Journal of Scientific Computing manuscript No.
(will be inserted by the editor)

Relative efficiency of finite-difference and discontinuous spectral-element summation-by-parts methods on distorted meshes

Pieter D. Boom · David C. Del Rey Fernández · David W. Zingg

Received: date / Accepted: date

Abstract Conventional wisdom suggests that high-order finite-difference methods are more efficient than high-order discontinuous spectral-element methods on smooth meshes, but less efficient as the mesh becomes increasingly distorted because of a significant loss of accuracy on such meshes. This paper investigates the influence of mesh distortion on the relative efficiency of different implementations of generalized summation-by-parts (GSBP) methods, with emphasis on comparing finite-difference and discontinuous spectral-element approaches. These include discretizations built using classical finite-difference SBP operators, with and without optimized boundary closures, as well as both Legendre-Gauss and Legendre-Gauss-Lobatto operators. The traditionally finite-difference operators are also applied as discontinuous spectral-element operators by selecting a fixed number of nodes per element and performing mesh refinement by increasing the number of elements rather than the number of mesh nodes. Using the linear convection equation and nonlinear Euler equations as models, solutions are obtained on meshes with different types and severity of distortion. Contrary to expectation, the results show that finite-difference implementations are no more sensitive to mesh distortion than discontinuous spectral-element implementations, maintaining their relative ef-

Pieter D. Boom
Department of Mechanical Engineering and Interdisciplinary Center for Construction and Building Materials, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
Tel.: +966-55-823-6308
E-mail: pieter.boom@kfupm.edu.sa

David C. Del Rey Fernández
Department of Applied Mathematics, University of Waterloo, Waterloo, ON, N2L 3G1, Canada
E-mail: ddelreyfernandez@uwaterloo.ca

David W. Zingg
University of Toronto Institute for Aerospace Studies, North York, ON, M3H 5T6, Canada
E-mail: david.zingg@utoronto.ca

ficiency in most cases. The results also show that the operators of Mattsson et al. (J. Comp. Phys., 264, 2014) with optimized boundary operators are often the most efficient for a given implementation strategy (finite-difference or discontinuous spectral-element). While their accuracy as finite-difference operators might be expected, their superior accuracy to LG and LGL nodal distributions when implemented as discontinuous spectral-element operators is not well known.

Keywords Summation-by-parts property · Finite-difference methods · Discontinuous spectral-element methods

Mathematics Subject Classification (2020) 65M06 · 65M60 · 65M70

1 Introduction

Summation-by-parts (SBP) is the discrete analogue of integration-by-parts. It can therefore be used to relate the inner product of discretized functions and partial differential equations over some domain to their solution along the boundary of that domain. This facilitates proofs of both linear and nonlinear stability for various discretization strategies. Numerical methods constructed to have the SBP property are classified as *SBP methods* (See review papers [19,4]). Two well-known strategies that can be used to implement SBP methods on complex domains are the multiblock finite-difference and discontinuous spectral-element approaches. A key characteristic of the finite-difference approach is the use of multiple instances of a repeating interior point operator. This requires a uniform nodal distribution away from boundaries in computational space, but allows the operator to grow arbitrarily large to meet resolution requirements. Classical SBP (CSBP) operators with a diagonal norm implemented in the traditional manner with repeating centered difference interior point operators of degree $2p$ and biased boundary point operators of minimum degree p provide an example. Closely related are the optimized operators of Mattsson et al. [14], which use the same repeating centered difference interior point operators but with a non-uniform nodal spacing and optimized coefficients near the boundary to minimize truncation error.

Alternatively, SBP methods can be implemented as spectral-element methods with a fixed number of nodes interior to an element, albeit without necessarily requiring basis functions or an underlying analytical representation of the solution in their construction [3]. This enables the use of non-uniform nodal distributions over the entire element to optimize the efficiency of the operator. However, it also requires new instances of the operator (and additional interfaces) to be added to meet resolution requirements. Operators constructed on the Legendre-Gauss (LG) and Legendre-Gauss-Lobatto (LGL) nodal distributions are popular examples of this approach [7]. It is interesting to note that SBP operators typically implemented in the traditional finite-difference manner, such as the CSBP and Mattsson et al. operators, can be implemented as spectral-element methods with a nearly arbitrary fixed number of interior nodes per element. For example, a single instance of the interior operator produces the minimum size of such operators.

Given the diverse nature of the manner in which SBP methods can be implemented, it is important to develop an understanding of the advantages of each and how their efficiency depends on the nature of the problem being solved. For example, one important factor relevant to the selection of a method is the smoothness of the mesh. If the computational domain is very regular, it is often possible to generate a smooth mesh in a straightforward manner. Under these circumstances one expects finite-difference methods, i.e. those with a repeating interior operator, to offer the lowest cost for a given accuracy (without adaptation for example). However, when the geometry under consideration is highly complex or the solution is expected to have sharp gradients, it can be challenging to generate a sufficiently smooth mesh, and a degree of mesh distortion must be anticipated. In this case, the use of discontinuous spectral-elements is expected to lead to better accuracy per unit cost [18]. The objective of this paper is to examine how the relative efficiency of finite-difference and discontinuous spectral-element SBP operators is affected by the degree of distortion of the mesh. The goal is to provide information useful to the selection of a method for a given problem or class of problems. Note here that we are leveraging the SBP property to minimize or eliminate the potential differences in robustness between finite-difference and spectral-element methods, allowing us to focus on efficiency.

This article is organized as follows: Sections 2 and 3 introduce basic notation and the model equations used for this analysis: the linear convection equation and nonlinear Euler equations. Generalized SBP (GSBP) theory applied to the model equations is presented in Section 4, along with simultaneous approximation terms (SATs) for weak enforcement of boundary conditions and interface coupling. Section 5 describes the different types of GSBP operators considered in this article as well as their finite-difference and discontinuous spectral-element implementations. Finally, Section 6 presents a discussion of the numerical results, and the conclusions of the study are summarized in Section 7.

2 Notation

In this article, continuous functions are represented with script uppercase characters and their restriction onto a mesh is represented by lower case bold characters, for example $\mathcal{U}(x,t) \in L^2\left([x_l, x_r] \times [0,T]\right)$ and $\boldsymbol{u} = \left[\mathcal{U}\left(x_1\right), \ldots, \mathcal{U}\left(x_N\right)\right]^{\mathrm{T}}$. Lower case bold characters are also used for general vectors, while matrices are denoted with uppercase sans-serif characters, for example H. Exponents should be understood as element-wise operations, for example $\boldsymbol{x}^k = \left[x_1^k, \ldots, x_N^k\right]^{\mathrm{T}}$, with the convention that $\boldsymbol{x}^k = \boldsymbol{0}$ for $k < 0$. For simplicity this article presents theory and results in two-dimensions, with the extension to three dimensions being straightforward.

3 Governing Equations

3.1 Linear Convection Equation

The study presented in this article is carried out predominantly using the two-dimensional[1] linear convection equation

$$\frac{\partial \mathcal{U}}{\partial t} + \beta_x \frac{\partial \mathcal{U}}{\partial x} + \beta_y \frac{\partial \mathcal{U}}{\partial y} = 0, \quad \forall (x, y) \in \mathit{\Omega}, \quad t \geq 0, \tag{1}$$

with constant convection velocities β_x and β_y, as well as initial and boundary conditions:

$$\begin{aligned} \mathcal{U}(x, y, 0) &= \mathcal{F}(x, y), \quad \forall (x, y) \in \mathit{\Omega}, \quad t \geq 0, \\ \mathcal{U}(x, y, t) &= \mathcal{B}_x(x, y, t), \quad \forall (x, y) \in \partial\mathit{\Omega}, \quad t \geq 0. \end{aligned}$$

The simulation domain $\mathit{\Omega}$ can be discretized directly or partitioned into K nonoverlapping blocks or elements $\mathit{\Omega}_i$. In either case, a time-invariant curvilinear coordinate transformation is applied to map the physical coordinates (x, y) of the domain, block, or element to regular orthogonal computational coordinates (ξ, η). The resulting equation can be written in strong conservation form as

$$\frac{\partial \left(\mathcal{J}^{-1}\mathcal{U}\right)}{\partial t} + \frac{\partial \left(\lambda_\xi \mathcal{U}\right)}{\partial \xi} + \frac{\partial \left(\lambda_\eta \mathcal{U}\right)}{\partial \eta} = 0, \tag{2}$$

where variable coefficients containing the metrics of the transformation are

$$\lambda_\xi = \beta_x \frac{\partial y}{\partial \eta} - \beta_y \frac{\partial x}{\partial \eta} \quad \text{and} \quad \lambda_\eta = -\beta_x \frac{\partial y}{\partial \xi} + \beta_y \frac{\partial x}{\partial \xi},$$

and the metric Jacobian is

$$\mathcal{J}^{-1} = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}.$$

This can be recast into the skew-symmetric form as follows:

$$\frac{\partial \left(\mathcal{J}^{-1}\mathcal{U}\right)}{\partial t} + \frac{1}{2}\frac{\partial \left(\lambda_\xi \mathcal{U}\right)}{\partial \xi} + \frac{1}{2}\frac{\partial \left(\lambda_\eta \mathcal{U}\right)}{\partial \eta} + \frac{\lambda_\xi}{2}\frac{\partial \mathcal{U}}{\partial \xi} + \frac{\lambda_\eta}{2}\frac{\partial \mathcal{U}}{\partial \eta} = 0, \tag{3}$$

which is the form used in this article. The skew-symmetric form is required in general for constructing provably stable semi-discrete forms [17, 16, 12, 5].

[1] We expect that three-dimensional results will exhibit similar trends.

3.2 Euler Equations

To validate whether the trends observed for linear PDEs also hold for nonlinear PDEs, some additional simulations are carried out using the two-dimensional Euler equations governing compressible inviscid fluid flow. In curvilinear coordinates, the Euler equations can be written in strong conservation form as:

$$\frac{\partial \mathcal{Q}}{\partial t} + \frac{\partial \mathcal{F}}{\partial \xi} + \frac{\partial \mathcal{G}}{\partial \eta} = 0, \tag{4}$$

where the solution and flux vector are:

$$\mathcal{Q} = \mathcal{J}^{-1}\begin{pmatrix} \rho \\ \rho\mathcal{U} \\ \rho\mathcal{V} \\ \mathcal{E} \end{pmatrix}, \mathcal{F} = \mathcal{J}^{-1}\begin{pmatrix} \rho\tilde{\mathcal{U}} \\ \rho\mathcal{U}\tilde{\mathcal{U}} + \mathcal{P}\frac{\partial \xi}{\partial x} \\ \rho\mathcal{V}\tilde{\mathcal{U}} + \mathcal{P}\frac{\partial \xi}{\partial y} \\ (\mathcal{E}+\mathcal{P})\tilde{\mathcal{U}} \end{pmatrix}, \mathcal{G} = \mathcal{J}^{-1}\begin{pmatrix} \rho\tilde{\mathcal{V}} \\ \rho\mathcal{U}\tilde{\mathcal{V}} + \mathcal{P}\frac{\partial \eta}{\partial x} \\ \rho\mathcal{V}\tilde{\mathcal{V}} + \mathcal{P}\frac{\partial \eta}{\partial y} \\ (\mathcal{E}+\mathcal{P})\tilde{\mathcal{V}} \end{pmatrix}, \tag{5}$$

ρ is the density, $\mathcal{U}$ and $\mathcal{V}$ are the flow velocities, $\mathcal{E}$ is the total energy per unit volume,

$$\mathcal{P} = (\gamma - 1)\left(\mathcal{E} - \tfrac{1}{2}\rho\left(\mathcal{U}^2 + \mathcal{V}^2\right)\right) \tag{6}$$

is the pressure derived from the ideal gas law, γ is the ratio of specific heat capacities, and $\tilde{\mathcal{U}} = \mathcal{U}\frac{\partial \xi}{\partial x} + \mathcal{V}\frac{\partial \xi}{\partial y}$ and $\tilde{\mathcal{V}} = \mathcal{U}\frac{\partial \eta}{\partial x} + \mathcal{V}\frac{\partial \eta}{\partial y}$ are the contravariant velocities.

