

## ENTROPY STABLE STAGGERED GRID DISCONTINUOUS SPECTRAL COLLOCATION METHODS OF ANY ORDER FOR THE COMPRESSIBLE NAVIER–STOKES EQUATIONS\*

MATTEO PARSANI<sup>†</sup>, MARK H. CARPENTER<sup>‡</sup>, TRAVIS C. FISHER<sup>§</sup>,  
AND ERIC J. NIELSEN<sup>¶</sup>

**Abstract.** Staggered grid, entropy stable discontinuous spectral collocation operators of any order are developed for the compressible Euler and Navier–Stokes equations on unstructured hexahedral elements. This generalization of previous entropy stable spectral collocation work [M. H. Carpenter, T. C. Fisher, E. J. Nielsen, and S. H. Frankel, *SIAM J. Sci. Comput.*, 36 (2014), pp. B835–B867, M. Parsani, M. H. Carpenter, and E. J. Nielsen, *J. Comput. Phys.*, 292 (2015), pp. 88–113], extends the applicable set of points from tensor product, Legendre–Gauss–Lobatto (LGL), to a combination of tensor product Legendre–Gauss (LG) and LGL points. The new semidiscrete operators discretely conserve mass, momentum, energy, and satisfy a mathematical entropy inequality for the compressible Navier–Stokes equations in three spatial dimensions. They are valid for smooth as well as discontinuous flows. The staggered LG and conventional LGL point formulations are compared on several challenging test problems. The staggered LG operators are significantly more accurate, although more costly from a theoretical point of view. The LG and LGL operators exhibit similar robustness, as is demonstrated using test problems known to be problematic for operators that lack a nonlinear stability proof for the compressible Navier–Stokes equations (e.g., discontinuous Galerkin, spectral difference, or flux reconstruction operators).

**Key words.** high-order accurate discontinuous methods, entropy stability, SBP-SAT, compressible Navier–Stokes, staggered grid, conservation

**AMS subject classifications.** 65N12, 65N35, 65P40

**DOI.** 10.1137/15M1043510

**1. Introduction.** Numerous strategies exist for constructing high-order accurate discontinuous Galerkin (DG) spectral element methods. Popular variants adopt either the weak (integral) or strong (differential) form of the governing equations derived by integrating the equations once or twice against a test function. Various interior and interface flux approximations are used (e.g., quadrature free fluxes [1], skew-symmetric [21]), as are various quadrature rules (e.g., Legendre–Gauss (LG),<sup>1</sup> Legendre–Gauss–Lobatto (LGL), or Legendre–Gauss–Radau points). Each design choice is motivated by desirable goals, such as efficiency, accuracy, flexibility, etc.

---

\*Submitted to the journal’s Methods and Algorithms for Scientific Computing section October 27, 2015; accepted for publication (in revised form) July 27, 2016; published electronically October 4, 2016. Some content of this paper appeared in 54th AIAA Aerospace Sciences Meeting, AIAA 2016-1058, Curran, Red Hook, NY, 2016 [13].

<http://www.siam.org/journals/sisc/38-5/M104351.html>

**Funding:** This work was partially supported by King Abdullah University of Science & Technology (KAUST) in Thuwal, Saudi Arabia.

<sup>†</sup>King Abdullah University of Science and Technology (KAUST), Extreme Computing Research Center (ECRC), Computer, Electrical and Mathematical Sciences & Engineering (CEMSE), Applied Mathematics and Computational Science (AMCS), Thuwal, 23955–6900, Saudi Arabia (matteo.parsani@kaust.edu.sa).

<sup>‡</sup>Computational AeroSciences Branch (CASB), NASA Langley Research Center (LaRC), Hampton, VA 23681 (mark.h.carpenter@nasa.gov).

<sup>§</sup>Computational Thermal and Fluid Mechanics, Sandia National Labs, Albuquerque, NM 87185 (tcfishe@sandia.gov).

<sup>¶</sup>CASB, NASA LaRC, Hampton, VA 23681 (eric.j.nielsen@nasa.gov).

<sup>1</sup>These points are also referred to as Gauss points in the literature.

Kopriva and Gassner [27] presented a survey of design decisions made when constructing nodal DG algorithms, as well as their advantages and disadvantages.

A popular design philosophy for the incompressible Navier–Stokes equations is to stagger the velocity and pressure fields on independent point sets. For example, Bernardi and Maday [2] and Malay, Patera, and Rønquist [32] develop a staggered approach for the Stokes problem, while Fischer and Rønquist [16] uses a similar formulation in large-scale computational studies. Kopriva and Kalias [28] extend the staggered ideas to the compressible Navier–Stokes equations. They collocate the solution variables at the Legendre–Gauss–Chebyshev quadrature points  $(0, \dots, N - 1)$ , and evaluate the fluxes at the Legendre–Gauss–Chebyshev points  $(0, \dots, N)$ . This conservative combination of over-located fluxes proves to be more robust in practical problems than conventional LGL techniques [28]. Staggered collocation operators have evolved over the past two decades to include a rich set of approaches. For example, the spectral difference (SD) [31, 33] and flux reconstruction (FR) [25, 45] approaches, discretize the compressible Navier–Stokes equations in strong form on a staggered set of solution (order  $p$ ) and flux points (order  $p + 1$ ). The observed convergence rate is reported to be  $(p + 1)$  for a sequence of nested uniform and quasi-uniform grids.

An alternate design strategy based on a summation-by-parts (SBP), simultaneous-approximation-term (SAT) framework (i.e., SBP-SAT operators), is used in references [8, 39] to construct discontinuous collocation (DC) spectral element methods of any order, and are referred to as entropy stable (SS) DC algorithms (i.e., SSDC algorithms). Therein, the primary design motivation is a semidiscrete spatial operator that supports a nonlinear (entropy) stability proof for the compressible three-dimensional (3D) Navier–Stokes equations, on curvilinear unstructured hexahedral elements.

The SSDC operators discretize the governing equations in strong form at the 3D tensor product LGL points, and adjoining elements are coupled using a provably nonlinearly stable SAT penalty approach technique [38]. The resulting algorithm is similar to the strong form nodal DG method reported in reference [23], although it substantially differs in the treatment of the nonlinear Euler fluxes and the interface couplings. A novel choice of nonlinear fluxes ensures conservation of mass, momentum, and energy as well as the entropy inequality within each element; hence elementwise entropy stability.<sup>2</sup>

The SSDC operators are distinguished from conventional finite element methods (FEMs) (e.g., DG, SD, FR) in several important ways. The nonlinear  $L^2$  stability proof is sharp; indeed, elementwise entropy conservation and entropy conservative interface fluxes guarantee global entropy conservation (neutral nonlinear stability). This sharpness is achievable because hyperviscosity dissipation, dealiasing, over-integration, or filtering of the fluxes/solution are not needed.<sup>3</sup> (Strong form SBP operators do not require assumptions of integral exactness, a common obstacle confronting weak formulations such as conventional FEM.) A second distinguishing feature is in the treatment of solid wall boundary conditions. Nonlinearly stable solid wall BCs that maintain the nonlinear stability properties of the interior operator exist for SBP-SAT operators of any order; for details see references [39, 41]. The sharp  $L^2$  stability properties of the SSDC operators provide enhanced robustness for under-resolved smooth flows and even enable capturing shocks of moderate strength (e.g., normal shock

<sup>2</sup>Dissipation is required for shocked flows to enforce a physical entropy inequality.

<sup>3</sup>Aliasing is still present but does not directly lead to nonlinear instability.

strengths  $M \leq 1.8$ ). The captured shocks are fully consistent with the Lax–Wendroff theorem [29] for weak solutions.<sup>4</sup>

Although the formulations presented in references [8, 39] are a significant step towards operational SSDC spatial discretizations of any order, noteworthy challenges remain: (1) arbitrary collocation points, (2) spatial- ( $h$ -) and order- ( $p$ -) adaptive refinement of hexahedral elements, and (3) triangular, prismatic and tetrahedral elements. Herein, an SBP-SAT framework is used to develop a staggered grid, SS spectral element formulation that includes a broader selection of collocation points. The SS mechanics developed in [8, 39] are extended to include solutions collocated at the LG points with fluxes computed at the LGL points.

The new staggered SSDC operators based on the LG points have several advantages relative to the original SSDC LGL operators. First, the staggered SSDC algorithms store the solution at the LG points, thus breaking the link between the solution storage points and the associated (suboptimal:  $2p-1$ ) LGL quadrature rules. A solution polynomial of order  $p$  can (in principle) be used with flux polynomials of arbitrary order constructed on the LGL points. Second, operators with desirable properties exist for prolonging and restricting data between the LG and LGL points, thus allowing many possibilities to move data around an element while maintaining entropy stability. Developing additional element types and connectivities can benefit from this important capability. For example, the closely related problems of  $h$ -refinement at a  $2:1$  element interface compression or a  $p$ -refinement interface, require data movement along nonconforming interfaces onto a common intermediate mortar [3]. The quadrature points do not, in general, coincide on either side of the interface, and thus, the nonlinear stability proofs presented in [8, 39] do not immediately extend to this extremely important capability. Recently, however, formulations based on the LG points were developed that naturally extend to nonconforming interfaces [11]. Third, the LG points are an interior point distribution and as such, the solution variables are not collocated at the boundaries of the element. Thus, geometric boundary discontinuities at corners are handled in an integral sense without explicit knowledge of the boundary singularity. Neither is it necessary to evaluate formulas that are known to be singular on the boundaries (e.g., turbulence source terms).

Extensive numerical tests, presented herein, reveal that the new staggered SSDC operators are significantly more accurate than the LGL SSDC operators [8, 39] of equivalent polynomial order. They are, however, more costly to implement. Simple counting arguments based on inviscid and viscous flux evaluations indicate that the cost of the staggered algorithm for a solution polynomial order  $p$  is comparable to that of an LGL operator [8, 39], with a solution polynomial order of  $(p+1)$ . Preliminary studies using an explicit temporal integrator indicate that the increased accuracy of the staggered approach partially offsets the additional cost, particularly at low polynomial order. Further investigation is needed to convincingly establish whether overcollocating the fluxes (including the SD [28, 31] or FR [25, 45] operators) can be justified from a cost perspective. An ongoing investigation continues that includes the effects of implicit temporal integrators as well as the impact of data movement; computationally intensive yet extremely accurate algorithms will be competitive in the future [24].

The paper is organized as follows. Section 2 includes the general theory of SBP operators, and differentiation and interpolation spectral collocation operators. Section 3

<sup>4</sup> The authors are not aware of a proof that density is bounded away from zero at the continuous level! Thus, departure from the real manifold is still problematic for strong shocks.

includes a brief survey of entropy stability at the continuous and semidiscrete level. In section 4, the data mechanics of the staggered approach, an entropy stability proof for the one-dimensional (1D) Burgers' equation, and its extension to the 3D compressible Navier–Stokes equations, including the interface SAT coupling approach, are presented. The comparison approach used to capture shocks and maintain nonlinear stability at the semidiscrete level is briefly presented in section 5. In section 6, a theoretical cost comparison is made between the conventional collocated [8, 39] and the staggered SSDC operators. Section 7 presents numerous numerical studies. The Euler vortex and viscous shock propagation problems are used to demonstrate the superior accuracy of the staggered algorithm. The Taylor–Green vortex problem and supersonic flow past a 3D square cylinder (Mach number = 1.5 and Reynolds number =  $10^4$ ), are used to demonstrate the robustness in the limit of order one discretization errors. Sod's problem and the sine-shock interaction problem are used to test the staggered operators on exact discontinuous solutions. The square cylinder in supersonic crossflow is used to demonstrate algorithmic robustness in three dimensions. Section 8 summarizes the results of the paper.

## 2. Methodology.

**2.1. Governing equations.** Consider the (nonlinear) 3D compressible Navier–Stokes equations for a calorically perfect gas expressed in the form

$$(2.1) \quad \begin{aligned} \frac{\partial q}{\partial t} + \frac{\partial f_i}{\partial x_i} &= \frac{\partial f_i^{(V)}}{\partial x_i}, & x \in \Omega, & \quad t \in [0, \infty), \\ Bq &= g^{(B)}(x, t), & x \in \partial\Omega, & \quad t \in [0, \infty), \\ q(x, 0) &= g^{(0)}(x), & x \in \Omega, & \end{aligned}$$

where the Cartesian coordinates,  $x = (x_1, x_2, x_3)^\top$ , and time,  $t$ , are independent variables, and index sums are implied. The vectors  $q$ ,  $f_i$ , and  $f_i^{(V)}$  are the conserved variables, and the conserved inviscid and viscous fluxes, respectively. A 3D box

$$\Omega = [x_1^L, x_1^H] \times [x_2^L, x_2^H] \times [x_3^L, x_3^H]$$

is chosen as our computational domain with  $\partial\Omega$  representing the boundary of the domain. The boundary vector  $g^{(B)}$  is assumed to contain linearly well-posed Dirichlet and/or Neumann data. Herein, we have omitted a detailed description of the 3D compressible Navier–Stokes equations because it can easily be found in the literature.

## 2.2. SBP Operators: General properties.

**2.2.1. First derivative.** First derivative operators that satisfy the SBP convention, discretely mimic the integration-by-parts property

$$(2.2) \quad \int_{x^L}^{x^H} \phi \frac{\partial q}{\partial x} dx = \phi q|_{x^L}^{x^H} - \int_{x^L}^{x^H} \frac{\partial \phi}{\partial x} q dx$$

with  $\phi$  an arbitrary scalar test function. At the discrete level, this mimetic property is achieved by constructing the first derivative approximation,  $\mathcal{D}\phi$ , with an operator in the form

$$(2.3) \quad \begin{aligned} \mathcal{D} &= \mathcal{P}^{-1} \mathcal{Q}, & \mathcal{P} &= \mathcal{P}^\top, & \zeta^\top \mathcal{P} \zeta &> 0, & \zeta \neq \mathbf{0}, \\ \mathcal{Q}^\top &= \mathcal{B} - \mathcal{Q}, & \mathcal{B} &= \text{Diag}(-1, 0, \dots, 0, 1), \end{aligned}$$

where  $\zeta$  is an arbitrary vector. The matrix  $\mathcal{P}$  can be thought of as a mass matrix (or integrator) much like in the finite element framework, or a volume that contains local grid information in the context of finite volume or finite difference numerical methods. The nearly skew-symmetric matrix  $\mathcal{Q}$ , is an undivided differencing operator; all rows sum to zero, as do all columns save the first and last, which sum to  $-1$  and  $1$ , respectively.

While the matrix  $\mathcal{P}$  need not be diagonal, the class of diagonal norm SBP operators plays a crucial role in the development of SS SBP SAT operators (see [18, 8, 39, 40, 41]).

Integration in the approximation space is conducted using an inner product with the integration weights contained in the norm  $\mathcal{P}$ ,

$$(2.4) \quad \int_{x^L}^{x^H} \phi \frac{\partial q}{\partial x} dx \approx \phi^\top \mathcal{P} \mathcal{D} \mathbf{q},$$

where

$$(2.5) \quad \mathbf{q}(\mathbf{x}) = (q(x_1), q(x_2), \dots, q(x_N))^\top, \quad \text{with } \mathbf{x} = (x_1, \dots, x_N), \quad x_1 = x^L, \quad x_N = x^H,$$

is the projection of continuous variables  $q$  onto the grid  $\mathbf{x}$ . Substituting (2.3) into (2.4), the mimetic SBP property is demonstrated,

$$(2.6) \quad \phi^\top \mathcal{P} \mathcal{P}^{-1} \mathcal{Q} \mathbf{q} = \phi^\top (\mathcal{B} - \mathcal{Q}^\top) \mathbf{q} = \phi_N q_N - \phi_1 q_1 - \phi^\top \mathcal{D}^\top \mathcal{P} \mathbf{q}.$$

**2.2.2. Complementary grids.** Entropy analysis is frequently performed in indicial notation on a staggered set of solution and flux points. Conventional SBP operators are not directly applicable to a staggered form of analysis. Reusing existing entropy analysis mechanics necessitates recasting (not modifying) conventional SBP operators into a staggered form.

Define on the interval  $x^L \leq x \leq x^H$ , the vectors of discrete solution points

$$(2.7) \quad \mathbf{x} = (x_1, x_2, \dots, x_{N-1}, x_N)^\top; \quad x^L \leq x_1, x_2, \dots, x_{N-1}, x_N \leq x^H.$$

Since the approximate solution is constructed at these points, they are denoted as the *solution points*. Furthermore, create a set of  $(N + 1)$  intermediate points  $\bar{\mathbf{x}}$  defining control volume bounds around each solution point. These points are denoted *flux points* as they are similar in nature to the control volume edges employed in the finite volume method and are used to prove the nonlinear stability (entropy stability) as briefly shown in section 3.2 (see [8, 39] for a more detailed discussion).

An example of the complete discretization operator for the fourth-order accurate polynomial interpolation ( $p = 4$ ) using the LGL points in the standard 1D element (i.e.,  $x^L = -1, x^H = 1$ ) is illustrated in Figure 1. In this figure, the solution points are identified with  $\bullet$  and the flux points are identified with  $|$ .

