \frac{1}{8}d_{7,j,2}$ for $j \in 1, 8]$.
- Internal point operator: $[-\frac{1}{2}, 3, -7, 7, 0, -7, 7, -3, \frac{1}{2}]]$.

The matrix $\widetilde{\mathrm{D}}_{8,4}^{(2,1,:)}$ is defined by a 8 × 12 block, the entries of which are denoted by $d_{8,i,j}$, and an interior repeating point operator given as

- $d_{8,j1} = 1 - d_{8,j,10} - 9d_{8,j,11} - 45d_{8,j,12}$,
- $d_{8,j,2} = -8 + 9d_{8,j,10} + 80d_{8,j,11} + 396d_{8,j,12}$,
- $d_{8,j,3} = 28 - 36d_{8,j,10} - 315d_{8,j,11} - 1540d_{8,j,12}$,
- $d_{8,j,4} = -56 + 84 * d_{8,j,10} + 720d_{8,j,11} + 3465d_{8,j,12}$,
- $d_{8,j,5} = 70 - 126d_{8,j,10} - 1050d_{8,j,11} - 4950d_{8,j,12}$,
- $d_{8,j,6} = -56 + 126d_{8,j,10} + 1008d_{8,j,11} + 4620d_{8,j,12}$,
- $d_{8,j,7} = 28 - 84d_{8,j,10} - 630d_{8,j,11} - 2772d_{8,j,12}$,
- $d_{8,j,8} = -8 + 36d_{8j,j10} + 240d_{8,j,11} + 990d_{8j,j12}$,
- $d_{8,j,9} = 1 - 9d_{8,j,10} - 45d_{8,j,11} - 165d_{8,j,12}$, for $j \in [0, 8]$.
- Internal point operator: $[1, -8, 28, -56, 70, -56, 28, -8, 1]$.

