EFFICIENT ENTROPY STABLE GAUSS COLLOCATION METHODS

JESSE CHAN, DAVID C. DEL REY FERNANDEZ, MARK H. CARPENTER

Abstract. The construction of high order entropy stable collocation schemes on quadrilateral and hexahedral elements has relied on the use of Gauss-Legendre-Lobatto collocation points [1, 2, 3] and their equivalence with summation-by-parts (SBP) finite difference operators [4]. In this work, we show how to efficiently generalize the construction of semi-discretely entropy stable schemes on tensor product elements to Gauss points and generalized SBP operators. Numerical experiments suggest that the use of Gauss points can significant improve accuracy on curved meshes.

1. Introduction. Time dependent nonlinear conservation laws are ubiquitous in computational fluid dynamics (CFD), for which high order methods are increasingly of interest. Such methods are more accurate per degree of freedom than low order methods, while also possessing much smaller numerical dispersion and dissipation errors. This makes high order methods especially well suited to time-dependent simulations. In this work, we focus specifically on discontinuous Galerkin methods on unstructured quadrilateral and hexahedral meshes. These methods combine properties of high order approximations with the geometric flexibility of unstructured meshing.

However, high order methods are notorious for being more prone to instability compared to low order methods [5]. This instability is addressed through various stabilization techniques (e.g. artificial viscosity, filtering, slope limiting). However, these techniques often reduce accuracy to first or second order, and can prevent solvers from realizing the advantages of high order approximations. Moreover, it is often not possible to prove that a high order scheme does not blow up even in the presence of stabilization. This ambiguity can necessitate the re-tuning of stabilization parameters, as parameters which are both stable and accurate for one problem or discretization setting may provide either too little or too much numerical dissipation for another.

The instability of high order methods is rooted in the fact that discretizations of nonlinear conservation laws do not typically satisfy a discrete analogue of the conservation or dissipation of energy (entropy). While the numerical dissipation present for low order methods serves as a stabilizing mechanism, the low of numerical dissipation in high order methods reveals the lack of an inherent statement of stability. The dissipation of entropy serves as an energetic principle for nonlinear conservation laws [6], and requires the use of the chain rule in its proof. The instability of high order methods is tied to the fact that, when discretizing systems of nonlinear PDEs, the chain rule does not typically hold at the discrete level.

The lack of a chain rule was circumvented by using a non-standard "flux differencing" formulation [1, 2, 3, 7], which is key to constructing semi-discretely entropy stable high order schemes on unstructured quadrilateral and hexahedral meshes. Flux differencing replaces the derivative of the nonlinear flux with the discrete differentiation of an auxiliary quantity. This auxiliary quantity is computed through the evaluation of a two-point entropy conservative flux [8] using pairs of solution values at quadrature points. These entropy stable schemes were later extended to non-tensor product elements using GLL-like quadrature points on triangles and tetrahedra [9, 10]. More recently, the construction of efficient entropy stable schemes was extended to more arbitrary choices of basis and quadrature [11, 12].

This paper focuses on the construction of efficient semi-discretely entropy stable collocation schemes based on Gauss nodes, using the framework introduced in [11, 12]. While entropy stable collocation schemes have been constructed on Gauss-like quadra-

ture points without boundary nodes [13], the inter-element coupling terms for such schemes introduce significantly more communication and computation than collocation points which contain boundary nodes. This work introduces new inter-element coupling terms which avoid this all-to-all coupling, requiring only communication of face values between neighboring elements.

In Section 2, we briefly review the derivation of continuous entropy inequalities for systems of nonlinear conservation laws. In Section 3, we describe how to construct entropy stable discretizations of nonlinear conservation laws using different quadrature points on affine tensor product elements. In Section 4, we describe how to extend this construction to curvilinear elements, and Section 5 presents numerical results which confirm the high order accuracy and stability of the proposed method for smooth, discontinuous, and under-resolved (turbulent) solutions of the compressible Euler equations in two and three dimensions.

2. A brief review of entropy stability theory. We are interested in methods for the numerical solution of systems of conservation laws in d dimensions

$$\frac{\partial \boldsymbol{u}}{\partial t} + \sum_{i=1}^{d} \frac{\partial \boldsymbol{f}_i(\boldsymbol{u})}{\partial x_i} = 0, \tag{2.1}$$

where $\boldsymbol{u}$ denotes the conservative variables, $\boldsymbol{f}_i(\boldsymbol{u})$ are nonlinear fluxes, and x_i denotes the ith coordinate. Many physically motivated conservation laws admit a statement of stability involving a convex scalar entropy $S(\boldsymbol{u})$. We first define the entropy variables $\boldsymbol{v}(\boldsymbol{u})$ to be the gradient of the entropy $S(\boldsymbol{u})$ with respect to the conservative variables

$$\boldsymbol{v} = \frac{\partial S(\boldsymbol{u})}{\partial \boldsymbol{u}}.$$

For a convex entropy, $\boldsymbol{v}(\boldsymbol{u})$ defines an invertible mapping from conservative to entropy variables. We denote the inverse of this mapping (from entropy to conservative variables) by $\boldsymbol{u}(\boldsymbol{v})$.

At the continuous level, it can be shown (for example, in [6]) that vanishing viscosity solutions to (2.1) satisfy the strong form of an entropy inequality

$$\frac{\partial S(\boldsymbol{u})}{\partial t} + \sum_{i=1}^{d} \frac{\partial F_i(\boldsymbol{u})}{\partial x_i} \leq 0, \tag{2.2}$$

where F_i denotes the ith scalar entropy flux function and $\psi_i(\boldsymbol{u})$ denotes the ith entropy potential

$$F_i(\boldsymbol{u}) = \boldsymbol{v}^T \frac{\partial \boldsymbol{f}_i}{\partial x_i}, \qquad \psi_i(\boldsymbol{u}) = \boldsymbol{v}^T \boldsymbol{f}_i(\boldsymbol{u}) - F_i(\boldsymbol{u}).$$

Integrating (2.2) over a domain Ω and applying the divergence theorem yields an integrated entropy inequality

$$\int_{\Omega} \frac{\partial S(\boldsymbol{u})}{\partial t} + \int_{\partial\Omega} \sum_{i=1}^{d} \boldsymbol{n}_i \left(\boldsymbol{v}^T \boldsymbol{f}_i(\boldsymbol{u}) - \psi_i(\boldsymbol{u}) \right) \leq 0, \tag{2.3}$$

where $\partial\Omega$ denotes the boundary of Ω and $\boldsymbol{n}_i$ denotes the ith component of the outward normal on $\partial\Omega$. Roughly speaking, this implies that the time rate of change of entropy is less than or equal to the flux of entropy through the boundary.

3. Entropy stable Gauss and Gauss-Legendre-Lobatto collocation methods. This focus of this paper is on entropy stable high order collocation methods which satisfy a semi-discrete version of the entropy inequality (2.3). These methods collocate the solution at some choice of collocation nodes, and are applicable to tensor product meshes consisting of quadrilateral and hexahedral elements.

Entropy stable collocation methods have largely utilized Gauss-Legendre-Lobatto (GLL) nodes [1, 2, 3, 7], which contain points on the boundary. The popularity of GLL nodes can be attributed in part to a connection made in [4], where it was shown by Gassner that collocation DG discretizations based on GLL nodes could be recast in terms of summation-by-parts (SBP) operators. This equivalence allowed Gassner to leverage existing finite difference formulations to produce stable high order discretizations of the nonlinear Burgers' equation.

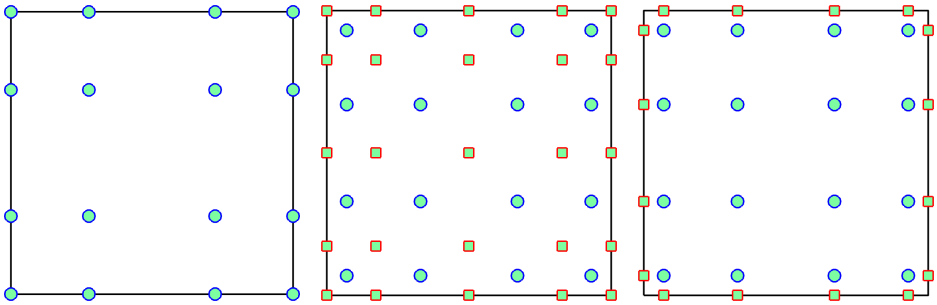
GLL quadratures contain boundary points, which greatly simplifies the construction of inter-element coupling terms for entropy stable collocation schemes. However, it is also known that the use of GLL quadrature within DG methods under-integrates the mass matrix, which can lead to solution "aliasing" and lower accuracy [14]. In this work, we explore entropy stable collocation schemes based on Gauss quadrature points instead of GLL points.

This comparison is motivated by the accuracy of each respective quadrature rule. While $(N+1)$-point GLL quadrature rules are exact for polynomial integrands of degree $(2N-1)$, $(N+1)$-point Gauss quadrature rules are exact for polynomials of degree $(2N+1)$. This higher accuracy of Gauss quadrature has been shown to translate to lower errors and slightly improved rates of convergence in simulations of wave propagation and fluid flow [15, 16, 17]. However, Gauss points have not been widely used to construct entropy stable discretizations due to the lack of efficient, stable, and high order accurate inter-element coupling terms, known as simultaneous approximation terms (SBP-SAT) in the finite difference literature [18, 13, 19]. SBP-SATs for Gauss points are non-compact, in the sense that they introduce all-to-all coupling between degrees of freedom on neighboring elements. This results in greater communication between elements, as well as a significantly larger number of two-point flux evaluations and floating point operations.

It is possible to realize the improved accuracy of Gauss points while avoiding non-compact SBP-SATs through a staggered grid formulation, where the solution is stored at Gauss nodes but interpolated to a set of higher degree $(N+2)$ GLL "flux" points for computation [14]. Because GLL nodes include boundary points, compact and high order accurate SBP-SAT terms can be constructed for the flux points. After performing computations on the flux points, the results are interpolated back to Gauss points and used to evolve the solution forward in time. Figure 1 shows an illustration of GLL, staggered grid, and Gauss point sets for a 2D quadrilateral element.