The distribution of the flux points depends on the discretization operator  $\mathcal{D} = \mathcal{P}^{-1} \mathcal{Q}$ . The spacing between the flux points is implicitly defined by the norm  $\mathcal{P} = (p_{ij})$ ; the diagonal elements of  $\mathcal{P}$  (i.e.,  $p_{ii}, i = 1, 2, \dots, N$ ) are equal to the spacing between flux points and their values depend on the LGL quadrature weights,  $\omega$ ,<sup>5</sup>

$$(2.8) \quad \bar{\mathbf{x}} = (\bar{x}_0, \bar{x}_1, \dots, \bar{x}_N)^\top, \quad \bar{x}_0 = x_1, \quad \bar{x}_N = x_N, \quad \bar{x}_i - \bar{x}_{i-1} = p_{ii}, \quad i = 1, 2, \dots, N.$$

<sup>5</sup>See section 2.3.1 for the functional form of the elements of the matrix  $\mathcal{P}$  and the their dependence on the quadrature weights,  $\omega$ .

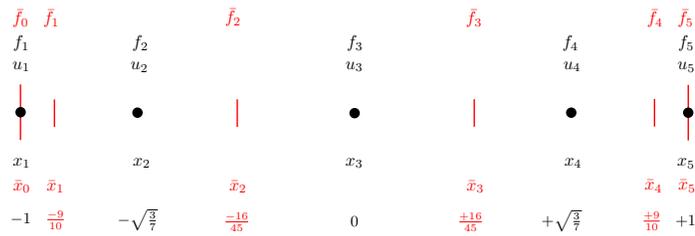


FIG. 1. The 1D discretization for the fourth-order accurate polynomial interpolation ( $p = 4$ ) LGL collocation in the standard element is illustrated. Solution points are identified with  $\bullet$  and the flux points are identified with  $|$ .

In operator notation, this is equivalent to

$$(2.9) \quad \Delta \bar{\mathbf{x}} = \mathcal{P} \mathbf{1}; \quad \mathbf{1} = (1, 1, \dots, 1)^\top,$$

where the  $N \times (N + 1)$  matrix  $\Delta$  is defined as

$$(2.10) \quad \Delta = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix},$$

and calculates the undivided difference of the two adjacent flux points.

Note that in (2.8), the first and last flux points are coincident with the first and last solution points. This fact enables the endpoint fluxes to be consistent,

$$(2.11) \quad \bar{f}_0 = f(q_1), \quad \bar{f}_N = f(q_N).$$

This duality is needed to define unique operators and is important in proving entropy stability.

*Remark 2.1.* Herein, all vectors are represented in bold font while matrices (when possible) are represented using calligraphy font. The “overbar” symbol is reserved for flux point vectors; the  $(N + 1)$ -dimensional flux point vectors are numbered from  $0 \leq i \leq N$ . In general, matrix ranks are inferred by the dimension of the vectors on which they operate. If the rank of a matrix is ambiguous, then the rank is provided in the text.

**2.2.3. Telescopic flux form.** [19] proves that any SBP matrix  $\mathcal{Q}$  can be expressed as  $\mathcal{Q} = \Delta_{(N+1)} I_N$ : the product of the  $N \times (N + 1)$  difference operator  $\Delta$  and a unique  $(N + 1) \times N$  interpolation operator  ${}_{(N+1)}I_N$ . Thus, all SBP derivative operators  $\mathcal{D} = \mathcal{P}^{-1} \mathcal{Q}$  can be manipulated into the telescopic flux form,

$$(2.12) \quad \frac{\partial}{\partial x} f(\mathbf{q}) = \mathcal{P}^{-1} \mathcal{Q} \mathbf{f} + \mathcal{T}_{(p+1)} = \mathcal{P}^{-1} \Delta \bar{\mathbf{f}} + \mathcal{T}_{(p+1)},$$

where  $\bar{\mathbf{f}} = {}_{(N+1)}I_N \mathbf{f}$ , and  $\mathcal{T}_{(p+1)}$  is the truncation error of the approximation of  $\partial f(\mathbf{q})/\partial x$ . Conservation follows immediately for telescopic derivative operators in the form of (2.12).

**2.2.4. The semidiscrete operator.** Based on the previous discussion of SBP operators and their equivalent telescopic form, the semidiscrete form of (2.1) becomes

$$\begin{aligned}
 \frac{\partial \mathbf{q}}{\partial t} &= \mathcal{D}_{x_i} \left( -\mathbf{f}_i + c_{ij} \mathcal{D}_{x_j} \mathbf{q} \right) + \mathcal{P}_{x_i}^{-1} \left( \mathbf{g}_i^{(B)} + \mathbf{g}_i^{(Int)} \right) \\
 &= \mathcal{P}_{x_i}^{-1} \Delta_{x_i} \left( -\bar{\mathbf{f}}_i + \bar{\mathbf{f}}_i^{(V)} \right) + \mathcal{P}_{x_i}^{-1} \left( \mathbf{g}_i^{(B)} + \mathbf{g}_i^{(Int)} \right), \\
 \mathbf{q}(x, 0) &= \mathbf{g}^{(0)}(x), \quad x \in \Omega,
 \end{aligned}
 \tag{2.13}$$

where the subscript  $x_i$  indicates the coordinate direction to which the operators apply. The source terms  $\mathbf{g}_i^{(B)}$  and  $\mathbf{g}_i^{(Int)}$  with  $i = 1, 2, 3$ , are constructed using a penalty approach and enforce boundary and interface conditions between elements,<sup>6</sup> respectively; and  $\mathbf{g}^{(0)}$  represents the initial condition. Full implementation details of the interior operators, including the definition of the viscous coefficient tensors,  $c_{ij}$ , are included in previous works [19, 20, 8, 39, 38].

**2.3. Spectral collocation operators.**

**2.3.1. Differentiation.** Consider the SBP operators constructed at the LGL points [9], which include the end points of the interval  $x^L = -1$  and  $x^H = 1$  (see Figure 1).

Define the Lagrange basis polynomials relative to the  $N$  discrete LGL points,  $\mathbf{x}$ , as

$$L_j(x) = \prod_{\substack{k=1 \\ k \neq j}}^N \frac{x - x_k}{x_j - x_k}, \quad 1 \leq j \leq N.
 \tag{2.14}$$

A representation of the differentiation operator  $\mathcal{D}$ , which satisfies all the requirements for being an SBP operator, is given in the following theorem.

**THEOREM 2.1.** *The derivative operator that exactly differentiates an arbitrary  $p$ th-order polynomial ( $p = N - 1$ ) at the collocation points,  $\mathbf{x}$ , can be expressed as*

$$\mathcal{D} = \mathcal{P}^{-1} \mathcal{Q}
 \tag{2.15}$$

with

$$\mathcal{P} = \sum_{\ell} \mathbf{L}(\eta_{\ell}; \mathbf{x}) \mathbf{L}(\eta_{\ell}; \mathbf{x})^{\top} \omega_{\ell}, \quad \mathcal{Q} = \sum_{\ell} \mathbf{L}(\eta_{\ell}; \mathbf{x}) \frac{d\mathbf{L}}{dx}(\eta_{\ell}; \mathbf{x})^{\top} \omega_{\ell},
 \tag{2.16}$$

where  $\eta_{\ell}$  and  $\omega_{\ell}$ ,  $1 \leq \ell \leq N$ , are the abscissas of the LGL points and their quadrature weights, respectively.  $\mathbf{L}(x; \mathbf{x})$  is a column vector whose components are the Lagrange basis polynomials relative to the discrete nodes  $\mathbf{x}$  (i.e.,  $L_j(x)$  in (2.14))

*Proof.* The proof to this theorem can be found in [12, Appendix B]. □

The matrix  $\mathcal{P}$  in (2.16) is symmetric and positive definite for any vector  $\mathbf{x}$  [9]. Furthermore, because in the SSDC algorithms the discrete points  $\mathbf{x}$  are the LGL points, the matrix  $\mathcal{P}$  is a diagonal approximation (i.e., the so-called “mass lumped” approximation) of the full  $\widehat{\mathcal{P}}$ -norm, which is defined as  $\widehat{\mathcal{P}} = (\widehat{p}_{ij}) = \int_{-1}^1 \mathbf{L}(x; \mathbf{x}) \mathbf{L}(x; \mathbf{x})^{\top}$  (see Appendix B in [12]). To the best of our knowledge, diagonal norm SBP operators are necessary to prove strict entropy conservation or entropy stability [17, 20, 12].

<sup>6</sup>Herein, the solution between adjoining elements is allowed to be discontinuous.

**2.3.2. Interpolation.** Define on the interval  $-1 = x^L \leq x \leq x^H = 1$  the vectors of discrete points,

$$(2.17) \quad \begin{aligned} \tilde{\mathbf{x}} &= (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{M-1}, \tilde{x}_M)^\top, & -1 \leq \tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{M-1}, \tilde{x}_M \leq 1, \\ \mathbf{x} &= (x_1, x_2, \dots, x_{N-1}, x_N)^\top, & -1 \leq x_1, x_2, \dots, x_{N-1}, x_N \leq 1. \end{aligned}$$

Herein, the discrete points  $\tilde{\mathbf{x}}$  and  $\mathbf{x}$  are the LG points (i.e., the so-called Gauss points) and the LGL points, respectively. All the scalars, vectors, and matrices associated with the LG points are denoted with a “tilde” symbol. Next, define the interpolation operators that move data between  $\tilde{\mathbf{x}}$  and  $\mathbf{x}$ :

$$(2.18) \quad \begin{aligned} \mathcal{I}_{LGL \rightarrow LG} &= \tilde{\mathcal{P}}^{-1} \mathcal{R}_{LG-LGL}, \\ \mathcal{I}_{LG \rightarrow LGL} &= \mathcal{P}^{-1} \mathcal{R}_{LG-LGL}^\top, \\ \tilde{\mathcal{P}} \mathcal{I}_{LGL \rightarrow LG} &= \mathcal{I}_{LG \rightarrow LGL}^\top \mathcal{P}, \end{aligned}$$

where

$$(2.19) \quad \mathcal{R}_{LG-LGL} = \int_{-1}^1 \mathbf{L}(x; \tilde{\mathbf{x}}) \mathbf{L}(x; \mathbf{x})^\top dx.$$

In [12], it is shown that these polynomial interpolation operators exist and satisfy the relations (2.18), provided that the LGL points are of higher polynomial orders than the LG points (i.e.,  $N > M$ ).

Note that because the  $\tilde{\mathbf{x}}$  are the LG points, the full  $\tilde{\mathcal{P}}$ -norm is already a positive definite diagonal matrix.

### 3. Entropy consistent and SS SBP operators for the compressible Navier–Stokes equations.

**3.1. Overview of the continuous entropy analysis.** The nonlinear compressible Navier–Stokes equations given in (2.1) have a convex extension of their original form, that when integrated over the physical domain,  $\Omega$ , depends only on boundary data and dissipative terms. This convex extension is motivated by an entropy function and provides a mechanism for proving the stability of the nonlinear system in the  $L^2$  norm. Indeed, Dafermos [14] shows that if a system of conservation laws is endowed with a convex entropy function,  $S = S(q)$ , a bound on the global estimate of  $S$  can be converted into an a priori estimate of the solution vector  $q$  (e.g., the solution of system (2.1)). Interested readers should also refer to the work of Svård [42] for the proof of the stability of system (2.1) in the  $L^2$  norm.

**DEFINITION 3.1.** *A scalar function  $S = S(q)$  is an entropy function of system (2.1) if it satisfies the following conditions:*

- *Differentiation of the convex function  $S(q)$  simultaneously contracts all the inviscid spatial fluxes as follows:*

$$(3.1) \quad \frac{\partial S}{\partial q} \frac{\partial f_i}{\partial x_i} = \frac{\partial S}{\partial q} \frac{\partial f_i}{\partial q} \frac{\partial q}{\partial x_i} = \frac{\partial F_i}{\partial q} \frac{\partial q}{\partial x_i} = \frac{\partial F_i}{\partial x_i}, \quad i = 1, 2, 3.$$

*The components of the contracting vector,  $\partial S / \partial q$ , are the entropy variables denoted as  $w^\top = \partial S / \partial q$ . The  $F_i(q)$  are the entropy fluxes in the  $i$ -direction.*

- The entropy variables,  $w$ , symmetrize system (2.1) if  $w$  assumes the role of a new independent variable (i.e.,  $q = q(w)$ ). Expressing (2.1) in terms of  $w$  yields

$$(3.2) \quad \frac{\partial q}{\partial t} + \frac{\partial f_i}{\partial x_i} - \frac{\partial f_i^{(V)}}{\partial x_i} = \frac{\partial q}{\partial w} \frac{\partial w}{\partial t} + \frac{\partial f_i}{\partial w} \frac{\partial w}{\partial x_i} - \frac{\partial}{\partial x_i} \left( \widehat{c}_{ij} \frac{\partial w}{\partial x_j} \right) = 0, \quad i = 1, 2, 3,$$

with the symmetry conditions:  $\partial q/\partial w = (\partial q/\partial w)^\top$ ,  $\partial f_i/\partial w = (\partial f_i/\partial w)^\top$ , and  $\widehat{c}_{ij} = \widehat{c}_{ij}^\top$ .<sup>7</sup>

An extensive entropy analysis of the Euler and compressible Navier–Stokes equations can be found, for instance, in [43, 44, 8, 20, 39, 40, 34], and the references therein.

Contracting system (2.1) with the entropy variables,  $w$ , results in the differential form of the (scalar) entropy equation,

$$(3.3) \quad \begin{aligned} \frac{\partial S}{\partial q} \frac{\partial q}{\partial t} + \frac{\partial S}{\partial q} \frac{\partial f_i}{\partial x_i} &= \frac{\partial S}{\partial t} + \frac{\partial F_i}{\partial x_i} = \frac{\partial S}{\partial q} \frac{\partial f_i^{(V)}}{\partial x_i} = \frac{\partial}{\partial x_i} \left( w^\top f_i^{(V)} \right) - \left( \frac{\partial w}{\partial x_i} \right)^\top f_i^{(V)} \\ &= \frac{\partial}{\partial x_i} \left( w^\top f_i^{(V)} \right) - \left( \frac{\partial w}{\partial x_i} \right)^\top \widehat{c}_{ij} \frac{\partial w}{\partial x_j}. \end{aligned}$$

Integrating (3.3) over the domain  $\Omega$  and accounting for local sources of entropy dissipation yields a global conservation statement for the entropy in the domain

$$(3.4) \quad \frac{d}{dt} \int_{\Omega} S \, d\mathbf{x} \leq \left[ w^\top f_i^{(V)} - F_i \right]_{\partial\Omega} - \int_{\Omega} w_{x_i}^\top \widehat{c}_{ij} w_{x_j} \, d\mathbf{x}.$$

Equation (3.4) is sharp (i.e., =) for smooth flows. Physical arguments require that the five-by-five matrices  $\widehat{c}_{ij}$  be positive semidefinite [17, 20]. Thus, the sign of the entropy change from viscous dissipation is always negative. An increase in entropy within the domain can only result from data that convect and/or diffuse through the boundaries,  $\partial\Omega$ .

*Remark 3.1.* The density and temperature are assumed to be strictly positive, i.e.,  $\rho, T > 0$ . This ensures the convexity of the entropy function  $S = S(q)$  and a one-to-one mapping between  $w$  and  $q$ . (The proof can be found in Appendix B.1 of [18]). An additional numerical mechanism must be employed to bound  $\rho$  and  $T$  away from zero to guarantee positivity; positivity preservation is not considered herein.

*Remark 3.2.* The scalar equation (3.4) is not a sharp integral statement of entropy conservation in the presence of shocks, as it does not precisely account for the dissipation of entropy at a shock. All that is known a priori is the sign of the jump in entropy.

### 3.2. Overview of the semidiscrete entropy analysis for the LGL points.

The semidiscrete, nonlinear entropy estimate is achieved by mimicking term by term the continuous estimate given in (3.3) and (3.4). The analysis begins by contracting the discrete entropy variables,  $\mathbf{w}^\top$  with the semidiscrete equation (2.13). The resulting global equation that governs the semidiscrete decay of entropy is given by [8, 39]

$$(3.5) \quad \mathbf{w}^\top \mathcal{P} \frac{\partial \mathbf{q}}{\partial t} + \mathbf{w}^\top \Delta \bar{\mathbf{f}}_i = \mathbf{w}^\top \Delta \bar{\mathbf{f}}_i^{(V)} + \mathbf{w}^\top \mathbf{g}_i^{(B)} + \mathbf{w}^\top \mathbf{g}_i^{(Int)}, \quad i = 1, 2, 3,$$

<sup>7</sup>Using the entropy variables, the viscous fluxes in the  $i$ -direction are defined as  $f_i^{(V)} = \widehat{c}_{ij} \frac{\partial w}{\partial x_j}$ . The explicit form of the  $\widehat{c}_{ij}$  matrices can be found in [17, 37].

where

$$\mathbf{w} = (w(q_{(1),(1),(1)})^\top, w(q_{(2),(1),(1)})^\top, \dots, w(q_{(N),(N),(N)})^\top)^\top,$$

and the source terms  $\mathbf{g}_i^{(B)}$  and  $\mathbf{g}_i^{(Int)}$  account for the enforcement of boundary and interface conditions, respectively. The entropy variables,  $\mathbf{w}$ , are defined at the solution points whereas the quantities with an overbar, i.e.,  $\bar{\mathbf{f}}$  and  $\bar{\mathbf{f}}_i^{(V)}$ , are defined at the flux points (see Figure 1).