B.1. Script for $D_2^{(4,2,:)}(B)$ operator

Appendix C. Interface SAT for the NS equations

Here we summarize the interface SAT discussed in Section 7.1.1 and give the relevant matrices (see [29]). The interface SAT is composed of three SATs, one for the inviscid portion of the NS equa-

```
function [D2]=D2_4(n,b)
% optimizdd solution
Db5 = .75725475037538196966175487651920415770 64d0;
Db6 = −.21372624812309015169128269293821111150 51d0;
c32 = 0.d0;c33 = 0.d0;
c34 = 3.53751476671083003791075269800745173161 0d0;
c41 = 0.d0;c42 = 0.d0;c43 = 0.d0;
c44 = 0.738550162248569558667778702591121467688 6d-2;
d325 = 32.5104847173842160451928625425472293318 6d0;
d326 = −7.6208581461957501923244938165043350273 86d0;
d335 = −13.887775022606329661389668702058272136 99d0;
d336 = 3.4689338221413310927002634155693378690 41d0;
d345 = −0.7823237972910208591778157593262697095 161d-2;
d346 = .13740855635915709988744444708621360951 42d0;
d416 = 4.2293448915482683832172582126853495740 41d0;
d426 = 9.5790168774233310870220804851602047727 41d0;
d436 = 23.256976729411852392939662640336114993 84d0;
d446 = 146.12848221117202302306915389087675911 80d0;
% optimum family of solutions
c31 = 1030347.d0/286552.d0-c32-c33-c34;
d315=(34.d0*(675.d0+8428.d0*d325*c32+8428.d0*d335*c33+8428.d0*d345*c34))/...
(−1030347.d0+286552.d0*c32+286552.d0*c33+286552.d0*c34);
d316 = (1462.d0*(196.d0*d326*c32+196.d0*d336*c33+196.d0*d346*c34-87.d0))/...
(−1030347.d0+286552.d0*c32+286552.d0*c33+286552.d0*c34);
Db1 = −11.d0/6.d0+Db5+4.d0*Db6; Db2 = 3.d0-4.d0*Db5-15.d0*Db6;
Db3 = −3.d0/2.d0+6.d0*Db5+20.d0*Db6; Db4 = 1.d0/3.d0-4.d0*Db5-10.d0*Db6;

% solution to Db
d311 = −1.d0+d315+4.d0*d316;d312 = 3.d0-4.d0*d315-15.d0*d316;
d313 = −3.d0+6.d0*d315+20.d0*d316;d314 = 1.d0-4.d0*d315-10.d0*d316;
d321 = −1.d0+d325+4.d0*d326;d322 = 3.d0-4.d0*d325-15.d0*d326;
```

```
d323 = −3.d0+6.d0∗d325+20.d0∗d326;d324 = 1.d0-4.d0∗d325-10.d0∗d326;
d331 = −1.d0+d335+4.d0∗d336;d332 = 3.d0-4.d0∗d335-15.d0∗d336;
d333 = −3.d0+6.d0∗d335+20.d0∗d336;d334 = 1.d0-4.d0∗d335-10.d0∗d336;
d341 = −1.d0+d345+4.d0∗d346;d342 = 3.d0-4.d0∗d345-15.d0∗d346;
d343 = −3.d0+6.d0∗d345+20.d0∗d346;d344 = 1.d0-4.d0∗d345-10.d0∗d346;

% construct D3
d411 = 1.d0-d416;d412 = −4.d0+5.d0∗d416;d413 = 6.d0-10.d0∗d416;
d414 = −4.d0+10.d0∗d416;d415 = 1.d0-5.d0∗d416;d421 = 1.d0-d426;
d422 = 5.d0∗d426-4.d0;d423 = 6.d0-10.d0∗d426;d424 = −4.d0+10.d0∗d426;
d425 = 1.d0-5.d0∗d426;d431 = 1.d0-d436;d432 = −4.d0+5.d0∗d436;
d433 = 6.d0-10.d0∗d436;d434 = −4.d0+10.d0∗d436;d435 = 1.d0-5.d0∗d436;
d441 = 1.d0-d446;d442 = −4.d0+5.d0∗d446;d443 = 6.d0-10.d0∗d446; d444 = 10.d0∗d446-4.d0;d445 = 1.d0-5.d0∗d446;

% construct D4
M11 = zeros(1,4);
M11(1) = 0.12d2 / 0.17d2 + d311 ∧ 2 ∗ c31 / 0.18d2 + d411 ∧ 2 ∗ c41 / 0.48d2;
M11(2) = 0.59d2 / 0.192d3 + d321 ∧ 2 ∗ c32 / 0.18d2 + d421 ∧ 2 ∗ c42 / 0.48d2;
M11(3) = 0.1d1 / 0.129d3 + d331 ∧ 2 ∗ c33 / 0.18d2 + d431 ∧ 2 ∗ c43 / 0.48d2;
M11(4) = 0.3d1 / 0.3136d4 + d341 ∧ 2 ∗ c34 / 0.18d2 + d441 ∧ 2 ∗ c44 / 0.48d2;

M12 = zeros(1,4);
M12(1) = −0.59d2 / 0.68d2 + d311 ∗ c31 ∗ d312 / 0.18d2 + d411 ∗ c41 ∗ d412 /
   0.48d2;
M12(2) = d321 ∗ c32 ∗ d322 / 0.18d2 + d421 ∗ c42 ∗ d422 / 0.48d2;
M12(3) = −0.59d2 / 0.1032d4 + d331 ∗ c33 ∗ d332 / 0.18d2 + d431 ∗ c43 ∗ d432 /
   0.48d2;
M12(4) = d341 ∗ c34 ∗ d342 / 0.18d2 + d441 ∗ c44 ∗ d442 / 0.48d2;

M13 = zeros(1,4);
M13(1) = 0.2d1 / 0.17d2 + d311 ∗ c31 ∗ d313 / 0.18d2 + d411 ∗ c41 ∗ d413 / 0.48d2;
M13(2) = −0.59d2 / 0.192d3 + d321 ∗ c32 ∗ d323 / 0.18d2 + d421 ∗ c42 ∗ d423 /
   0.48d2;
M13(3) = d331 ∗ c33 ∗ d333 / 0.18d2 + d431 ∗ c43 ∗ d433 / 0.48d2;
M13(4) = −0.59d2 / 0.3136d4 + d341 ∗ c34 ∗ d343 / 0.18d2 + d441 ∗ c44 ∗ d443 /
   0.48d2;

M14 = zeros(1,4);
M14(1) = 0.3d1 / 0.68d2 + d311 ∗ c31 ∗ d314 / 0.18d2 + d411 ∗ c41 ∗ d414 / 0.48d2;
M14(2) = d321 ∗ c32 ∗ d324 / 0.18d2 + d421 ∗ c42 ∗ d424 / 0.48d2;
M14(3) = 0.59d2 / 0.1032d4 + d331 ∗ c33 ∗ d334 / 0.18d2 + d431 ∗ c43 ∗ d434 / 0.48d2;
M14(4) = d341 ∗ c34 ∗ d344 / 0.18d2 + d441 ∗ c44 ∗ d444 /
   0.48d2;

M15 = zeros(1,4);
M15(1) = d311 ∗ c31 ∗ d315 / 0.18d2 + d411 ∗ c41 ∗ d415 / 0.48d2;
M15(2) = d321 ∗ c32 ∗ d325 / 0.18d2 + d421 ∗ c42 ∗ d425 / 0.48d2;
M15(3) = −0.1d1 / 0.129d3 + d331 ∗ c33 ∗ d335 / 0.18d2 + d431 ∗ c43 ∗ d435 / 0.48d2;
M15(4) = 0.1d1 / 0.49d2 + d341 ∗ c34 ∗ d345 / 0.18d2 + d441 ∗ c44 ∗ d445 /
   0.48d2;

M16 = zeros(1,4);
M16(1) = d311 ∗ c31 ∗ d316 / 0.18d2 + d411 ∗ c41 ∗ d416 / 0.48d2;
M16(2) = d321 ∗ c32 ∗ d326 / 0.18d2 + d421 ∗ c42 ∗ d426 / 0.48d2;
M16(3) = d331 ∗ c33 ∗ d336 / 0.18d2 + d431 ∗ c43 ∗ d436 / 0.48d2;
M16(4) = −0.1d1 / 0.392d3 + d341 ∗ c34 ∗ d346 / 0.18d2 + d441 ∗ c44 ∗ d446 /
   0.48d2;

M22 = zeros(1,4);
M22(1) = 0.3481d4 / 0.3264d4 + d312 ∧ 2 ∗ c31 / 0.18d2 + d412 ∧ 2 ∗ c41 / 0.48d2;
M22(2) = d322 ∧ 2 ∗ c32 / 0.18d2 + d422 ∧ 2 ∗ c42 / 0.48d2;
M22(3) = 0.3481d4 / 0.8256d4 + d332 ∧ 2 ∗ c33 / 0.18d2 + d432 ∧ 2 ∗ c43 / 0.48d2;
M22(4) = d342 ∧ 2 ∗ c34 / 0.18d2 + d442 ∧ 2 ∗ c44 / 0.48d2;

M23 = zeros(1,4);
M23(1) = −0.59d2 / 0.408d3 + d312 ∗ c31 ∗ d313 / 0.18d2 + d412 ∗ c41 ∗ d413 /
   0.48d2;
M23(2) = d322 ∗ c32 ∗ d323 / 0.18d2 + d422 ∗ c42 ∗ d423 / 0.48d2;
```