The following sections will describe how to construct efficient high order entropy stable schemes using Gauss points. These schemes are based on "decoupled" SBP operators introduced in [11, 12], which are applicable to general choices of basis and quadrature. By choosing a tensor product Lagrange polynomial basis and $(N+1)$ point Gauss quadrature rules, we recover a Gauss collocation scheme. The high order accuracy and entropy stability of this scheme are direct results of theorems presented in [11, 12]. However, we will also present a different proof of entropy stability in one dimension for completeness.

3.1. Gauss nodes and generalized summation by parts operators. We assume the solution is collocated at $(N+1)$ quadrature points x_i with associated



(a) GLL nodes (b) Staggered grid nodes (c) Gauss nodes

Fig. 1: Examples of nodal sets under which efficient entropy stable schemes can be constructed. This work focuses on the construction of efficient and accurate SBP-SAT terms for Gauss nodal sets.

quadrature weights w_i. We do not make any assumptions on the points, in order to accommodate both GLL and Gauss nodes using this notation. The collocation assumption is equivalent to approximating the solution using a degree N Lagrange basis $\ell_j(x)$ defined over the $(N+1)$ quadrature points.

Let $\boldsymbol{D}$ denote the nodal differentiation matrix, and let $\boldsymbol{V}_f$ denote the $(N+1)\times 2$ matrix which interpolates polynomials at Gauss nodes to values at endpoints. These two matrices are defined entrywise as

$$\boldsymbol{D}_{ij} = \left.\frac{\partial \ell_j}{\partial x}\right|_{x=x_i}, \qquad (\boldsymbol{V}_f)_{i1} = \ell_i(-1), \qquad (\boldsymbol{V}_f)_{i2} = \ell_i(1).$$

We also introduce the diagonal matrix of quadrature weights $\boldsymbol{W}_{ij} = \delta_{ij} w_i$. In this work, $\boldsymbol{W}$ is identical to the mass matrix, whose entries are evaluated as L^2 inner products of basis functions. This equality holds so long as the basis functions are Lagrange polynomials and integrals are approximated using quadrature (either GLL or Gauss). It can be show that the mass and differentiation matrices for Gauss nodes fall under the class of generalized SBP (GSBP) operators [20].

LEMMA 3.1. *$\boldsymbol{Q} = \boldsymbol{W}\boldsymbol{D}$ satisfies the generalized summation by parts property*

$$\boldsymbol{Q} = \boldsymbol{V}_f^T \boldsymbol{B} \boldsymbol{V}_f - \boldsymbol{Q}^T, \qquad \boldsymbol{B} = \begin{bmatrix} -1 & \\ & 1 \end{bmatrix}.$$

The proof is a direct restatement of integration by parts, and can be found in [21, 22, 23]. Lemma 3.1 holds for both GLL and Gauss nodes, and switching between these two nodal sets simply results in a redefinition of the matrices $\boldsymbol{D}, \boldsymbol{V}_f$. For example, because GLL nodes include boundary points, the interpolation matrix $\boldsymbol{V}_f$ reduces to a generalized permutation matrix which extracts the nodal values associated with the left and right endpoints.

3.2. Existing entropy stable SBP-SATs for generalized SBP operators. In this section, we will review the construction of semi-discretely entropy stable discretizations. Entropy stable discretizations can be constructed by first introducing an entropy conservative scheme, then adding appropriate interface dissipation to produce

an entropy inequality. The construction of entropy conservative schemes relies on the existence of an two-point (dyadic) entropy conservative flux [8].

DEFINITION 3.2. *Let $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ be a bivariate function which is symmetric and consistent with the flux function $\boldsymbol{f}(\boldsymbol{u})$*

$$\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R) = \boldsymbol{f}_S(\boldsymbol{u}_R, \boldsymbol{u}_L), \qquad \boldsymbol{f}_S(\boldsymbol{u}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u})$$

The numerical flux $\boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R)$ is entropy conservative if, for entropy variables $\boldsymbol{v}_L = \boldsymbol{v}(\boldsymbol{u}_L), \boldsymbol{v}_R = \boldsymbol{v}(\boldsymbol{u}_R)$, the Tadmor "shuffle" condition is satisfied

$$(\boldsymbol{v}_L - \boldsymbol{v}_R)^T \boldsymbol{f}_S(\boldsymbol{u}_L, \boldsymbol{u}_R) = (\psi_L - \psi_R), \qquad \psi_L = \psi(\boldsymbol{v}(\boldsymbol{u}_L)), \quad \psi_R = \psi(\boldsymbol{v}(\boldsymbol{u}_R)).$$

For illustrative purposes, we will prove a semi-discrete entropy inequality on a one-dimensional mesh consisting of two elements of degree N. We assume both meshes are translations of a reference element $[-1, 1]$, such that derivatives with respect to physical coordinates are identical to derivatives with respect to reference coordinates. The extension to multiple elements and variable mesh sizes is straightforward.

The construction of entropy conservative schemes relies on appropriate SBP-SATs for Gauss collocation schemes [18, 13, 19]. Let the rows of $\boldsymbol{V}_f$ be denoted as $\boldsymbol{t}_L, \boldsymbol{t}_R$

$$\boldsymbol{V}_f = \begin{bmatrix} — & \boldsymbol{t}_L & — \\ — & \boldsymbol{t}_R & — \end{bmatrix}, \qquad (\boldsymbol{t}_L)_j = \ell_j(-1), \qquad (\boldsymbol{t}_R)_j = \ell_j(1).$$

The inter-element coupling terms in [18, 13, 19] utilize a decomposition of the surface mass matrix $\boldsymbol{V}_f^T \boldsymbol{B} \boldsymbol{V}_f$ as

$$\boldsymbol{V}_f^T \boldsymbol{B} \boldsymbol{V}_f = \boldsymbol{t}_R \boldsymbol{t}_R^T - \boldsymbol{t}_L \boldsymbol{t}_L^T. \tag{3.1}$$

The construction of entropy conservative schemes on multiple elements requires appropriate inter-element coupling terms (SBP-SATs) involving $\boldsymbol{t}_L, \boldsymbol{t}_R$. We consider a two element mesh, and show how SBP-SATs can be interpreted as constructing a global SBP operator.

Let $\boldsymbol{u}_N^1, \boldsymbol{u}_N^2$ denote nodal degrees of freedom of the vector valued solution $\boldsymbol{u}(x)$ on the first and second element, respectively. To simplify notation, we assume that all following operators are defined in terms of Kronecker products, such that they are applied to each component of $\boldsymbol{u}_N^1, \boldsymbol{u}_N^2$. We first define the matrix

$$\boldsymbol{S} = \boldsymbol{Q} - \frac{1}{2} \boldsymbol{V}_f^T \boldsymbol{B} \boldsymbol{V}_f. \tag{3.2}$$

It is straightforward to show (using Lemma 3.1) that $\boldsymbol{S}$ is skew-symmetric. We can now define an SBP operator $\boldsymbol{D}_h = \boldsymbol{W}_h^{-1} \boldsymbol{Q}_h$ over two elements

$$\boldsymbol{Q}_h = \underbrace{\begin{bmatrix} \boldsymbol{S} & \frac{1}{2}\boldsymbol{t}_R \boldsymbol{t}_L^T \\ -\frac{1}{2}\boldsymbol{t}_L \boldsymbol{t}_R^T & \boldsymbol{S} \end{bmatrix}}_{\boldsymbol{S}_h} + \underbrace{\begin{bmatrix} -\frac{1}{2}\boldsymbol{t}_L \boldsymbol{t}_L^T & \\ & \frac{1}{2}\boldsymbol{t}_R \boldsymbol{t}_R^T \end{bmatrix}}_{\frac{1}{2}\boldsymbol{B}_h}, \qquad \boldsymbol{W}_h = \begin{bmatrix} \boldsymbol{W} & \\ & \boldsymbol{W} \end{bmatrix}. \tag{3.3}$$

It can be shown that $\boldsymbol{D}_h$ is high order accurate such that, if $\boldsymbol{u}_h$ is a polynomial of degree N, it is differentiated exactly. Straightforward computations show that $\boldsymbol{Q}_h$ also satisfies an SBP property $\boldsymbol{Q}_h + \boldsymbol{Q}_h^T = \boldsymbol{B}_h$.

Ignoring boundary conditions, an entropy conservative scheme for (2.1) on two elements can then be given as

$$\frac{\mathrm{d}}{\mathrm{dt}}\boldsymbol{W}_h\boldsymbol{u}_h + 2\left(\boldsymbol{Q}_h \circ \boldsymbol{F}_S\right)\mathbf{1} = 0, \qquad \boldsymbol{u}_h = \begin{bmatrix}\boldsymbol{u}_N^1 \\ \boldsymbol{u}_N^2\end{bmatrix} \tag{3.4}$$

$$\left(\boldsymbol{F}_S\right)_{ij} = \boldsymbol{f}_S\left(\left(\boldsymbol{u}_h\right)_i, \left(\boldsymbol{u}_h\right)_j\right), \qquad 1 \le i,j \le 2(N+1).$$

where $\circ$ denotes the Hadamard product [24]. Multiplying (3.4) by $\boldsymbol{v}_h^T = \boldsymbol{v}\left(\boldsymbol{u}_h\right)^T$ will yield a semi-discrete version of the conservation of entropy (mimicking (2.3) with the inequality replaced by an equality)

$$\frac{\mathrm{d}}{\mathrm{dt}}\boldsymbol{W}_h S(\boldsymbol{u}_h) + \boldsymbol{v}_h^T\left(\boldsymbol{B}_h \circ \boldsymbol{F}_S\right)\mathbf{1} - \mathbf{1}^T\boldsymbol{B}_h\psi\left(\boldsymbol{u}_h\right) = 0. \tag{3.5}$$

We refer to [13, 10] for the proof of (3.5).