Simplifying (3.5) into a semidiscrete convection-diffusion equation for the entropy, proceeds as follows. The semidiscrete time term is manipulated for diagonal-norm SBP operators as  $\mathbf{w}^\top \mathcal{P} \partial \mathbf{q} / \partial t = \mathbf{1}^\top \mathcal{P} \partial \mathbf{S} / \partial t$ , (or the pointwise statement:  $w_i^\top \mathcal{P}_i \partial q_i / \partial t = \mathcal{P}_i \partial S_i / \partial t \forall i$ ). The viscous terms are first symmetrized as in (3.2) and then manipulated as

$$(3.6) \quad \mathbf{w}^\top \Delta \bar{\mathbf{f}}_i^{(V)} = \mathbf{w}^\top \mathcal{B} \hat{c}_{ij} \mathcal{D}_{x_j} \mathbf{w} - (\mathcal{D}_{x_i} \mathbf{w})^\top \mathcal{P} \hat{c}_{ij} (\mathcal{D}_{x_j} \mathbf{w}).$$

The terms  $\mathbf{w}^\top \mathcal{B} \hat{c}_{ij} \mathcal{D}_{x_j} \mathbf{w}$  account for diffusive fluxes on boundaries while the terms  $-(\mathcal{D}_{x_i} \mathbf{w})^\top \mathcal{P} \hat{c}_{ij} (\mathcal{D}_{x_j} \mathbf{w})$  account for viscous dissipation. See [8] for a complete derivation of the temporal and viscous manipulations.

The inviscid terms of (3.5) are entropy conservative if in each coordinate direction  $i$ , they satisfy

$$(3.7) \quad \mathbf{w}^\top \Delta \bar{\mathbf{f}}_i = \bar{F}_i(q_N) - \bar{F}_i(q_1) = F_i(q_N) - F_i(q_1) = \mathbf{1}^\top \Delta \bar{\mathbf{F}}_i.$$

A design-order nonentropy conservative discrete flux  $\bar{\mathbf{f}}_i$  will in general only satisfy (3.7) to design order, and could lead to numerical instability if it is not SS. A general procedure for constructing design-order, entropy conservative fluxes,  $\bar{\mathbf{f}}^{(SC)}$ , appears in [17, 18], where it is shown that a pointwise, entropy conservative flux  $\bar{f}_m^{(SC)}$  can be constructed in each coordinate direction  $i$  by using a linear combination of  $q_{\ell k}$ -weighted, two-point entropy conservative fluxes  $\bar{f}_S = \bar{f}_S(q_\ell, q_k)$ . The implementation formula is

$$(3.8) \quad \bar{f}_m^{(SC)} = \sum_{k=m+1}^N \sum_{\ell=1}^m 2 q_{\ell k} \bar{f}_S(q_\ell, q_k), \quad 1 \leq m \leq N-1,$$

where the coefficient  $q_{\ell k}$  corresponds to the  $\ell$  row and  $k$  column in the SBP matrix  $\mathcal{Q}$ . Herein, the two-point entropy conservative flux  $\bar{f}_S$  of Ismail and Roe [26] is used in (3.8).

Discontinuous formulations do not enforce solution continuity across adjoining element interfaces. Thus, the interface penalty terms  $\mathbf{g}_i^{(Int)}$  in (3.5) account for interface connections at adjoining elements and are formulated using an SAT approach [38]. Inviscid coupling terms are first constructed that ensure entropy conservation across the interface. Then a characteristic based dissipation term is added, that results in an upwinding of the interface fluxes. The viscous coupling terms are patterned after a local discontinuous Galerkin-type (LDG-type) approach with additional interior penalty (IP) terms to damp neutral eigenvalues.

An extensive analysis of all semidiscrete terms can be found in [18, 8, 7, 38, 37, 39].

*Remark 3.3.* The semidiscrete entropy stability does not necessarily lead to fully discrete stability as is usually the case for linear partial differential equations. However, as noted by Tadmor [43], entropy stability is enhanced by fully implicit time discretization. In contrast, explicit time discretization, leads to entropy production.

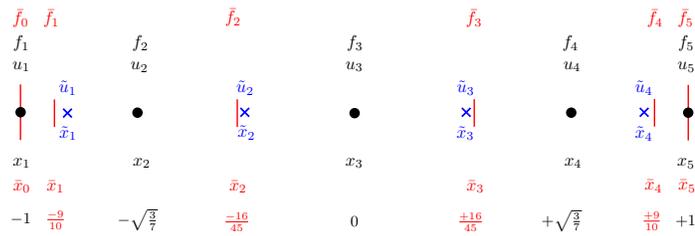


FIG. 2. The 1D discretization for the fourth-order accurate polynomial interpolation ( $p = 4$ ) with the staggered approach is illustrated. Solution points  $\tilde{\mathbf{x}}$  are identified with  $\times$  and auxiliary points  $\mathbf{x}$  are identified with  $\bullet$ . Flux points  $\bar{\mathbf{x}}$  (used to prove the entropy stability) are identified with  $|$ .

Thus, the entropy stability of explicit schemes hinges on a delicate balance between temporal entropy production and spatial entropy dissipation. Fully discrete entropy conservation is offered, for instance, by Crank–Nicolson time differencing [43].

#### 4. Staggered SBP operators.

**4.1. Data mechanics of the staggered grid approach.** Define a staggered grid algorithm (Sta-Grd-Alg) for building discrete differentiation operators using two sets of collocation points:  $\tilde{\mathbf{x}}$  and  $\mathbf{x}$  of dimension  $M$  and  $N$ , respectively (see Figure 2). The proposed algorithm is similar to that proposed by Kopriva and Kolas [28]. Assume that the time-dependent solution is stored at the points  $\tilde{\mathbf{x}}$ . Furthermore, assume that the extrema of  $\mathbf{x}$  coincide with the endpoints of the domain,  $\mathbf{x}_1 = x^L$ ,  $\mathbf{x}_N = x^H$ , to facilitate the imposition of interface or boundary data.

Discrete differentiation of first- or second-order spatial terms (e.g.,  $\partial f/\partial x$  or  $\partial f^{(V)}/\partial x$ ), by using the Sta-Grd-Alg is accomplished as follows:

- Interpolate the discrete entropy variables from  $\tilde{\mathbf{x}}$  to  $\mathbf{x}$ .
- Build the nonlinear fluxes  $f$  and  $f^{(V)}$  on the set of points  $\mathbf{x}$ .
- Build the interface and/or boundary penalties at the extrema of  $\mathbf{x}$ .
- Differentiate the fluxes on  $\mathbf{x}$ , and impose the penalties by using the SAT approach.
- Interpolate the discrete flux derivatives and penalties back to  $\tilde{\mathbf{x}}$ .
- Advance the solution with a time integration scheme by using the interpolated flux derivative on  $\tilde{\mathbf{x}}$ .

Tensor product arithmetic extends the approach directly to three spatial dimensions.<sup>8</sup> An SBP-SAT stability proof is now presented for the Sta-Grd-Alg. It is valid for all diagonal-norm SBP operators.

Define tilde variables and operators that act on the set of points  $\tilde{\mathbf{x}}$  (e.g.,  $\tilde{\mathbf{u}}$ ,  $\tilde{\mathcal{P}}$ , and  $\tilde{\mathcal{D}}$ ), and the pair of interpolation operators  $\mathcal{I}_{LG \rightarrow LGL}$  and  $\mathcal{I}_{LGL \rightarrow LG}$  introduced in section 2.3.2. The  $\mathcal{I}_{LG \rightarrow LGL}$  operator transfers data from  $\tilde{\mathbf{x}}$  to  $\mathbf{x}$ , while the  $\mathcal{I}_{LGL \rightarrow LG}$  transfers data from  $\mathbf{x}$  to  $\tilde{\mathbf{x}}$ .

An entropy analysis of the 1D Burgers’ equation is presented before that of the compressible Navier–Stokes equations in multiple dimensions. The matrix manipulations used to prove entropy stability for Burgers’ equation, are structurally equivalent to those used in the compressible Navier–Stokes equations, although they are much easier to follow.

<sup>8</sup>The Sta-Grd-Alg is valid for other grid distributions that do not support tensor product arithmetic.

**4.2. Entropy stability of Burgers’ equation.** Consider the 1D viscous Burgers’ equation

$$\begin{aligned}
 & \frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \frac{\partial (\varepsilon f^{(V)} (\frac{\partial u}{\partial x}))}{\partial x}, \quad f(u) = \frac{u^2}{2}, \\
 & f^{(V)} \left( \frac{\partial u}{\partial x} \right) = \frac{\partial u}{\partial x}, \quad x \in [x^L = -1, x^H = 1], \quad t \in [0, \infty), \\
 (4.1) \quad & \frac{u(x^L, t) + |u(x^L, t)|}{3} u(x^L, t) - \varepsilon \frac{\partial u}{\partial x}(x^L, t) - g(x^L, t) = 0, \\
 & \frac{u(x^H, t) - |u(x^H, t)|}{3} u(x^H, t) - \varepsilon \frac{\partial u}{\partial x}(x^H, t) - g(x^H, t) = 0, \\
 & u(x, 0) = g^{(0)}(x),
 \end{aligned}$$

where  $u = u(x, t)$  is the continuous solution vector,  $f = f(u)$  is a nonlinear flux function. The boundary conditions in (4.1) are constructed such that the semidiscrete energy only increases with respect to the imposed data and maintains the same form in the inviscid limit  $\varepsilon \rightarrow 0$ .

Define the interpolated solution vector  $\mathbf{u}$ , and the diagonal velocity and viscosity matrices  $[\Gamma]$  and  $[\epsilon]$  as

$$(4.2) \quad \mathbf{u} = \mathcal{I}_{LG \rightarrow LGL} \tilde{\mathbf{u}}; \quad [\Gamma] = \text{Diag}[\mathcal{I}_{LG \rightarrow LGL} \tilde{\mathbf{u}}]; \quad [\epsilon] = \text{Diag}[\mathcal{I}_{LG \rightarrow LGL} \tilde{\epsilon}];$$

and the boundary operator nomenclature

$$(4.3) \quad \mathbf{e}(x^L) = (1, 0, \dots, 0)_M^\top, \quad u(x^L) = \mathbf{u}|_{x=-1} = \mathbf{u}^\top \mathbf{e}(x^L), \\
 (\mathcal{D}\mathbf{u})(x^L) = (\mathcal{D}\mathbf{u})|_{x=-1} = (\mathcal{D}\mathbf{u})^\top \mathbf{e}(x^L), \quad g(x^L) = \mathbf{g}|_{x=-1} = \mathbf{g}^\top \mathbf{e}(x^L)$$

with similar definitions for  $\mathbf{e}(x^H)$ ,  $u(x^H)$ ,  $(\mathcal{D}\mathbf{u})(x^H)$ , and  $g(x^H)$ . The subscript  $M$  in (4.3) denotes the size of the vector  $\mathbf{e}(x^L)$ .

An entropy-entropy flux pair,  $(S, F)$ , and the entropy variable,  $w$ , for Burgers’ equation are [43]

$$(4.4) \quad (S, F) = \left( \frac{u^2}{2}, \frac{u^3}{3} \right); \quad u = w.$$

Note that the entropy is guaranteed to be convex for all  $u$  (i.e.,  $\partial^2 S / \partial u^2 = 1$ ).<sup>9</sup>

Apply the Sta-Grd-Alg to construct the quadratic inviscid and linear viscous fluxes as well as the boundary penalties. The entropy analyses for the St-Grd-Alg proceeds as expected for the semidiscrete temporal, and discrete viscous, and SAT terms (e.g., see the collocated analysis given in reference [8]). Differences, however, appear in the analysis of the quadratic flux term. The quadratic flux is discretized using a diagonal norm SBP operator,  $\mathcal{D} = \mathcal{P}^{-1} \mathcal{Q}$ , with the  $\mathcal{Q}$  matrix rearranged into telescoping form [12, 30]. The resulting discretized expression for the quadratic term in (4.1) is

$$\frac{\partial}{\partial x} f(u) = \frac{1}{2} \frac{\partial u^2}{\partial x} \approx \mathcal{P}^{-1} \Delta \bar{\mathbf{f}}(\mathbf{u}).$$

The resulting semidiscrete staggered grid operator is then [12]

<sup>9</sup>Note that the entropy for the Burgers’ equation is not unique.

(4.5)

$$\begin{aligned} \frac{d}{dt} \tilde{\mathbf{u}} + \mathcal{I}_{LGL \rightarrow LG} \mathcal{P}^{-1} \Delta \bar{\mathbf{f}}(\mathbf{u}) &= \mathcal{I}_{LGL \rightarrow LG} \mathcal{D} [\epsilon] \mathcal{D} \mathbf{u} \\ &- \left( \frac{u(x^L) + |u(x^L)|}{3} u(x^L) - \epsilon(\mathcal{D}\mathbf{u})(x^L) - g(x^L) \right) \mathcal{I}_{LGL \rightarrow LG} \mathcal{P}^{-1} \mathbf{e}(x^L) \\ &+ \left( \frac{u(x^H) - |u(x^H)|}{3} u(x^H) - \epsilon(\mathcal{D}\mathbf{u})(x^H) - g(x^H) \right) \mathcal{I}_{LGL \rightarrow LG} \mathcal{P}^{-1} \mathbf{e}(x^H) \end{aligned}$$

with the initial data  $\mathbf{u}(x, 0) = \mathbf{g}^{(0)}(x)$ . To simplify the notation in (4.5), the dependence on the time,  $t$ , has been omitted. What remains is to construct an entropy conserving flux  $\bar{\mathbf{f}}(\mathbf{u}) = \bar{\mathbf{f}}_i^{(SC)}$ .

In [20] it is shown that an entropy conserving flux for the delta form operator  $\mathcal{P}^{-1} \Delta \bar{\mathbf{f}}(\mathbf{u})$  can be constructed, provided there exists a two-point entropy flux relation and a diagonal-norm SBP operator is used for  $\mathcal{D}$ . Using the definition for the entropy variable  $u$  given in (4.4) and Tadmor’s integral relation [43],

$$(4.6) \quad \bar{f}_S(u_k, u_\ell) = \int_0^1 g(w(u_k) + \xi(w(u_\ell) - w(u_k))) \, d\xi, \quad g(w(u)) = f(u),$$

yields a two-point entropy conservative flux

$$(4.7) \quad \bar{f}_S(u_\ell, u_k) = \frac{1}{6} (u_\ell^2 + u_\ell u_k + u_k^2).$$

The entropy conserving flux is given by

$$(4.8) \quad \begin{aligned} \bar{f}_i^{(SC)} &= \sum_{k=i+1}^N \sum_{\ell=1}^i 2q_{\ell k} \bar{f}_S(u_\ell, u_k) \\ &= 2 \sum_{k=i+1}^N \sum_{\ell=1}^i q_{\ell k} \frac{1}{6} (u_\ell^2 + u_\ell u_k + u_k^2), \quad 1 \leq i \leq N-1, \end{aligned}$$

and is of design-order accuracy [12].

Contracting the quadratic term  $\mathcal{I}_{LGL \rightarrow LG} \mathcal{P}^{-1} \Delta \bar{\mathbf{f}}^{(SC)}$  in (4.5) with the discrete vector  $(\tilde{\mathcal{P}}\tilde{\mathbf{u}})^\top$  yields the telescoping condition

$$(4.9) \quad \mathbf{u}^\top \Delta \bar{\mathbf{f}}^{(SC)} = \mathbf{1}^\top [\square] \Delta \bar{\mathbf{f}}^{(SC)} = \mathbf{1}^\top \Delta \bar{\mathbf{F}} = (\bar{F}_N - \bar{F}_1)$$

with

$$(4.10) \quad \begin{aligned} \bar{F}_i &= \sum_{k=i+1}^N \sum_{\ell=1}^i q_{\ell k} \left[ (u_\ell + u_k) \bar{f}_S(u_\ell, u_k) - \frac{1}{6} (u_\ell^3 + u_k^3) \right], \quad 1 \leq i \leq N-1, \\ \bar{F}_0 &= \frac{1}{3} u_0^3, \quad \bar{F}_N = \frac{1}{3} u_N^3. \end{aligned}$$

Collecting all the terms in this entropy analysis yields  $\frac{d}{dt} (\tilde{\mathbf{u}}^\top \tilde{\mathcal{P}}\tilde{\mathbf{u}}) \leq -\epsilon(\mathcal{D}\mathbf{u})^\top \mathcal{P} \mathcal{D} \mathbf{u}$  which establishes boundedness of the  $L^2$  norm of  $\mathbf{u}$  provided that the boundary data  $g(x^L)^2$  and  $g(x^H)^2$  are well behaved [12].