(*continued on next page*)

```
M23(3) = d332 * c33 * d333 / 0.18d2 + d432 * c43 * d433 / 0.48d2;
M23(4) = d342 * c34 * d343 / 0.18d2 + d442 * c44 * d443 / 0.48d2;

M24 = zeros(1,4);
M24(1) = −0.59d2 / 0.1088d4 + d312 * c31 * d314 / 0.18d2 + d412 * c41 * d414 /
   0.48d2;
M24(2) = d322 * c32 * d324 / 0.18d2 + d422 * c42 * d424 / 0.48d2;
M24(3) = −0.3481d4 / 0.8256d4 + d332 * c33 * d334 / 0.18d2 + d432 * c43 * d434 /
   0.48d2;
M24(4) = d342 * c34 * d344 / 0.18d2 + d442 * c44 * d444 / 0.48d2;

M25 = zeros(1,4);
M25(1) = d312 * c31 * d315 / 0.18d2 + d412 * c41 * d415 / 0.48d2;
M25(2) = d322 * c32 * d325 / 0.18d2 + d422 * c42 * d425 / 0.48d2;
M25(3) = 0.59d2 / 0.1032d4 + d332 * c33 * d335 / 0.18d2 + d432 * c43 * d435 /
   0.48d2;
M25(4) = d342 * c34 * d345 / 0.18d2 + d442 * c44 * d445 / 0.48d2;

M26 = zeros(1,4);
M26(1) = d312 * c31 * d316 / 0.18d2 + d412 * c41 * d416 / 0.48d2;
M26(2) = d322 * c32 * d326 / 0.18d2 + d422 * c42 * d426 / 0.48d2;
M26(3) = d332 * c33 * d336 / 0.18d2 + d432 * c43 * d436 / 0.48d2;
M26(4) = d342 * c34 * d346 / 0.18d2 + d442 * c44 * d446 / 0.48d2;

M33 = zeros(1,5);
M33(1) = 0.1d1 / 0.51d2 + d313 ∧ 2 * c31 / 0.18d2 + d413 ∧ 2 * c41 / 0.48d2;
M33(2) = 0.59d2 / 0.192d3 + d323 ∧ 2 * c32 / 0.18d2 + d423 ∧ 2 * c42 / 0.48d2;
M33(3) = d333 ∧ 2 * c33 / 0.18d2 + d433 ∧ 2 * c43 / 0.48d2;
M33(4) = 0.3481d4 / 0.9408d4 + d343 ∧ 2 * c34 / 0.18d2 + d443 ∧ 2 * c44 / 0.48d2;
M33(5) = 0.1d1 / 0.24d2;

M34 = zeros(1,5);
M34(1) = 0.1d1 / 0.136d3 + d313 * c31 * d314 / 0.18d2 + d413 * c41 * d414 / 0.48d2;
M34(2) = d323 * c32 * d324 / 0.18d2 + d423 * c42 * d424 / 0.48d2;
M34(3) = d333 * c33 * d334 / 0.18d2 + d433 * c43 * d434 / 0.48d2;
M34(4) = d343 * c34 * d344 / 0.18d2 + d443 * c44 * d444 / 0.48d2;
M34(5) = −0.1d1 / 0.6d1;

M35 = zeros(1,5);
M35(1) = d313 * c31 * d315 / 0.18d2 + d413 * c41 * d415 / 0.48d2;
M35(2) = d323 * c32 * d325 / 0.18d2 + d423 * c42 * d425 / 0.48d2;
M35(3) = d333 * c33 * d335 / 0.18d2 + d433 * c43 * d435 / 0.48d2;
M35(4) = −0.59d2 / 0.147d3 + d343 * c34 * d345 / 0.18d2 + d443 * c44 * d445 / 0.48d2;
M35(5) = 0.1d1 / 0.8d1;

M36 = zeros(1,5);
M36(1) = d313 * c31 * d316 / 0.18d2 + d413 * c41 * d416 / 0.48d2;
M36(2) = d323 * c32 * d326 / 0.18d2 + d423 * c42 * d426 / 0.48d2;
M36(3) = d333 * c33 * d336 / 0.18d2 + d433 * c43 * d436 / 0.48d2;
M36(4) = 0.59d2 / 0.1176d4 + d343 * c34 * d346 / 0.18d2 + d443 * c44 * d446 /
   0.48d2;

M44 = zeros(1,6);
M44(1) = 0.3d1 / 0.1088d4 + d314 ∧ 2 * c31 / 0.18d2 + d414 ∧ 2 * c41 / 0.48d2;
M44(2) = d324 ∧ 2 * c32 / 0.18d2 + d424 ∧ 2 * c42 / 0.48d2;
M44(3) = 0.3481d4 / 0.8256d4 + d334 ∧ 2 * c33 / 0.18d2 + d434 ∧2 * c43 / 0.48d2;
M44(4) = d344 ∧ 2 * c34 / 0.18d2 + d444 ∧ 2 * c44 / 0.48d2;
M44(5) = 0.5d1 / 0.6d1;
M44(6) = 0.1d1 / 0.24d2;

M45 = zeros(1,6);
M45(1) = d314 * c31 * d315 / 0.18d2 + d414 * c41 * d415 / 0.48d2;
M45(2) = d324 * c32 * d325 / 0.18d2 + d424 * c42 * d425 / 0.48d2;
M45(3) = −0.59d2 / 0.1032d4 + d334 * c33 * d335 / 0.18d2 + d434 * c43 * d435 /
   0.48d2;