The drawback of the SBP-SATs introduced in this section lies in the nature of the off-diagonal matrices $\boldsymbol{t}_R\boldsymbol{t}_L$ and $-\boldsymbol{t}_L\boldsymbol{t}_R$. For Gauss nodes, these blocks are dense, which implies that inter-element coupling terms produce a non-compact stencil. Evaluating (3.4) requires computing two-point fluxes $\boldsymbol{f}_S$ between all nodes on two neighboring elements, which significantly increases both the computational work, as well as communication between neighboring elements. The main goal of this work is to circumvent the all-to-all coupling of degrees of freedom introduced by the SBP-SATs described in this section, which can be done through the use of "decoupled" SBP operators.

3.3. Decoupled SBP operators. Decoupled SBP operators were first introduced in [11] and used to construct entropy stable schemes on simplicial elements. These operators (and simplifications under a collocation assumption) are presented in a more general setting in [11, 12] and in Appendix A. In this section, decoupled SBP operators are introduced in one dimension for GLL and Gauss nodal sets.

Decoupled SBP operators build upon the GSBP matrices $\boldsymbol{W}, \boldsymbol{Q}$, interpolation matrix $\boldsymbol{V}_f$, and boundary matrix $\boldsymbol{B}$ introduced in Section 3.1. The decoupled SBP operator $\boldsymbol{Q}_N$ is defined as the block matrix

$$\boldsymbol{Q}_N = \begin{bmatrix}\boldsymbol{Q} - \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B}\boldsymbol{V}_f & \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B} \\ -\frac{1}{2}\boldsymbol{B}\boldsymbol{V}_f & \frac{1}{2}\boldsymbol{B}\end{bmatrix}. \tag{3.6}$$

Lemma 3.1 and straightforward computations show that $\boldsymbol{Q}_N$ also satisfies the following SBP property

LEMMA 3.3. *Let $\boldsymbol{Q}_N$ be defined through (3.6). Then,*

$$\boldsymbol{Q}_N + \boldsymbol{Q}_N^T = \begin{bmatrix}\mathbf{0} & \\ & \boldsymbol{B}\end{bmatrix}.$$

The matrix $\boldsymbol{Q}_N$ acts not only on volume nodes, but on both volume and surface nodes. However, $\boldsymbol{Q}_N$ does not directly define an approximation to the derivative. Let $f(x), g(x)$ denote two functions, and let $\boldsymbol{f}, \boldsymbol{g}$ denote the values of f, g at interior nodal points. We also define vectors $\boldsymbol{f}_N, \boldsymbol{g}_N$ denoting the values of f, g at both interior and

boundary points

$$\boldsymbol{f}_N = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_{N+1}) \\ f(-1) \\ f(1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}_f \end{bmatrix}, \qquad \boldsymbol{g}_N = \begin{bmatrix} g(x_1) \\ \vdots \\ g(x_{N+1}) \\ g(-1) \\ g(1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{g} \\ \boldsymbol{g}_f \end{bmatrix}. \tag{3.7}$$

Then, a polynomial approximation to $f\frac{\partial g}{\partial x}$ can be computed using $\boldsymbol{Q}_N$. Let $\boldsymbol{u}$ denote the nodal values of the polynomial $u(x) \approx f\frac{\partial g}{\partial x}$. These coefficients are computed via

$$\boldsymbol{W}\boldsymbol{u} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{V}_f \end{bmatrix}^T \operatorname{diag}(\boldsymbol{f}_N)\, \boldsymbol{Q}_N \boldsymbol{g}_N. \tag{3.8}$$

The expression (3.8) can be rewritten in "strong" form as follows

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{W}^{-1} \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{V}_f \end{bmatrix}^T \operatorname{diag}(\boldsymbol{f}_N)\, \boldsymbol{Q}_N \boldsymbol{g}_N \\ &= \operatorname{diag}(\boldsymbol{f})\, \boldsymbol{D}\boldsymbol{g} + \frac{1}{2}\operatorname{diag}(\boldsymbol{f})\, \boldsymbol{W}^{-1}\boldsymbol{V}_f^T \boldsymbol{B}\,(\boldsymbol{g}_f - \boldsymbol{V}_f \boldsymbol{g}) \\ &\qquad + \frac{1}{2}\boldsymbol{W}^{-1}\boldsymbol{V}_f^T \boldsymbol{B} \operatorname{diag}(\boldsymbol{f}_f)\,(\boldsymbol{g}_f - \boldsymbol{V}_f \boldsymbol{g}), \end{aligned}$$

where we have used the fact that diagonal matrices commute to simplify expressions. The decoupled SBP operator $\boldsymbol{Q}_N$ can thus be interpreted as adding boundary corrections to the GSBP operator $\boldsymbol{D}$ in a skew-symmetric fashion.

It was shown in [11] that $\boldsymbol{u}$ is a high order accurate approximation to the quantity $f\frac{\partial g}{\partial x}$. Both the generalized SBP operator $\boldsymbol{D}$ and the expression in (3.8) involving the decoupled SBP operator recover exact derivatives of high order polynomials. However, when applied to non-polynomial functions, the decoupled SBP operator $\boldsymbol{Q}_N$ improves accuracy near the boundaries. Figure 2 illustrates this by using both operators to approximate the derivative of a Gaussian e^{-4x^2}. The decoupled SBP operator results in an improved approximation near the boundary.

3.4. An entropy stable Gauss collocation scheme based on decoupled SBP operators. We can now construct an entropy conservative Gauss collocation scheme with compact SBP-SATs using decoupled SBP operators. As in Section 3.2, we will construct a Gauss collocation scheme and provide a proof of semi-discrete entropy conservation for a two-element mesh.

(see Figure **??**) Add pics of two-element mesh

We first note that $\boldsymbol{B}$ can be trivially decomposed into the sum of two outer products as in (3.1)

$$\boldsymbol{B} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = \boldsymbol{e}_R \boldsymbol{e}_R^T - \boldsymbol{e}_L \boldsymbol{e}_L^T, \qquad \boldsymbol{e}_L = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \boldsymbol{e}_R = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.9}$$

The vectors $\boldsymbol{e}_L, \boldsymbol{e}_R$ are related to $\boldsymbol{t}_L, \boldsymbol{t}_R$ through the interpolation matrix $\boldsymbol{V}_f$. Because $\boldsymbol{t}_L, \boldsymbol{t}_R$ are rows of $\boldsymbol{V}_f$, $\boldsymbol{t}_L \boldsymbol{1} = \boldsymbol{t}_R \boldsymbol{1} = 1$. This can be used to show, for example, that

$$\boldsymbol{B}\boldsymbol{V}_f \boldsymbol{1} = \left(\boldsymbol{e}_R \boldsymbol{e}_R^T - \boldsymbol{e}_L \boldsymbol{e}_L\right) \boldsymbol{1} = \boldsymbol{e}_R - \boldsymbol{e}_L. \tag{3.10}$$

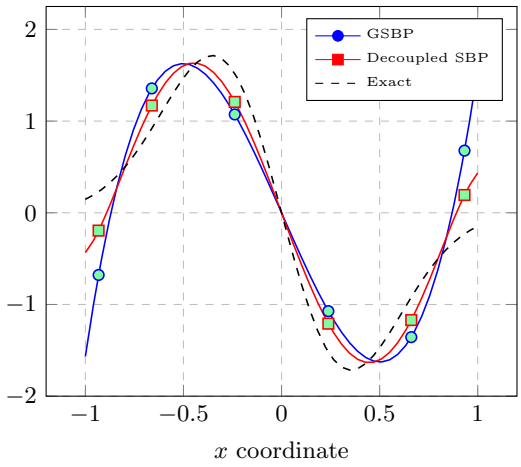


Fig. 2: Degree $N = 5$ approximations of derivatives of a Gaussian e^{-4x^2} using the generalized SBP operator $\boldsymbol{D}$ and the decoupled SBP operator $\boldsymbol{Q}_N$ via (3.8). The colored circles and squares denote values at Gauss points.

We can define a decoupled SBP matrix $\boldsymbol{Q}_h$ over two elements as follows

$$\boldsymbol{Q}_h = \left[\begin{array}{cc|cc} \boldsymbol{S} & \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B} & & \\ -\frac{1}{2}\boldsymbol{B}\boldsymbol{V}_f & -\frac{1}{2}\boldsymbol{e}_L\boldsymbol{e}_L^T & & \frac{1}{2}\boldsymbol{e}_R\boldsymbol{e}_L^T \\ \hline & & \boldsymbol{S} & \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B} \\ & -\frac{1}{2}\boldsymbol{e}_L\boldsymbol{e}_R^T & -\frac{1}{2}\boldsymbol{B}\boldsymbol{V}_f & \frac{1}{2}\boldsymbol{e}_R\boldsymbol{e}_R^T \end{array}\right] \tag{3.11}$$

We can also show that $\boldsymbol{Q}_h\mathbf{1} = 0$. Using (3.10), we have that

$$\boldsymbol{Q}_h\mathbf{1} = \begin{bmatrix} \left(\boldsymbol{S} + \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B}\right)\mathbf{1} \\ \frac{1}{2}\boldsymbol{B}\boldsymbol{V}_f\mathbf{1} + \frac{1}{2}\left(\boldsymbol{e}_R - \boldsymbol{e}_L\right) \\ \left(\boldsymbol{S} + \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B}\right)\mathbf{1} \\ \frac{1}{2}\boldsymbol{B}\boldsymbol{V}_f\mathbf{1} + \frac{1}{2}\left(\boldsymbol{e}_R - \boldsymbol{e}_L\right) \end{bmatrix} = \begin{bmatrix} \left(\boldsymbol{Q} - \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B}\boldsymbol{V}_f + \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B}\boldsymbol{V}_f\right)\mathbf{1} \\ \frac{1}{2}\boldsymbol{B}\left(-\mathbf{1} + \mathbf{1}\right) \\ \left(\boldsymbol{Q} - \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B}\boldsymbol{V}_f + \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B}\boldsymbol{V}_f\right)\mathbf{1} \\ \frac{1}{2}\boldsymbol{B}\left(-\mathbf{1} + \mathbf{1}\right) \end{bmatrix} = \mathbf{0}. \tag{3.12}$$

Here, we have used the definition of $\boldsymbol{B}$ in (3.9), the fact that $\boldsymbol{V}_f\mathbf{1} = \mathbf{1}$ and $\boldsymbol{Q}\mathbf{1} = 0$ [20], and the definition of $\boldsymbol{S}$ in (3.2). This property (3.12) will be used in the proof of entropy conservation.