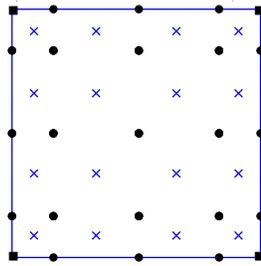


FIG. 3. Fully staggered two-dimensional (2D) tensor product elements. Solution LG points are identified with  $\times$ , flux LGL points are identified with  $\bullet$  and  $\blacksquare$ . Note that black square symbols are used only for corner LGL flux points.

**4.3. Staggered grid in two dimensions.** Figure 3 shows a popular staggered data structure in two spatial dimensions, with the solution stored at the tensor product LG points (i.e., the so-called Gauss points) of order  $p$  (blue crosses). Herein, this data structure is referred to as the fully staggered approach.

The fully staggered approach moves the entropy variables via 3D tensor product interpolations from LG (supporting a polynomial of order  $p$ ) to LGL (supporting a polynomial of order  $p+1$ ) points (black circles). The discrete operators reported in [8, 7, 38, 39] are used on the LGL points to construct the spatial residual. The temporal updates needed on the LG points are obtained by restricting the LGL residuals back to the LG points. Extension to general curvilinear coordinates follows immediately on the LGL points [8, 38].

**4.4. Tensor operators in three dimensions.** Consider a single tensor product element and an SSDC discretization with  $M = p + 1$  LG solution points in each coordinate direction<sup>10</sup> [8, 7, 39, 37]; the following elementwise matrices are needed:

(4.11)

$$\tilde{\mathcal{P}} = \left( \tilde{\mathcal{P}}_M \otimes \tilde{\mathcal{P}}_M \otimes \tilde{\mathcal{P}}_M \otimes I_5 \right),$$

$$\mathcal{I}_{LG \rightarrow LGL} = (\mathcal{I}_{LG \rightarrow LGL})_{x_1 x_2 x_3} = (\mathcal{I}_{LG \rightarrow LGL} \otimes \mathcal{I}_{LG \rightarrow LGL} \otimes \mathcal{I}_{LG \rightarrow LGL} \otimes I_5),$$

$$\mathcal{I}_{LGL \rightarrow LG} = (\mathcal{I}_{LGL \rightarrow LG})_{x_1 x_2 x_3} = (\mathcal{I}_{LGL \rightarrow LG} \otimes \mathcal{I}_{LGL \rightarrow LG} \otimes \mathcal{I}_{LGL \rightarrow LG} \otimes I_5),$$

$$\mathcal{D}_{x_1} = (\mathcal{D}_N \otimes I_N \otimes I_N \otimes I_5), \quad \dots \quad \mathcal{D}_{x_3} = (I_N \otimes I_N \otimes \mathcal{D}_N \otimes I_5),$$

$$\mathcal{P}_{x_1} = (\mathcal{P}_N \otimes I_N \otimes I_N \otimes I_5), \quad \dots \quad \mathcal{P}_{x_3} = (I_N \otimes I_N \otimes \mathcal{P}_N \otimes I_5),$$

$$\mathcal{P}_{x_1 x_2} = (\mathcal{P}_N \otimes \mathcal{P}_N \otimes I_N \otimes I_5), \quad \dots \quad \mathcal{P}_{x_2 x_3} = (I_N \otimes \mathcal{P}_N \otimes \mathcal{P}_N \otimes I_5),$$

<sup>10</sup>Recall from section 2.3.2 that with the staggered algorithm the number of LG and LGL points in one dimension are denoted by  $M$  and  $N$ , respectively.

$$\mathcal{P} = \mathcal{P}_{x_1 x_2 x_3} = (\mathcal{P}_N \otimes \mathcal{P}_N \otimes \mathcal{P}_N \otimes I_5),$$

$$\mathcal{B}_{x_1} = (\mathcal{B}_N \otimes I_N \otimes I_N \otimes I_5), \quad \dots \quad \mathcal{B}_{x_3} = (I_N \otimes I_N \otimes \mathcal{B}_N \otimes I_5),$$

$$\Delta_{x_1} = (\Delta_N \otimes I_N \otimes I_N \otimes I_5), \quad \dots \quad \Delta_{x_3} = (I_N \otimes I_N \otimes \Delta_N \otimes I_5),$$

where  $\tilde{\mathcal{P}}_M$  is the norm of the LG points, while  $\mathcal{D}_N$ ,  $\mathcal{P}_N$ ,  $\Delta_N$ , and  $\mathcal{B}_N$  are the 1D SBP operators [37] defined on the LGL points, and  $I_N$  is the identity matrix of dimension  $N$ .  $I_5$  denotes the identity matrix of dimension five.<sup>11</sup> The subscripts in (4.11) indicate the coordinate directions to which the operators apply (e.g.,  $\mathcal{D}_{x_1}$  is the differentiation matrix in the  $x_1$  direction). The symbol  $\otimes$  represents the Kronecker product. When applying these operators to the scalar entropy equation in space at the LG points, a hat is used to differentiate the scalar operator from the full vector operator. For example,

$$(4.12) \quad \hat{\tilde{\mathcal{P}}} = (\tilde{\mathcal{P}}_M \otimes \tilde{\mathcal{P}}_M \otimes \tilde{\mathcal{P}}_M); \quad \hat{\mathcal{P}} = (\mathcal{P}_M \otimes \mathcal{P}_M \otimes \mathcal{P}_M).$$

The vector of conservative variables of each element is ordered as

$$(4.13) \quad \tilde{\mathbf{q}} = \left( \tilde{q}(\tilde{x}_{(1)(1)(1)})^\top, \tilde{q}(\tilde{x}_{(1)(1)(2)})^\top, \dots, \tilde{q}(\tilde{x}_{(M)(M)(M)})^\top \right) = \left( \tilde{q}_{(1)}^\top, \tilde{q}_{(2)}^\top, \dots, \tilde{q}_{(M^3)}^\top \right),$$

where the subscripts denote the ordering of the solution points in the coordinate directions. Assume an equivalent definition and order for the entropy variables  $\tilde{\mathbf{w}}$ , and an analogous definition for the variables at the LGL points.<sup>12</sup>

The  $\mathcal{I}_{LG \rightarrow LGL}$  operator transfers data from  $\tilde{\mathbf{x}}$  to  $\mathbf{x}$ , while the  $\mathcal{I}_{LGL \rightarrow LG}$  transfers data from  $\mathbf{x}$  to  $\tilde{\mathbf{x}}$ . Define

$$(4.14) \quad \mathbf{w} = \mathcal{I}_{LG \rightarrow LGL} \tilde{\mathbf{w}}; \quad \hat{c}_{ij} = \mathcal{I}_{LG \rightarrow LGL} \tilde{c}_{ij}; \quad [\hat{c}_{ij}] = \text{Diag}[\mathcal{I}_{LG \rightarrow LGL} \tilde{c}_{ij}].$$

**4.5. SS semidiscrete operator on one domain.** Using these definitions and the SBP operators (4.11), system (2.1) is discretized locally on an isolated element as [37, 38]

$$(4.15) \quad \frac{d\tilde{\mathbf{q}}}{dt} + \mathcal{I}_{LGL \rightarrow LG} \left[ \mathcal{P}_{x_i}^{-1} \Delta_{x_i} \bar{\mathbf{f}}_i - \mathcal{D}_{x_i} \bar{\mathbf{f}}_i^{(V)} \right] = \mathcal{I}_{LGL \rightarrow LG} \mathcal{P}_{x_i}^{-1} \mathbf{g}_i^{(Int)},$$

where Einstein notation is used to express the coordinate directions. The penalty interface terms  $\mathbf{g}_i^{(Int)}$  with  $i = 1, 2, 3$  are used to connect neighboring elements (see section 4.6).

The inviscid fluxes  $\bar{\mathbf{f}}_i$  of any order in (4.15) are computed using linear combinations of  $q_{\ell k}$ -weighted, two-point entropy conservative fluxes ( $\bar{f}_S(u_\ell, u_k)$  as in (3.8). Thus, they are entropy conservative. The interpolated entropy variables  $\mathbf{w} = \mathcal{I}_{LG \rightarrow LGL} \tilde{\mathbf{w}}$  are used to build the fluxes on the LGL points. The viscous fluxes are also computed using interpolated entropy variables and the operators  $\mathcal{D}_{x_i}$ ,  $i = 1, 2, 3$ , defined in (4.11). The viscous coefficient matrices  $\tilde{c}_{ij}$  are again formed using interpolated data on the LGL points.

<sup>11</sup>The 3D compressible Navier–Stokes equations form a system of five nonlinear partial differential equations.

<sup>12</sup>With the staggered algorithm, the number of LGL points in one dimension is denoted by  $N$  (see section 2.3.2). Thus,  $N^3$  is the number of LGL points in a 3D tensor-product element.

*Remark 4.1.* The interpolations from and to the LG points are carried out in computational space by using an efficient tensor-product algorithm that require only the knowledge of the 1D  $\mathcal{I}_{LG \rightarrow LGL}$  and  $\mathcal{I}_{LGL \rightarrow LG}$  operators. The extension to general curvilinear coordinates follows immediately on the LGL points [8, 39].

**4.6. SS interface coupling.** Consider two cubic tensor product elements by extending (4.15) to two adjoining elements denoted with the subscripts  $l$  (left) and  $r$  (right), respectively. Without loss of generality assume that all their faces are orthogonal to the three coordinate directions and are not boundary faces, i.e., they are not part of the boundary surface  $\partial\Omega$ . The resulting expressions become [37, 38]

$$(4.16a) \quad \frac{d\tilde{\mathbf{q}}_\ell}{dt} + \mathcal{I}_{LGL \rightarrow LG\ell} \left[ \mathcal{P}_{x_i,\ell}^{-1} \Delta_{x_i,\ell} \bar{\mathbf{f}}_{i,\ell} - \mathcal{D}_{x_i,\ell} [\hat{c}_{ij,\ell}] \Theta_{j,\ell} \right] = \mathcal{I}_{LGL \rightarrow LG\ell} \mathcal{P}_{x_i,\ell}^{-1} \mathbf{g}_{i,\ell}^{(Int),q},$$

$$(4.16b) \quad \Theta_{i,\ell} - \mathcal{D}_{x_i} \mathbf{w}_l = \mathcal{P}_{x_i,\ell}^{-1} \mathbf{g}_{i,\ell}^{(Int),\Theta},$$

$$(4.16c) \quad \frac{d\tilde{\mathbf{q}}_r}{dt} + \mathcal{I}_{LGL \rightarrow LGr} \left[ \mathcal{P}_{x_i,r}^{-1} \Delta_{x_i,r} \bar{\mathbf{f}}_{i,r} - \mathcal{D}_{x_i,r} [\hat{c}_{ij,r}] \Theta_{j,r} \right] = \mathcal{I}_{LGL \rightarrow LGr} \mathcal{P}_{x_i,r}^{-1} \mathbf{g}_{i,r}^{(Int),q},$$

$$(4.16d) \quad \Theta_{i,r} - \mathcal{D}_{x_i} \mathbf{w}_l = \mathcal{P}_{x_i,r}^{-1} \mathbf{g}_{i,r}^{(Int),\Theta}.$$

In (4.16),  $\Theta_{i,\ell}$  and  $\Theta_{i,r}$  are the vectors of the gradient of the entropy variables on the left and right elements in the  $i$  direction, whereas  $\mathbf{g}_{i,(\cdot)}^{(Int),q}$  and  $\mathbf{g}_{i,(\cdot)}^{(Int),\Theta}$  are the penalty interface terms on the conservative variable and the gradient of the entropy variable, respectively [37, 38]. As indicated in (4.14), the matrices  $[\hat{c}_{ij}]$  are block diagonal matrices with  $N^3$  five-by-five blocks corresponding to the viscous coefficients of each LGL point.<sup>13</sup> Note that (4.16) is obtained by using  $f_i^{(V)} = \hat{c}_{ij} w_{x_j} = \hat{c}_{ij} \Theta_j$ .

To obtain an equation for the entropy of the system, we follow the entropy stability analysis presented in [8, 37, 38]. Therefore, multiplying the two discrete equations in the left element by  $\tilde{\mathbf{w}}_l^\top \tilde{\mathcal{P}}_l$  and  $([\hat{c}_{ij,l}] \Theta_{j,l})^\top \mathcal{P}_l$ , respectively, and the two discrete equations in the right element by  $\tilde{\mathbf{w}}_r^\top \tilde{\mathcal{P}}_r$  and  $([\hat{c}_{ij,r}] \Theta_{j,r})^\top \mathcal{P}_r$ , respectively, the expression for the time derivative of the entropy function  $S$  in each element is

$$(4.17a) \quad \begin{aligned} & \frac{d}{dt} \tilde{\mathbf{1}}^\top \tilde{\mathcal{P}}_l \tilde{\mathbf{S}}_l + 2 \left\| \sqrt{[\hat{c}_{ij,l}]} \Theta_{j,l} \right\|_{\mathcal{P}_l}^2 \\ & + \mathbf{1}^\top \left( \hat{\mathcal{P}}_{x_2 x_3, l} \hat{\mathcal{B}}_{x_1, l} \bar{\mathbf{F}}_{1, l} + \hat{\mathcal{P}}_{x_1 x_3, l} \hat{\mathcal{B}}_{x_2, l} \bar{\mathbf{F}}_{2, l} + \hat{\mathcal{P}}_{x_1 x_2, l} \hat{\mathcal{B}}_{x_3, l} \bar{\mathbf{F}}_{3, l} \right) \\ & = \mathbf{w}_l^\top \left( \mathcal{P}_{x_2 x_3, l} \mathcal{B}_{x_1, l} [\hat{c}_{1j,l}] \Theta_{j,l} + \mathcal{P}_{x_1 x_3, l} \mathcal{B}_{x_2, l} [\hat{c}_{2j,l}] \Theta_{j,l} + \mathcal{P}_{x_1 x_2, l} \mathcal{B}_{x_3, l} [\hat{c}_{3j,l}] \Theta_{j,l} \right) \\ & + \mathbf{w}_l^\top \left( \mathcal{P}_{x_2 x_3, l} \mathbf{g}_{1, l}^{(Int),q} + \mathcal{P}_{x_1 x_3, l} \mathbf{g}_{2, l}^{(Int),q} + \mathcal{P}_{x_1 x_2, l} \mathbf{g}_{3, l}^{(Int),q} \right) \\ & + ([\hat{c}_{1j,l}] \Theta_{j,l})^\top \mathcal{P}_{x_2 x_3, l} \mathbf{g}_{1, l}^{(Int),\Theta} + ([\hat{c}_{2j,l}] \Theta_{j,l})^\top \mathcal{P}_{x_1 x_3, l} \mathbf{g}_{2, l}^{(Int),\Theta} \\ & + ([\hat{c}_{3j,l}] \Theta_{j,l})^\top \mathcal{P}_{x_1 x_2, l} \mathbf{g}_{3, l}^{(Int),\Theta}, \end{aligned}$$

<sup>13</sup>In the staggered algorithm framework,  $N^3$  is the number of LGL points in a 3D tensor-product element.

(4.17b)

$$\begin{aligned}
 & \frac{d}{dt} \tilde{\mathbf{1}}^\top \widehat{\mathcal{P}}_r \tilde{\mathbf{S}}_r + 2 \left\| \sqrt{[\widehat{c}_{ij,r}]} \boldsymbol{\Theta}_{j,r} \right\|_{\mathcal{P}_r}^2 \\
 & + \mathbf{1}^\top \left( \widehat{\mathcal{P}}_{x_2x_3,r} \widehat{\mathcal{B}}_{x_1,r} \overline{\mathbf{F}}_{1,r} + \widehat{\mathcal{P}}_{x_1x_3,r} \widehat{\mathcal{B}}_{x_2,r} \overline{\mathbf{F}}_{2,r} + \widehat{\mathcal{P}}_{x_1x_2,r} \widehat{\mathcal{B}}_{x_3,r} \overline{\mathbf{F}}_{3,r} \right) \\
 & = \mathbf{w}_r^\top \left( \mathcal{P}_{x_2x_3,r} \mathcal{B}_{x_1,r} [\widehat{c}_{1j,r}] \boldsymbol{\Theta}_{j,r} + \mathcal{P}_{x_1x_3,r} \mathcal{B}_{x_2,r} [\widehat{c}_{2j,r}] \boldsymbol{\Theta}_{j,r} + \mathcal{P}_{x_1x_2,r} \mathcal{B}_{x_3,r} [\widehat{c}_{3j,r}] \boldsymbol{\Theta}_{j,r} \right) \\
 & + \mathbf{w}_r^\top \left( \mathcal{P}_{x_2x_3,r} \mathbf{g}_{1,r}^{(Int),q} + \mathcal{P}_{x_1x_3,r} \mathbf{g}_{2,r}^{(Int),q} + \mathcal{P}_{x_1x_2,r} \mathbf{g}_{3,r}^{(Int),q} \right) \\
 & + ([\widehat{c}_{1j,r}] \boldsymbol{\Theta}_{j,r})^\top \mathcal{P}_{x_2x_3,r} \mathbf{g}_{1,r}^{(Int),\Theta} + ([\widehat{c}_{2j,r}] \boldsymbol{\Theta}_{j,r})^\top \mathcal{P}_{x_1x_3,r} \mathbf{g}_{2,r}^{(Int),\Theta} \\
 & + ([\widehat{c}_{3j,r}] \boldsymbol{\Theta}_{j,r})^\top \mathcal{P}_{x_1x_2,r} \mathbf{g}_{3,r}^{(Int),\Theta},
 \end{aligned}$$

where the vectors  $\tilde{\mathbf{1}}$  and  $\mathbf{1}$  represent vectors with  $M^3$  and  $N^3$  elements, respectively (i.e.,  $\tilde{\mathbf{1}} = (1, 1, \dots, 1)_{M^3}^\top$ ).