M45(4) = d344 * c34 * d345 / 0.18d2 + d444 * c44 * d445 / 0.48d2;
M45(5) = −0.1d1 / 0.2d1;
M45(6) = −0.1d1 / 0.6d1;
```

```
M46 = zeros(1,6);
M46(1) = d314 ∗ c31 ∗ d316 / 0.18d2 + d414 ∗ c41 ∗ d416 / 0.48d2;
M46(2) = d324 ∗ c32 ∗ d326 / 0.18d2 + d424 ∗ c42 ∗ d426 / 0.48d2;
M46(3) = d334 ∗ c33 ∗ d336 / 0.18d2 + d434 ∗ c43 ∗ d436 / 0.48d2;
M46(4) = d344 ∗ c34 ∗ d346 / 0.18d2 + d444 ∗ c44 ∗ d446 / 0.48d2;
M46(5) = −0.1d1 / 0.6d1;M46(6) = 0.1d1 / 0.8d1;

M55 = zeros(1,7);
M55(1) = d315 ∧ 2 ∗ c31 / 0.18d2 + d415 ∧ 2 ∗ c41 / 0.48d2;
M55(2) = d325 ∧ 2 ∗ c32 / 0.18d2 + d425 ∧ 2 ∗ c42 / 0.48d2;
M55(3) = 0.1d1 / 0.129d3 + d335 ∧ 2 ∗ c33 / 0.18d2 + d435 ∧ 2 ∗ c43 / 0.48d2;
M55(4) = 0.64d2 / 0.147d3 + d345 ∧ 2 ∗ c34 / 0.18d2 + d445 ∧ 2 ∗ c44 / 0.48d2;
M55(5) = 0.3d1 / 0.4d1;M55(6) = 0.5d1 / 0.6d1;M55(7) = 0.1d1 / 0.24d2;

M56 = zeros(1,7);
M56(1) = d315 ∗ c31 ∗ d316 / 0.18d2 + d415 ∗ c41 ∗ d416 / 0.48d2;
M56(2) = d325 ∗ c32 ∗ d326 / 0.18d2 + d425 ∗ c42 ∗ d426 / 0.48d2;
M56(3) = d335 ∗ c33 ∗ d336 / 0.18d2 + d435 ∗ c43 ∗ d436 / 0.48d2;
M56(4) = −0.8d1 / 0.147d3 + d345 ∗ c34 ∗ d346 / 0.18d2 + d445 ∗ c44 ∗ d446 /
   0.48d2;
M56(5) = −0.1d1 / 0.2d1;M56(6) = −0.1d1 / 0.2d1;M56(7) = −0.1d1 / 0.6d1;

M66 = zeros(1,8);
M66(1) = d316 ∧ 2 ∗ c31 / 0.18d2 + d416 ∧ 2 ∗ c41 / 0.48d2;
M66(2) = d326 ∧ 2 ∗ c32 / 0.18d2 + d426 ∧ 2 ∗ c42 / 0.48d2;
M66(3) = d336 ∧ 2 ∗ c33 / 0.18d2 + d436 ∧ 2 ∗ c43 / 0.48d2;
M66(4) = 0.1d1 / 0.147d3 + d346 ∧ 2 ∗ c34 / 0.18d2 + d446 ∧ 2 ∗ c44 / 0.48d2;
M66(5) = 0.5d1 / 0.6d1;M66(6) = 0.3d1 / 0.4d1;M66(7) = 0.5d1 / 0.6d1;
M66(8) = 0.1d1 / 0.24d2;

Mjm2 = zeros(1,5);
Mjm2(1) = 0.1D1 / 0.8D1;Mjm2(2) = −0.1D1 / 0.6D1;Mjm2(3) = 0.1D1 / 0.8D1;
Mjm2(4) = 0.d0;Mjm2(5) = 0.d0;

Mjm1 = zeros(1,5);
Mjm1(1) = −0.1D1 / 0.6D1;Mjm1(2) = −0.1D1 / 0.2D1;Mjm1(3) = −0.1D1 / 0.2D1;
Mjm1(4) = −0.1D1 / 0.6D1;Mjm1(5) = 0.d0;

Mj = zeros(1,5);
Mj(1) = 0.1D1 / 0.24D2;Mj(2) = 0.5D1 / 0.6D1;Mj(3) = 0.3D1 / 0.4D1;
Mj(4) = 0.5D1 / 0.6D1;Mj(5) = 0.1D1 / 0.24D2;

Mjp1 = zeros(1,5);
Mjp1(1) = 0.d0;Mjp1(2) = −0.1D1 / 0.6D1;Mjp1(3) = −0.1D1 / 0.2D1;
Mjp1(4) = −0.1D1 / 0.2D1;Mjp1(5) = −0.1D1 / 0.6D1;

Mjp2 = zeros(1,5);
Mjp2(1) = 0.d0;Mjp2(2) = 0.d0;Mjp2(3) = 0.1D1 / 0.8D1;
Mjp2(4) = −0.1D1 / 0.6D1;Mjp2(5) = 0.1D1 / 0.8D1;

H11 = 17.d0/48.d0; H22 = 59.d0/48.d0; H33 = 43.d0/48.d0; H44 = 49.d0/48.d0;
%sdtting up H
H = eye(n,n);
H(1,1) = H11;H(n,n) = H11;H(2,2) = H22;H(n-1,n-1) = H22;
H(3,3) = H33;H(n-2,n-2) = H33;H(4,4) = H44;H(n-3,n-3) = H44;

M = zeros(n,n);
Db = zeros(n,n);
Db(1,1:6) = b(1)∗[Db1,Db2,Db3,Db4,Db5,Db6];
Db(n,n:-1:n-5) = -b(n)∗[Db1,Db2,Db3,Db4,Db5,Db6];
E = zeros(n,n);
E(1,1) = −1;
E(n,n) = 1;
M(1,1:6) = [M11∗b(1:4),M12∗b(1:4),M13∗b(1:4),M14∗b(1:4),M15∗b(1:4),M16∗b(1:4)];
M(2,1:6) = [M12∗b(1:4),M22∗b(1:4),M23∗b(1:4),M24∗b(1:4),M25∗b(1:4),M26∗b(1:4)];
M(3,1:6) = [M13∗b(1:4),M23∗b(1:4),M33∗b(1:5),M34∗b(1:5),M35∗b(1:5),M36∗b(1:5)];
M(4,1:6) = [M14∗b(1:4),M24∗b(1:4),M34∗b(1:5),M44∗b(1:6),M45∗b(1:6),M46∗b(1:6)];
M(5,1:7) = [M15∗b(1:4),M25∗b(1:4),M35∗b(1:5),M45∗b(1:6),M55∗b(1:7),M56∗b(1:7),...
```