It can be helpful to split up $\boldsymbol{Q}_h$ into two matrices

$$\boldsymbol{Q}_h = \boldsymbol{S}_h + \frac{1}{2}\boldsymbol{B}_h, \qquad \boldsymbol{S}_h = \left[\begin{array}{cc|cc} \boldsymbol{S} & \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B} & & \\ -\frac{1}{2}\boldsymbol{B}\boldsymbol{V}_f & & & \frac{1}{2}\boldsymbol{e}_R\boldsymbol{e}_L^T \\ \hline & & \boldsymbol{S} & \frac{1}{2}\boldsymbol{V}_f^T\boldsymbol{B} \\ & -\frac{1}{2}\boldsymbol{e}_L\boldsymbol{e}_R^T & -\frac{1}{2}\boldsymbol{B}\boldsymbol{V}_f & \end{array}\right]$$

$$\boldsymbol{B}_h = \left[\begin{array}{cc|cc} -\boldsymbol{e}_L\boldsymbol{e}_L^T & & & \\ & & & \\ \hline & & & \\ & & & \boldsymbol{e}_R\boldsymbol{e}_R^T \end{array}\right].$$

The matrix $\boldsymbol{S}_h$ is skew-symmetric, while the matrix $\boldsymbol{B}_h$ functions as a boundary operator which extracts boundary values over the two-element domain (i.e. $\boldsymbol{B}_h$ extracts

a left boundary value from element 1 and a right boundary value from element 2). Note that here, $\boldsymbol{B}_h$ is diagonal, unlike the boundary operator defined in Section 3.2.

The two-element decoupled SBP operator $\boldsymbol{Q}_h$ in (3.11) is related to the GSBP operator in (3.3) through a block interpolation matrix

$$\boldsymbol{I}_h = \begin{bmatrix} \boldsymbol{I} & \\ \boldsymbol{V}_f & \\ & \boldsymbol{I} \\ & \boldsymbol{V}_f \end{bmatrix}, \qquad \boldsymbol{I}_h^T \boldsymbol{Q}_h \boldsymbol{I}_h = \begin{bmatrix} \boldsymbol{S} & \frac{1}{2}\boldsymbol{t}_R \boldsymbol{t}_L^T \\ -\frac{1}{2}\boldsymbol{t}_L \boldsymbol{t}_R^T & \boldsymbol{S} \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}\boldsymbol{t}_L \boldsymbol{t}_L^T & \\ & \frac{1}{2}\boldsymbol{t}_R \boldsymbol{t}_R^T \end{bmatrix}.$$

Let $\boldsymbol{u}_N^1, \boldsymbol{u}_N^2$ denote the values of the conservative variables at Gauss points on elements 1 and 2, respectively, and let $\boldsymbol{u}_h$ denote their concatenation as defined in (3.4). Let $\boldsymbol{v}(\boldsymbol{u}_N^i)$ denote the evaluation of entropy variables at Gauss points on element i, and define the "entropy-projected conservative variables" $\widetilde{\boldsymbol{u}}_f^1, \widetilde{\boldsymbol{u}}_f^2$ by evaluating the conservative variables in terms of the interpolated values of the entropy variables at element boundaries

$$\widetilde{\boldsymbol{u}}_f^i = \boldsymbol{u}(\boldsymbol{v}_f^i), \qquad \boldsymbol{v}_f^i = \boldsymbol{V}_f \boldsymbol{v}(\boldsymbol{u}_N^i), \qquad i = 1, 2.$$

We now introduce $\boldsymbol{u}_h = \begin{bmatrix} \boldsymbol{u}_N^1 & \boldsymbol{u}_N^2 \end{bmatrix}^T$ as the concatenated vector of solution values at Gauss points, and define $\boldsymbol{v}_h, \widetilde{\boldsymbol{v}}$, and $\widetilde{\boldsymbol{u}}$ as follows:

$$\boldsymbol{v}_h = \boldsymbol{v}\left(\boldsymbol{u}_h\right) = \boldsymbol{v}\left(\begin{bmatrix} \boldsymbol{u}_N^1 \\ \boldsymbol{u}_N^2 \end{bmatrix}\right), \qquad \widetilde{\boldsymbol{v}} = \boldsymbol{I}_h \boldsymbol{v}_h, \qquad \widetilde{\boldsymbol{u}} = \boldsymbol{u}\left(\widetilde{\boldsymbol{v}}\right) = \begin{bmatrix} \boldsymbol{u}_N^1 \\ \widetilde{\boldsymbol{u}}_f^1 \\ \boldsymbol{u}_N^2 \\ \widetilde{\boldsymbol{u}}_f^2 \end{bmatrix}.$$

The term $\boldsymbol{v}_h$ corresponds to the vector of entropy variables at $\boldsymbol{u}_h$, while $\widetilde{\boldsymbol{v}}$ corresponds to the concatenated vector of Gauss point values and interpolated boundary values $\boldsymbol{v}_f^i$ of the entropy variables. The term $\widetilde{\boldsymbol{u}}$ denotes the evaluation of the conservative variables in terms of $\widetilde{\boldsymbol{v}}$.

Finally, we define $\boldsymbol{F}_S$ as the matrix of evaluations of the two-point flux $\boldsymbol{f}_S$ at combinations of values of $\widetilde{\boldsymbol{u}}$

$$\left(\boldsymbol{F}_S\right)_{ij} = \boldsymbol{f}_S\left(\widetilde{\boldsymbol{u}}_i, \widetilde{\boldsymbol{u}}_j\right), \quad 1 \le i, j \le 2(N+3).$$

Note that, due to the consistency of $\boldsymbol{f}_S$, the diagonal of $\boldsymbol{F}_S$ reduces to flux evaluations

$$\left(\boldsymbol{F}_S\right)_{ii} = \boldsymbol{f}_S\left(\widetilde{\boldsymbol{u}}_i, \widetilde{\boldsymbol{u}}_j\right) = \boldsymbol{f}\left(\widetilde{\boldsymbol{u}}_i\right). \tag{3.13}$$

We can now construct a semi-discretely entropy conservative formulation based on decoupled SBP operators:

THEOREM 3.4. *Let $\boldsymbol{Q}_h$ be defined by (3.11), and let $\boldsymbol{u}_h$ denote the two-element solution of the following formulation*

$$\boldsymbol{W}_h \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}_h + \boldsymbol{I}_h^T \left(\boldsymbol{Q}_h \circ \boldsymbol{F}_S\right) \boldsymbol{1} = 0. \tag{3.14}$$

Then, $\boldsymbol{u}_h$ satisfies a semi-discrete conservation of entropy

$$\boldsymbol{1}^T \boldsymbol{W}_h \frac{\mathrm{d}S(\boldsymbol{u}_h)}{\mathrm{d}t} + \boldsymbol{1}^T \boldsymbol{B}_h \left(\boldsymbol{v}_f^T \boldsymbol{f}(\widetilde{\boldsymbol{u}}_f) - \psi\left(\widetilde{\boldsymbol{u}}_f\right)\right) = 0$$

$$\boldsymbol{v}_f = \begin{bmatrix} \boldsymbol{t}_L \boldsymbol{v}\left(\boldsymbol{u}_N^1\right) \\ \boldsymbol{t}_R \boldsymbol{v}\left(\boldsymbol{u}_N^2\right) \end{bmatrix}, \qquad \widetilde{\boldsymbol{u}}_f = \boldsymbol{u}(\boldsymbol{v}_f).$$

Proof. The proof results from testing with $\boldsymbol{v}_h^T$. Since $\boldsymbol{v}(\boldsymbol{u}) = \frac{\partial S(\boldsymbol{u})}{\partial \boldsymbol{u}}$ and $\boldsymbol{W}_h$ is diagonal, the time term yields

$$\boldsymbol{v}_h^T \frac{\mathrm{d}}{\mathrm{dt}} \boldsymbol{W}_h \boldsymbol{u}_h = \mathbf{1}^T \frac{\mathrm{d}}{\mathrm{dt}} \boldsymbol{W}_h \mathrm{diag}\left(\boldsymbol{v}_h\right) \boldsymbol{u}_h = \mathbf{1}^T \frac{\mathrm{d}}{\mathrm{dt}} \boldsymbol{W}_h S(\boldsymbol{u}_h).$$

The spatial term can be manipulated as follows

$$\begin{aligned} 2\boldsymbol{v}_h^T \boldsymbol{I}_h^T \left(\boldsymbol{Q}_h \circ \boldsymbol{F}_S\right) \mathbf{1} &= 2\left(\boldsymbol{I}_h \boldsymbol{v}_h\right)^T \left(\left(\boldsymbol{S}_h \circ \boldsymbol{F}_S\right) \mathbf{1} + \frac{1}{2} \left(\boldsymbol{B}_h \circ \boldsymbol{F}_S\right) \mathbf{1} \right) \\ &= \widetilde{\boldsymbol{v}}^T \left(\boldsymbol{B}_h \circ \boldsymbol{F}_S\right) \mathbf{1} + \widetilde{\boldsymbol{v}}^T \left(\boldsymbol{S}_h \circ \boldsymbol{F}_S\right) \mathbf{1} - \mathbf{1}^T \left(\boldsymbol{S}_h \circ \boldsymbol{F}_S\right) \widetilde{\boldsymbol{v}}, \end{aligned}$$

where we have used the skew-symmetry of $\boldsymbol{S}_h$ in the last step. The boundary term reduces to

$$\widetilde{\boldsymbol{v}}^T \left(\boldsymbol{B}_h \circ \boldsymbol{F}_S\right) \mathbf{1} = \mathbf{1}^T \boldsymbol{B}_h \widetilde{\boldsymbol{v}}_f^T \boldsymbol{f}\left(\widetilde{\boldsymbol{u}}\right),$$

where we have used (3.13) and the fact that $\boldsymbol{B}_h$ is diagonal. Here, $\widetilde{\boldsymbol{v}}_f^T \boldsymbol{f}\left(\widetilde{\boldsymbol{u}}\right)$ denotes a vector whose entries are the inner product of $\widetilde{\boldsymbol{v}}$ and $\boldsymbol{f}\left(\widetilde{\boldsymbol{u}}\right)$ at face points.