To simplify the notation, assume that the interface between the two tensor-product cells lies at  $x_1 = 0$ . We also assume that all the points that lie on the other faces of the two cubes are treated in an SS fashion; their contributions can then be neglected without loss of generality. Then, for our analysis, we can just focus on a pair of LGL interface nodes at  $x_1 = 0$ . We then introduce the operators  $\mathbf{e}^{(-)}$  and  $\mathbf{e}^{(+)}$ , which “extract” from the cellwise solution and flux vectors only the variables associated with these LGL points (i.e., the node at the left,  $(-)$ , and at the right,  $(+)$ , of the interface).<sup>14</sup> Therefore, (4.17) reduces to

(4.18a)

$$\begin{aligned}
 & \frac{d}{dt} \tilde{\mathbf{1}}^\top \widehat{\mathcal{P}}_l \tilde{\mathbf{S}}_l + \mathbf{1}^\top \widehat{\mathcal{P}}_{x_2x_3,l} \overline{\mathbf{F}}_{1,l} + 2 \left\| \sqrt{[\widehat{c}_{ij,l}]} \boldsymbol{\Theta}_{j,l} \right\|_{\mathcal{P}_l}^2 \\
 & = \mathbf{w}_l^\top \mathcal{P}_{x_2x_3,l} [\widehat{c}_{1j,l}] \boldsymbol{\Theta}_{j,l} \mathbf{e}^{(-)} + \mathbf{w}_l^\top \mathcal{P}_{x_2x_3,l} \mathbf{g}_{1,l}^{(Int),q} \\
 & + ([\widehat{c}_{1j,l}] \boldsymbol{\Theta}_{j,l})^\top \mathcal{P}_{x_2x_3,l} \mathbf{g}_{1,l}^{(Int),\Theta},
 \end{aligned}$$

(4.18b)

$$\begin{aligned}
 & \frac{d}{dt} \tilde{\mathbf{1}}^\top \widehat{\mathcal{P}}_r \tilde{\mathbf{S}}_r - \mathbf{1}^\top \widehat{\mathcal{P}}_{x_2x_3,r} \overline{\mathbf{F}}_{1,r} + 2 \left\| \sqrt{[\widehat{c}_{ij,r}]} \boldsymbol{\Theta}_{j,r} \right\|_{\mathcal{P}_r}^2 \\
 & = -\mathbf{w}_r^\top \mathcal{P}_{x_2x_3,r} [\widehat{c}_{1j,r}] \boldsymbol{\Theta}_{j,r} \mathbf{e}^{(+)} + \mathbf{w}_r^\top \mathcal{P}_{x_2x_3,r} \mathbf{g}_{1,r}^{(Int),q} \\
 & + ([\widehat{c}_{1j,r}] \boldsymbol{\Theta}_{j,r})^\top \mathcal{P}_{x_2x_3,r} \mathbf{g}_{1,r}^{(Int),\Theta}.
 \end{aligned}$$

The interface penalty terms are constructed as a combination of a LDG-type approach and an IP technique [38]:

(4.19a)

$$\begin{aligned}
 \mathbf{g}_{1,l}^{(Int),q} & = \left[ +\overline{\mathbf{f}}_1^{(-)} - f^{(SS)} \left( q^{(-)}, q^{(+)} \right) \right] \mathbf{e}^{(-)} \\
 & + \left[ -\frac{1}{2} (1 + \alpha) \left( [\widehat{c}_{1,j}^{(-)}] \boldsymbol{\Theta}_j^{(-)} - [\widehat{c}_{1,j}^{(+)}] \boldsymbol{\Theta}_j^{(+)} \right) \right] \mathbf{e}^{(-)} \\
 & + \left[ \frac{1}{2} [L] \left( \mathbf{w}^{(-)} - \mathbf{w}^{(+)} \right) \right] \mathbf{e}^{(-)},
 \end{aligned}$$

<sup>14</sup>These two vectors are zero at all points except at the “ $(-), (+)$ ” interface points.

$$(4.19b) \quad \mathbf{g}_{1,l}^{(Int),\Theta} = \left[ -\frac{1}{2}(1 - \alpha) \left( \mathbf{w}^{(-)} - \mathbf{w}^{(+)} \right) \right] \mathbf{e}^{(-)},$$

$$(4.19c) \quad \begin{aligned} \mathbf{g}_{1,r}^{(Int),q} &= \left[ -\bar{\mathbf{f}}_1^{(+)} + f^{(SS)} \left( q^{(-)}, q^{(+)} \right) \right] \mathbf{e}^{(+)} \\ &+ \left[ +\frac{1}{2}(1 - \alpha) \left( \left[ \hat{c}_{1,j}^{(+)} \right] \Theta_j^{(+)} - \left[ \hat{c}_{1,j}^{(-)} \right] \Theta_j^{(-)} \right) \right] \mathbf{e}^{(+)} \\ &+ \left[ \frac{1}{2}[L] \left( \mathbf{w}^{(+)} - \mathbf{w}^{(-)} \right) \right] \mathbf{e}^{(+)}, \end{aligned}$$

$$(4.19d) \quad \mathbf{g}_{1,r}^{(Int),\Theta} = \left[ +\frac{1}{2}(1 + \alpha) \left( \mathbf{w}^{(+)} - \mathbf{w}^{(-)} \right) \right] \mathbf{e}^{(+)}.$$

The LDG penalty terms involve the coefficients  $\frac{1}{2}(1 \pm \alpha)$  and act only in the normal direction to the face. The IP terms involve the block diagonal parameter matrix,  $[L] = \text{Diag}[L]$ , with  $N^3$  five-by-five blocks,  $L$ , which are left unspecified for the moment.<sup>15</sup>

Herein, the solution between adjoining elements is allowed to be discontinuous. An inviscid interface flux that preserves the entropy consistency of the interior high-order accurate spatial operators [8] on either side of the interface  $f^{(SS)}(q^{(-)}, q^{(+)})$  is constructed as

$$(4.20) \quad f^{(SS)}(q^{(-)}, q^{(+)}) = f^{(SC)}(q^{(-)}, q^{(+)}) + \Lambda \left( w^{(+)} - w^{(-)} \right),$$

where  $f^{(SC)}(q^{(-)}, q^{(+)})$  is the entropy *conservative* inviscid interface flux of any order [18, 8, 7, 37, 39] constructed as in (3.8).  $\Lambda$  is a negative semidefinite interface matrix with zero or negative eigenvalues. The superscripts  $(-)$  and  $(+)$  denote the collocated values on the left and right sides of the interface, respectively. The SS flux  $f^{(SS)}(q^{(-)}, q^{(+)})$  is more dissipative than the entropy conservative inviscid flux  $f^{(SC)}(q^{(-)}, q^{(+)})$ , as can be easily verified by contracting  $f^{(SS)}(q^{(-)}, q^{(+)})$  against the entropy variables [8, 39]. Note that in [41], grid interfaces for SS finite difference schemes are studied and interface fluxes similar to (4.20) are proposed.

Substituting expressions (4.19) and (4.20) into (4.18) and summing all the contributions of the two elements result in

$$(4.21) \quad \frac{d}{dt} \tilde{\mathbf{1}}^\top \tilde{\mathcal{P}}_l \tilde{\mathbf{S}}_l + \frac{d}{dt} \tilde{\mathbf{1}}^\top \tilde{\mathcal{P}}_r \tilde{\mathbf{S}}_r + 2 \left[ \left\| \sqrt{[\hat{c}_{ij,l}]} \Theta_{j,l} \right\|_{\mathcal{P}_l}^2 + \left\| \sqrt{[\hat{c}_{ij,r}]} \Theta_{j,r} \right\|_{\mathcal{P}_r}^2 \right] = \Upsilon^{(I)} + \Upsilon^{(V)},$$

where  $\Upsilon^{(I)}$  and  $\Upsilon^{(V)}$  are the inviscid and the viscous interface terms. At the two interface nodes, these terms are

$$(4.22) \quad \begin{aligned} \Upsilon^{(I)} &= \left( w^{(+)} - w^{(-)} \right)^\top f^{(SS)} \left( q^{(-)}, q^{(+)} \right) - \left( \psi^{(+)} - \psi^{(-)} \right) \\ &= \left( w^{(+)} - w^{(-)} \right)^\top \Lambda \left( w^{(+)} - w^{(-)} \right), \end{aligned}$$

$$(4.23) \quad \Upsilon^{(V)} = \left( w^{(+)} - w^{(-)} \right)^\top L \left( w^{(+)} - w^{(-)} \right).$$

<sup>15</sup>In the staggered algorithm framework,  $N^3$  is the number of LGL points in a 3D tensor-product element.

Clearly, the interface contributions are dissipative if both the five-by-five matrices  $\Lambda$  and  $L$  are negative semidefinite. The matrix  $\Lambda$  can be constructed using different approaches. Herein, an upwind operator that dissipates each characteristic wave based on the magnitude of its eigenvalue is used [34, 8, 37, 39].

We are then left with the viscous interface term,  $\Upsilon^{(V)}$ . The parameter values  $\alpha = 0$  and  $\alpha = \pm 1$  yield a symmetric LDG and a “flip-flop” narrow stencil (nearest neighbor) LDG penalty, respectively [37, 39]. An LDG value of  $\alpha = 0$  produces a global discrete operator that has a neutrally damped spurious eigenmode. Herein, when  $\alpha = 0$ , this mode is damped using the IP dissipation. For a Reynolds number that approaches  $\infty$ , we would like the five-by-five matrix  $L$  to go to zero so that only the inviscid SS penalty contributions in (4.19) are recovered. To achieve that, the matrix  $L$  is constructed as

$$(4.24) \quad L = -\beta^{(Int)} \frac{\widehat{c}_{11}^{(-)} + \widehat{c}_{11}^{(+)}}{2p_{x_1,11}}, \quad \beta^{(Int)} > 0,$$

where  $\widehat{c}_{11}^{(-)}$  and  $\widehat{c}_{11}^{(+)}$  are the positive semidefinite viscous coefficient matrices at the left and right side of the interface in the normal direction. The coefficient  $\beta^{(Int)}$  can be used to modify the strength of the IP penalty term, although excessively large values of  $\beta^{(Int)}$  reduce the maximum stable time step. In references [37, 39, 38],  $\beta^{(Int)}$  is selected to be the maximum value for which the explicit stability constraint remains unaffected. The factor  $p_{x_1,11}$  in the denominator represents the normal local grid spacing which is incorporated in the diagonal SBP operator  $\mathcal{P}$ .<sup>16</sup> This term is introduced to get the correct dimension, and, as for the standard IP finite element approach, it increases the strength of  $L$  with increased resolution.

**5. The comparison approach using entropy conservative schemes.** Entropy conservative formulations suffer breakdown when used without dissipation to capture shocks. The entropy discontinuity at the shock is inconsistent with the underlying premise of an isentropic algorithm, and large amplitude oscillations around the shock are the result. More problematic, however, is that entropy conservative formulations cannot converge to the weak solution; there is no mechanism to admit the dissipation physically required at the shock.

Herein the comparison approach presented in [8] is used to facilitate the addition of dissipation to the entropy conservative baseline formulation. A comparison approach uses a companion algorithm in conjunction with an entropy conservative formulation (or other entropy datum). At every point in the domain, the entropy generated by the companion scheme is compared with that of the entropy datum. If the entropy condition is violated (i.e., the inequality derived in (3.4)), then more dissipation is added locally. (A popular example of schemes developed using a comparison approach are the E-schemes of Osher [36] that use a Godunov scheme as the entropy datum.)

The sufficient local conditions for entropy stability are [8]

$$(5.1) \quad (w_{i+1} - w_i)^\top \left( \bar{f}_i - \bar{f}_i^{(SC)} \right) \leq 0, \quad i = 1, 2, \dots, N - 1,$$

where  $\bar{f}_i$  is the inviscid Euler flux constructed using a high-order accurate operator which might not (always) be SS (e.g., a nodal DG method) and  $\bar{f}_i^{(SC)}$  is the entropy

<sup>16</sup>Herein,  $p_{x_i,11}$  with  $i = 1$  appears in the definition of  $L$  because the interface between the two elements is orthogonal to the  $x_1$  direction [37, 38].

conservative flux constructed as given in (3.8). These conditions can be enforced by using a limiter function of the form

$$(5.2) \quad \bar{f}_i^{(SS)} = \bar{f}_i + \delta(\bar{f}_i^{(SC)} - \bar{f}_i), \delta = \frac{\sqrt{b^2 + c^2} - b}{2\sqrt{b^2 + c^2}}, b = (w_{i+1} - w_i)^\top (\bar{f}_i^{(SC)} - \bar{f}_i), c = 10^{-12}$$

with  $\bar{f}_i^{(SS)}$  the entropy stabilized flux. The pointwise entropy stability conditions given in (5.1) and (5.2) are valid for any pair of comparison fluxes  $\bar{f}_i$  and  $\bar{f}_i^{(SC)}$ , provided that both can be expressed in telescoping flux form [8]. Note, however, that the local conditions in (5.1) do not provide insight into the magnitude of dissipation required to achieve a nonoscillatory shock.

**6. Discussion: A theoretical cost analysis.** A  $p$ th-order 3D tensor-product LGL element has  $(p+1)^3$  collocation points. The total cost of the fully discrete, explicit-in-time, operator predominantly results from the inviscid and viscous differentiation operators. Thus, the operation count for a conventional nonstaggered element [8, 39] scales as  $(p+1)^4$ . By a similar accounting, the operation count for the staggered operator scales as  $(p+2)^4$ . Assuming that the explicit time stepping stability constraint is equivalent for both schemes, the work ratio between the conventional and staggered algorithm is given by

$$(6.1) \quad \left(\frac{p+2}{p+1}\right)^4 = W_{st}.$$

Note that this theoretical cost estimate does not account for hardware effects such as cache efficiency and data movement. At present, the current implementation of the staggered SSDC algorithm is approximately 1.7–2.0 times more costly than that of conventional SSDC algorithms. Optimization of both SSDC algorithms for current and future computing hardware architectures is the goal of future work. After optimization, a thorough comparison between the two approaches will be performed.

The staggered formulation is indeed more costly than the conventional SSDC algorithm. Alternatively, the results shown in section 7 demonstrate that the fully staggered SSDC approach is significantly more accurate than the conventional LGL SSDC operator for the same solution polynomial and grid density. Thus, a significantly finer mesh is needed by the conventional LGL SSDC operator to achieve a comparable error level. Simple accounting arguments that compare the respective number of elements needed to achieve comparable error tolerances, suggest that the staggered formulation will be competitive, particularly for low polynomial orders (e.g.,  $0 \leq p \leq 3$ ). We reiterate that more testing is needed before establishing the viability of the staggered approach.

## 7. Numerical results: Accuracy and robustness.

**7.1. Isentropic Euler vortex propagation.** The test case considered in this section is the (inviscid) propagation of a vortex for which an exact solution is known. This is an excellent test problem for verifying the accuracy and functionality of the inviscid components of a compressible Navier–Stokes solver. It is fully described by

(7.1)

$$\begin{aligned}
 f(x_1, x_2, x_3, t) &= 1 - \left[ (x_1 - x_{1,0} - U_\infty \cos(\alpha) t)^2 + (x_2 - x_{2,0} - U_\infty \sin(\alpha) t)^2 \right], \\
 T(x_1, x_2, x_3, t) &= \left[ 1 - \epsilon_v^2 M_\infty^2 \frac{\gamma - 1}{8\pi^2} \exp(f(x_1, x_2, x_3, t)) \right], \\
 \rho(x_1, x_2, x_3, t) &= T^{\frac{1}{\gamma-1}}, \\
 u_1(x_1, x_2, x_3, t) &= U_\infty \cos(\alpha) - \epsilon_v \frac{x_2 - x_{2,0} - U_\infty \sin(\alpha) t}{2\pi} \exp\left(\frac{f(x_1, x_2, x_3, t)}{2}\right), \\
 u_2(x_1, x_2, x_3, t) &= U_\infty \sin(\alpha) - \epsilon_v \frac{x_1 - x_{1,0} - U_\infty \cos(\alpha) t}{2\pi} \exp\left(\frac{f(x_1, x_2, x_3, t)}{2}\right), \\
 u_3(x_1, x_2, x_3, t) &= 0,
 \end{aligned}$$

where  $U_\infty$ ,  $M_\infty$ , and  $(x_{1,0}, x_{2,0}, x_{3,0})$  are the module of the freestream velocity, the Mach number, and the coordinates of the vortex center, respectively. In this study, the values  $U_\infty = M_\infty c_\infty$ ,  $\epsilon_v = 5.0$ ,  $M_\infty = 0.5$ ,  $\gamma = 1.4$ , and  $\alpha = 45^\circ$  are used; the domain is described by

$$x_1 \in (-5, 5), \quad x_2 \in (-5, 5), \quad x_3 \in (-1, 1), \quad (x_{1,0}, x_{2,0}, x_{3,0}) = (0, 0, 0), \quad t \geq 0.$$

The boundary conditions are prescribed by penalizing the numerical solution against the exact solution by using an SAT approach [10], whereas the interface coupling between two adjoining elements is imposed using the SS treatment proposed in references [8, 38].