(continued on next page)

```
    Mjp2*b(5-2:5+2)];
M(6,1:8) = [M16*b(1:4),M26*b(1:4),M36*b(1:5),M46*b(1:6),M56*b(1:7),M66*b(1:8),...
    Mjp1*b(6-2:6+2),Mjp2*b(6-2:6+2)];
%intdrnal rows
for i = 7:n-6
M(i,i-2:i+2) = [Mjm2*b(i-2:i+2),Mjm1*b(i-2:i+2),Mj*b(i-2:i+2),Mjp1*b(i-2:i+2),Mjp2*b(i-2:i+2)];
end
M(n-5,n:-1:n-7) =
      [M16*b(n:-1:n-3),M26*b(n:-1:n-3),M36*b(n:-1:n-4),M46*b(n:-1:n-5),M56*b(n:-1:n-6),...
    M66*b(n:-1:n-7),Mjm1*b((n-5)-2:(n-5)+2),Mjm2*b((n-5)-2:(n-5)+2)];
M(n-4,n:-1:n-6) =
      [M15*b(n:-1:n-3),M25*b(n:-1:n-3),M35*b(n:-1:n-4),M45*b(n:-1:n-5),M55*b(n:-1:n-6),...
    M56*b(n:-1:n-6),Mjm2*b((n-4)-2:(n-4)+2)];
M(n-3,n:-1:n-5) =
      [M14*b(n:-1:n-3),M24*b(n:-1:n-3),M34*b(n:-1:n-4),M44*b(n:-1:n-5),M45*b(n:-1:n-5),...
    M46*b(n:-1:n-5)];
M(n-2,n:-1:n-5) =
      [M13*b(n:-1:n-3),M23*b(n:-1:n-3),M33*b(n:-1:n-4),M34*b(n:-1:n-4),M35*b(n:-1:n-4),...
    M36*b(n:-1:n-4)];
M(n-1,n:-1:n-5) =
      [M12*b(n:-1:n-3),M22*b(n:-1:n-3),M23*b(n:-1:n-3),M24*b(n:-1:n-3),M25*b(n:-1:n-3),...
    M26*b(n:-1:n-3)];
M(n,n:-1:n-5) =
      [M11*b(n:-1:n-3),M12*b(n:-1:n-3),M13*b(n:-1:n-3),M14*b(n:-1:n-3),M15*b(n:-1:n-3),...
    M16*b(n:-1:n-3)];
D2 = inv(H)*(-M+E*Db);
```