4 Spatial discretization

To discretize the linear convection and Euler equations, we apply GSBP operators, along with SATs to weakly enforce boundary and interface conditions.

Consider first the definition of a generalized summation-by-parts (GSBP) operators for the first derivative applied to general nodal distributions [3]:

Definition 1 Summation-by-parts operator for the first derivative: A matrix operator, $\mathsf{D}_\xi \in \mathbb{R}^{N\times N}$, is an SBP operator of order and degree p approximating the derivative $\frac{\partial}{\partial \xi}$ on the nodal distribution $\boldsymbol{\xi} \in [\xi_l, \xi_r]$ if

1. $\mathsf{D}_\xi \boldsymbol{\xi}^k = \mathsf{H}_\xi^{-1}\mathsf{Q}_\xi \boldsymbol{\xi}^k = k\boldsymbol{\xi}^{k-1}, \quad k = 0, 1, \ldots, p;$
2. H_ξ, denoted the norm matrix, is symmetric positive definite;
3. $\mathsf{Q}_\xi + \mathsf{Q}_\xi^{\mathrm{T}} = \mathsf{E}_\xi$ where $\left(\boldsymbol{\xi}^i\right)^{\mathrm{T}} \mathsf{E}_\xi \boldsymbol{\xi}^j = \xi_r^{i+j} - \xi_l^{i+j}, \quad i,\ j = 0, 1, \ldots, r,\ r \geq p.$

While this definition allows for arbitrary symmetric positive definite norm matrices, the focus in this article will be on diagonal norms.

In addition, to develop proofs for linearly stable and conservative discretizations, appropriate implementation of boundary conditions and interface coupling between subdomains is critical. A common approach is to use SATs, which weakly impose these conditions. In this case, it is convenient to further decompose E_ξ from Definition 1 as [3]

$$\mathsf{E}_\xi = \boldsymbol{t}_{\xi_r}\boldsymbol{t}_{\xi_r}^{\mathrm{T}} - \boldsymbol{t}_{\xi_l}\boldsymbol{t}_{\xi_l}^{\mathrm{T}},$$

where

$$\boldsymbol{t}_{\xi_l}^{\mathrm{T}}\boldsymbol{\xi}^k = \xi_l^k, \quad \boldsymbol{t}_{\xi_r}\boldsymbol{\xi}^k = \xi_r^k, \quad k = 0, 1, \ldots, r.$$

Combining these definitions, the GSBP-SAT discretization of the skew-symmetric form of the linear convection equation (3) is presented in terms of two blocks with a shared interface along constant ξ. Here it is assumed that the two blocks are conforming and have the same nodal distribution in the η direction. The discretization in the left (L) block is

$$\begin{aligned}\frac{\mathrm{d}\mathsf{J}_l^{-1}\boldsymbol{u}_l}{\mathrm{d}t} &+ \frac{1}{2}\left(\mathsf{D}_{\xi_l}\otimes\mathsf{I}_\eta\right)\Lambda_{\xi_l}\boldsymbol{u}_l + \frac{1}{2}\left(\mathsf{I}_{\xi_l}\otimes\mathsf{D}_\eta\right)\Lambda_{\eta_l}\boldsymbol{u}_l \\ &+ \frac{1}{2}\Lambda_{\xi_l}\left(\mathsf{D}_{\xi_l}\otimes\mathsf{I}_\eta\right)\boldsymbol{u}_l + \frac{1}{2}\Lambda_{\eta_l}\left(\mathsf{I}_{\xi_l}\otimes\mathsf{D}_\eta\right)\boldsymbol{u}_l = \\ &\qquad \frac{1}{2}\left(\mathsf{H}_{\xi_l}^{-1}\boldsymbol{t}_{l,\xi_r}\otimes\mathit{\Sigma}_{\mathrm{l}}\right)\left[\left(\boldsymbol{t}_{l,\xi_r}^{\mathrm{T}}\otimes\mathsf{I}_\eta\right)\Lambda_{\xi_l}\boldsymbol{u}_l - \left(\boldsymbol{t}_{r,\xi_l}^{\mathrm{T}}\otimes\mathsf{I}_\eta\right)\Lambda_{\xi_r}\boldsymbol{u}_r\right].\end{aligned}$$

where $\mathit{\Sigma}_{\mathrm{l}}$ is a diagonal matrix of SAT coefficients, I_{ξ_l} and I_η are identity matrices the size of the number of nodes in the ξ_l and η directions, respectively, and $\otimes$ denotes the tenor product. The variable coefficients λ_ξ and λ_η, as well as the metric Jacobian, are constructed as follows

$$\begin{aligned}\Lambda_{\xi_l} &= \mathrm{diag}\left[\beta_x\left(\mathsf{I}_{\xi_l}\otimes\mathsf{D}_\eta\right)\boldsymbol{y}_l - \beta_y\left(\mathsf{I}_{\xi_l}\otimes\mathsf{D}_\eta\right)\boldsymbol{x}_l\right], \\ \Lambda_{\eta_l} &= \mathrm{diag}\left[-\beta_x\left(\mathsf{D}_{\xi_l}\otimes\mathsf{I}_\eta\right)\boldsymbol{y}_l + \beta_y\left(\mathsf{D}_{\xi_l}\otimes\mathsf{I}_\eta\right)\boldsymbol{x}_l\right], \\ \mathsf{J}_l^{-1} &= \mathrm{diag}\left[\left(\mathsf{D}_{\xi_l}\otimes\mathsf{I}_\eta\right)\boldsymbol{x}_l\odot\left(\mathsf{I}_{\xi_l}\otimes\mathsf{D}_\eta\right)\boldsymbol{y}_l - \left(\mathsf{I}_{\xi_l}\otimes\mathsf{D}_\eta\right)\boldsymbol{x}_l\odot\left(\mathsf{D}_{\xi_l}\otimes\mathsf{I}_\eta\right)\boldsymbol{y}_l,\right],\end{aligned}$$

where $\odot$ denotes the Hadamard product. Note that the vectors $\boldsymbol{x}_l$ and $\boldsymbol{y}_l$ contain the x and y locations of the nodes. Likewise, the discretization in the right (R) block is given as

$$\begin{aligned}\frac{\mathrm{d}\mathsf{J}_r^{-1}\boldsymbol{u}_r}{\mathrm{d}t} &+ \frac{1}{2}\left(\mathsf{D}_{\xi_r}\otimes\mathsf{I}_\eta\right)\Lambda_{\xi_r}\boldsymbol{u}_r + \frac{1}{2}\left(\mathsf{I}_{\xi_r}\otimes\mathsf{D}_\eta\right)\Lambda_{\eta_r}\boldsymbol{u}_r \\ &+ \frac{1}{2}\Lambda_{\xi_r}\left(\mathsf{D}_{\xi_r}\otimes\mathsf{I}_\eta\right)\boldsymbol{u}_r + \frac{1}{2}\Lambda_{\eta_r}\left(\mathsf{I}_{\xi_r}\otimes\mathsf{D}_\eta\right)\boldsymbol{u}_r = \\ &\qquad \frac{1}{2}\left(\mathsf{H}_{\xi_r}^{-1}\boldsymbol{t}_{r,\xi_r}\otimes\mathit{\Sigma}_{\mathrm{r}}\right)\left[\left(\boldsymbol{t}_{r,\xi_r}^{\mathrm{T}}\otimes\mathsf{I}_\eta\right)\Lambda_{\xi_r}\boldsymbol{u}_r - \left(\boldsymbol{t}_{l,\xi_l}^{\mathrm{T}}\otimes\mathsf{I}_\eta\right)\Lambda_{\xi_l}\boldsymbol{u}_l\right].\end{aligned}$$

There are two common choices for the coefficients in the SAT matrices $\mathit{\Sigma}_{\mathrm{l}}$ and $\mathit{\Sigma}_{\mathrm{r}}$, symmetric and upwind. Symmetric SATs introduce no dissipation but sometimes lead to reduced convergence rates [9,7,8]. Therefore, here we exclusively use upwind SATs in the form

$$\mathit{\Sigma}_{\mathrm{l}} = \frac{-\tilde{\Lambda}_{\xi_l} + |\tilde{\Lambda}_{\xi_l}|}{|\tilde{\Lambda}_{\xi_l}|} \quad \text{and} \quad \mathit{\Sigma}_{\mathrm{r}} = \frac{-\tilde{\Lambda}_{\xi_r} - |\tilde{\Lambda}_{\xi_r}|}{|\tilde{\Lambda}_{\xi_r}|},$$

where $\tilde{\Lambda}_{\xi_l} = \left(\boldsymbol{t}_{l,\xi_r}^{\mathrm{T}}\otimes\mathsf{I}_\eta\right)\Lambda_{\xi_l}\left(\boldsymbol{1}_{\xi_l}\otimes\mathsf{I}_\eta\right)$, $\tilde{\Lambda}_{\xi_r} = \left(\boldsymbol{t}_{r,\xi_r}^{\mathrm{T}}\otimes\mathsf{I}_\eta\right)\Lambda_{\xi_r}\left(\boldsymbol{1}_{\xi_r}\otimes\mathsf{I}_\eta\right)$, and $\boldsymbol{1}$ denotes a column vector of ones.

For the Euler equations in divergence form, the discretization in the left (L) block is

$$\frac{\mathrm{d}\mathsf{J}_l^{-1}\boldsymbol{q}_l}{\mathrm{d}t} + \left(\mathsf{D}_{\xi_l}\otimes\mathsf{I}_\eta\otimes\mathsf{I}_4\right)\mathsf{F}_{\xi_l} + \left(\mathsf{I}_{\xi_l}\otimes\mathsf{D}_\eta\otimes\mathsf{I}_4\right)\mathsf{G}_{\eta_l} = \frac{1}{2}\left(\mathsf{H}_{\xi_l}^{-1}\boldsymbol{t}_{l,\xi_r}\otimes\mathit{\Sigma}_{\mathrm{l}}\right)\left[\left(\boldsymbol{t}_{l,\xi_r}^{\mathrm{T}}\otimes\mathsf{I}_\eta\right)\boldsymbol{q}_l - \left(\boldsymbol{t}_{r,\xi_l}^{\mathrm{T}}\otimes\mathsf{I}_\eta\right)\boldsymbol{q}_r\right].$$

where F_{ξ_l} and G_{η_l} are the local fluxes evaluated with $\boldsymbol{q}_l$, and I_4 is a 4×4 identity matrix to account for the four solution variables per node. For simplicity we keep the Euler equations in divergence form since we would need an entropy stable implementation to formally guarantee stability.

The upwind SAT for the Euler equations has the form

$$\mathit{\Sigma}_{\mathrm{l}} = -\tilde{\mathsf{A}}_\xi + |\tilde{\mathsf{A}}_\xi|$$

with $\tilde{\mathsf{A}}_\xi = \frac{1}{2}\mathsf{A}_\xi\left[\left(\boldsymbol{t}_{l,\xi_r}^{\mathrm{T}}\otimes\mathsf{I}_\eta\right)\boldsymbol{q}_l + \left(\boldsymbol{t}_{r,\xi_l}^{\mathrm{T}}\otimes\mathsf{I}_\eta\right)\boldsymbol{q}_r\right]$, A_ξ being the flux Jacobian ($\mathcal{A} = \frac{\partial\mathcal{G}}{\partial\mathcal{Q}}$) evaluated at the Roe averaged state of the interface, and $|\tilde{\mathsf{A}}_\xi| = \mathsf{X}|\mathsf{\Lambda}|\mathsf{X}^{-1}$, where X is the matrix of right eigenvectors of $\tilde{\mathsf{A}}_\xi$ and $|\mathit{\Lambda}|$ a diagonal matrix containing the absolute values of the eigenvalues of $\tilde{\mathsf{A}}_\xi$. The discretization of the Euler equations in the right (R) block follows similarly, as in the case of the linear convection equation.

5 GSBP Operators and Implementations

Spatial discretizations using GSBP operators can be constructed in a number of different ways. This section describes the classes of operators considered in this article and their compatibility with two different implementation strategies, one typically associated with finite-difference methods, the other with finite-element methods, including discontinuous spectral-element methods, labeled here *FD* and *DSE*.