The volume terms can be manipulated using the definition of $\boldsymbol{F}_S$ and the Tadmor shuffle condition in Definition 3.2

$$\begin{aligned} \widetilde{\boldsymbol{v}}^T \left(\boldsymbol{S}_h \circ \boldsymbol{F}_S\right) \mathbf{1} - \mathbf{1}^T \left(\boldsymbol{S}_h \circ \boldsymbol{F}_S\right) \widetilde{\boldsymbol{v}} &= \sum_{ij} \left(\boldsymbol{S}_h\right)_{ij} \left(\widetilde{\boldsymbol{v}}_i - \widetilde{\boldsymbol{v}}_j\right)^T \boldsymbol{f}_S\left(\widetilde{\boldsymbol{u}}_i, \widetilde{\boldsymbol{u}}_j\right) \\ &= \sum_{ij} \left(\boldsymbol{S}_h\right)_{ij} \left(\psi\left(\widetilde{\boldsymbol{u}}_i\right) - \psi\left(\widetilde{\boldsymbol{u}}_j\right)\right) \\ &= \psi\left(\widetilde{\boldsymbol{u}}\right)^T \boldsymbol{S}_h \mathbf{1} - \mathbf{1}^T \boldsymbol{S}_h \psi\left(\widetilde{\boldsymbol{u}}\right) = 2\left(\psi\left(\widetilde{\boldsymbol{u}}\right)^T \boldsymbol{S}_h \mathbf{1}\right). \end{aligned}$$

where we have again used the skew-symmetry of $\boldsymbol{S}_h$ in the last step. Substituting $\boldsymbol{S}_h = \boldsymbol{Q}_h - \frac{1}{2}\boldsymbol{B}_h$ and using (3.12) yields

$$2\left(\psi\left(\widetilde{\boldsymbol{u}}\right)^T \boldsymbol{S}_h \mathbf{1}\right) = \psi\left(\widetilde{\boldsymbol{u}}\right)^T \left(2\boldsymbol{Q}_h - \boldsymbol{B}_h\right) \mathbf{1} = -\psi\left(\widetilde{\boldsymbol{u}}\right)^T \boldsymbol{B}_h \mathbf{1} = -\mathbf{1}^T \boldsymbol{B}_h \psi\left(\widetilde{\boldsymbol{u}}\right),$$

where we have used the symmetry of $\boldsymbol{B}_h$ in the last step. ∎

REMARK. *The proof of Theorem 3.4 follows directly from choosing either GLL or Gauss quadratures in Theorem 4 of [11]. The proof is reproduced here for clarity, as the two-element case illuminates the skew-symmetric nature and structure of the inter-element coupling more explicitly.*

It is possible to convert the semi-discrete entropy equality in Theorem 3.4 to a semi-discrete entropy inequality by adding appropriate interface dissipation terms, such as Lax-Friedrichs or matrix dissipation [25]. We note that these terms must be computed in terms of $\widetilde{\boldsymbol{u}}_f$ in order to ensure a discrete dissipation of entropy [9, 11].

4. Extension to higher dimensions and curved meshes. The formulation in Theorem 3.4 can be naturally extended to Cartesian meshes in higher dimensions through a tensor product construction. We consider first the construction of higher dimensional differentiation matrices on a two-dimensional reference element $\widehat{D}$, assuming a two dimensional tensor product grid of quadrature nodes (the construction of decoupled SBP operators in three dimensions is straightforward and similar to the two-dimensional case). We then construct physical differentiation matrices on mapped elements D^k, through which we construct an entropy conservative scheme.

Let $\boldsymbol{D}_{1\mathrm{D}}, \boldsymbol{W}_{1\mathrm{D}}$ denote the 1D differentiation and mass matrices, respectively, on on the reference interval $[-1, 1]$. Let $\boldsymbol{W}$ denote the 2D reference mass matrix, and let $\boldsymbol{D}^i$ denote the differentiation matrices with respect to the ith reference coordinate. These matrices can be expressed in terms of Kronecker products

$$\boldsymbol{D}^1 = \boldsymbol{D}_{1\mathrm{D}} \otimes \boldsymbol{I}_{N+1}, \qquad \boldsymbol{D}^2 = \boldsymbol{I}_{N+1} \otimes \boldsymbol{D}_{1\mathrm{D}}, \qquad \boldsymbol{W} = \boldsymbol{W}_{1\mathrm{D}} \otimes \boldsymbol{W}_{1\mathrm{D}}.$$

where $\boldsymbol{I}_{N+1}$ denotes the $(N+1) \times (N+1)$ identity matrix.

We also construct higher dimensional face interpolation matrices. Let $\boldsymbol{V}_{f,1\mathrm{D}}$ denote the one-dimensional interpolation matrix, and let $\boldsymbol{B}_{1\mathrm{D}}$ denote the boundary matrix defined in Lemma 3.1. For an appropriate ordering of face quadrature points, the two-dimensional face interpolation matrix $\boldsymbol{V}_f$ and reference boundary matrices $\boldsymbol{B}^1, \boldsymbol{B}^2$ can be expressed as the concatenation of Kronecker product matrices

$$\boldsymbol{V}_f = \begin{bmatrix} \boldsymbol{V}_{f,1\mathrm{D}} \otimes \boldsymbol{I}_2 \\ \boldsymbol{I}_2 \otimes \boldsymbol{V}_{f,1\mathrm{D}} \end{bmatrix}, \qquad \boldsymbol{B}^1 = \begin{bmatrix} \boldsymbol{B}_{1\mathrm{D}} \otimes \boldsymbol{I}_2 & \\ & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{B}^2 = \begin{bmatrix} \boldsymbol{0} & \\ & \boldsymbol{I}_2 \otimes \boldsymbol{B}_{1\mathrm{D}} \end{bmatrix}.$$

In three dimensions, the face interpolation matrix would be expressed as the concatenation of three Kronecker products involving $\boldsymbol{V}_{f,1\mathrm{D}}$. The higher dimensional differentiation and interpolation matrices $\boldsymbol{D}^i, \boldsymbol{V}_f$ can now be used to construct higher dimensional decoupled SBP operators. Let $\boldsymbol{Q}_N^i$ denote the decoupled SBP operator for the ith coordinate on the reference element, where $\boldsymbol{Q}_N^i$ is defined as

$$\boldsymbol{Q}_N^i = \begin{bmatrix} \boldsymbol{Q}^i - \frac{1}{2}\boldsymbol{V}_f^T \boldsymbol{B}^i \boldsymbol{V}_f & \frac{1}{2}\boldsymbol{V}_f^T \boldsymbol{B}^i \\ -\frac{1}{2}\boldsymbol{B}^i \boldsymbol{V}_f & \frac{1}{2}\boldsymbol{B}^i \end{bmatrix}, \qquad \boldsymbol{Q}^i = \boldsymbol{W}\boldsymbol{D}^i.$$

Let the domain now be decomposed into non-overlapping elements D^k, such that D^k is the image of $\widehat{D}$ under a degree N polynomial mapping $\boldsymbol{\Phi}^k$. We define geometric terms $\boldsymbol{G}_{ij}^k$ as scaled derivatives of reference coordinates $\widehat{\boldsymbol{x}}$ with respect to physical coordinates $\boldsymbol{x}$

$$J^k \boldsymbol{G}_{ij}^k = J^k \frac{\partial \widehat{x}_j}{\partial \boldsymbol{x}_j},$$

where J^k is the determinant of the Jacobian of $\boldsymbol{\Phi}^k$. We also introduce the face Jacobian J_f^k, which is computed from the mapping of a physical face to a reference face. We define matrices over each physical element D^k

$$\boldsymbol{W}_k = \boldsymbol{W}\mathrm{diag}\left(\boldsymbol{J}^k\right), \qquad \boldsymbol{B}_k^i = \boldsymbol{B}_N^i \mathrm{diag}\left(\boldsymbol{n}_i \circ \boldsymbol{J}^f\right)$$

$$\boldsymbol{Q}_k^i = \frac{1}{2}\sum_{j=1}^{d} \left(\mathrm{diag}\left(\widetilde{\boldsymbol{G}}_{ij}^k\right) \boldsymbol{Q}_N^i + \boldsymbol{Q}_N^i \mathrm{diag}\left(\widetilde{\boldsymbol{G}}_{ij}^k\right) \right),$$

where we have discretized the differentiation matrix in split form [26]. Here, $\boldsymbol{J}^k$ denotes the vector of values of J^k at volume quadrature points, $\widetilde{\boldsymbol{G}}_{ij}^k$ denotes the vector of values of $J^k \boldsymbol{G}_{ij}^k$ at volume and face quadrature points, and $\boldsymbol{n}_i^k, \boldsymbol{J}_f^k$ denote the ith outward unit normal and J_f^k at face quadrature points. Then, an entropy conservative scheme in two dimensions is given as follows:

THEOREM 4.1. *Assume that* $\boldsymbol{Q}^k \boldsymbol{1} = 0$. *Let* $\boldsymbol{u}_N^k$ *denote the solution on* D^k *satis-*

fying

$$\sum_k \left(\boldsymbol{W}_k \frac{\mathrm{d}\boldsymbol{u}_N^k}{\mathrm{dt}} + \sum_{i=1}^{d} \left(2\left(\boldsymbol{Q}_k^i \circ \boldsymbol{F}_S\right) + \boldsymbol{V}_f^T \boldsymbol{B}_k^i \boldsymbol{f}_S\left(\widetilde{\boldsymbol{u}}_f^+, \widetilde{\boldsymbol{u}}_f^k\right)\right)\right) = 0$$

$$(\boldsymbol{F}_S)_{ij} = \boldsymbol{f}_S\left(\widetilde{\boldsymbol{u}}_i, \widetilde{\boldsymbol{u}}_j\right), \qquad \widetilde{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_N^k \\ \widetilde{\boldsymbol{u}}_f^k \end{bmatrix}, \qquad \widetilde{\boldsymbol{u}}_f^k = \boldsymbol{u}\left(\boldsymbol{v}_f^k\right), \qquad \boldsymbol{v}_f^k = \boldsymbol{V}_f \boldsymbol{v}\left(\boldsymbol{u}_N^k\right),$$

where $\widetilde{\boldsymbol{u}}^+$ denotes boundary values of the entropy-projected conservative variables on neighboring elements. Then, $\boldsymbol{u}_N^k$ satisfies the discrete conservation of entropy

$$\sum_k \left(\mathbf{1}^T \boldsymbol{W} \frac{\mathrm{d}\boldsymbol{u}_N^k}{\mathrm{dt}} + \sum_{i=1}^{d} \mathbf{1}^T \boldsymbol{B}^i \left(\boldsymbol{v}_f^T \boldsymbol{f}(\widetilde{\boldsymbol{u}}_f) - \psi\left(\widetilde{\boldsymbol{u}}_f\right)\right)\right) = 0.$$