The accuracy of the following SS schemes is investigated using uniform Cartesian and unstructured nonuniform grids:

- Fully staggered SSDC algorithm with  $p_{LG} = 1, 2, 3, 4, 5, 10$  and  $p_{LGL} = 2, 3, 4, 5, 6, 11$ .
- Conventional LGL SSDC algorithm [8, 39] with  $p_{LGL} = 1, 2, 3, 4, 5, 10$ .

**7.1.1. Uniform Cartesian grid.** Different grid resolutions are examined, and the vortex is halfway out of the domain when the error measure is evaluated. This measures the effect of the penalty boundary conditions and the interior scheme. Figure 4 shows the convergence study for a sequence of maximum nine nested grids  $(2 \times 2 \times 2, 4 \times 4 \times 2, \dots, 512 \times 512 \times 2)^{17}$ . The  $L^2$  norms of the error decay asymptotically towards the designed rate in each case (i.e., second order, third order, fourth order, fifth order, sixth order, and eleventh order, respectively), for both fully staggered and the conventional SSDC approaches. We highlight that the fully staggered approach is significantly more accurate than the conventional algorithm for the same grid resolution. However, simple theoretical counting arguments based on inviscid flux evaluations (see section 6), seem to indicate that the cost of the staggered algorithm for a solution polynomial order  $p$  is comparable to that of an LGL operator [8, 39] with a solution polynomial order of  $(p + 1)$ .

**7.1.2. Unstructured nonuniform grids.** The goal of this study is to investigate whether the fully staggered algorithm is effectively  $(p + 1)$ -order accurate for more realistic meshes ( $p$  is the order of the polynomial supported by the LG points). As for the case of uniform Cartesian grids, different grid resolutions are examined, and the vortex is halfway out of the domain when the error measure is evaluated. A nested

<sup>17</sup>The number of grid cells is doubled every time in each coordinate direction.

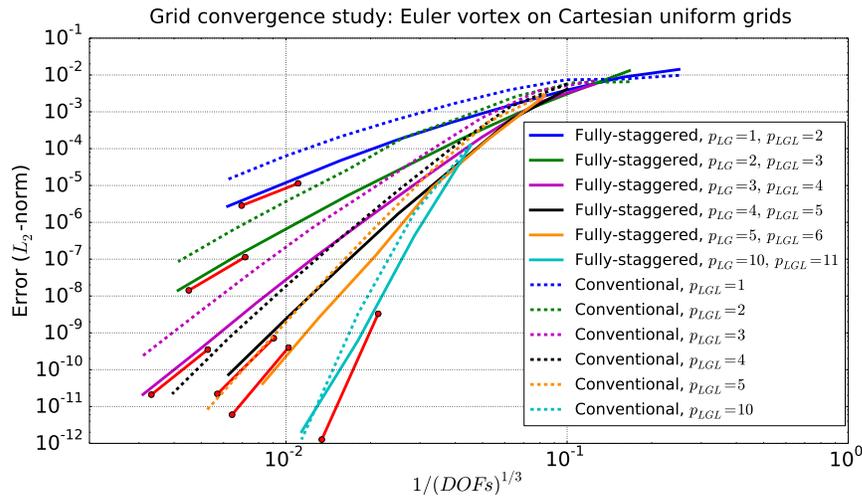


FIG. 4. Uniform grid convergence study for the inviscid vortex test case. Fully staggered SSDC algorithm (solid lines); conventional LGL SSDC algorithm [8, 39]: Dashed lines; reference curves ( $p+1$ ): Red solid lines.

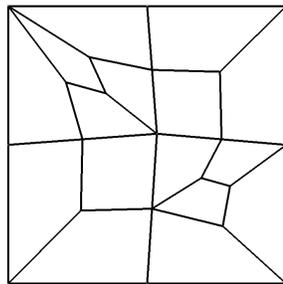


FIG. 5. “Unstructured grid kernel” used to construct a sequence of nested grids for the inviscid vortex test case.

family of grids is constructed by replicating  $N$  times the “unstructured grid kernel” shown in Figure 5. The grids used are:  $1 \times 1 \times 2$ ,  $2 \times 2$ ,  $\dots$ ,  $16 \times 16 \times 2$ , where the first integers represent the number of times that the kernel is repeated in the  $x_1$  and  $x_2$  directions; and the last integer indicates the number elements in the  $x_3$  direction.

Figure 6, show the convergence study for the sequence of maximum sixteen nested grids. For the fully staggered SSDC algorithm, the  $L^2$  norms of the error decay asymptotes towards the designed rate (i.e.,  $(p+1)$ ) and, in each case, is more accurate than the conventional LGL SSDC algorithm for the same grid resolution. The conventional path converges instead to  $p$ .

**7.2. Viscous shock.** The second test case considered in this section is the propagation of a viscous shock for which an exact time-dependent solution is known. The compressible Navier–Stokes equations support an exact solution for the viscous shock profile, under the assumption that the Prandtl number  $Pr = \frac{3}{4}$ . Mass and total enthalpy are constant across a shock. Furthermore, if  $Pr = \frac{3}{4}$  then the momentum and

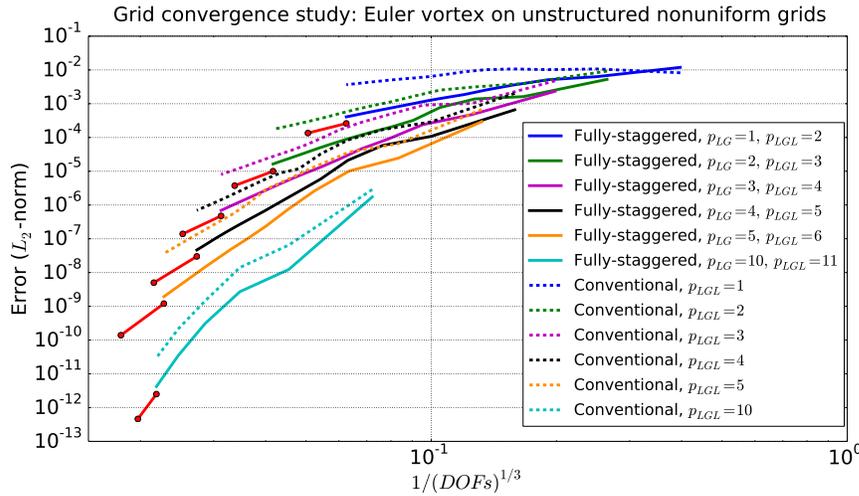


FIG. 6. Nonuniform grid convergence study for the inviscid vortex test case. Fully-staggered SSDC algorithm (solid lines); conventional LGL SSDC algorithm [8, 39]: Dashed lines; reference curves  $(p + 1)$ : Red solid lines.

energy equations are redundant. If we assume that the shock is propagating along the  $x_1$  coordinate direction,<sup>18</sup> the single momentum equation across the shock is given by

$$(7.2) \quad \begin{aligned} \alpha v \frac{\partial v}{\partial x_1} - (v - 1)(v - v_f) &= 0; & -\infty \leq x_1 \leq \infty, & \quad t \geq 0; \\ v = \frac{u_1}{u_{1,left}}; v_f = \frac{u_{1,right}}{u_{1,left}}; \alpha &= \frac{\gamma - 1}{2\gamma} \frac{\mu}{Pr \dot{m}}, \end{aligned}$$

where  $\dot{m}$  is the constant mass flow across the shock. An exact solution is obtained by solving the momentum equation (7.2) for the velocity profile,  $v$ :

$$(7.3) \quad x_1 = \frac{1}{2}\alpha \left( \text{Log} |(v - 1)(v - v_f)| + \frac{1+v_f}{1-v_f} \text{Log} \left| \frac{v-1}{v-v_f} \right| \right).$$

A moving shock is recovered by applying a uniform translation to the solution. A full derivation of this solution appears in the thesis of Fisher [17]. In this study, the values  $U_\infty = M_\infty c_\infty$ ,  $M_\infty = 2.5$ ,  $Re_\infty = 10$ , and  $\gamma = 1.4$  are used. The viscous shock, which at  $t = 0$  is located at the center of the computational domain, is propagated in the direction parallel to the horizontal axis. The domain is described by

$$x_1 \in (-10, 10), \quad x_2 \in (-10, 10), \quad x_3 \in (-1, 1), \quad t \geq 0.$$

The boundary conditions on the faces perpendicular to the horizontal axis are prescribed by penalizing the numerical solution against the exact solution; periodic boundary conditions are used on the remaining four boundary faces of the computational domain. This is an excellent test problem for verifying the accuracy and functionality of the inviscid and viscous components of a compressible Navier–Stokes solver.

The accuracy of the following SS schemes is investigated using uniform Cartesian and unstructured nonuniform grids:

<sup>18</sup>This assumption is just used to simplify the notation.

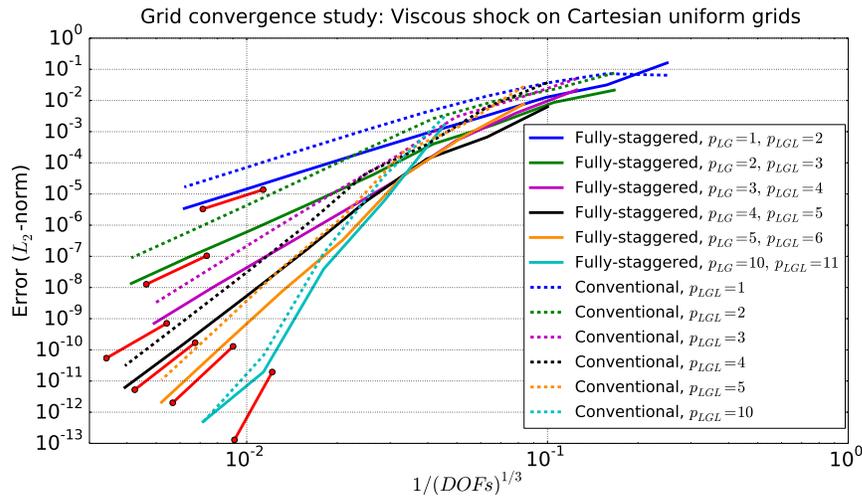


FIG. 7. Uniform grid convergence study for the viscous shock test case. Fully staggered SSDC algorithm (solid lines) versus a conventional LGL SSDC algorithm [8, 39] (dashed lines). The red lines show the  $(p + 1)$  slopes.

- Fully staggered SSDC algorithm with  $p_{LG} = 1, 2, 3, 4, 5, 10$  and  $p_{LGL} = 2, 3, 4, 5, 6, 11$ .
- Conventional LGL SSDC algorithm [8, 39] with  $p_{LGL} = 1, 2, 3, 4, 5, 10$ .

**7.2.1. Uniform Cartesian grids.** Different grid resolutions are examined, and the viscous shock has been propagated halfway out of the domain when the error measure is evaluated. This measures the effect of the penalty boundary condition and the interior scheme. Figure 7 shows the convergence study for a sequence of maximum nine nested grids ( $2 \times 2 \times 2, 4 \times 4 \times 2, \dots, 512 \times 512 \times 2$ ).<sup>19</sup> The  $L^2$  norms of the error decay asymptotically towards the designed rate in each case, for both fully staggered and conventional LGL SSDC schemes. As for the Euler vortex test case, we note that the fully staggered approach is more accurate than the conventional algorithm for the same grid resolution. However, simple theoretical counting arguments based on inviscid flux and viscous flux evaluations (see section 6), seem to indicate that the cost of the staggered algorithm for a solution polynomial order  $p$  is comparable to that of an LGL operator [8, 39] with a solution polynomial order of  $(p + 1)$ .

**7.2.2. Unstructured nonuniform grids.** A family of nested grids is constructed for the grid convergence study, by replicating  $N$  times the “unstructured grid kernel” shown in Figure 8. Figure 9 shows the convergence study for the family of maximum sixteen nested grids. As was the case in the previous study performed with the inviscid propagation of the Euler vortex, the fully staggered SSDC approach is more accurate than the conventional SSDC algorithm for the same grid resolution. Furthermore, for the fully staggered algorithm, the  $L^2$  norms of the error decay asymptotically towards the designed rate (i.e.,  $(p + 1)$ ) whereas the conventional path converges to order  $p$ .

<sup>19</sup>The number of grid cells is doubled every time in each coordinate direction.

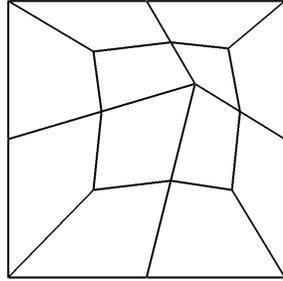


FIG. 8. Unstructured grid kernel used to construct a sequence of nested grids for the viscous shock test case.

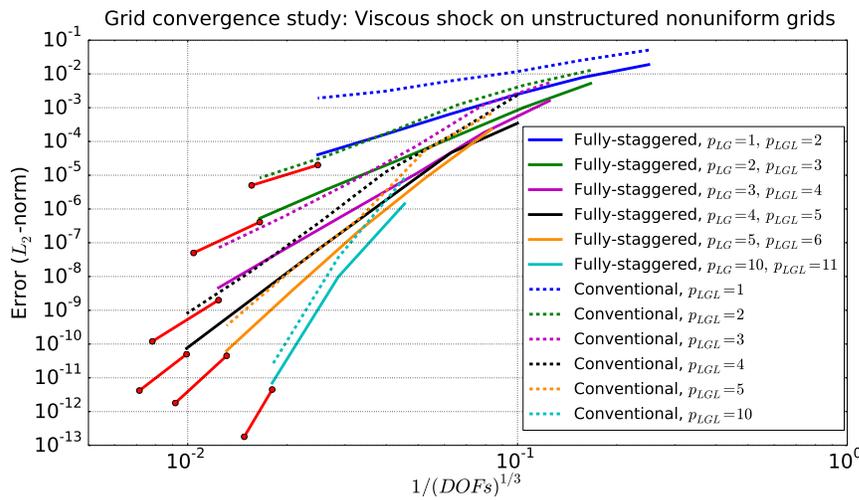


FIG. 9. Nonuniform grid convergence study for the viscous shock test case. Fully staggered SSDC algorithm (solid lines) vs. conventional LGL SSDC algorithm [8, 39] (dashed lines). The red lines show the  $(p + 1)$  slopes.

**7.3. Taylor–Green vortex.** The goal of this section is to demonstrate that nonlinearly stable schemes do not require stabilization techniques (e.g., dealiasing, filtering, limiting, overintegration procedures) for successful computations of under-resolved turbulent flows.

The Taylor–Green vortex test case is used as a benchmark model problem, and is solved on the periodic cube  $[-\pi L \leq x, y, z \leq +\pi L]$ . The initial condition is given by the following analytical expressions,

$$\begin{aligned}
 (7.4) \quad u_1 &= V_0 \sin\left(\frac{x_1}{L}\right) \cos\left(\frac{x_2}{L}\right) \cos\left(\frac{x_3}{L}\right), \\
 u_2 &= -V_0 \cos\left(\frac{x_1}{L}\right) \sin\left(\frac{x_2}{L}\right) \cos\left(\frac{x_3}{L}\right), \\
 u_3 &= 0, \\
 p &= p_0 + \frac{\rho_0 V_0^2}{16} \left[ \cos\left(\frac{2x_1}{L}\right) + \cos\left(\frac{2x_2}{L}\right) \right] \left[ \cos\left(\frac{2x_3}{L}\right) + 2 \right],
 \end{aligned}$$

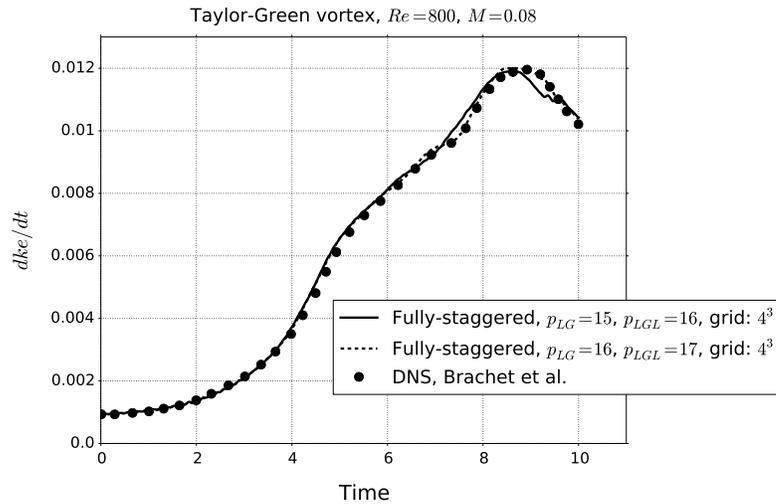


FIG. 10. Evolution of the time derivative of the kinetic energy for the Taylor-Green vortex at  $Re = 800$ ,  $M = 0.08$ ; fully-staggered SSDC algorithm.

where  $u_1$ ,  $u_2$ ,  $u_3$  are the components of the velocity in the  $x_1$ -,  $x_2$ -, and  $x_3$ -directions,  $p$  is the pressure, and  $\rho$  is the density. The flow is initialized to be isothermal, i.e.,  $p/\rho = p_0/\rho_0 = RT_0$ . The Reynolds number for this flow is defined as  $Re = (\rho_0 V_0 L)/\mu$ , where  $\mu$  is the dynamic viscosity. Starting from the initial condition, the nonlinear interactions of different flow scales yield vortex breakdowns. This nonlinear process is initially anisotropic and laminar, but subsequently, it develops into fully anisotropic turbulence that decays with the typical spectral energy distribution.