5.1 Finite-Difference Implementation

Definition 1 of a GSBP operator encompasses many known operators; however, it was originally developed as a generalization of classical finite-difference summation-by-parts (CSBP) operators. CSBP operators are defined on a regular computational mesh, using repeating centered finite-difference stencils of order $2p$ everywhere except near boundaries[2], where a fixed number of special biased finite-difference stencils are used of minimum order p. The quadrature rules often associated with these operators are Gregory rules, which are based

[2] For first derivative operators, the order and polynomial degree of the operator are the same.

on the composite trapezoidal rule with end corrections. The $p = 1$ CSBP operator is:

$$\mathsf{D}_x = \frac{1}{h}\begin{bmatrix} -1 & 1 & & & \\ -\frac{1}{2} & 0 & \frac{1}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{2} & 0 & \frac{1}{2} \\ & & & -1 & 1 \end{bmatrix}, \; \mathsf{H}_x = h\begin{bmatrix} \frac{1}{2} & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & \frac{1}{2} \end{bmatrix}, \; \mathsf{Q}_x = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & & & \\ -\frac{1}{2} & 0 & \frac{1}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{2} & 0 & \frac{1}{2} \\ & & & -\frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

where $h = \frac{x_R - x_l}{N-1}$ is the mesh spacing.

The minimum order p of the boundary stencils is a proven limit relative to the interior $2p$ order stencils; however, it is possible to reduce the magnitude of the leading truncation error terms [3,14,15,13]. This can be done by optimizing the number and location of boundary nodes, as well as the coefficients of the boundary stencils. This was first done by Mattsson et al. [14], who optimized the location of several nodes near the boundary to minimize truncation error. For example, one optimized $p = 1$ operator with boundary node locations $\boldsymbol{x} = \frac{1}{h}\left[0,\ 0.789,\ 1.748,\ 1.748+1,\ 1.748+2, \ldots\right]^T$ at the left boundary is defined as follows[3]:

$$\mathsf{D}_x = \frac{1}{h}\begin{bmatrix} -1.48 & 1.66 & -0.176 & & & \\ -0.572 & 0 & 0.572 & & & \\ 0.064 & -0.600 & 0.536 & & & \\ & -\frac{1}{2} & 0 & \frac{1}{2} & & \\ & & \ddots & \ddots & \ddots & \end{bmatrix}, \; \mathsf{H}_x = h\begin{bmatrix} 0.337 & & & & \\ & 0.978 & & & \\ & & 0.933 & & \\ & & & 1 & \\ & & & & \ddots \end{bmatrix},$$

$$\mathsf{Q}_x = \begin{bmatrix} -\frac{1}{2} & 0.559 & -0.059 & & & \\ -0.559 & 0 & 0.559 & & & \\ 0.059 & -0.559 & \frac{1}{2} & & & \\ & -\frac{1}{2} & 0 & \frac{1}{2} & & \\ & & \ddots & \ddots & \ddots & \end{bmatrix}.$$

where $h = \frac{x_R - x_l}{N-1.504}$ is the average mesh spacing and $N - 1.504$ is the location of the last node of the mesh in computational space.

Refinement for these classes of operators (CSBP, Mattsson et al.) is often achieved simply by increasing the instances of the repeating interior centered finite-difference operator. For complex geometries, the domain is typically subdivided into multiple subdomains, or blocks, both to enable a tensor-product structure in each block and possibly as part of a parallelization strategy. Creating a block topology that enables high mesh quality can be a difficult and time-consuming task. However, once a block topology has been developed and a structured mesh generated in each block, this approach has been shown to lead to excellent computational solution efficiency [18].

[3] The values presented here are rounded to 3 decimal places for the sake of space. Please refer to [14] for higher precision values of the operator's coefficients.

5.2 Discontinuous Spectral-Element Implementation

In each of the previous cases, the operator can be arbitrarily expanded by inserting additional instances of the repeating $2p$ order centered finite-difference point operator away from boundaries. However, many discontinuous spectral-element operators can also be described using Definition 1. The simplest are associated with Newton-Cotes quadratures, at least those with positive weights when using diagonal norms; however, these are not often used in practice.

Similar to the finite-difference operators, the location of nodes and the coefficients of the individual stencils can be optimized. If the merit function is quadrature strength, then one obtains the LG and LGL nodes, where in the latter case the first and last node are aligned with the element boundaries, and result in $2n-1$ and $2n-3$ quadrature rules (LG and LGL respectively), where n is number of quadrature points. These nodal distributions are advantageous for diagonal-norm SBP operators as they enable the construction of maximum degree operators (e.g., $p = n - 1$). An example is the three-node Legendre-Gauss operator on the nodal distribution $\boldsymbol{x} = \left[-\sqrt{\frac{3}{5}}, 0, \sqrt{\frac{3}{5}}\right]^T$ shown here rounded to three decimal places:

$$\mathsf{D}_x = \frac{1}{h}\begin{bmatrix} -3.873 & 5.164 & -1.291 \\ -1.291 & 0 & 1.291 \\ 1.291 & -5.164 & 3.873 \end{bmatrix}, \quad \mathsf{H}_x = \frac{h}{9}\begin{bmatrix} 5 & & \\ & 8 & \\ & & 5 \end{bmatrix},$$

$$\mathsf{Q}_x = \begin{bmatrix} -1.076 & 1.434 & -0.359 \\ -0.574 & 0 & 0.574 \\ 0.359 & -1.434 & 1.076 \end{bmatrix}, \quad \mathsf{E}_x = \begin{bmatrix} -2.152 & 0.860 & 0 \\ 0.860 & 0 & -0.860 \\ 0 & -0.860 & 2.152 \end{bmatrix}.$$

where $h = \frac{x_R - x_l}{2}$ is the average mesh spacing. A second example is the four-node Legendre-Gauss-Lobatto operator on the nodal distribution $\boldsymbol{x} = \frac{1}{h}\left[-1, -\frac{\sqrt{5}}{5}, \frac{\sqrt{5}}{5}, 1\right]^T$:

$$\mathsf{D}_x = \frac{1}{h}\begin{bmatrix} -3 & \frac{5+5\sqrt{5}}{4} & \frac{5-5\sqrt{5}}{4} & \frac{1}{2} \\ -\frac{1+\sqrt{5}}{4} & 0 & \frac{\sqrt{5}}{2} & \frac{1-\sqrt{5}}{4} \\ \frac{\sqrt{5}-1}{4} & -\frac{\sqrt{5}}{2} & 0 & \frac{\sqrt{5}+1}{4} \\ -\frac{1}{2} & \frac{5\sqrt{5}-5}{4} & -\frac{5+5\sqrt{5}}{4} & 3 \end{bmatrix}, \quad \mathsf{H}_x = h\begin{bmatrix} \frac{1}{6} & & & \\ & \frac{5}{6} & & \\ & & \frac{5}{6} & \\ & & & \frac{1}{6} \end{bmatrix},$$

$$\mathsf{Q}_x = \begin{bmatrix} -\frac{1}{2} & \frac{5+5\sqrt{5}}{24} & \frac{5-5\sqrt{5}}{24} & \frac{1}{12} \\ -\frac{5+5\sqrt{5}}{24} & 0 & \frac{5\sqrt{5}}{12} & \frac{5-5\sqrt{5}}{24} \\ \frac{5\sqrt{5}-5}{24} & -\frac{5\sqrt{5}}{12} & 0 & \frac{5+5\sqrt{5}}{24} \\ -\frac{1}{12} & \frac{5\sqrt{5}-5}{24} & -\frac{5+5\sqrt{5}}{24} & \frac{1}{2} \end{bmatrix}.$$

where $h = \frac{x_R - x_l}{2}$ is the average mesh spacing. Note that the CSBP and Mattsson et al. operators described in the previous section can also be used as discontinuous spectral-element operators by selecting a fixed number of interior operators in each subdomain.

In contrast to the finite-difference implementation, h-refinement with discontinuous spectral-element operators requires that the number of elements be increased. The number and topology of the subdomains in this case is largely dictated by the desired mesh density, as well as geometric complexity to a lesser extent. A consequence of this approach is that the number of interfaces increases rapidly with refinement. The GSBP operators are multi-valued at interfaces, increasing the number of solution points rapidly with discontinuous spectral-element implementations. This also requires an additional mechanism to couple the solution in adjacent subdomains, achieved weakly in this article using SATs.

6 Results and discussion

6.1 Linear convection

The results presented in this subsection are for the two-dimensional linear convection equation described in equation (1) with periodic boundary conditions initially on a unit square. The convection velocities are set to unit values $\beta_x = \beta_y = 1$ without loss of generality, and the initial condition chosen is a sine wave given by:

$$\mathcal{U}_0(x, y) = \sin(2\pi x) + \sin(2\pi y).$$

The computational domain is Cartesian with $\xi, \eta \subset [0, 1]$, with each block or element discretized with the nodal distribution associated with the applied operator. The properties of the operators evaluated can be found in Table 1.

Low-order Mattsson et al. operators ($p = 1-4$) are taken from the original 2014 paper [14] and the higher-order operators ($p = 5, 6$) from the subsequent 2018 paper [15], specifically those optimized for accuracy. The operator labeled “Circulant” is a circulant operator obtained by applying the centered finite-difference method without lower-order boundary schemes with the order selected to match the interior order of the SBP methods. This provides a best-case scenario for evaluating the error introduced by boundary operators; it is not intended to be seen as an option for most practical problems.

Typically after decomposing a domain into blocks or elements, significant effort is required to place submesh nodes. In multiblock approaches this can be done edge-by-edge by setting minimum and/or maximum node spacings along with stretching functions, and determining how these properties vary through the subdomain from one edge to other parallel edges. With operators like Mattsson et al.’s that have a non-uniform nodal distribution, an additional step is required to account for this non-uniformity [2]. One strategy is to fit the multiblock mesh with B-splines and redistribute the nodes according

Operator	$\min p$	$\max p$	$\min N$	Bndry node	Opt
Circulant	(2,4,6,8)	(2,4,6,8)	N/A	N/A	no
CSBP	(1,2,3,4,5,6)	(2,4,6,8,10,12)	(5,9,13,17,23,31)	yes	no
Mattsson et al. [14,15]	(1,2,3,4,5,6)	(2,4,6,8,10,12)	(5,9,13,17,21,25)	yes	yes
LGL	(1,2,3,4,5,6)	(1,2,3,4,5,6)	(2,3,4,5,6,7)	yes	yes
LG	(1,2,3,4,5,6)	(1,2,3,4,5,6)	(2,3,4,5,6,7)	no	yes

Table 1 Description of operators. "Bndry node" denotes whether the operator includes nodes at the boundary of the computational domain or element; and "Opt" denotes whether or not the nodal distribution of the operator in the computational domain is optimized (as opposed to uniform). p is used to denote the pointwise order or degree of the difference stencils at each node in the submesh an operator, and N is used to denote the total number of nodes in the submesh of an operator.

to the non-uniform nodal distribution. With discontinuous spectral-elements, submesh nodal placement can be done with high-order elements, fitting boundaries, and imposing mesh gradients [11].

In this paper, the distorted meshes are obtained by applying analytical perturbations to the computational mesh directly. These transformations are described later in Table 4. This approach enables the methods to be compared directly, without having to account for the variety and complexity of mesh generation strategies. The metrics of the transformations are computed numerically using the same operators used for discretization following the approach of Diener et al [6]. It is expected that the error introduced by the metric computation will be on the same order as the operator used for the discretization.

The solutions are integrated from $t = 0$ to $t = 1$ (one full period) using the standard explicit fourth-order Runge-Kutta time-marching method and 3.2×10^4 time steps for the finest meshes. The number of time steps is halved on successively coarser meshes. With this choice, the error from the time-marching method is found to be negligible relative to the error from the spatial discretization.