Proof. The proof is a special case of the proof of Theorem 1 in [12]. ∎

We note that Theorem 4.1 relies on the assumption that $\boldsymbol{Q}^k \mathbf{1} = 0$. This condition is equivalent to ensuring that the scaled geometric terms $\widetilde{\boldsymbol{G}}_{ij}^k$ satisfy a discrete geometric conservation law (GCL) [2, 7, 10, 12]. For two-dimensional degree N (isoparametric) mappings, the GCL is automatically satisfied. However, in three dimensions, the GCL is not guaranteed to be satisfied at the discrete level, due to the fact that geometric terms for isoparametric mappings are polynomials of degree higher than N.

It is possible to ensure the satisfaction of a discrete GCL by using a sub-parametric polynomial geometric mapping. Let N_{geo} denote the degree of a polynomial geometric mapping. In three dimensions, the exact geometric terms for a degree N_{geo} polynomial mapping are polynomials of degree $(2N_{\text{geo}} - 2)$ [27, 16, 10]. If $(2N_{\text{geo}} - 2) \leq N$, or if $N_{\text{geo}} \leq \left\lfloor \frac{N}{2} \right\rfloor + 1$, then the discrete GCL is automatically satisfied.

For $N_{\text{geo}} \geq \left\lfloor \frac{N}{2} \right\rfloor + 1$, modifications to the computation of geometric terms are required to ensure that the GCL is satisfied at the discrete level. For general SBP operators, the discrete GCL can be enforced through the solution of a local least squares problem [10]. We take an alternative approach and construct geometric terms using the approach of Kopriva [27]. This construction takes advantage of the fact that GLL and Gauss collocation methods correspond to polynomial discretizations. Kopriva's construction is based on rewriting the geometric terms as the curl of some auxiliary quantity. By approximating this auxiliary quantity using polynomial interpolation prior to applying the curl, the geometric terms are approximated by degree N polynomials which satisfy the discrete GCL by construction.

To summarize, entropy stable Gauss collocation schemes on curved meshes require the following additional steps:

1. Construct polynomial approximations of the geometric terms at GLL nodes using (**??**) [27].
2. Evaluate the geometric terms at volume and surface quadrature points.
3. Compute normal vectors using (**??**).

5. Numerical results. The compressible Euler equations in d dimensions are given as follows:

$$\frac{\partial \rho}{\partial t} + \sum_{j=1}^{d} \frac{\partial \left(\rho \boldsymbol{u}_j\right)}{\partial x_j} = 0,$$

$$\frac{\partial \rho \boldsymbol{u}_i}{\partial t} + \sum_{j=1}^{d} \frac{\partial \left(\rho \boldsymbol{u}_i \boldsymbol{u}_j + p\delta_{ij}\right)}{\partial x_j} = 0, \qquad i = 1, \ldots, d$$

$$\frac{\partial E}{\partial t} + \sum_{j=1}^{d} \frac{\partial \left(\boldsymbol{u}_j (E+p)\right)}{\partial x_j} = 0.$$

Here, ρ is density, $\boldsymbol{u} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d)$ is the vector of velocities, and E is the total energy. The pressure p and specific internal energy ρe are given by

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho \left|\boldsymbol{u}\right|^2\right), \qquad \rho e = E - \frac{1}{2}\rho \left|\boldsymbol{u}\right|^2, \qquad \left|\boldsymbol{u}\right|^2 = \left(\sum_{j=1}^{d} \boldsymbol{u}_j^2\right).$$

There exists an infinite family of suitable convex entropies for the compressible Euler equations [28]. However, there is only a single unique entropy which appropriately treats the viscous heat conduction term in the compressible Navier-Stokes equations [29]. This entropy $S(\boldsymbol{u})$ is given in d dimensions by

$$S(\boldsymbol{u}) = -\frac{\rho s}{\gamma - 1},$$

where $s = \log\left(\frac{p}{\rho^\gamma}\right)$ is the physical specific entropy. The entropy variables in d dimensions are given by

$$v_1 = \frac{\rho e(\gamma + 1 - s) - E}{\rho e},$$

$$v_{1+i} = \frac{\rho \boldsymbol{u}_i}{\rho e}, \qquad i = 1, \ldots, d,$$

$$v_{d+2} = -\frac{\rho}{\rho e},$$

while the conservation variables in terms of the entropy variables are given by

$$\rho = -(\rho e) v_{d+2},$$

$$\rho \boldsymbol{u}_i = (\rho e) v_{1+i}, \qquad i = 1, \ldots, d$$

$$E = (\rho e)\left(1 - \frac{\sum_{j=1}^{d} v_{1+j}^2}{2 v_{d+2}}\right),$$

where the quantities ρe and s in terms of the entropy variables are

$$\rho e = \left(\frac{(\gamma - 1)}{(-v_{d+2})^\gamma}\right)^{1/(\gamma-1)} e^{\frac{-s}{\gamma - 1}}, \qquad s = \gamma - v_1 + \frac{\sum_{j=1}^{d} v_{1+j}^2}{2 v_{d+2}}.$$

Explicit expressions for entropy conservative numerical fluxes in two dimensions are

given by Chandrashekar [30]

$$
\begin{aligned}
f^1_{1,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= \{\{\rho\}\}^{\log}\{\{u\}\}, & f^1_{2,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= \{\{\rho\}\}^{\log}\{\{v\}\},\\
f^2_{1,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= f^1_{1,S}\{\{u\}\} + p_{\text{avg}}, & f^2_{2,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= f^1_{2,S}\{\{u\}\},\\
f^3_{1,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= f^2_{2,S}, & f^3_{2,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= f^1_{2,S}\{\{v\}\} + p_{\text{avg}},\\
f^4_{1,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= (E_{\text{avg}}+p_{\text{avg}})\{\{u\}\}, & f^4_{2,S}(\boldsymbol{u}_L,\boldsymbol{u}_R) &= (E_{\text{avg}}+p_{\text{avg}})\{\{v\}\},
\end{aligned}
$$

where we have defined the auxiliary quantities

$$
p_{\text{avg}} = \frac{\{\{\rho\}\}}{2\{\{\beta\}\}}, \qquad E_{\text{avg}} = \frac{\{\{\rho\}\}^{\log}}{2\{\{\beta\}\}^{\log}(\gamma-1)} + \frac{\|\boldsymbol{u}\|^2_{\text{avg}}}{2},
$$
$$
\|\boldsymbol{u}\|^2_{\text{avg}} = 2(\{\{u\}\}^2 + \{\{v\}\}^2) - \left(\{\{u^2\}\} + \{\{v^2\}\}\right).
$$

Expressions for entropy conservative numerical fluxes for the three-dimensional compressible Euler equations can also be explicitly written as

$$
\boldsymbol{f}_{1,S} = \begin{pmatrix} \{\{\rho\}\}^{\log}\{\{u\}\} \\ \{\{\rho\}\}^{\log}\{\{u\}\}^2 + p_{\text{avg}} \\ \{\{\rho\}\}^{\log}\{\{u\}\}\{\{v\}\} \\ \{\{\rho\}\}^{\log}\{\{u\}\}\{\{w\}\} \\ (E_{\text{avg}}+p_{\text{avg}})\{\{u\}\} \end{pmatrix}, \qquad \boldsymbol{f}_{2,S} = \begin{pmatrix} \{\{\rho\}\}^{\log}\{\{v\}\} \\ \{\{\rho\}\}^{\log}\{\{u\}\}\{\{v\}\} \\ \{\{\rho\}\}^{\log}\{\{v\}\}^2 + p_{\text{avg}} \\ \{\{\rho\}\}^{\log}\{\{v\}\}\{\{w\}\} \\ (E_{\text{avg}}+p_{\text{avg}})\{\{v\}\} \end{pmatrix},
$$
$$
\boldsymbol{f}_{3,S} = \begin{pmatrix} \{\{\rho\}\}^{\log}\{\{w\}\} \\ \{\{\rho\}\}^{\log}\{\{u\}\}\{\{w\}\} \\ \{\{\rho\}\}^{\log}\{\{v\}\}\{\{w\}\} \\ \{\{\rho\}\}^{\log}\{\{w\}\}^2 + p_{\text{avg}} \\ (E_{\text{avg}}+p_{\text{avg}})\{\{w\}\} \end{pmatrix}.
$$

with auxiliary quantities

$$
p_{\text{avg}} = \frac{\{\{\rho\}\}}{2\{\{\beta\}\}}, \qquad E_{\text{avg}} = \frac{\{\{\rho\}\}^{\log}}{2(\gamma-1)\{\{\beta\}\}^{\log}} + \frac{1}{2}\{\{\rho\}\}^{\log}\|\boldsymbol{u}\|^2_{\text{avg}}
$$
$$
\|\boldsymbol{u}\|^2_{\text{avg}} = 2(\{\{u\}\}^2 + \{\{v\}\}^2 + \{\{w\}\}^2) - \left(\{\{u^2\}\} + \{\{v^2\}\} + \{\{w^2\}\}\right).
$$