Because our framework solves the compressible Navier–Stokes equations, we choose a Mach number of  $M = 0.08$ , which leads to a flow field that is essentially incompressible. This allows for a fair comparison with some of the incompressible simulations reported in literature. Herein, two Reynolds numbers are considered:  $Re = 800$  and  $Re = 1,600$ . The Prandtl number is set to  $Pr = 0.71$ .

**7.3.1.  $Re = 800$ .** In this section we report the results for  $Re = 800$  on a uniform Cartesian grid with four hexahedrons in each coordinate direction. Therefore, the total number of elements in the grid is also  $N^e = 4^3 = 64$ . We compute the numerical solution with the sixteenth- ( $p_{LG} = 15$ ,  $p_{LGL} = 16$ ), and seventeenth-order ( $p_{LG} = 16$ ,  $p_{LGL} = 17$ ) accurate fully staggered algorithms. The numerical solutions presented herein are compared with the direct numerical simulation (DNS) of Brachet et al. [5]. Figure 10 shows the kinetic energy dissipation rate of our computations and the reference data. The computation with a formally seventeenth-order accurate scheme compares very well with the DNS results, even on this very relatively coarse grid.

**7.3.2.  $Re = 1,600$ .** The simulation is run using a fully unstructured grid which contains 42 hexahedrons. The distribution of the element in the plane is shown in Figure 11.

Figure 12 shows the kinetic energy dissipation rate of our computations with  $p_{LG} = 15$  and  $p_{LGL} = 16$ , and  $p_{LG} = 25$  and  $p_{LGL} = 26$ , and the reference data of Carton de Wiart et al. [15]. The DOFs for the computation with  $p_{LG} = 15$  and  $p_{LGL} = 16$  are too few to accurately resolve the flow field, however, the computation with a formally seventeenth-order accurate scheme is stable throughout all the

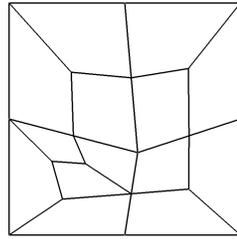


FIG. 11. Plane distribution of the elements used for the Taylor-Green vortex at  $Re = 1,600$ ,  $M = 0.08$ .

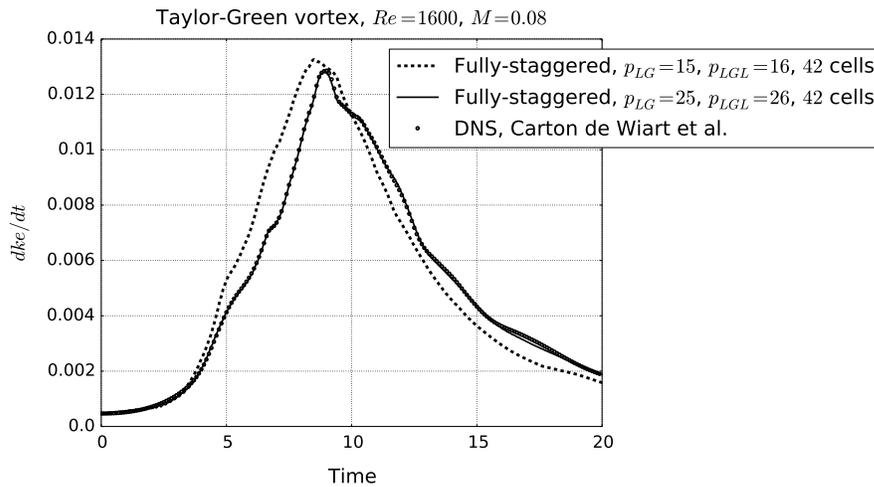


FIG. 12. Evolution of the time derivative of the kinetic energy for the Taylor-Green vortex at  $Re = 1,600$ ,  $M = 0.08$ ; fully staggered SSDC algorithm.

simulation. This is a feat unattainable with alternative approaches based on high-order accurate linear stable schemes without filtering, dealiasing, limiters, etc. [6, 22]. The robustness and accuracy of the SS fully staggered algorithm is attained also for higher-order discretization as indicated in Figure 12 for the solution computed with  $p_{LG} = 25$  and  $p_{LGL} = 26$ . In this case, the kinetic energy dissipation rate computed with a formally twenty-seventh-order accurate scheme compares very well with the DNS results. However, the dissipation rate typically converges relatively fast. Thus, to assess if the numerical solution is effectively converged it is good practice to compare enstrophy [15] as reported in Figure 13. At this resolution the enstrophy is not converged yet (spectral methods require a resolution of about  $256^3$  to converge at this Reynolds number). Thus, a finer grid or a larger polynomial order is required.

**7.4. Sod’s shock tube.** Sod’s shock tube problem is a classical Riemannian problem that evaluates the behavior of a numerical method when a shock, expansion, and contact discontinuity are present. Of particular interest is smearing in the shock and contact, or oscillations at any of the discontinuities. The governing equations are the time-dependent 1D Euler equations. Sod’s problem is initialized with

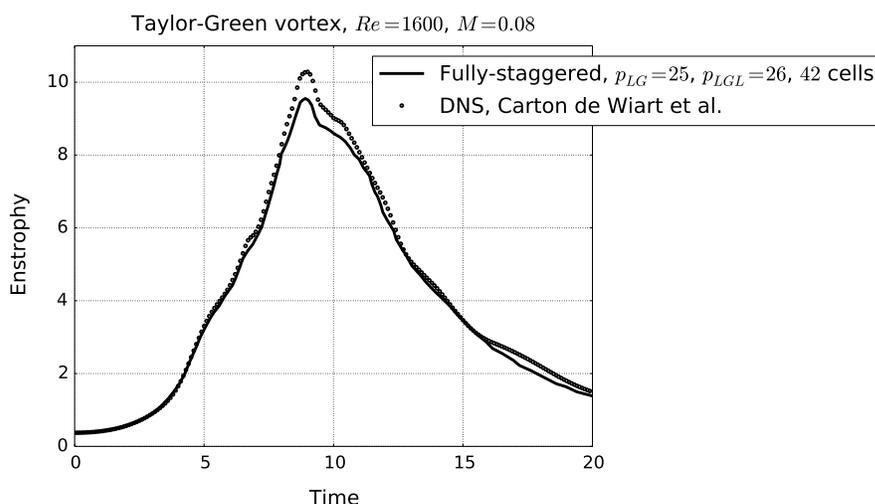


FIG. 13. Evolution of the enstrophy for the Taylor-Green vortex at  $Re = 1,600$ ,  $M = 0.08$ ; fully staggered SSDC algorithm.

$$(7.5) \quad \begin{aligned} \rho(x) &= \begin{cases} 1 & x < 0.5, \\ 1/8 & x \geq 0.5, \end{cases} & p(x) &= \begin{cases} 1, & x < 0.5, \\ 1/10, & x \geq 0.5, \end{cases} \\ u(x) &= 0, & x &\in (0, 1), \quad t \geq 0 \quad . \end{aligned}$$

All simulations use the value  $\gamma = 7/5$  for the ratio of specific heats. The governing equations are integrated in time up to  $t = 0.2$ . Neither the corrected nor the uncorrected schemes experienced instability on this test problem for elements of polynomial order  $1 \leq p_{LG} \leq 9$ .

Figures 14 and 15 show the density distribution computed with the fully staggered SSDC algorithm with  $p_{LG} = 3$ ,  $p_{LGL} = 4$  and  $p_{LG} = 9$ ,  $p_{LGL} = 10$ , respectively. Resolution of the expansion and contact discontinuity is adequate with either approach. Both the provably SS and baseline algorithms exhibit oscillations at the shock, although they capture the location correctly. Note, however, that when the entropy flux comparison is activated, the amplitudes of the oscillations are significantly reduced. The only mechanisms for dissipation in the fully staggered SSDC operators scheme are (1) boundary conditions, (2) element interface upwinding, (3) pointwise entropy correction of the fluxes; this is a remarkably low level of dissipation.

**7.5. Sine-shock interaction.** Numerical results are presented for the shock entropy-wave interaction problem. The solution of this benchmark problem contains both strong discontinuities and smooth structures and is well suited for testing high-order shock-capturing schemes. The governing equations are the time-dependent 1D Euler equations subject to the following initial conditions:

$$(7.6) \quad (\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{if } 0 \leq x < 4.5, \\ (1 + 0.2 \sin 5x, 0, 1) & \text{if } 4.5 \leq x \leq 9. \end{cases}$$

The governing equations are integrated in time up to  $t = 1.8$ . The exact solution to this problem is not available. Therefore, a numerical solution that is obtained using the conventional SSDC LGL algorithm [8, 39] with  $p_{LGL} = 3$  on a uniform grid with 1024 elements is used as the reference solution.

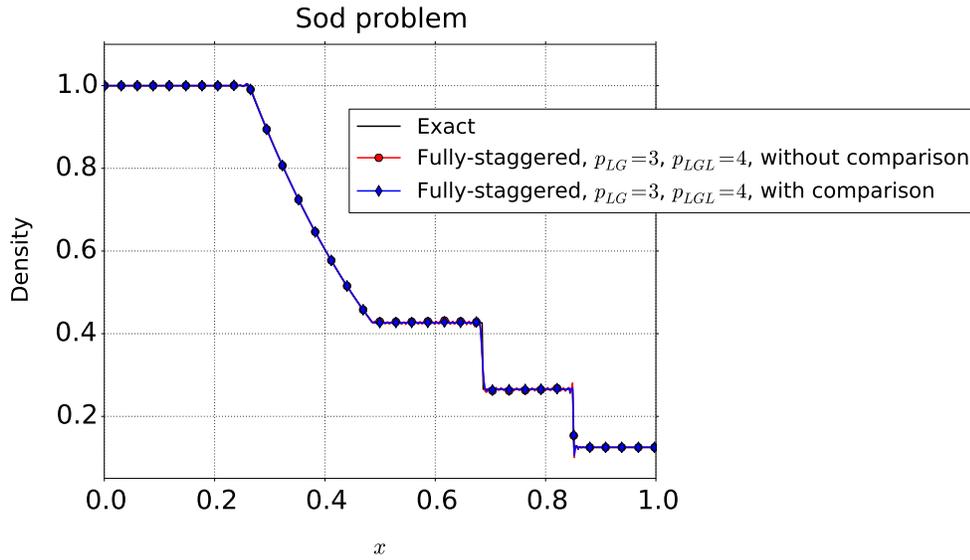


FIG. 14. Density profiles for Sod's shock tube problem at  $t = 0.2$ . Fully-staggered SSDC algorithm with  $p_{LG} = 3$ ,  $p_{LGL} = 4$  on 128 uniform elements. Line markers are plotted every 0 solution LG points

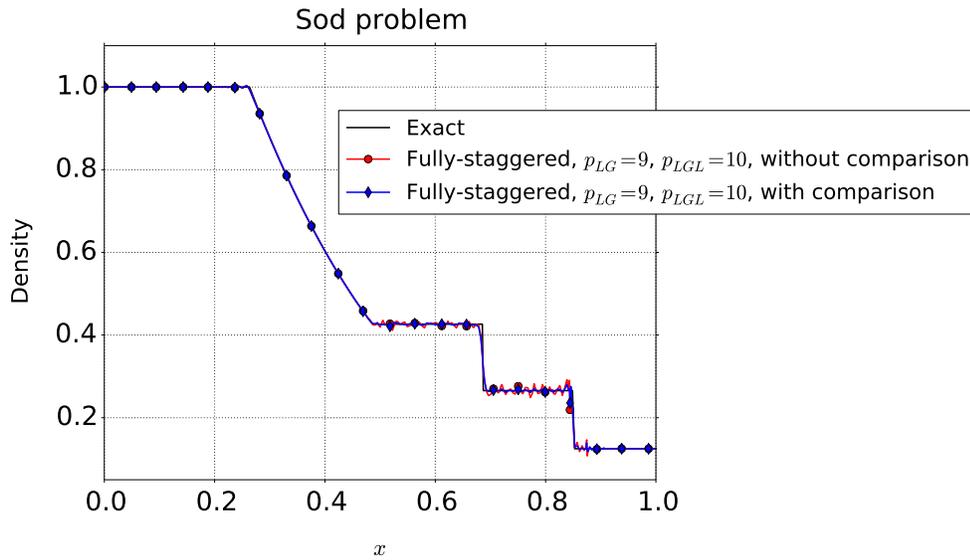


FIG. 15. Density profiles for Sod's shock tube problem at  $t = 0.2$ . Fully staggered SSDC approach with  $p_{LG} = 9$ ,  $p_{LGL} = 10$  on 32 uniform elements. Line markers are plotted every fifteen solution LG points.

Figures 16 and 17 show the density distribution computed with the fully staggered SSDC algorithm with  $p_{LG} = 3$ ,  $p_{LGL} = 4$  and  $p_{LG} = 9$ ,  $p_{LGL} = 10$ , respectively. The algorithmic observations made in Sod's problem are equally valid for the

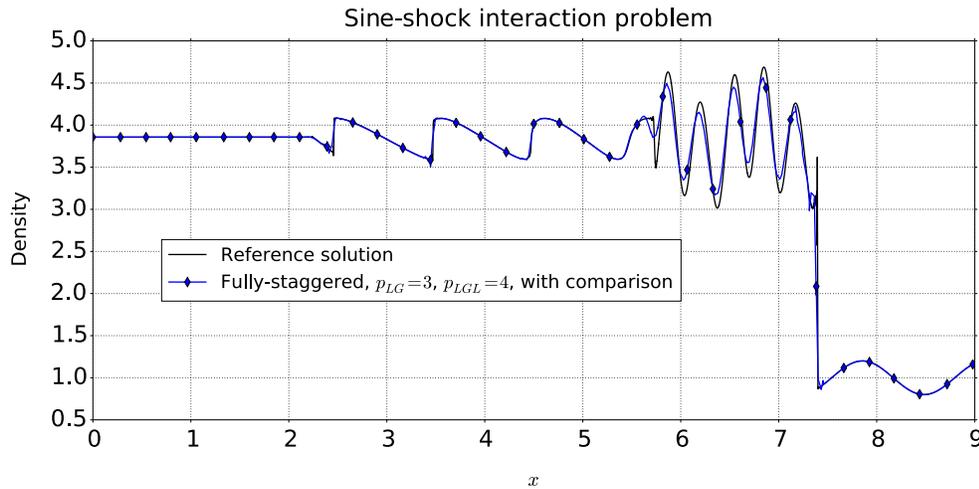


FIG. 16. Density profiles for sine-shock interaction problem at  $t = 1.8$ . Fully staggered SSDC algorithm with  $p_{LG} = 3$ ,  $p_{LGL} = 4$  on 128 uniform elements. Line markers are plotted every fifteen solution LG points.

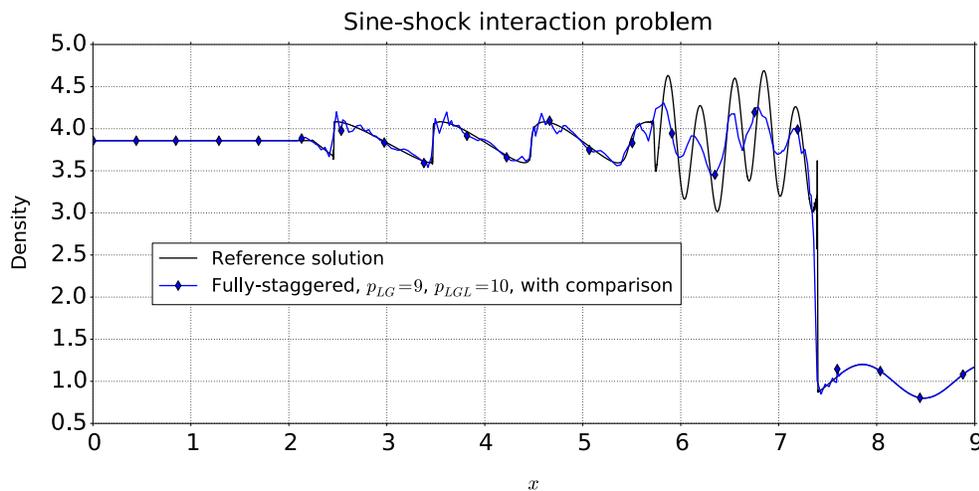


FIG. 17. Density profiles for sine-shock interaction problem at  $t = 1.8$ . Fully staggered SSDC approach with  $p_{LG} = 9$ ,  $p_{LGL} = 10$  on 32 uniform elements. Line markers are plotted every fifteen solution LG points.

sine-shock interaction problem. The coarse grid  $p_{LG} = 9$ ,  $p_{LGL} = 10$  simulation shows inadequate resolution as well as oscillations in the vicinity of the shock. None of the simulations exhibited numerical instability triggered by negative pressures.