For each simulation, the solution error is evaluated using the global norm of the discretization, H_g, assembled from the local element norm matrices scaled by the appropriate transformation Jacobian on each element. Therefore, the error of the simulation is computed relative to the exact solution of the PDE as

$$||\boldsymbol{u}_g - \boldsymbol{u}_{\text{exact}}||_{\mathsf{H}_g} = \sqrt{(\boldsymbol{u}_g - \boldsymbol{u}_{\text{exact}})^T \mathsf{H}_g (\boldsymbol{u}_g - \boldsymbol{u}_{\text{exact}})},$$

where $\boldsymbol{u}_{\text{exact}}$ is the vector constructed by evaluating the analytical solution at mesh nodes.

6.1.1 Cartesian meshes

As a baseline, the operators and implementation strategies are applied on a uniform Cartesian mesh. This is the ideal case of a smooth mesh, allowing an evaluation of the performance degradation of each method as a function of the different mesh distortions. Tables 2 and 3 display rankings of the methods tested (higher in the table is better). Two separate columns are shown in each subtable ranking the methods for:

1. **A minimally stringent error tolerance:** the region of the convergence plots with the largest error, where the discretizations may not exhibit their design order; and
2. **The asymptotic behavior:** the region of the convergence plots where the mesh spacing is sufficiently small that the methods exhibit their design order, i.e. the error is dominated by the leading error term.

The tolerances are reduced as the order increases to account for the change in accuracy of the methods on the given meshes. The tables display the ranking based on the required 1) number of nodes[4]; 2) number of floating-point operations to evaluate the discrete equations; and 3) number of floating-point operations multiplied by spectral radius of the full discretization under consideration, including the specific operator used in the discretization and all of the SATs. The values used to determine the ranking in the tables are estimated values required to achieve a prescribed error tolerance, and are obtained via interpolation of the raw data. Some values are extrapolated from the data, though they are largely restricted to the optimized Mattsson et al. operators at the minimally stringent tolerance and other operators in the asymptotic region. The circulant operator is included as a best-case reference to quantify the effect of the numerical boundary operators and SATs. The use of floating point operations and spectral radius, rather than CPU time, is done to minimize the influence of software implementation choices on the results.

The results show that the optimized Mattsson et al. operators are nearly always the most efficient GSBP operator, independent of implementation. This is further supported by plotting the data (See Figure 1 for example or the *Supplementary Files*), which show that any lower ranking in the tables is due to a negligible difference, with two exceptions.

The first exception is the circulant discretizations, which do not have subdomain boundaries, eliminating the need for low-order boundary point operators and SATs. The discretizations are not SBP, nor of broad practical use; however, they highlight the penalty incurred by other methods at every subdomain boundary. This is the same reason the Mattsson et al. discretizations perform so well relative to other schemes: 1) they minimize the truncation error of the lowest-order boundary point operators; and 2) facilitate the use of larger elements, and therefore fewer SATs, in discontinuous spectral-element implementations. It is also the reason why finite-difference discretizations always become the most efficient, for a particular operator, in the asymptotic region: 1) fewer low-order point operators; and 2) fewer SATs.

[4] Figures show the square root of the number of nodes; accounting for the simulations being in two dimensions. This is to enable reference convergence slopes to be added to the figures.

# nodes		FLOPs		$\lambda_{\max}$ × FLOPs	
p = 1		p = 1		p = 1	
10^0	10^{-3}	10^0	10^{-3}	10^0	10^{-3}
▨ LG	▨ LG	CirI	CirI	CirI	CirI
CirI	CirI	▨ LG	Mat.	CSBP	Mat.
CSBP	Mat.	CSBP	CSBP	▨ CSBP	CSBP
▨ CSBP	CSBP	▨ CSBP	▨ LG	▨ Mat.	▨ Mat.
▨ Mat.	▨ Mat.	▨ Mat.	▨ Mat.	▨ LG	▨ LG
▨ LGL	▨ CSBP	Mat.	▨ CSBP	Mat.	▨ CSBP
Mat.	▨ LGL	▨ LGL	▨ LGL	▨ LGL	▨ LGL
p = 2		p = 2		p = 2	
10^{-1}	10^{-4}	10^{-1}	10^{-4}	10^{-1}	10^{-4}
▨ LG	▨ Mat.	CirI	CirI	CirI	CirI
Mat.	Mat.	▨ Mat.	Mat.	Mat.	Mat.
▨ Mat.	CirI	Mat.	▨ Mat.	▨ Mat.	▨ Mat.
CirI	▨ LG	CSBP	CSBP	▨ CSBP	CSBP
CSBP	CSBP	▨ CSBP	▨ CSBP	CSBP	▨ CSBP
▨ CSBP	▨ CSBP	▨ LGL	▨ LGL	▨ LGL	▨ LGL
▨ LGL	▨ LGL	▨ LG	▨ LG	▨ LG	▨ LG
p = 3		p = 3		p = 3	
10^{-3}	10^{-5}	10^{-3}	10^{-5}	10^{-3}	10^{-5}
CirI	CirI	CirI	CirI	CirI	CirI
▨ Mat.	Mat.	▨ Mat.	Mat.	▨ Mat.	Mat.
Mat.	▨ Mat.	Mat.	▨ Mat.	Mat.	▨ Mat.
▨ LG	▨ LG	▨ LGL	CSBP	CSBP	CSBP
▨ LGL	CSBP	CSBP	▨ LGL	▨ LGL	▨ CSBP
CSBP	▨ LGL	▨ CSBP	▨ CSBP	▨ CSBP	▨ LGL
▨ CSBP	▨ CSBP	▨ LG	▨ LG	▨ LG	▨ LG

□	FD
▨	DSE

Table 2 Efficiency ranking of operators ($p = 1$ to $p = 3$) applied to the linear convection equation simulation on Cartesian meshes relative to an estimate of the required: (left) number of nodes (middle) number of floating point operations to evaluate the RHS (right) number of floating point operations to evaluate the RHS multiplied by spectral radius of the discretization.

The second exception to the dominance of the Mattsson et al. operators is with respect to the required number of nodes for $p = 1$ LG, which is significantly lower than all other $p = 1$ methods at all error tolerances. However, this advantage quickly disappears once the additional computational work associated with the SATs is taken into account. SATs for the LG method require extrapolation of the solution from all nodes in adjacent elements since the operator does not include points on the boundary of the element. This additional cost makes the LG method no more efficient than the finite-difference operators (CSBP, Mattsson et al.) implemented as discontinuous spectral-elements.

Next we compare the operators other than the optimized Mattsson et al. operators. For $p = 1$, operators using the same implementation strategy generally have similar performance, with results generated with the finite-difference approach slightly more efficient than the discontinuous spectral-element approach. The two exceptions to this are the significantly fewer nodes required by $p = 1$ LG, though this does not translate to noticeably higher efficiency when considering the number of floating point operations, as noted earlier,

# nodes		FLOPs		$\lambda_{\max}$ × FLOPs	
p = 4		p = 4		p = 4	
10^{-4}	10^{-7}	10^{-4}	10^{-7}	10^{-4}	10^{-7}
CirI	CirI	CirI	CirI	CirI	CirI
▨ Mat.	Mat.	▨ Mat.	Mat.	▨ Mat.	Mat.
Mat.	▨ Mat.	Mat.	▨ Mat.	Mat.	▨ Mat.
▨ LG	▨ LG	▨ LGL	▨ LGL	▨ LGL	▨ LGL
▨ LGL	▨ LGL	CSBP	CSBP	CSBP	CSBP
CSBP	CSBP	▨ CSBP	▨ CSBP	▨ CSBP	▨ CSBP
▨ CSBP	▨ CSBP	▨ LG	▨ LG	▨ LG	▨ LG
p = 5		p = 5		p = 5	
10^{-5}	10^{-7}	10^{-5}	10^{-7}	10^{-5}	10^{-7}
▨ Mat.	Mat.	▨ Mat.	Mat.	▨ Mat.	Mat.
Mat.	▨ Mat.	Mat.	▨ Mat.	Mat.	▨ Mat.
▨ LG	▨ LG	▨ LGL	▨ LGL	▨ LGL	▨ LGL
▨ LGL	▨ LGL	▨ LG	▨ LG	▨ LG	▨ LG
p = 6		p = 6		p = 6	
10^{-8}	10^{-10}	10^{-8}	10^{-10}	10^{-8}	10^{-10}
▨ Mat.	Mat.	▨ Mat.	Mat.	▨ Mat.	Mat.
Mat.	▨ Mat.	Mat.	▨ Mat.	Mat.	▨ Mat.
▨ LG	▨ LG	▨ LGL	▨ LGL	▨ LGL	▨ LGL
▨ LGL	▨ LGL	▨ LG	▨ LG	CSBP	CSBP
CSBP	CSBP	CSBP	CSBP	▨ LG	▨ LG
▨ CSBP	▨ CSBP	▨ CSBP	▨ CSBP	▨ CSBP	▨ CSBP

□	FD
▨	DSE

Table 3 Efficiency ranking of operators ($p = 4$ to $p = 6$) applied to the linear convection equation simulation on Cartesian meshes relative to an estimate of the required: (left) number of nodes (middle) number of floating point operations to evaluate the RHS (right) number of floating point operations to evaluate the RHS multiplied by spectral radius of the discretization.

and consistently poor performance of $p = 1$ LGL relative to all metrics. The latter is primarily due to the fact that the operator only has $p = 1$ stencils, i.e. it has no higher order stencils on the interior of the operator, and the high number of SATs required relative to the size of the operator.

For $p = 2$, CSBP is the next most efficient operator after Mattsson et al., independent of implementation, followed by LGL. These two methods have more-or-less equal performance for $p = 3$, then LGL becomes more efficient than CSBP for $p = 4$ and above. LG remains the least efficient scheme for $p = 2$ through $p = 4$, then has similar performance to CSBP at $p = 5$ and above. This highlights the increasingly poor performance of CSBP schemes as the order increases.

The results also highlight the value of optimizing the location and coefficients of the boundary stencils. These modifications, though small in the finite-difference context, have an enormous impact on the performance of the schemes. The plotted data (See Figure 1 for example or the *Supplementary Files*), show that for these Cartesian simulations, Mattsson et al.'s operators initially achieve a rate of convergence higher than the expected $p+1$, creating significant separation from the other methods, before recovering the expected rate well into the most stringent accuracy region.

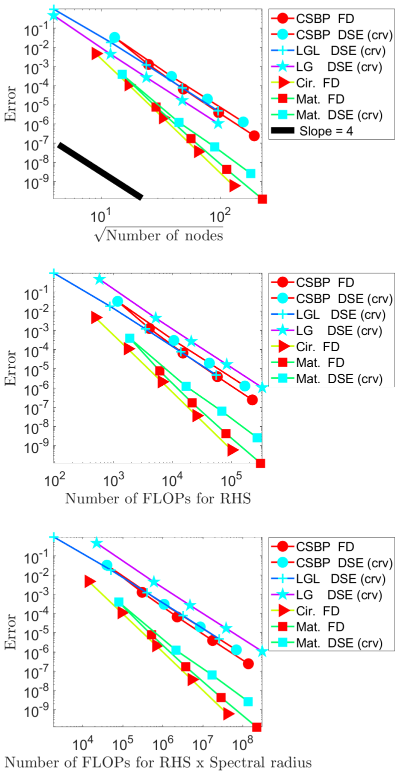

Fig. 1 Convergence of linear convection equation simulation on Cartesian mesh for $p = 3$ operators

6.1.2 Mesh distortions

We now investigate the impact of mesh distortion on the relative efficiency and robustness of the operators and implementation strategies using global transformations to create distorted meshes. Three different transformations, found in Table 4 and shown in Figure 2, are applied to accomplish this. The properties of meshes with 25 and 385 unique nodes in each spatial dimension are provided in Tables 5 and 6. This includes the maximum ratio of edge lengths in any one cell[5] of a mesh (aspect ratio), the maximum ratio of sequential edge lengths in a mesh, for example edges from node i to $(i+1)$ and $(i+1)$ to $(i+2)$ or edges from node j to $(j+1)$ and $(j+1)$ to $(j+2)$ (stretching), and the maximum ratio of cell diagonals in any one cell of a mesh (skewness).