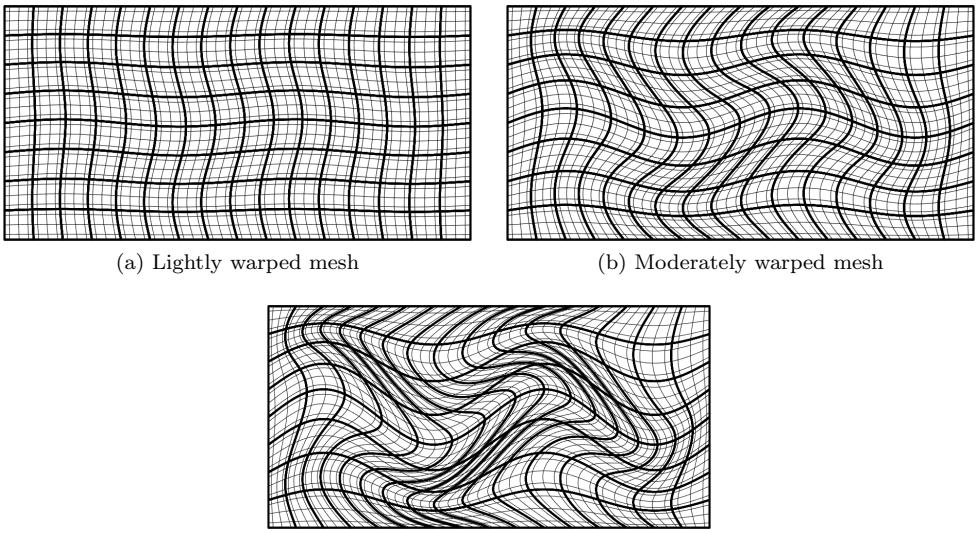
In all problems, we estimate the timestep size dt using J, J^f, and degree-dependent L^2 trace constants C_N

$$
dt = C_{\text{CFL}}\frac{h}{C_N}, \qquad h = \frac{1}{\|J^{-1}\|_{L^\infty}\|J^f\|_{L^\infty}},
$$

where C_{CFL} is some user-defined CFL constant. For isotropic elements, the ratio of J to J^f scales as the mesh size h, while C_N captures the dependence of the maximum stable timestep on the polynomial degree N. For hexahedral elements, C_N varies depending on the choice of quadrature. It was shown in [17] that

$$
C_N = \begin{cases} d\frac{N(N+1)}{2} & \text{for GLL nodes} \\ d\frac{(N+1)(N+2)}{2} & \text{for Gauss nodes} \end{cases}.
$$

Thus, based on this rough estimate of the maximum stable timestep, GLL collocation schemes should be able to take a timestep which is roughly $(1+2/N)$ times larger than



(a) Lightly warped mesh

(b) Moderately warped mesh

(c) Heavily warped mesh

Fig. 3: Lightly, moderately, and heavily warped meshes for $N = 4, K = 16$.

the maximum stable timestep for Gauss collocation schemes. We do not account for this discrepancy in this work, and instead set the timestep for both GLL and Gauss collocation schemes based on the more conservative Gauss collocation estimate of dt.

Numerical results in 1D are similar to those presented in [11]. Thus, we focus on two and three dimensional problems and comparisons of entropy stable GLL and Gauss collocation schemes.

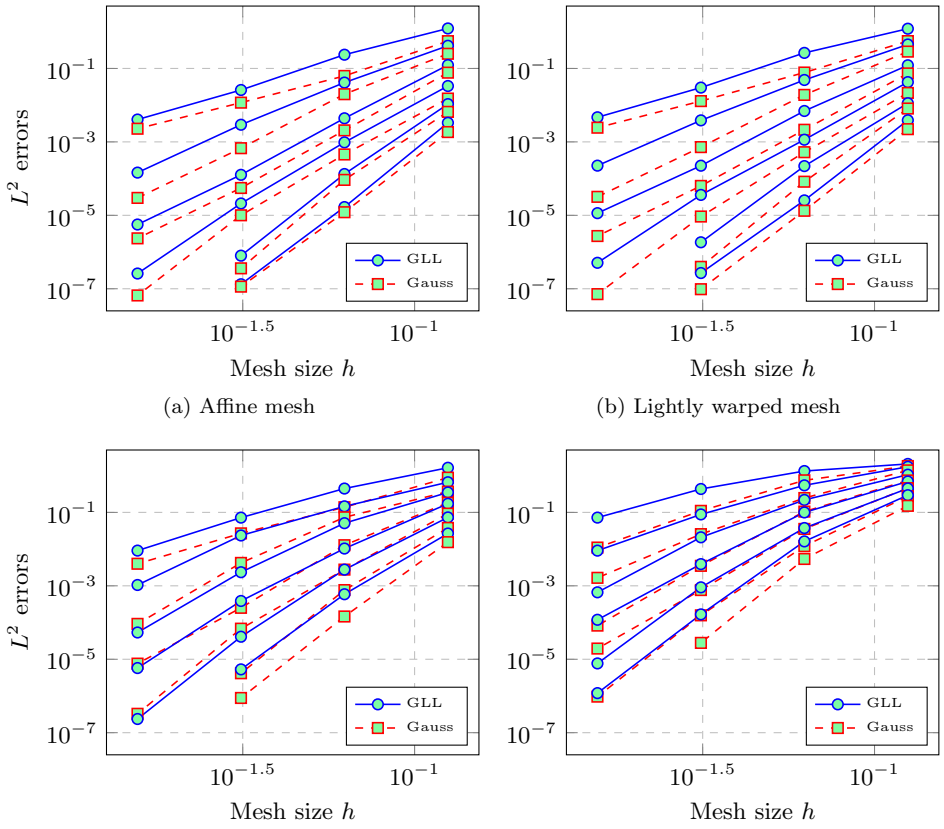
5.1. 2D isentropic vortex problem. We begin by examining high order convergence of the proposed methods in two dimensions using the isentropic vortex problem [31, 13]. The analytical solution is

$$\rho(\boldsymbol{x},t) = \left(1 - \frac{\frac{1}{2}(\gamma-1)(\beta e^{1-r(\boldsymbol{x},t)^2})^2}{8\gamma\pi^2}\right)^{\frac{1}{\gamma-1}}, \qquad p = \rho^\gamma, \tag{5.1}$$

$$u(\boldsymbol{x},t) = 1 - \frac{\beta}{2\pi} e^{1-r(\boldsymbol{x},t)^2}(y-y_0), \qquad v(\boldsymbol{x},t) = \frac{\beta}{2\pi} e^{1-r(\boldsymbol{x},t)^2}(y-y_0),$$

where u, v are the x and y velocity and $r(\boldsymbol{x},t) = \sqrt{(x-x_0-t)^2 + (y-y_0)^2}$. Here, we take $x_0 = 5, y_0 = 0$ and $\beta = 5$.

We solve on a periodic rectangular domain $[0,20] \times [-5,5]$ until final time $T = 5$, and compute errors over all all solution fields. For a degree N approximation, we approximate the L^2 error using an $(N+2)$ point Gauss quadrature rule. We also examine the influence of element curvature for both GLL and Gauss collocation schemes by examining L^2 errors on a sequence of moderately and heavily warped curvilinear meshes (see Figure 3). These warpings are constructed by modifying



(a) Affine mesh

(b) Lightly warped mesh

(c) Moderately warped mesh

(d) Heavily warped mesh

Fig. 4: L^2 errors for the 2D isentropic vortex at time $T = 5$ for degree $N = 2, \ldots, 7$ GLL and Gauss collocation schemes.

nodal positions according to the following mapping

$$\widetilde{x} = x + \alpha \cos\left(\frac{\pi}{20}(x-10)\right)\cos\left(\frac{3\pi}{10}y\right)$$
$$\widetilde{y} = y + \alpha \sin\left(\frac{\pi}{5}(\widetilde{x}-10)\right)\cos\left(\frac{\pi}{10}y\right).$$

The lightly warped mesh corresponds to $\alpha = 1/32$, the moderately warped mesh corresponds to $\alpha = 1/8$, and the heavily warped mesh corresponds to $\alpha = 1/4$. All results are computed using $C_{\text{CFL}} = 1/2$ and Lax-Friedrichs interface dissipation.

Figure 4 shows the L^2 errors for affine, moderately warped, and heavily warped meshes. For affine meshes, Gauss collocation results in a lower errors than GLL collocation at all orders. However, the difference between both schemes decreases as N increases. This is not too surprising: on a Cartesian domain, the discrete L^2 inner product resulting from GLL quadrature converges to exact L^2 inner product over the space of polynomials as N increases [32]. However, GLL and Gauss collocation differ more significantly on curved meshes. For both moderately and heavily warped meshes, the errors for a degree N Gauss collocation scheme are nearly identical to

errors for a higher order GLL collocation scheme of degree $(N+1)$. These results are in line with numerical experiments in [14], which show that GLL collocation schemes lose one order of convergence in the L^2 norm on unstructured non-uniform meshes. Both results show that increasing quadrature accuracy significantly reduces the effect of polynomial aliasing due to curved meshes and spatially varying geometric terms.

We note that L^2 approximation estimates on curved meshes [33, 34, 35] assume that the mesh size is small enough to be in the asymptotic regime (such that asymptotic error estimates hold). On curved meshes, this requires that the mesh is sufficiently fine to resolve both the solution and geometric mapping. The numerical results in Figure 4 suggest that, compared to Gauss collocation schemes, under-integrated GLL collocation schemes may require a much finer mesh resolution to reach the asymptotic regime.