**7.6. Computation of a square cylinder in supersonic freestream.** The development of a high-order accurate SS discretization aims to provide the next generation of robust high-fidelity numerical solvers for complex fluid flow simulations, for

which standard suboptimal algorithms suffer greatly or fail completely. By computing the flow past a 3D square cylinder at  $Re_\infty = 10^4$  and  $M_\infty = 1.5$ , we provide numerical evidence of such robustness for the complete SS high-order spatial discretization. This supersonic flow is characterized by a very large range of length scales, strong shocks, and expansion regions that interact with each other, leading to complex flow patterns. During the past three decades, this fluid flow problem has been thoroughly investigated by several researchers for aerodynamic applications (see for instance, [35, 4]).

The domain of interest spans one square cylinder edge in the  $x_3$  direction, and at the two planes perpendicular to this coordinate direction, periodic boundary conditions are used. The flow is computed using an unstructured grid with 43,936 hexahedrons. A fourth-order accurate ( $p_{LG} = 3$ ,  $p_{LGL} = 4$ ) fully staggered SS discretization without any stabilization technique is used. The body surface is considered adiabatic, and the wall BCs are imposed using the SS approach presented in [39]. The solution is initialized using a uniform flow at  $M_\infty = 1.5$  with zero angle of attack. This test case was also used in reference [39] to assess the robustness of the conventional SSDC algorithm.

A strong shock is formed in front of the bluff body at the beginning of the simulation. Subsequently, the discontinuity moves upstream until it reaches a “stationary” position that is about 2.15 square cylinder edges from the frontal surface of the body. During this phase, additional weaker shocks, which originate from the four sharp corners of the body, interact with the subsonic regions formed near the walls. This complicated flow pattern yields the formation of shocklets in the wake of the square cylinder.

A global view of the “high-order grid,” the Mach number, density, temperature, and entropy contours at  $t = 100$  are shown in Figure 18. The shock has already reached a stationary position at  $t = 100$ , and the flow past the square cylinder is completely unsteady, characterized by subsonic and supersonic regions. The formation of shocklets in the near wake region are clearly visible.

An extensive parametric study of Mach numbers ( $1.1 < M < 1.8$ ) and grids ( $10K - 40K$  elements) was performed using the supersonic square cylinder. All high Mach number simulations performed to date suggest that the staggered and conventional approaches have comparable robustness. The failure mode for both approaches is a negative density that develops in the vicinity of a strong shock.

**8. Conclusions.** An SBP-SAT framework is used to develop a generalized SS spectral element formulation that includes a broader selection of collocation points. A necessary condition for an SS, staggered operator is a restriction/prolongation interpolation pair that satisfies a precise Galerkin constraint (see (2.18)). This constraint is automatically satisfied when interpolating between the LG and LGL points, provided the polynomial order of the LGL points exceeds that of the LG points (as well as other less desirable combinations). Thus, the primary goals herein are to (1) extend the SS mechanics to a staggered configuration that collocates the solution variables at the LG points and the fluxes at the LGL points, and (2) begin to investigate the competitive advantages (if any) of the new SS staggered algorithm relative to the algorithms presented in [8, 39].

The entropy analysis techniques presented in [8, 39] are used to develop an entropy conservative, staggered grid, spectral collocation operator for the 3D compressible Navier–Stokes equation. Entropy conservative/stable inviscid interface operators are then developed for the staggered formulation. The viscous interface coupling is based

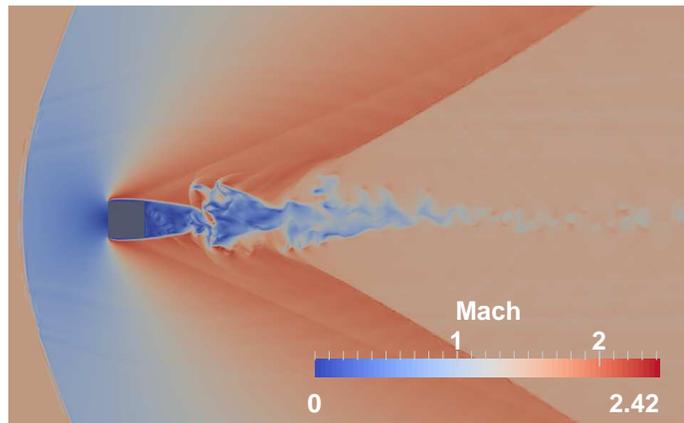


FIG. 18. *Unsteady flow past a 3D square cylinder at  $Re_\infty = 10^4$  and  $M_\infty = 1.5$ ; fourth-order accurate fully staggered SSDC method ( $p_{LG} = 3$ ,  $p_{LGL} = 4$ ) without stabilization technique;  $t = 100$ .*

on an LDG/IP approach analogous to the coupling operators presented in [8, 39]. The resulting spectral collocation operators are design-order accurate for arbitrary order, conservative on the LGL points, and satisfy the additional secondary constraint of entropy stability. Extension to curvilinear meshes [17, 20] and SS solid wall BCs [37, 39] follow immediately.

The new operators are tested on both smooth and discontinuous test problems. The 1D Sod's and sine-shock interaction discontinuous test problems are simulated with and without entropy corrections. It is shown that the base entropy conservative algorithm with upwinded interface fluxes is remarkably robust despite large oscillations at flow discontinuities. Remarkably little dissipation is needed to achieve satisfactory solutions for strong shocks. Extensive numerical tests for the 3D Euler and compressible Navier–Stokes equations reveal that the new staggered grid SS algorithms are significantly more accurate than the collocated LGL operators of equivalent polynomial order, that are presented in [8, 39]. They are however, more costly to implement from a theoretical point of view. The cost of a flux evaluation for the staggered algorithm of solution polynomial order  $p$  is comparable to that of an LGL points operator [8, 39] of solution polynomial order  $(p + 1)$ . Preliminary studies using an explicit temporal integrator indicate that the increased accuracy of the staggered approach nearly offsets the additional cost. Further, thorough investigation is required to fully establish the cost efficacy of overcollocated fluxes (including the SD and the FR operators), relative to conventional collocated fluxes. Such an analysis will undoubtedly require a deep code optimization for current and emerging data-centric computing architectures. An ongoing investigation continues and includes (1) the effects of implicit temporal integrators and (2) data movement on current and next generation hardware; computationally intensive yet extremely accurate, low memory footprint algorithms could be competitive in the future.

This work provides an essential step towards an operational SS framework of arbitrary order. An obvious extension of this work is the development of SS,  $h$ - and/or  $p$ -refinement operators. Both scenarios require data interpolation from adjoining interfaces onto a common intermediate mortar, with quadrature points that do not in general coincide. A long-term goal includes SS operators for other element types including triangles, prisms, and tetrahedra.

**Acknowledgments.** For computer time, this research partially used the resources of the Extreme Computing Research Center at King Abdullah University of Science & Technology. Special thanks are extended to Dr. Mujeeb R. Malik for partially funding this work as part of NASA's Transformational Tools and Technologies ( $T^3$ ) project.

## REFERENCES

- [1] H. L. ATKINS AND C.-W. SHU, *Quadrature-free implementation of discontinuous Galerkin method for hyperbolic equations*, AIAA J., 36 (1998), pp. 775–782.
- [2] C. BERNARDI AND Y. MADAY, *A collocation method over staggered grids for the Stokes problem*, Internat. J. Numer. Methods Fluids, 8 (1988), pp. 537–557.
- [3] C. BERNARDI, Y. MADAY, AND A. T. PATERA, *A new nonconforming approach to domain decomposition: The mortar element method*, vol., in Nonlinear Partial Differential Equations and their Applications, Collège de France Seminar, Vol. XI Pitman Res. Notes Math. Ser., Longman Scientific and Technical, Harlow, England 299 (1994), pp. 13–51.
- [4] T. BIRCH, S. A. PRINCE, AND G. M. SIMPSON, *An Experimental and Computational Study of the Aerodynamics of a Square Cross-Section Body at Supersonic Speeds*, Technical report RTO-MP-069(I), DERA, United Kingdom, 2003.
- [5] M. E. BRACHET, D. MEIRON, S. ORSZAG, B. NICKEL, R. MORF, AND U. FRISCH, *The Taylor–Green vortex and fully developed turbulence*, J. Stat. Phys., 34 (1984), pp. 1049–1063.
- [6] J. R. BULL AND A. JAMESON, *Simulation of the compressible Taylor–Green vortex using high-order flux reconstruction schemes*, in 7th AIAA Theoretical Fluid Mechanics Conference, AIAA 2014-3210, Curran, Red Hook, NY, 2014.
- [7] M. H. CARPENTER AND T. C. FISHER, *Entropy Stable Spectral Collocation Schemes for the Navier–Stokes Equations: Discontinuous Interfaces*, Technical report NASA TM 218039, 2013.
- [8] M. H. CARPENTER, T. C. FISHER, E. J. NIELSEN, AND S. H. FRANKEL, *Entropy stable spectral collocation schemes for the Navier–Stokes equations: Discontinuous interfaces*, SIAM Journal on Scientific Computing, 36 (2014), pp. B835–B867.
- [9] M. H. CARPENTER AND D. GOTTLIEB, *Spectral methods on arbitrary grids*, J. Comput. Phys., 129 (1996), pp. 74–86.
- [10] M. H. CARPENTER, D. GOTTLIEB, AND S. ABARBANEL, *Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes*, J. Comput. Phys., 111 (1994), pp. 220–236.
- [11] M. H. CARPENTER, M. PARSANI, T. C. FISHER, AND E. J. NIELSEN, *Towards an entropy stable spectral element framework for computational fluid dynamics*, in 54th AIAA Aerospace Sciences Meeting, AIAA 2016-1058, Curran, Red Hook, NY, 2016.
- [12] M. H. CARPENTER, M. PARSANI, T. C. FISHER, AND E. J. NIELSEN, *Entropy Stable Staggered Grid Spectral Collocation for the Burgers' and Compressible Navier–Stokes Equations*, Technical report NASA TM 218990, 2015.
- [13] M. H. CARPENTER, M. PARSANI, T. C. FISHER, AND E. J. NIELSEN, *Towards an entropy stable spectral element framework for computational fluid dynamics*, in 54th AIAA Aerospace Sciences Meeting, AIAA 2016-1058, Curran, Red Hook, NY, 2016.
- [14] C. M. DAFERMOS, *Hyperbolic Conservation Laws in Continuum Physics*, Springer-Verlag, Berlin, 2010.
- [15] C. DE WIART, K. HILLEWAERT, M. DUPONCHEEL, AND G. WINCKELMANS, *Assessment of a discontinuous Galerkin method for the simulation of vortical flows at high Reynolds number*, Internat. J. Numer. Methods Fluids, 74 (2014), pp. 469–493.
- [16] P. F. FISCHER AND E. M. RØNQUIST, *Spectral element methods for large scale parallel Navier–Stokes calculations*, Comput. Methods Appl. Mech. Engrg, 116 (1994), pp. 69–76.
- [17] T. C. FISHER, *High-order  $L^2$  Stable Multi-Domain Finite Difference Method for Compressible Flows*, PhD thesis, Purdue University, West Lafayette, IN, 2012.
- [18] T. C. FISHER AND M. H. CARPENTER, *High-order entropy stable finite difference schemes for nonlinear conservation laws: Finite domains*, J. Comput. Phys., 252 (2013), pp. 518–557.
- [19] T. C. FISHER, M. H. CARPENTER, J. NORDSTRÖM, N. K. YAMALEEV, AND R. C. SWANSON, *Discretely Conservative Finite-Difference Formulations for Nonlinear Conservation Laws in Split Form: Theory and Boundary Conditions*, Technical report NASA TM 2011-217307, 2011.

- [20] T. C. FISHER, M. H. CARPENTER, J. NORDSTRÖM, N. K. YAMALEEV, AND R. C. SWANSON, *Discretely conservative finite-difference formulations for nonlinear conservation laws in split form: Theory and boundary conditions*, J. Comput. Phys., 234 (2013), pp. 353–375.
- [21] G. J. GASSNER, *A kinetic energy preserving nodal discontinuous Galerkin spectral element method*, Internat. J. Numer. Methods Fluids, 76 (2014), pp. 28–50.
- [22] G. J. GASSNER AND A. D. BECK, *On the accuracy of high-order discretizations for underresolved turbulence simulations*, Theoretical and Computational Fluid Dynamics, 27 (2012), pp. 221–237.
- [23] J. S. HESTHAVEN AND T. WARBURTON, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Texts Appl. Math. 54 New York, Springer, 2008.
- [24] M. HUTCHINSON, A. HEINECKE, H. PABST, G. HENRY, M. PARSANI, AND D. KEYES, *Efficiency of high order spectral element methods on petascale architectures*, in Proceedings of the International Supercomputing Conference, ISC’16, 2016.
- [25] H. T. HUYNH, *A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods*, in 18th AIAA Computational Fluid Dynamics Conference, AIAA 2007-4079, AIAA, Recton, VA, 2007.
- [26] F. ISMAIL AND P. L. ROE, *Affordable, entropy-consistent Euler flux functions II: Entropy production at shocks*, J. Comput. Phys., 228 (2009), pp. 5410–5436.
- [27] D. A. KOPRIVA AND G. J. GASSNER, *On the quadrature and weak form choices in collocation type discontinuous Galerkin spectral element methods*, J. Sci. Comput., 44 (2010), pp. 136–155.
- [28] D. A. KOPRIVA AND J. H. KOLIAS, *A conservative staggered-grid Chebyshev multidomain method for compressible flows*, J. Comput. Phys., 125 (1996), pp. 244–261.
- [29] P. LAX AND B. WENDROFF, *Systems of conservation laws*, Comm. Pure Appl. Math., 13 (1960), pp. 217–237.
- [30] P. D. LAX, *Hyperbolic systems of conservation laws and the mathematical theory of shock waves*, CBMS-NSF Regional Conf, Ser. in Appl. Math 11, SIAM, Philadelphia, 1973.
- [31] Y. LIU, M. VINOKUR, AND Z. J. WANG, *Spectral difference method for unstructured grids I: Basic formulation*, J. Comput. Phys., 216 (2006), pp. 780–801.
- [32] Y. MADAY, A. T. PATERA, AND E. M. RONQUIST, *The  $pn$   $pn_2$  Method for the Approximation of the Stokes Problem*, Technical report 4, Laboratoire d’Analyse Numerique, Paris, 1992.
- [33] G. MAY AND A. JAMESON, *A spectral difference method for the Euler and Navier–Stokes equations on unstructured meshes*, in 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA 2006-304, AIAA, Reston, VA, 2006.
- [34] M. L. MERRIAM, *An Entropy-Based Approach to Nonlinear Stability*, Technical report NASA TM 101086, 1989.
- [35] T. NAKAGAWA, *Vortex shedding behind a square cylinder in transonic flows*, J. Fluid Mech., 178 (1987), pp. 303–323.
- [36] S. OSHER, *Riemann solvers, the entropy condition, and difference*, SIAM J. Numer. Anal., 21 (1984), pp. 217–235.
- [37] M. PARSANI, M. H. CARPENTER, AND E. J. NIELSEN, *Entropy Stable Wall Boundary Conditions for the Compressible Navier–Stokes Equations*, Technical report NASA TM 218282, 2014.
- [38] M. PARSANI, M. H. CARPENTER, AND E. J. NIELSEN, *Entropy stable discontinuous interfaces coupling for the three-dimensional compressible Navier–Stokes equations*, J. Comput. Phys., 290 (2015), pp. 132–138.
- [39] M. PARSANI, M. H. CARPENTER, AND E. J. NIELSEN, *Entropy stable wall boundary conditions for the three-dimensional compressible Navier–Stokes equations*, J. Comput. Phys., 292 (2015), pp. 88–113.
- [40] M. SVÄRD AND S. MISHRA, *Entropy stable schemes for initial-boundary-value conservation laws*, Z. Angew. Math. Phys., 63 (2012), pp. 985–1003.
- [41] M. SVÄRD AND H. ÖZCAN, *Entropy-stable schemes for the Euler equations with far-field and wall boundary conditions*, J. Sci. Comput., 58 (2014), pp. 61–89.
- [42] M. SVÄRD, *Weak solutions and convergent numerical schemes of modified compressible Navier–Stokes equations*, J. Comput. Phys., 288 (2015), pp. 19–51.
- [43] E. TADMOR, *Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems*, Acta Numer., 12 (2003), pp. 451–512.
- [44] E. TADMOR AND W. ZHONG, *Entropy stable approximations of the Navier–Stokes equations with no artificial numerical viscosity*, J. Hyperbolic Differ. Equ., 3 (2006), pp. 529–559.
- [45] P. E. VINCENT, P. CASTONGUAY, AND A. JAMESON, *A new class of high-order energy stable flux reconstruction schemes*, J. Sci. Comput., 47 (2011), pp. 50–72.