[5] The term cell is used here to indicate the quadrilateral formed by 4 nodes in a mesh with indices (i,j), $(i+1,j)$, $(i+1,j+1)$, and $(i,j+1)$ - not a spectral-element.

Name	Transformation	α
Sinusoidal perturbation	$x = \xi + \alpha_1 \sin(2\pi\eta)$ $y = \eta + \lvert\alpha_1\rvert \sin(2\pi\xi)$	$\alpha_1 = \pm 0.15, \pm 0.158$
Sinusoidal stretching	$x = \xi + \alpha_2 \sin^2(2\pi\xi)$ $y = \eta + \alpha_2 \sin^2(2\pi\eta)$	$\alpha_2 = 0.15, 0.158$
Transformation from Ref. [5]	$x = \xi + \alpha_3 \sin(\pi\xi)\sin(\pi\eta)$ $y = \eta + \alpha_3 \exp(1-\eta)\sin(\pi\xi)\sin(\pi\eta)$	$\alpha_3 = 0.2$

Table 4 Global mesh transformations

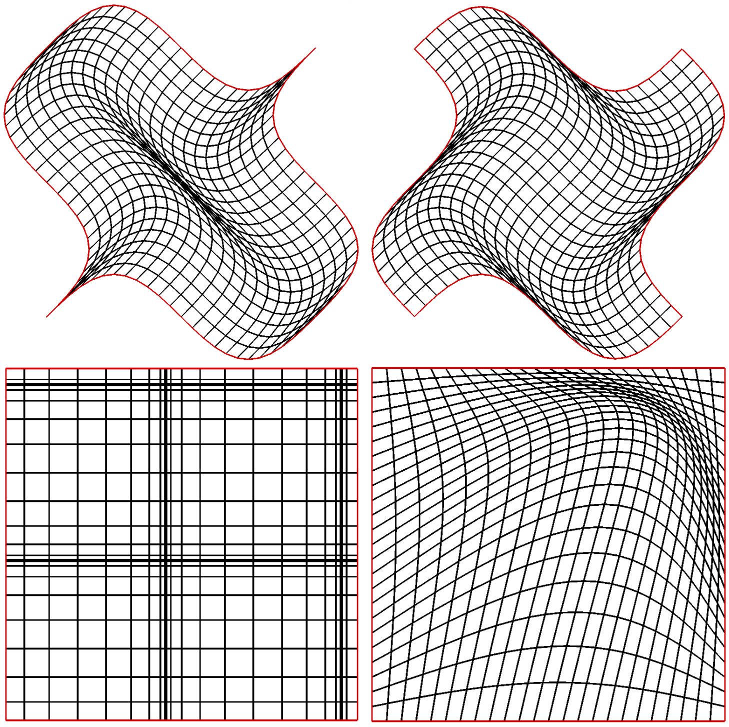

Fig. 2 Meshes for global transformations using 25-node CSBP nodal distribution: sinusoidal perturbation (top; $\alpha_1 = \pm 0.158$); sinusoidal stretching (bottom left $\alpha_2 = 0.158$) and transformation from Ref. [5] (bottom right $\alpha_3 = 0.2$)

The study was carried out for the discontinuous spectral-element operators using both curved submeshes, where the nodes within the element are placed according to the transformation applied to the nodal distribution associated with the operator, and linear submeshes, where only the corners of the element are placed according to the transformation, and the remaining nodes within the element are linearly interpolated according to the nodal distribution associated with the element.

Normally meshes are generated to mirror the complex features of the solution such that suitable resolution is achieved throughout the flow domain. In contrast, these meshes have been created solely to examine the effect of mesh

Transformation	Operator	Max Aspect ratio	Max Stretching	Max Skewness
Sinusoidal perturbation ($\alpha_1 \pm 0.158$)	CSBP	53.4	1.09	> 0.99
	Mattsson et al. (p4)	104.3	1.86	> 0.99
	LGL (p4) - curved submesh	78.5	2.07	> 0.99
	LGL (p4) - linear submesh	9.7	1.90	0.90
Sinusoidal stretching ($\alpha_2 \pm 0.158$)	CSBP	37.5	5.88	0
	Mattsson et al. (p4)	47.7	8.11	0
	LGL (p4) - curved submesh	48.5	8.71	0
	LGL (p4) - linear submesh	11.2	1.90	0
Transformation from Ref. [5] ($\alpha_3 \pm 0.2$)	CSBP	4.6	1.2	0.81
	Mattsson et al. (p4)	6.6	1.89	0.81
	LGL (p4) - curved submesh	8.5	2.27	0.82
	LGL (p4) - linear submesh	12.0	1.90	0.82

Table 5 Properties of meshes constructed for select operators with 25 unique nodes per spatial dimension

Transformation	Operator	Max Aspect ratio	Max Stretching	Max Skewness
Sinusoidal perturbation ($\alpha_1 \pm 0.158$)	CSBP	136.5	1.01	> 0.99
	Mattsson et al. (p4)	303.2	1.89	> 0.99
	LGL (p4) - curved submesh	249.5	1.91	> 0.99
	LGL (p4) - linear submesh	267.0	1.90	> 0.99
Sinusoidal stretching ($\alpha_2 \pm 0.158$)	CSBP	268.0	1.31	0
	Mattsson et al. (p4)	267.1	1.93	0
	LGL (p4) - curved submesh	513.9	2.46	0
	LGL (p4) - linear submesh	373.9	1.90	0
Transformation from Ref. [5] ($\alpha_3 \pm 0.2$)	CSBP	4.7	1.01	0.82
	Mattsson et al. (p4)	6.5	1.89	0.82
	LGL (p4) - curved submesh	8.8	1.92	0.82
	LGL (p4) - linear submesh	8.9	1.90	0.82

Table 6 Properties of meshes constructed for select operators with 385 unique nodes per spatial dimension

distortion on the accuracy of the operators under study for a uniformly and smoothly varying solution.

Tables 7 and 8 show the efficiency ranking of methods applied to meshes with extreme sinusoidal perturbations ($\alpha_1 = 0.158$), which is representative of results obtained for other sinusoidally perturbed meshes.

The first observation is that using a linear submesh in the discontinuous spectral-element strategy appears to have a significant advantage over curved submeshes. Viewing the plotted data (See Figure 3 for example or the *Supplemental Files*), the impact of this increases with order, but diminishes with increased refinement. These results are due in part to the linear submesh creating a more even distribution of submesh nodes (reduced clustering) and reducing the maximum angles between mesh lines. Since the solution is uniformly and smoothly varying, this makes sense - the largest and most distorted cells are increasingly regularized and the growth of the smallest cells has minimal impact on the integrated error. As the meshes are refined, the difference between the linear and curved submeshes diminishes, yielding similar efficiency.

# nodes		FLOPs		$\lambda_{\max}$ × FLOPs	
p = 1		p = 1		p = 1	
10^{-1}	10^{-3}	10^{-1}	10^{-3}	10^{-1}	10^{-3}
LG ░	LG ░	CirI □	CirI □	Mat. ░	Mat. ░
LG ▓	LG ▓	CSBP □	Mat. □	CSBP ░	CirI □
Mat. □	CirI □	Mat. □	CSBP □	CirI □	LG ░
CirI □	Mat. □	Mat. ░	LG ░	LG ░	Mat. □
Mat. ░	CSBP □	Mat. ▓	LG ▓	Mat. □	LG ▓
Mat. ▓	Mat. ▓	CSBP ░	Mat. ▓	LG ▓	CSBP □
CSBP □	Mat. ░	CSBP ▓	Mat. ░	Mat. ▓	Mat. ▓
CSBP ░	CSBP ░	LG ░	CSBP ░	CSBP □	CSBP ░
CSBP ▓	CSBP ▓	LG ▓	CSBP ▓	CSBP ▓	CSBP ▓
LGL ▓	LGL ▓	LGL ▓	LGL ▓	LGL ▓	LGL ▓
p = 2		p = 2		p = 2	
10^{-3}	$10^{-4.5}$	10^{-3}	$10^{-4.5}$	10^{-3}	$10^{-4.5}$
Mat. ░	Mat. ░	CirI □	CirI □	CSBP □	Mat. ░
CirI □	CirI □	Mat. ░	Mat. □	Mat. ░	CirI □
Mat. ▓	Mat. ▓	Mat. □	Mat. ░	CSBP ░	Mat. □
Mat. □	Mat. □	Mat. ▓	Mat. ▓	CirI □	Mat. ▓
LG ░	LG ░	CSBP □	CSBP □	Mat. □	CSBP ░
LG ▓	LG ▓	CSBP ░	CSBP ░	Mat. ▓	CSBP □
CSBP □	CSBP □	CSBP ▓	CSBP ▓	LG ░	CSBP ▓
CSBP ░	CSBP ░	LGL ░	LGL ░	LGL ░	LGL ▓
CSBP ▓	CSBP ▓	LGL ▓	LGL ▓	LG ▓	LG ░
LGL ░	LGL ░	LG ░	LG ░	LGL ▓	LGL ░
LGL ▓	LGL ▓	LG ▓	LG ▓	CSBP ▓	LG ▓

□	FD
░	DSE (lin)
▓	DSE (crv)

p = 3		p = 3		p = 3	
10^{-3}	10^{-5}	10^{-3}	10^{-5}	10^{-3}	10^{-5}
Mat. ░	CirI □	Mat. ░	CirI □	Mat. ░	Mat. ░
CirI □	Mat. ░	CirI □	Mat. ░	CirI □	CirI □
LG ░	Mat. □	Mat. □	Mat. □	LGL ░	Mat. □
Mat. □	Mat. ▓	Mat. ▓	Mat. ▓	CSBP ░	CSBP ░
Mat. ▓	LG ░	LGL ░	LGL ░	LG ░	Mat. ▓
LG ▓	LG ▓	CSBP ░	CSBP ░	Mat. □	LGL ░
LGL ░	LGL ░	LGL ▓	LGL ▓	Mat. ▓	LG ░
CSBP ░	CSBP ░	CSBP □	CSBP □	CSBP □	LGL ▓
LGL ▓	LGL ▓	LG ░	CSBP ▓	CSBP ▓	CSBP ▓
CSBP □	CSBP □	CSBP ▓	LG ░	LGL ▓	CSBP □
CSBP ▓	CSBP ▓	LG ▓	LG ▓	LG ▓	LG ▓

Table 7 Efficiency ranking of operators ($p = 1$ to $p = 3$) applied to the linear convection equation simulation on sinusoidally perturbed meshes ($\alpha_1 = 0.158$) relative to an estimate of the required: (left) number of nodes (middle) number of floating point operations to evaluate the RHS (right) number of floating point operations to evaluate the RHS multiplied by spectral radius of the discretization.

Another factor influencing the higher relative efficiency of linear submeshes is the lower spectral radius for a given number of elements, which is driven by the same mechanisms discussed above.