tions (the Euler equations), and two from the viscous portion of the equations. Thus, the full SAT has a general form:

$$\mathbf{SAT} = \mathbf{SAT}_I + \mathbf{SAT}_{v1} + \mathbf{SAT}_{v2}, \tag{C.1}$$

where

$$\mathbf{SAT} = \sigma_{0/N}\widehat{\mathsf{H}}\widehat{\mathsf{E}}_{0/N,\xi}(A_\xi^\pm)\left(\widehat{\mathbf{Q}} - \widehat{\mathbf{Q}}_2\right), \tag{C.2}$$

$$\mathbf{SAT} = \frac{\sigma_{v,0/N}}{Re}\widehat{\mathsf{H}}\mathsf{E}_{0,\xi}\left(\widehat{\mathbf{E}}_v - \widehat{\mathbf{E}}_2\right), \tag{C.3}$$

and

$$\mathbf{SAT}_{v,\text{int}} = \frac{\sigma_{v2}}{JRe}\widehat{\mathsf{H}}\mathsf{E}_{0,\xi}\mathsf{B}_{\xi,\text{int}}\left(\widehat{\mathbf{Q}}_v - \widehat{\mathbf{Q}}_{\text{target}}\right), \tag{C.4}$$

where, for stability, $\sigma_{0/N} = \mp 1, \sigma_{v,0/N} = \pm 1$, and $\sigma_{v2} \leqslant \frac{1}{2}$. The form of $\mathsf{B}_{\xi,\text{int}}$ is given as

$$\mathsf{B}_{\xi,\text{int}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ (-a_1u - a_2v - a_3w) & a_1 & a_2 & a_3 & 0 \\ (-a_2u - a_4v - a_5w) & a_2 & a_4 & a_5 & 0 \\ (-a_3u - a_5v - a_6w) & a_3 & a_5 & a_6 & 0 \\ b_{51} & b_{52} & b_{53} & b_{54} & a_7 \end{bmatrix}, \tag{C.5}$$

where

$$a_1 = \frac{\mu}{\rho}\left(\frac{4}{3}\xi_x^2 + \xi_y^2 + \xi_z^2\right), a_2 = \frac{\mu}{\rho}\left(\frac{1}{3}\xi_x\xi_y\right),$$

$$a_3 = \frac{\mu}{\rho}\left(\frac{1}{3}\xi_x\xi_z\right),$$

$$a_4 = \frac{\mu}{\rho}\left(\xi_x^2 + \frac{4}{3}\xi_y^2 + \xi_z^2\right), a_5 = \frac{\mu}{\rho}\left(\frac{1}{3}\xi_y\xi_z\right),$$

$$a_6 = \frac{\mu}{\rho}\left(\xi_x^2 + \xi_y^2 + \frac{4}{3}\xi_z^2\right),$$

$$a_7 = \frac{\mu}{\rho Pr}\left(\xi_x^2 + \xi_y^2 + \xi_z^2\right),$$

$$b_{52} = -a_7u + a_1u + a_2v + a_3w, b_{53} = -a_7u + a_2u + a_4v + a_5w,$$

$$b_{51} = a_7\left(-\frac{e}{\rho} + [u^2 + v^2 + w^2]\right) - a_1u^2 - a_4v^2 - a_6w^2 \\ - 2(a_2uv + a_3uw + a_5vw),$$

where Pr is the Prandtl number, μ is the viscosity, ρ is the density, u, v, w, are the x, y, and z, velocity components.

References

[1] Kreiss H-O, Oliger J. Comparison of accurate methods for the integration of hyperbolic equations. Tellus 1972;24(3):199–215.

[2] Swartz B, Wendroff B. The relative efficiency of finite difference and finite element methods. I: Hyperbolic problems and splines. SIAM J Num Anal 1974;11(5):979–93.

[3] De Rango S, Zingg DW. A high-order spatial discretization for turbulent aerodynamic computations. AIAA J 2001;39(7):1296–304.

[4] Ceze M, Fidkowski KJ. Pseudo-transient continuation, solution update methods, and CFL strategies for DG discretizations of the RANS-SA equations. AIAA Paper 2013:2013–686.

[5] Darmofal DL, Allmaras S, Yano M, Kudo J. Progress towards a higher-order adaptive solver for aerodynamics. AIAA Paper 2013:2013–871.

[6] Osusky M, Zingg DW. A parallel Newton–Krylov flow solver for the Navier–Stokes equations discretized using summation-by-parts operators. AIAA J 2013;51(12):2833–51.

[7] Shu C-W. High-order finite difference and finite volume WENO schemes and discontinuous Galerkin methods for CFD. Int J Comput Fluid Dynam 2003;17(2):107–18.