Using linear submeshes in the discontinuous spectral-element implementation yields a similar efficiency ranking as compared to the uniform undistorted mesh, Mattsson et al.'s operators being the most efficient, CSBP being next most efficient at low order and becoming increasingly poor with increased or-

# nodes		FLOPs		$\lambda_{\max}$ × FLOPs	
p = 4		p = 4		p = 4	
10^{-4}	10^{-6}	10^{-4}	10^{-6}	10^{-4}	10^{-6}
░ Mat.	□ CirI	░ Mat.	□ CirI	░ Mat.	░ Mat.
□ CirI	░ Mat.	□ CirI	░ Mat.	░ LGL	□ CirI
░ LG	□ Mat.	░ LGL	□ Mat.	□ CirI	░ LGL
□ Mat.	▓ Mat.	□ Mat.	▓ Mat.	░ CSBP	░ CSBP
▓ Mat.	░ LG	▓ Mat.	░ LGL	░ LG	□ Mat.
░ LGL	░ LGL	▓ LGL	▓ LGL	□ Mat.	▓ Mat.
▓ LG	▓ LG	░ CSBP	░ CSBP	▓ Mat.	░ LG
░ CSBP	░ CSBP	░ LG	░ LG	▓ LGL	▓ LGL
▓ LGL	▓ LGL	□ CSBP	□ CSBP	□ CSBP	▓ LG
□ CSBP	□ CSBP	▓ CSBP	▓ LG	▓ CSBP	□ CSBP
▓ CSBP	▓ CSBP	▓ LG	▓ CSBP	▓ LG	▓ CSBP

p = 5		p = 5		p = 5	
10^{-5}	10^{-7}	10^{-5}	10^{-7}	10^{-5}	10^{-7}
░ Mat.	░ Mat.	░ Mat.	░ Mat.	░ Mat.	░ Mat.
░ LG	░ LG	░ LGL	□ Mat.	░ LGL	░ LGL
░ LGL	□ Mat.	□ Mat.	░ LGL	░ LG	░ LG
□ Mat.	▓ Mat.	▓ Mat.	▓ Mat.	░ CSBP	□ Mat.
▓ Mat.	░ LGL	▓ LGL	▓ LGL	□ Mat.	▓ Mat.
▓ LG	▓ LG	░ LG	░ LG	▓ Mat.	░ CSBP
▓ LGL	▓ LGL	░ CSBP	░ CSBP	▓ LGL	▓ LGL
░ CSBP	░ CSBP	▓ LG	▓ LG	▓ LG	▓ LG
□ CSBP	□ CSBP	□ CSBP	□ CSBP	□ CSBP	□ CSBP
▓ CSBP	▓ CSBP	▓ CSBP	▓ CSBP	▓ CSBP	▓ CSBP

p = 6		p = 6		p = 6	
10^{-4}	10^{-7}	10^{-4}	10^{-7}	10^{-4}	10^{-7}
░ Mat.	░ Mat.	░ Mat.	░ Mat.	░ Mat.	░ Mat.
░ LG	░ LG	░ LGL	░ LGL	░ LGL	░ LGL
░ LGL	░ LGL	▓ LGL	□ Mat.	░ LG	░ LG
□ Mat.	▓ Mat.	▓ Mat.	▓ Mat.	░ CSBP	▓ Mat.
▓ Mat.	□ Mat.	□ Mat.	▓ LGL	□ Mat.	□ Mat.
▓ LG	▓ LG	░ LG	░ LG	▓ Mat.	░ CSBP
▓ LGL	▓ LGL	░ CSBP	░ CSBP	▓ LGL	▓ LGL
░ CSBP	░ CSBP	▓ LG	▓ LG	▓ LG	▓ LG
▓ CSBP	□ CSBP	□ CSBP	□ CSBP	▓ CSBP	□ CSBP
□ CSBP	▓ CSBP	▓ CSBP	▓ CSBP	□ CSBP	▓ CSBP

□	FD
░	DSE (lin)
▓	DSE (crv)

Table 8 Efficiency ranking of operators ($p = 4$ to $p = 6$) applied to the linear convection equation simulation on sinusoidally perturbed meshes ($\alpha_1 = 0.158$) relative to an estimate of the required: (left) number of nodes (middle) number of floating point operations to evaluate the RHS (right) number of floating point operations to evaluate the RHS multiplied by spectral radius of the discretization.

der, and LGL being more efficient than LG at and above $p = 2$, and CSBP at and above $p = 3$. Comparing the relative efficiency of operators using finite-difference or curved submesh discontinuous spectral-element implementations, they are noticeably less efficient than using linear submeshes, but follow the same trends as the Cartesian results.

Tables 9 and 10 show the efficiency ranking of methods applied to meshes with extreme sinusoidal stretching ($\alpha_2 = 0.158$). In this case, most trends are again similar to the cases above. A significant difference seen in the plot-

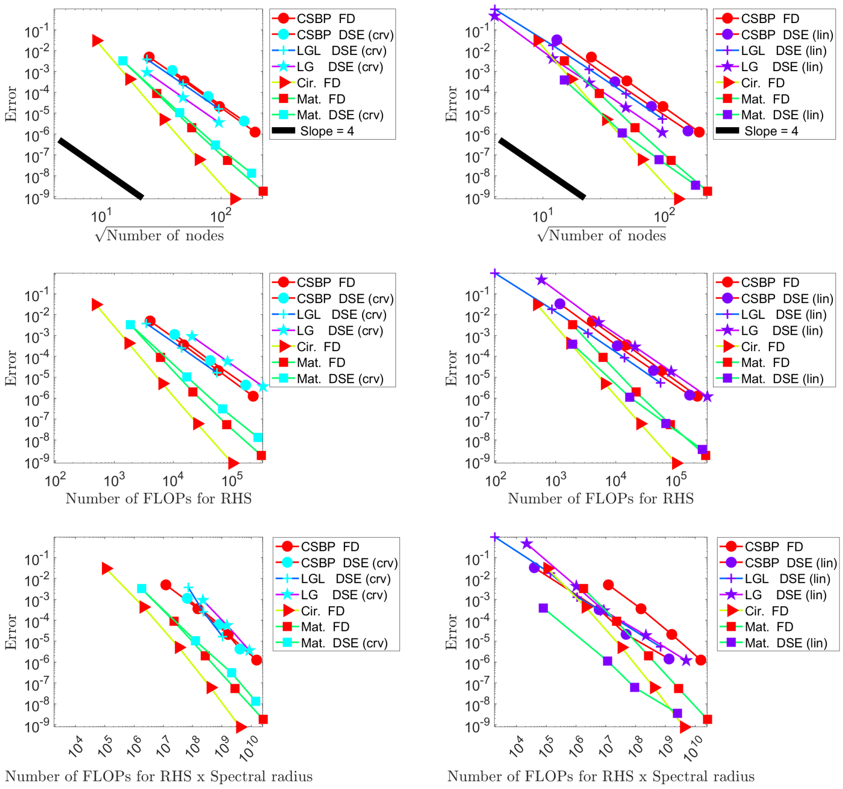

Fig. 3 Convergence of linear convection equation simulation on sinusoidally perturbed meshes ($\alpha_1 = 0.158$) for $p = 3$ operators

ted data (See Figure 4) is that $p = 6$ LGL using curved submeshes exhibits very similar efficiency to the Mattsson et al. schemes on curved elements. Furthermore, the impact of large finite-difference-compatible elements discussed previously is amplified, resulting in LGL becoming the most efficient scheme on curved elements at the least stringent error tolerances.

The final set of simulations using the transformation presented in Ref [5] produces similar results to the Cartesian case, with the exception of the circulant operators. The transformation introduces a slope discontinuity in the mesh at the periodic boundary, which severely reduces the convergence rate of the circulant operators that span the boundary. However, because the slope discontinuity is located at a subdomain boundary in both the finite-difference and discontinuous spectral-element strategies, and hence handled using a SAT, the other SBP schemes retain their default relative convergence performance. This has been demonstrated for multiblock finite-difference SBP discretiza-

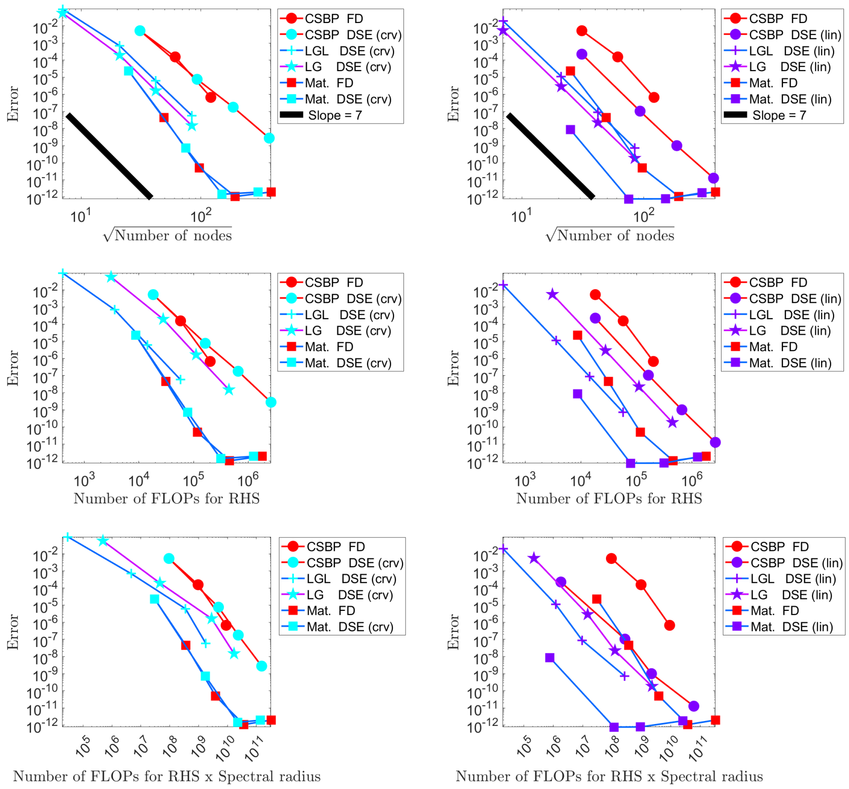

Fig. 4 Convergence of linear convection equation simulation on sinusoidally stretched meshes ($\alpha_2 = 0.158$) for $p = 6$ operators

tions in the past [10]. Therefore, as long as slope discontinuities in the mesh are located at block boundaries, the multiblock finite-difference implementations can handle this type of mesh as can a discontinuous spectral-element implementation.

# nodes		FLOPs		$\lambda_{\max}$ × FLOPs	
p = 1		p = 1		p = 1	
10^{-1}	10^{-3}	10^{-1}	10^{-3}	10^{-1}	10^{-3}
LG ░	LG ░	CirI □	CirI □	Mat. ░	Mat. ░
LG ▓	LG ▓	Mat. ░	CSBP □	CSBP ░	CirI □
Mat. ░	CirI □	CSBP □	Mat. □	CirI □	Mat. □
CirI □	Mat. □	Mat. □	Mat. ░	LG ░	LG ░
CSBP □	CSBP □	Mat. ▓	Mat. ▓	CSBP □	CSBP □
Mat. □	Mat. ░	CSBP ░	LG ░	Mat. □	LG ▓
Mat. ▓	Mat. ▓	CSBP ▓	LG ▓	Mat. ▓	CSBP ░
CSBP ░	CSBP ░	LG ░	CSBP ░	LG ▓	Mat. ▓
CSBP ▓	CSBP ▓	LG ▓	CSBP ▓	CSBP ▓	CSBP ▓
LGL ▓	LGL ▓	LGL ▓	LGL ▓	LGL ▓	LGL ▓
p = 2		p = 2		p = 2	
10^{-1}	10^{-4}	10^{-1}	10^{-4}	10^{-1}	10^{-4}
Mat. ░	Mat. ░	Mat. ░	Mat. ░	Mat. ░	Mat. ░
LG ░	Mat. ▓	CSBP ░	CirI □	CSBP ░	CSBP ░
LG ▓	Mat. □	CirI □	Mat. □	LG ░	Mat. ▓
CSBP ░	CirI □	Mat. □	Mat. ▓	LGL ░	CirI □
Mat. ▓	LG ░	Mat. ▓	CSBP □	LG ▓	Mat. □
Mat. □	CSBP □	CSBP □	CSBP ░	Mat. ▓	LG ░
CirI □	CSBP ░	LGL ░	CSBP ▓	Mat. □	CSBP □
LGL ░	LG ▓	CSBP ▓	LGL ░	CirI □	LGL ░
CSBP □	LGL ░	LG ░	LG ░	LGL ▓	CSBP ▓
CSBP ▓	CSBP ▓	LG ▓	LGL ▓	CSBP ▓	LG ▓
LGL ▓	LGL ▓	LGL ▓	LG ▓	CSBP □	LGL ▓
p = 3		p = 3		p = 3	
10^{-2}	10^{-4}	10^{-2}	10^{-4}	10^{-2}	10^{-4}
Mat. ░	Mat. ░	Mat. ░	Mat. ░	Mat. ░	Mat. ░
LG ░	CirI □	CSBP ░	CirI □	CSBP ░	CSBP ░
CSBP ░	Mat. □	CirI □	Mat. □	LGL ░	CirI □
CirI □	Mat. ▓	LGL ░	Mat. ▓	LG ░	Mat. □
Mat. □	LG ░	Mat. □	LGL ░	CirI □	LGL ░
LG ▓	LG ▓	Mat. ▓	CSBP □	Mat. □	Mat. ▓
Mat. ▓	LGL ░	CSBP □	CSBP ░	LGL ▓	LG ░
LGL ░	CSBP ░	LG ░	LGL ▓	Mat. ▓	CSBP □
CSBP □	CSBP □	LGL ▓	CSBP ▓	LG ▓	CSBP ▓
LGL ▓	LGL ▓	CSBP ▓	LG ░	CSBP ▓	LG ▓
CSBP ▓	CSBP ▓	LG ▓	LG ▓	CSBP □	LGL ▓

□	FD
░	DSE (lin)
▓	DSE (crv)

Table 9 Efficiency ranking of operators ($p = 1$ to $p = 3$) applied to the linear convection equation simulation on sinusoidally stretched meshes relative to an estimate of the required: (left) number of nodes (middle) number of floating point operations to evaluate the RHS (right) number of floating point operations to evaluate the RHS multiplied by spectral radius of the discretization.

# nodes		FLOPs		$\lambda_{\max}$ × FLOPs	
p = 4		p = 4		p = 4	
10^{-2}	10^{-6}	10^{-2}	10^{-6}	10^{-2}	10^{-6}
▨ Mat.	▨ Mat.	▨ Mat.	▨ Mat.	▨ Mat.	▨ Mat.
▨ LG	□ CirI	▨ LGL	□ CirI	▨ LGL	□ CirI
▨ LGL	□ Mat.	▨ CSBP	□ Mat.	▨ CSBP	▨ LGL
▨ CSBP	▨ LG	□ CirI	■ Mat.	▨ LG	□ Mat.
□ CirI	■ Mat.	▨ LG	▨ LGL	□ Mat.	▨ CSBP
□ Mat.	▨ LGL	□ Mat.	▨ CSBP	■ Mat.	■ Mat.
■ Mat.	■ LG	■ Mat.	▨ LG	□ CirI	▨ LG
■ LG	▨ CSBP	■ LGL	■ LGL	■ LGL	□ CSBP
■ LGL	□ CSBP	□ CSBP	□ CSBP	■ LG	■ LGL
□ CSBP	■ LGL	■ CSBP	■ CSBP	□ CSBP	■ CSBP
■ CSBP	■ CSBP	■ LG	■ LG	■ CSBP	■ LG

p = 5		p = 5		p = 5	
10^{-3}	10^{-5}	10^{-3}	10^{-5}	10^{-3}	10^{-5}
▨ Mat.	▨ Mat.	▨ Mat.	▨ Mat.	▨ Mat.	▨ Mat.
▨ LG	▨ LG	▨ LGL	▨ LGL	▨ LGL	▨ CSBP
▨ LGL	□ Mat.	□ Mat.	□ Mat.	▨ CSBP	▨ LGL
□ Mat.	■ Mat.	■ Mat.	■ Mat.	▨ LG	▨ LG
■ Mat.	▨ LGL	■ LGL	▨ CSBP	■ LGL	□ Mat.
▨ CSBP	▨ CSBP	▨ CSBP	■ LGL	□ Mat.	■ Mat.
■ LG	■ LG	▨ LG	▨ LG	■ Mat.	■ LGL
■ LGL	■ LGL	□ CSBP	□ CSBP	■ LG	■ LG
□ CSBP	□ CSBP	■ LG	■ LG	□ CSBP	□ CSBP
■ CSBP	■ CSBP	■ CSBP	■ CSBP	■ CSBP	■ CSBP

□	FD
▨	DSE (lin)
■	DSE (crv)

p = 6		p = 6		p = 6	
10^{-4}	10^{-7}	10^{-4}	10^{-7}	10^{-4}	10^{-7}
▨ Mat.	▨ Mat.	▨ Mat.	▨ Mat.	▨ Mat.	▨ Mat.
▨ LG	▨ LG	▨ LGL	▨ LGL	▨ LGL	▨ LGL
▨ LGL	▨ LGL	▨ LG	□ Mat.	▨ LG	▨ CSBP
□ Mat.	□ Mat.	□ Mat.	■ Mat.	▨ CSBP	▨ LG
■ Mat.	■ Mat.	■ Mat.	■ LGL	□ Mat.	□ Mat.
▨ CSBP	■ LG	■ LGL	▨ LG	■ Mat.	■ LGL
■ LG	■ LGL	▨ CSBP	▨ CSBP	■ LGL	■ Mat.
■ LGL	▨ CSBP	■ LG	■ LG	■ LG	■ LG
□ CSBP	□ CSBP	□ CSBP	□ CSBP	□ CSBP	□ CSBP
■ CSBP	■ CSBP	■ CSBP	■ CSBP	■ CSBP	■ CSBP

Table 10 Efficiency ranking of operators (p4 to p6) applied to the linear convection equation simulation on sinusoidally stretched meshes relative to an estimate of the required: (left) number of nodes (middle) number of floating point operations to evaluate the RHS (right) number of floating point operations to evaluate the RHS multiplied by spectral radius of the discretization.

6.1.3 Summary of linear convection results

The simulation results from the linear convection equations yield some expected and some unexpected results. First, while a finite-difference implementation was decidedly the most efficient scheme on Cartesian meshes, this is only true in general for the optimized schemes of Mattsson et al.. The results obtained using CSBP schemes become increasingly less efficient relative to LG/LGL, with the cross-over in efficiency happening around $p = 3$ or $p = 4$. Beyond this point LGL is the second most efficient scheme after the Mattsson et al. operator. There are free coefficients available in higher-order CSBP that can be potentially be optimized to improve these results.

With the introduction of mesh distortion, it is not surprising that the discontinuous spectral-element implementations with linear submeshes are the most efficient especially at looser error tolerances. However, the optimized schemes of Mattsson et al. applied with linear submesh elements are more efficient than LG/LGL schemes, a finding that is to some extent unexpected.

A further unexpected and significant result is the accuracy of the finite-difference implementations, even with the extreme mesh distortions tested. As long as slope discontinuities in the mesh are located at block interfaces, the multiblock finite-difference approach is as accurate as the discontinuous spectral-element approach on highly distorted meshes.

6.2 Euler equation results

To assess whether the results obtained with linear governing equations also hold for nonlinear equations, a smaller set of results is generated with the Euler equations. The simulation chosen is the periodic convection of an inviscid isotropic vortex. The domain is again square $[x, y] \in [0, 1]$ with periodic boundaries. The initial conditions are as follows:

$$\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} = \begin{bmatrix} \left(1 - \frac{1}{2} M^2 \beta^2 (\gamma - 1) e^{-r^2}\right)^{\frac{1}{\gamma-1}} \\ \rho M \left(1 - \beta \frac{y - Y_c}{R} e^{-r^2/2}\right) \\ \rho M \beta \frac{x - X_c}{R} e^{-r^2/2} \\ \frac{\rho^\gamma}{\gamma(\gamma-1)} + \frac{1}{2}\rho(u^2 + v^2) \end{bmatrix} \tag{7}$$

where ρ is the density, $M = 0.5$ is the Mach number, $\beta = 0.2$ is the vortex strength, $\gamma = 1.4$ is the ratio of specific heat capacities, $R = 0.05$ is the characteristic radius of the vortex, and $X_c = Y_c = 0.5$ are the coordinates of the vortex center. Time integration is as before with the linear convection equation simulations, integrated from $t = 0$ to $t = 2$ (one full period).

Operators of order $p = 1$ to $p = 4$ are compared, and the spectral radius was not taken into account (only the number of nodes and number of floating point operations to evaluate the right-hand side). Error is evaluated using the global norm of the discretization based on the difference in volumetric

x-momentum (ρu) component of the solution. In this case, many of the trends described earlier still hold (See Figure 5 for example or the *Supplementary Files*). For example,

- With $p = 1$, LG often requires the smallest number of nodes, but rarely the smallest number of floating point evaluations to evaluate the right-hand side.
- The finite-difference implemenations are often more efficient than the discontinuous spectral-element approach with curved submeshes for a given operator based on the relative to number of flops.
- The discontinuous spectral-element approach with linear submeshes has an advantage on the coarsest meshes, as observed previously. It also is able to handle the most extreme distortions without having to lower the time step size, indicating that the spectral radius of these simulations may be lower as was seen for the linear convection equation. Also as above, this advantage diminished with mesh refinement
- For a given implementation, Mattsson et al.'s optimized operators were nearly always the most efficient with respect to the number of flops
  - The second most efficient was CSBP for $p \leq 2$ and LGL for $p \geq 3$

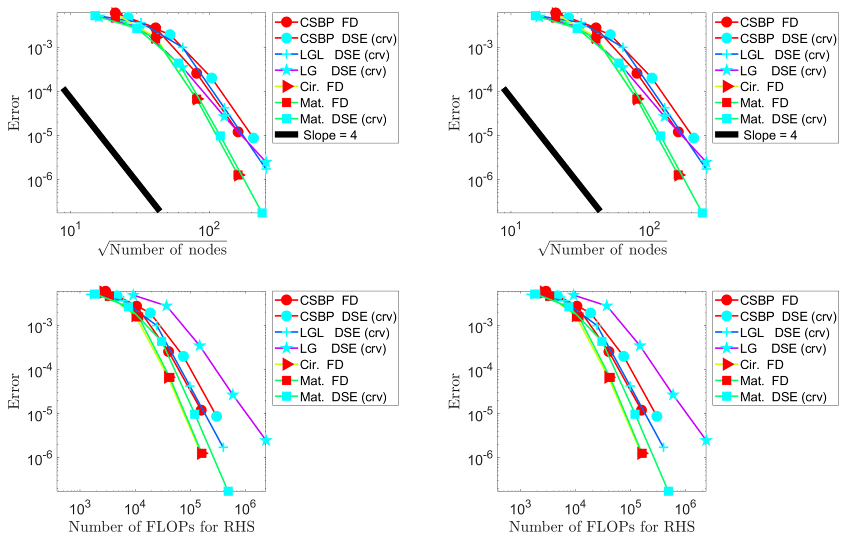

Fig. 5 Convergence of Euler's equation simulation on sinusoidally perturbed meshes ($\alpha_1 = 0.15$) for p3 operators

There are some difference that can be observed in the Euler equation results, which are as follows (See Figure 6 and the *Supplementary Files*):

- CSBP remains more efficient than LG relative to flops, even at $p = 4$, where previously LG was equally efficient or marginally more efficient.

Name	Transformation
Discontinuous stretching	$x = \xi$ $y = \begin{cases} 2\eta^* \dfrac{1}{\alpha_4} + 0.5 & \text{for } \lvert\eta^*\rvert \le 0.25 \\ 2\eta^* \left(1 - \dfrac{1}{\alpha_4}\right) - \dfrac{\eta^*}{\lvert\eta^*\rvert}\left(\dfrac{1}{\alpha_4} - 0.5\right) + 0.5 & \text{otherwise} \end{cases}$
Exponential stretching	$x = \xi$ $y = \dfrac{\eta^*}{\lvert\eta^*\rvert}\left(\dfrac{e^{\alpha_5\lvert\eta^*\rvert} - 1}{e^{\alpha_5} - 1}\right) + 0.5$

Table 11 Additional global transformations, where $\eta^* = \eta - 0.5$

- LGL $p = 4$ is often as efficient (or even marginally more efficient) than Mattsson et al.'s operators at the coarsest mesh levels. Mattsson et al.'s operators do eventually become more efficient at more stringent accuracy tolerances, but the advantage is delayed.