[8] Jurgens HM, Zingg DW. Numerical solution of the time-domain Maxwell equtions using high-accuracy finite-difference methods. SIAM J Sci Comput 2000;22(5).

[9] Hicken JE, Zingg DW. A parallel Newton–Krylov solver for the euler equations discretized using simultaneous approximation terms. AIAA J 2008;46(11):2773–86.
[10] Reist TA, Zingg DW. Aerodynamic shape optimization of a blended-wing-body regional transport for a short range mission. AIAA Paper 2013:2013–414.
[11] Svärd M, Nordström J. Review of summation-by-parts schemes for initial-boundary-value-problems; 2013. arXiv:1311.4984v1.
[12] Kreiss H-O, Scherer G. Finite element and finite difference methods for hyperbolic partial differential equations. In: Mathematical aspects of finite elements in partial differential equations. New York/London: Academic Press; 1974. p. 195–212.
[13] Gustafsson B. The convergence rate for difference approximations to mixed initial boundary value problems. Math Comput 1975;29(130):396–406.
[14] Kreiss H-O, Scherer G. On the existence of energy estimates for difference approximations for hyperbolic system. Tech. rep., Department of Information Technology, Uppsala University; 1977.
[15] Scherer G. On energy estimates for difference approximations to hyperbolic partial differential equations. Ph.D. thesis, Uppsala University; October 1977.
[16] Olsson P. High-order difference methods and dataparallel implementation. Ph.D. thesis, Uppsala University; 1992.
[17] Strand B. Summation by parts for finite difference approximations for d/dx. J Comput Phys 1994;110(1):47–67.
[18] Carpenter MH, Gottlieb D, Abarbanel S. Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes. J Comput Phys 1994;111(2):220–36.
[19] Funaro D. Domain decomposition methods for pseudo spectral approximations part I. Second order equations in one dimension. Numer Math 1987;52(3):329–44.
[20] Funaro D, Gottlieb D. A new method of imposing boundary conditions in pseudospectral approximations of hyperbolic equations. Math Comput 1988;51(184):599–613.
[21] Olsson P. Summation by parts, projections, and stability. I. Math Comput 1995;64(211):1035–65.
[22] Olsson P. Summation by parts, projections, and stability. II. Math Comput 1995;64(212):1473–93.
[23] Carpenter MH, Nordström J, Gottlieb D. A stable and conservative interface treatment of arbitrary spatial accuracy. J Comput Phys 1999;148(2):341–65.
[24] Nordström J, Carpenter MH. Boundary and interface conditions for high-order finite-difference methods applied to the Euler and Navier-Stokes equations. J Comput Phys 1999;148(2):621–45.
[25] Nordström J, Carpenter MH. High-order finite-difference methods, multidimensional linear problems, and curvilinear coordinates. J Comput Phys 2001;173(1):149–74.
[26] Mattsson K. Boundary procedures for summation-by-parts operators. J Sci Comput 2003;18(1):133–53.
[27] Svärd M, Carpenter MH, Nordström J. A stable high-order finite difference scheme for the compressible Navier–Stokes equations, far-field boundary conditions. J Comput Phys 2007;225(1):1020–38.
[28] Svärd M, Nordström J. A stable high-order finite difference scheme for the compressible Navier–Stokes equations: no-slip wall boundary conditions. J Comput Phys 2008;227(10):4805–24.
[29] Nordström J, Gong J, van der Weide E, Svärd M. A stable and conservative high order multi-block method for the compressible Navier–Stokes equations. J Comput Phys 2009;228(24):9020–35.
[30] Berg J, Nordström J. Stable robin solid wall boundary conditions for the Navier–Stokes equations. J Comput Phys 2011;230(19):7519–32.
[31] Diener P, Dorband EN, Schnetter E, Tiglio M. Optimized high-order derivative and dissipation operators satisfying summation by parts, and applications in three-dimensional multi-block evolutions. J Sci Comput 2007;32(1):109–45.
[32] Mattsson K, Nordström J. Summation by parts operators for finite difference approximations of second derivatives. J Comput Phys 2004;199:503–40.
[33] Svärd M, Nordström J. On the order of accuracy for difference approximation of initial-boundary value problems. J Comput Phys 2006;218(1):333–52.
[34] Mattsson K, Svärd M, Shoeybi M. Stable and accurate schemes for the compressible Navier–Stokes equations. J Comput Phys 2008;227(4):2293–316.
[35] Kamakoti R, Pantano C. High-order narrow stencil finite-difference approximations of second-derivatives involving variable coefficients. SIAM J Sci Comput 2009;31(6):4222–43.
[36] Mattsson K. Summation by parts operators for finite difference approximations of second-derivatives with variable coefficients. J Sci Comput 2012;51(3):650–82.
[37] Mattsson K, Svärd M, Nordström J. Stable and accurate artificial dissipation. J Sci Comput 2004;21(1):57–79.
[38] Nordström J. Conservative finite difference formulations, variable coefficients, energy estimates and artificial dissipation. J Sci Comput 2006;29(3):375–404.
[39] Olsson P, Oliger J. Energy and maximum norm estimates for nonlinear conservation laws. Tech. rep. 94–01, The Research Institute of Advanced Computer Science; 1994.
[40] Sandham ND, Li Q, Yee HC. Entropy splitting for high-order numerical simulation of compressible turbulence. J Comput Phys 2002;178(2):307–22.
[41] Yee HC, Vinokur M, Djomehri MJ. Entropy splitting and numerical dissipation. J Comput Phys 2000;162(1):33–81.
[42] Yee HC, Sjögreen B. Designing adaptive low-dissipative high order schemes for long-time integrations. Turbulent flow computation fluid mechanics and its applications, vol. 66. Springer; 2004. p. 141–98. Chapter 5.
[43] Fisher TC, Carpenter MH, Nordström J, Yamaleev NK. Discretely conservative finite-difference formulations for nonlinear conservation laws in split form: theory and boundary conditions. J Comput Phys 2013;234(1):353–75.
[44] Fisher TC, Carpenter MH. High-order entropy stable finite difference schemes for nonlinear conservation laws: finite domains. J Comput Phys 2013;252(1):518–57.
[45] Svärd M. On coordinate transformations for summation-by-parts operators. J Sci Comput 2004;20(1):29–42.
[46] Hicken JE, Zingg DW. Superconvergent functional estimates from summation-by-parts finite-difference discretizations. SIAM J Sci Comput 2011;33(2):893–922.
[47] Hicken JE, Zingg DW. Dual consistency and functional accuracy: a finite-difference perspective. J Comput Phys 2014;256(1):161–82.
[48] Hicken JE, Zingg DW. Summation-by-parts operators and high-order quadrature. J Comput Appl Math 2013;237(1):111–25.
[49] Berg J, Nordström J. Superconvergent functional output for time-dependent problems using finite differences on summation-by-parts. J Comput Phys 2012;231(20):6846–60.
[50] Carpenter MH, Nordström J, Gottlieb D. Revisiting and extending interface penalties for multi-domain summation-by-parts operators. J Sci Comput 2010;45(1–3):118–50.
[51] Mattsson K, Svärd M, Carpenter MH, Nordström J. High-order accurate computations for unsteady aerodynamics. Comput Fluids 2007;36(3):636–49.
[52] Svärd M, Lundberg J, Nordström J. A computational study of vortex-airfoil interaction using high-order finite difference methods. Comput Fluids 2010;39(8):1267–74.
[53] Osusky M, Boom PD, Del Rey Fernández DC, Zingg DW. An efficient Newton–Krylov–Shur parallel solution algorithm for the steady and unsteady Navier–Stokes equations. In: ICCFD7; 2012.
[54] Boom PD, Zingg DW. Time-accurate flow simulations using an efficient Newton–Krylov–Schur approach with high-order temporal and spatial discretization. AIAA paper 2013-0383; 2013.
[55] Sjögreen B, Yee HC. Grid convergence of high order methods for multiscale complex unsteady viscous compressible flows. J Comput Phys 2003;185(1):1–26.
[56] Lindström J, Nordström J. A stable and high-order accurate conjugate heat transfer problem. J Comput Phys 2010;229(14):5440–56.
[57] Nordström J, Berg J. Conjugate heat transfer for the unsteady compressible Navier–Stokes equations using a multi-block coupling. Comput Fluids 2013;72:20–9.
[58] Pettersson P, Iaccarino G, Nordström J. Numerical analysis of the Burgers equation in the presence of uncertainty. J Comput Phys 2009;228(22):8394–412.
[59] Pettersson P, Nordström J, Iaccarino G. Boundary procedures for the time-dependent Burgers equation under uncertainty. Acta Math Sci 2010;30B(2):539–50.
[60] Pettersson P, Doostan A, Nordström J. On stability and monotonicity requirements of finite difference approximations of stochastic conservation laws with random viscosity. Comput Methods Appl Mech Eng 2013;258(5):134–51.
[61] Koley U, Mishra S, Risebro NH, Svärd M. Higher order finite difference schemes for the magnetic induction equations. BIT Numer Math 2009;49(2):375–95.
[62] Mishra S, Svärd M. On stability of numerical schemes via forzen coefficients and magnetic induction equations. BIT Numer Math 2010;50(1):85–108.
[63] Koley U, Mishra S, Risebro NH, Svärd M. Higher-order finite difference schemes for the magnetic induction equations with resistivity. IMA J Numer Anal 2012;32(3):1173–93.
[64] Nordström J, Gustafsson R. High order finite difference approximations of electromagnetic wave propagation close to material discontinuities. J Sci Comput 2003;18(2):215–34.
[65] Mattsson K, Svärd M, Shoeybi M. Stable and compact finite difference schemes. Tech. rep., Centre for Turbulence Research; 2006.
[66] Mattsson K, Ham F, Iaccarino G. Stable and accurate wave-propagation in discontinuous media. J Comput Phys 2008;227(19):8753–67.
[67] Mattsson K, Ham F, Iaccarino G. Stable boundary treatment for the wave equation on second-order form. J Sci Comput 2009;41(3):366–83.
[68] Austin JM, Bodony DJ. Wave propogation in gaseous small-scale channel flows. Shock Waves 2011;21(6):547–57.
[69] Ostoich CM, Bodony DJ, Geubelle PH. Interaction of a mach 2.25 turbulent boundary layer with a fluttering panel using direct numerical simulation. Phys Fluids 2013;25.
[70] Zhang Q, Bodony DJ. Numerical investigation and modelling of acoustically excited flow through a circular orifice backed by a hexagonal cavity. J Fluid Mech 2012;693:367–401.
[71] Gustafsson B. High order difference methods for time dependent PDE. Springer; 2008.
[72] Gustafsson B, Kreiss H-O, Oliger J. Time-dependent problems and difference methods. Pure and applied mathematics. Wiley; 2013.
[73] Kozdon JE, Dunham EM, Nordström J. Interaction of waves with frictional interfaces using summation-by-parts difference operators: weak enforcement of nonlinear boundary conditions. J Sci Comput 2012;50(2):341–67.

[74] Carpenter MH, Gottlieb D. Spectral methods on arbitrary grids. J Comput Phys 1996;129(1):74–86.

[75] Gassner GJ. A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods. SIAM J Sci Comput 2013;35(3):A1233–53.

[76] Kreiss H-O, Lorenz J. Initial-boundary value problems and the Navier–Stokes equations. Classics in applied mathematics, vol. 47. SIAM; 2004.

[77] Sjögreen B, Yee HC. On tenth order central spatial schemes. In: Fifth international symposium on turbulence and shear flow phenomena; 2007.

[78] Mattsson K, Almquist M. A solution to the stability issues with block norm summation by parts operators. J Comput Phys 2013;15:418–42.

[79] Mattsson K, Almquist M, Carpenter MH. Optimal diagonal-norm SBP operators. Tech. rep. 016, Uppsala University; 2013.

[80] Del Rey Fernández DC, Zingg DW. High-order compact-stencil summation-by-parts operators for the second derivative with variable coefficients. In: ICCFD7-2803.

[81] Del Rey Fernández DC, Zingg DW. High-order compact-stencil summation-by-parts operators for the compressible Navier–Stokes equations. AIAA Paper 2013:2013–570.

[82] Hadamard J. Lectures on Cauchy's problem in linear partial differential equations. Silliman memorial lectures. Yale University Press; 1923.

[83] Pulliam TH, Zingg DW. Fundamental algorithms in computational fluid dynamics. Springer; 2014.

[84] Osusky M. A parallel Newton–Krylov–Schur algorithm for the Reynolds-averaged Navier–Stokes equations. Ph.D. thesis, University of Toronto, Institute for Aerospace Studies, 4925 Dufferin Street, Toronto, Ontario, Canada M3H 5T6; 2013.

[85] Svärd M, Mattsson K, Nordström J. Steady state computations using summation-by-parts operators. J Sci Comput 2005;24(1):79–95.

[86] Pulliam TH. Artificial dissipation models for the Euler equations. AIAA J 1986;24(12):1931–40.

[87] Swanson RC, Turkel E. On central-difference up wind schemes. J Comput Phys 1992;101(2):292–306.

[88] Vinokur M, Yee HC. Frontiers of computational fluid dynamics 2002. In: Frontiers of computational fluid dynamics. World Scientific; 2002. p. 129–63. No. 8, Chapter: Extension of efficient low dissipation high order schemes for 2-D curvilinear moving grids.

[89] Deng X, Min Y, Mao M, Liu H, Tu G, Zhang H. Further studies on geometric conservation law and applications to high-order finite difference schemes with stationary grids. J Comput Phys 2013;239:90–111.

[90] Svärd M. Personal communication; 2013.

[91] Arnold DN, Brezzi F, Cockburn B, Marini LD. Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J Numer Anal 2002;39(5):1749–79.

[92] Lu JC. An a posteriori error control framework for adaptive precision optimization using discontinuous Galerkin finite element method. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts; 2005.

[93] Berg J, Nordström J. On the impact of boundary conditions on dual consistent finite difference discretizations. J Comput Phys 2013;236:41–55.

[94] Hartmann R. Adjoint consistency analysis of discontinuous Galerkin discretizations. SIAM J Numer Anal 2007;45(6):2671–96. http://dx.doi.org/10.1137/060665117.

[95] Rannacher R. Adaptive Galerkin finite element methods for partial differential equations. J Comput Appl Math 2001;128(1–2):205–33. http://dx.doi.org/10.1016/S0377-042(00)00513-6.

[96] Fidkowski KJ, Darmofal DL. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. AIAA J 2011;49(4):673–94.

[97] Fidkowski KJ, Roe PL. An entropy adjoint approach to mesh refinement. SIAM J Sci Comput 2010;32(3):1261–87. http://dx.doi.org/10.1137/090759057.

[98] Venditti DA, Darmofal DL. Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow. J Comput Phys 2000;164(1):204–27. http://dx.doi.org/10.1006/jcph.2000.6600.

[99] Nemec M, Aftosmis MJ. Adjoint error estimation and adaptive refinement for embedded-boundary Cartesian meshes. In: 18th AIAA computational fluid dynamics conference, no. AIAA-2007-4187, Miami, Florida, United States; 2007.

[100] Hicken JE. Output error estimation for summation-by-parts finite-difference schemes. J Comput Phys 2013;231(9):3828–48.

[101] Del Rey Fernández DC, Boom PD, Zingg DW. A generalized framework for nodal first derivative summation-by-parts operators. J Comput Phys 2014;266(1):214–39. http://dx.doi.org/10.1016/j.jcp.2014.01.038.

[102] Tadmor E. Skew-selfadjoint form for systems of conservation laws. J Math Anal Appl 1984;103(2):428–42.

[103] Kitson A, McLachlan RI, Robidoux N. Skew-adjoint finite difference methods on nonuniform grids. New Zealand J Math 2003;32(2):139–59.

[104] Chiu EK-Y, Wang Q, Hu R, Jameson A. A conservative mesh-free scheme and generalized framework for conservation laws. SIAM J Sci Comput 2012;34(6):A2896–916.

[105] Reichert A, Heath MT, Bodony DJ. Energy stable numerical method for hyperbolic partial differential equations using overlapping domain decomposition. J Comput Phys 2012;231:5243–65.

[106] Chertock AE. Strict stability of high-order compact implicit finite-difference schemes: the role of boundary conditions for hyperbolic PDEs. Ph.D. thesis, Tel-Aviv University; 1998.

[107] Abarbanel SS, Chertock AE, Yefet A. Strict stability of high-order compact implicit finite-difference schemes: the role of boundary conditions for hyperbolic PDEs, I. J Comput Phys 2000;160:42–66.

[108] Abarbanel SS, Chertock AE, Yefet A. Strict stability of high-order compact implicit finite-difference schemes: the role of boundary conditions for hyperbolic PDEs, II. J Comput Phys 2000;160:67–86.

[109] Nordström J, Lundquist T. Summation-by-parts in time. J Comput Phys 2013;251:487–99.

[110] Mattsson K, Carpenter MH. Stable and accurate interpolation operators for high-order multiblock finite difference methods. SIAM J Sci Comput 2010;32(4):2298–320.

[111] Eriksson S, Abbas Q, Nordström J. A stable and conservative method for locally adapting the design order of finite difference schemes. J Comput Phys 2011;230(11):4216–31.

[112] Yamaleev NK, Carpenter MH. Third-order energy stable WENO scheme. J Comput Phys 2009;228(8):3025–47.

[113] Yamaleev NK, Carpenter MH. A systematic methodology for constructing high-order energy stable WENO schemes. J Comput Phys 2009;228(11):4248–72.

[114] Fisher TC, Carpenter MH, Yamaleev NK, Frankel SH. Boundary closures for fourth-order energy stable weighted essentially non-oscillatory finite-difference schemes. J Comput Phys 2011;230(10):3727–52.

[115] Carpenter MH, Fisher TC, Yamaleev NK. Boundary closures for sixth-order energy-stable weighted essentially non-oscillatory finite-difference schemes. In: Advances in applied mathematics, modeling, and computational science. Fields institute communications, vol. 66. US: Springer; 2013. p. 117–60.