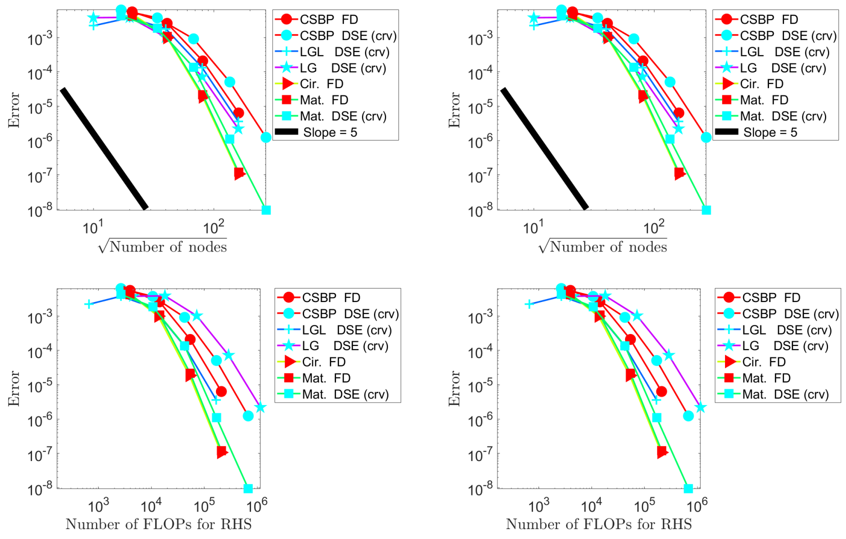

Fig. 6 Convergence of Euler's equation simulation on sinusoidally perturbed meshes ($\alpha_1 = 0.15$) for p4 operators

Some additional simulations are conducted using a discontinuous and exponential mesh stretching. The transformations used to impose this stretching are described in Tables 11 and 12, with samples shown in Figure 7.

In the discontinuous case, the finite-difference implementations do not fail but suffer a significant reduction in convergence rate, especially as the order increases (See Figure 8 or the *Supplemental Files*). This is to be expected because the mesh metrics are computed from the nodal locations in the mesh –

Transformation	Operator	Max Aspect ratio	Max Stretching	Max Skewness
Discontinuous stretching ($\alpha_4 = 3$)	CSBP	2	3	0
Exponential stretching ($\alpha_5 = 4$) [25 unique nodes]	CSBP	11.29	1.4	0
Exponential stretching ($\alpha_5 = 4$) [385 unique nodes]	CSBP	13.26	1.02	0

Table 12 Properties of meshes constructed for select operators

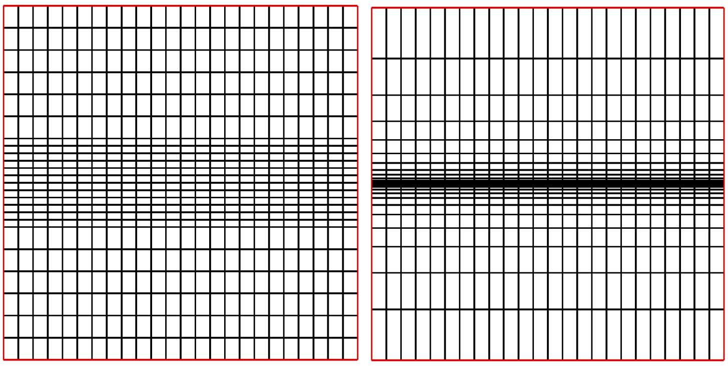

Fig. 7 Meshes for global transformations using 25 node CSBP nodal distribution: discontinuous stretching (left); and exponential stretching (right)

a discontinuity in mesh spacing will limit the accuracy of the metric computations. However, if a block boundary is introduced at the discontinuity, the strategy maintains the prescribed convergence rate, as well as a higher overall efficiency relative to flops observed in other simulations - this is indicated in the plots as FD ++SATs.

In the case of exponential mesh stretching, all simulations maintained a similar efficiency ranking to previous (non-discontinuous) results.

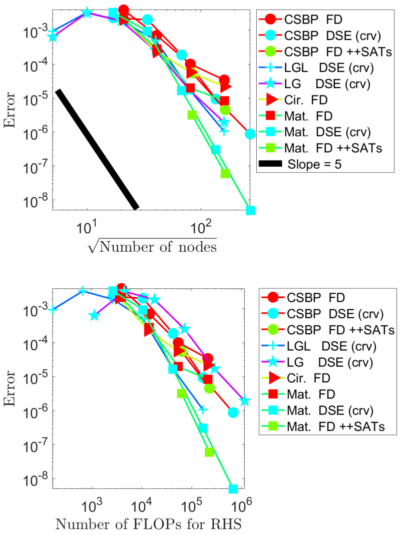

Fig. 8 Convergence of Euler's equation simulation on discontinuously stretched meshes ($\alpha_4 = 3$) for p4 operators. FD ++SATs indicates that a block boundary was introduced at the mesh discontinuity.

6.2.1 Summary of Euler equation results

In general, the results and conclusions obtained with the linear convection equations hold for the Euler equations. The relative efficiency rankings of the methods are nearly the same in all cases, even if the margins are different. This varies a little bit with the higher-order simulations where LGL is more competitive with Mattsson et al.'s operators and CSBP performs better than LG compared with previous results.

Extending the simulations to consider different types of mesh stretching functions again yielded very similar ranking of the operators for a given implementation strategy. In this case, finite-difference implementations require that discontinuities be located at block interfaces to maintain a high relative efficiency. Furthermore, the extreme exponential stretching did finally have an effect on the relative performance of the finite-difference implementations. While not the most efficient in all cases, it did not fail and was still competitive with LG/LGL discontinuous spectral-element schemes.

6.3 Other considerations

There are many different aspects to consider when choosing a discretization strategy for a new or upgraded solver. This paper concentrates on one aspect, the impact of severe mesh distortion on the accuracy and efficiency of two distinct strategies, finite-difference and discontinuous spectral-element. Other potential considerations include: generation of block or element topologies; and placement of submesh nodes; whether or not volume dissipation is needed; relative accuracy of superconvergent functionals; construction, implementation and impact of higher order derivatives; whether or not there is a benefit from having basis functions, i.e. an analytical representation of the solution as opposed to solely nodal values; mesh adaptivity strategies; representation of boundaries; and many more. Particularly relevant to the results presented in this paper, it has previously been shown that current volume dissipation operators can have a significant impact on the accuracy of results obtained with Mattsson et al. operators, and that the operators provide little or no benefit with respect to functional accuracy [1].

7 Conclusions

We have compared the accuracy and efficiency of several high-order operators with the SBP property on meshes with varying degrees of distortion for the linear convection and Euler equations in two dimensions. Some of the operators studied have been applied as multi-block finite-difference methods where mesh refinement is achieved by increasing the number of mesh nodes per block with a fixed number of blocks. All of the operators have been applied in a discontinuous spectral-element manner where mesh refinement is achieved by increasing the number of elements with a fixed number of degrees of freedom per element. For operators that have a variable number of submesh nodes (traditionally applied as finite-difference operators), the discontinuous spectral-element operators of minimum size are obtained by using the minimum possible number of submesh nodes in each element, which typically includes the point operators that have been modified due to their proximity to the element boundary and one instance of the interior point operator. The objective of this study is to understand how mesh distortion affects the accuracy and efficiency of such methods, in particular how mesh distortion affects the relative accuracy of finite-difference approaches compared to discontinuous spectral-element approaches, where the degrees of freedom are internal to the element. It was anticipated that the latter would become increasingly advantageous as mesh distortion increases.

Our results show that the accuracy of the finite-difference approach is not as sensitive to mesh distortion as expected and hence this approach is competitive with the discontinuous spectral-element approach even on highly distorted meshes. This conclusion holds when the mesh, although highly distorted, can be refined in a continuous manner. If the mesh contains a discontinuity, the conclusion holds if a block interface is located at the discontinuity.

A second conclusion of our study is that the optimized operators of Mattsson et al. are very efficient whether applied as finite-difference operators or as discontinuous spectral-element operators of minimum size. While their accuracy as finite-difference operators may be expected, their superior accuracy to

LG and LGL nodal distributions when implemented as discontinous spectral-element operators is surprising.

Finally, given that the multiblock finite-difference approach characterized by blocks of arbitrary size and the discontinuous spectral-element approach characterized by elements of fixed size for a given degree can be comparably efficient, this suggests that hybrid approaches combining the two could be optimal. For example, the number of blocks or elements affects the amount of numerical dissipation resulting, for example, from upwind SATs, and a particular block size or operator degree may be optimal for a given computer architecture. Therefore, the flexibility provided by the generalize summation-by-parts framework, specifically with multiblock finite-difference which decouples block size, i.e. the number of nodes in a block, from the operator degree, could be exploited to tailor the global operator for a given problem and computer architecture.

Conflict of interest

The authors have no competing interests to declare that are relevant to the content of this article.

Data availability

The authors declare that the data supporting the findings of this study are available within the paper and its supplementary files in graphical form. Raw data can be provided from the corresponding author on reasonable request.

References

1. Craig Penner, D.A.: Development and investigation of accurate high-order generalized summation-by-parts discretizations for computational fluid dynamics. Ph.D. thesis, University of Toronto (2022)
2. Craig Penner, D.A., Zingg, D.W.: Accurate high-order tensor-product generalized summation-by-parts discretizations of hyperbolic conservation laws: General curved domains and functional superconvergence. Journal Scientific Computing **93**(2) (2022)
3. Del Rey Fernández, D.C., Boom, P.D., Zingg, D.W.: A generalized framework for nodal first derivative summation-by-parts operators. Journal of Computational Physics **266**(1), 214–239 (2014)
4. Del Rey Fernández, D.C., Hicken, J.E., Zingg, D.W.: Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. Computers & Fluids **95**(22), 171–196 (2014)
5. Del Rey Fernández, D.C., Hicken, J.E., Zingg, D.W.: Simultaneous approximation terms for multidimensional summation-by-parts operators. Journal of Scientific Computing **75**(1), 83–110 (2018)
6. Diener, P., Dorband, E.N., Schnetter, E., Tiglio, M.: Optimized high-order derivative and dissipation operators satisfying summation by parts, and applications in three-dimensional multi-block evolutions. Journal of Scientific Computing **32**(1), 109–145 (2007)

7. Gassner, G.J.: A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods. SIAM Journal on Scientific Computing **35**(3), A1233–A1253 (2013)
8. Hesthaven, J.S., Warburton, T.: Nodal Discontinuous Galerkin Methods Algorithms, Analysis, and Applications. Springer (2008)
9. Hicken, J.E., Del Rey Fernández, D.C., Zingg, D.W.: Multi-dimensional summation-by-parts operators: General theory and application to simplex elements. SIAM Journal on Scientific Computing **38**(4), A1935–A1958 (2016)
10. Hicken, J.E., Zingg, D.W.: Parallel Newton-Krylov solver for the Euler equations discretized using simultaneous approximation terms. AIAA Journal **46**(11), 2773–2786 (2008)
11. Hindenlang, F.: Mesh curving techniques for high order parallel simulations on unstructured meshes. Ph.D. thesis, University of Stuttgart (2014)
12. Kopriva, D.A., Gassner, G.J.: An energy stable discontinuous Galerkin spectral element discretization for variable coefficient advection problems. SIAM Journal on Scientific Computing **4**(36), A2076–A2099 (2014)
13. Linders, V., Lundquist, T., Nordström, J.: On the order of accuracy of finite difference operators on diagonal norm based summation-by-parts form. SIAM Journal on Numerical Analysis **56**(2), 1048–1063 (2018)
14. Mattsson, K., Almquist, M., Carpenter, M.H.: Optimal diagonal-norm SBP operators. Journal of Computational Physics **264**(1), 91–111 (2014)
15. Mattsson, K., Almquist, M., van der Weide, E.: Boundary optimized diagonal-norm SBP operators. Journal of Computational Physics **374**, 1261–1266 (2018)
16. Nordström, J.: Conservative finite difference formulations, variable coefficients, energy estimates and artificial dissipation. Journal of Scientific Computing **29**(3), 375–404 (2006)
17. 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)
18. Shu, C.W.: High-order finite difference and finite volume WENO schemes and discontinuous Galerkin methods for CFD. International Journal of Computational Fluid Dynamics **17**(2), 107–118 (2003)
19. Svärd, M., Nordström, J.: Review of summation-by-parts schemes for initial-boundary-value-problems. Journal of Computational Physics **268**(1), 17–38 (2014)