5.2. 3D isentropic vortex problem. As in two dimensions, we test the accuracy of the proposed scheme using an isentropic vortex solution adapted to three dimensions. The solution is the extruded 2D vortex propagating in the y direction, with an analytic expression given in [36]

$$\begin{aligned}
\rho(\boldsymbol{x},t) &= \left(1-\frac{(\gamma-1)}{2}\Pi^2\right)^{\frac{1}{\gamma-1}} \\
\boldsymbol{u}(\boldsymbol{x},t) &= \Pi \boldsymbol{r}, \\
E(\boldsymbol{x},t) &= \frac{p_0}{\gamma-1}\left(1-\frac{\gamma-1}{2}\Pi^2\right)^{\frac{\gamma}{\gamma-1}} + \frac{\rho}{2}\left|\boldsymbol{u}\right|.
\end{aligned}$$

where $\boldsymbol{u} = (u,v,w)^T$ is the velocity vector and

$$\Pi = \Pi_{\max} e^{\frac{1-\boldsymbol{r}^T\boldsymbol{r}}{2}}, \qquad \boldsymbol{r} = \begin{pmatrix} -(x_2-c_2-t) \\ x_1-c_1 \\ 0 \end{pmatrix}.$$

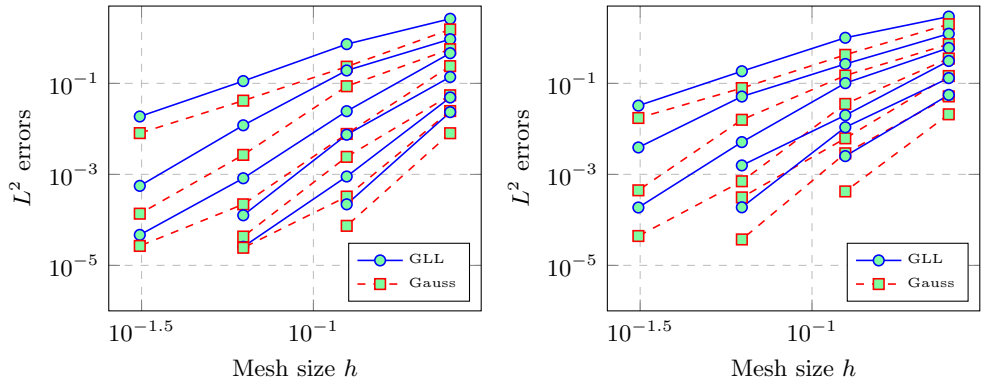
We take $c_1 = c_2 = 5$, $p_0 = 1/\gamma$, and $\Pi_{\max} = 0.4$, and solve on the domain $[0,10]\times[0,20]\times[0,10]$. As in the 2D case, we also examine the effect of curvilinear mesh warping. Following [12], we construct a curved warping of the initial Cartesian mesh by mapping nodes on each hexahedron to warped nodal positions $(\widetilde{x},\widetilde{y},\widetilde{z})$ through the transformation

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \sin\left(\pi\frac{(x-5)}{5}\right)\sin\left(2\pi\frac{(y-10)}{10}\right)\sin\left(\pi\frac{(z-5)}{5}\right).$$

On curved meshes, the geometric terms are constructed using the approach of Kopriva described in Section 4. A $C_{\text{CFL}} = .75$ is used for all experiments.

Figure 5 shows the L^2 errors for degrees $N = 2,3,4$. As in the 2D case, Gauss collocation schemes produce smaller errors than GLL collocation schemes, especially on on curved meshes. The difference between the two schemes on affine meshes is slightly more pronounced than in 2D. The difference between the two schemes on curved meshes is qualitatively similar to the 2D case, with Gauss collocation achieving higher accuracy than GLL collocation schemes at all polynomial degrees and mesh resolutions.

We also compared L^2 errors for both isoparametric and sub-parametric geometric mappings. For sub-parametric mappings, we chose the degree of approximation of geometry $N_{\text{geo}} = \left\lfloor \frac{N}{2} \right\rfloor + 1$, such that the geometric terms are computed exactly and



(a) Affine mesh (b) Curved mesh

Fig. 5: L^2 errors for the 3D isentropic vortex for $N = 2, \ldots, 7$ on sequences of Cartesian and curved meshes.

the GCL is satisfied by default. This test is intended to address the fact that the GCL-preserving interpolation of Kopriva introduces a small approximation error, as the geometric terms are no longer exact. For these sub-parametric mappings, the gap between GLL and Gauss collocation widens slightly at $N = 2$. However, the results for sub-parametric mappings are nearly identical to the results in the isoparametric case for higher polynomial degrees.

5.3. Shock-vortex interaction. describe shock-vortex problem

We modify the shock-vortex interaction problem such that wall boundary conditions are imposed at top and bottom boundaries. For left and right boundaries, we impose periodic boundary conditions.

Experiments using the matrix dissipation flux [25].

We note that entropy stable wall boundary conditionscan also be imposed when using decoupled SBP operators with over-integration as well [11]. As far as the authors know, the stable and high order accurate imposition of such boundary conditions for existing GSBP couplings [13] remains an open problem.

5.4. Inviscid Taylor-Green vortex. We conclude by investigating the behavior of entropy stable Gauss collocation schemes for the inviscid Taylor–Green vortex [37, 3, 10]. This problem is posed on the periodic box $[-\pi, \pi]^3$, with initial conditions

$$\begin{aligned} \rho &= 1 \\ u &= \sin(x_1)\cos(x_2)\cos(x_3), \\ v &= -\cos(x_1)\sin(x_2)\cos(x_3), \\ w &= 0, \\ p &= \frac{100}{\gamma} + \frac{1}{16}\left(\cos(2x_1) + \cos(2x_2)\right)\left(2 + \cos(2x_3)\right). \end{aligned}$$

The Taylor–Green vortex is used to study the transition and decay of turbulence [38]. In the absence of viscosity, the Taylor–Green vortex develops successively smaller scales as time increases. As a result, the solution is guaranteed to contain under-resolved features after a sufficiently large time. We study the evolution of the kinetic

energy $\kappa(t)$

$$\kappa(t) = \frac{1}{|\Omega|} \int_\Omega \rho \boldsymbol{u} \cdot \boldsymbol{u} \, \mathrm{d}\boldsymbol{x},$$

as well as the kinetic energy dissipation rate $-\frac{\partial \kappa}{\partial t}$, which is approximated by differencing $\kappa(t)$.

6. Conclusion. Give computational comparison: two point flux evaluations, FLOPS due to matrix operations.

Appendix A. Decoupled SBP operators for general choices of quadrature and basis.

For general choices of quadrature and basis, decoupled projection operators involve a volume quadrature interpolation matrix $\boldsymbol{V}_q$, a face quadrature interpolation matrix $\boldsymbol{V}_f$, and a quadrature-based L^2 projection matrix $\boldsymbol{P}_q$. Let $\{\phi_j\}_{j=1}^{N_p}$ denote a set of N_p basis functions, and let $\{\boldsymbol{x}_i, \boldsymbol{w}_i\}_{i=1}^{N_q}$ denote a set of N_q volume quadrature points and weights in d dimensions. Then, $\boldsymbol{V}_q, \boldsymbol{V}_f$ are given as

$$\begin{aligned} (\boldsymbol{V}_q)_{ij} &= \phi_j(\boldsymbol{x}_i), \quad 1 \leq i, j \leq N_q, \\ (\boldsymbol{V}_f)_{ij} &= \phi_j\left(\boldsymbol{x}_i^f\right), \quad 1 \leq i \leq N_q, \quad 1 \leq j \leq N_q^f. \end{aligned}$$

These matrices can be used to define the quadrature-based L^2 projection matrix $\boldsymbol{P}_q$. Let $\boldsymbol{W}$ denote the diagonal matrix of quadrature weights. Then,

$$\boldsymbol{M} = \boldsymbol{V}_q^T \boldsymbol{W} \boldsymbol{V}_q, \qquad \boldsymbol{P}_q = \boldsymbol{M}^{-1} \boldsymbol{V}_q^T \boldsymbol{W}.$$

Let $\boldsymbol{D}^i$ now denote a modal differentiation matrix with respect to the ith coordinate, which maps coefficients in the basis ϕ_j to coefficients of the ith derivative. By composing this matrix with interpolation and projection matrices, one can define differencing operators $\boldsymbol{D}_q^i = \boldsymbol{V}_q \boldsymbol{D}^i \boldsymbol{P}_q$ which map values at quadrature points to values of approximate derivatives at quadrature points. It was shown in [11] that $\boldsymbol{Q}^i = \boldsymbol{W} \boldsymbol{D}_q^i$ satisfies a generalized SBP property involving the face interpolation and projection matrices $\boldsymbol{V}_f, \boldsymbol{P}_q$.

The decoupled SBP operator $\boldsymbol{Q}_N^i$ is then given as

$$\boldsymbol{Q}_N^i = \begin{bmatrix} \boldsymbol{Q}^i - \frac{1}{2} \left(\boldsymbol{V}_f \boldsymbol{P}_q\right)^T \boldsymbol{W}_f \mathrm{diag}\left(\boldsymbol{n}_i\right) \boldsymbol{V}_f \boldsymbol{P}_q & \frac{1}{2} \left(\boldsymbol{V}_f \boldsymbol{P}_q\right)^T \boldsymbol{W}_f \mathrm{diag}\left(\boldsymbol{n}_i\right) \\ -\frac{1}{2} \boldsymbol{W}_f \mathrm{diag}\left(\boldsymbol{n}_i\right) \boldsymbol{V}_f \boldsymbol{P}_q & \frac{1}{2} \boldsymbol{W}_f \mathrm{diag}\left(\boldsymbol{n}_i\right) \end{bmatrix}.$$

A straightforward computation shows that $\boldsymbol{Q}_N^i$ satisfies an SBP property [11]. It is worth noting that the form of $\boldsymbol{Q}_N^i$ does not depend on the choice of basis. So long as the approximation space spanned by the basis ϕ_j does not change, the domain and range of $\boldsymbol{Q}_N^i$ depend solely on the choice of volume and surface quadrature points.

A collocation scheme assumes that the number of quadrature points is identical to the number of basis functions. If the solution is represented using degree N Lagrange polynomials at quadrature points, the matrices $\boldsymbol{V}_q, \boldsymbol{P}_q$ simplify

$$(\boldsymbol{V}_q)_{ij} = \delta_{ij}, \qquad \boldsymbol{M} = \boldsymbol{W}, \qquad \boldsymbol{P}_q = \boldsymbol{M}^{-1} \boldsymbol{V}_q^T \boldsymbol{W} = \boldsymbol{I}.$$

In summary, the collocation assumption removes the need to interpolate to and from a polynomial space to volume quadrature points within an entropy stable scheme.

REFERENCES

[1] Travis C Fisher and Mark H Carpenter. High-order entropy stable finite difference schemes for nonlinear conservation laws: Finite domains. *Journal of Computational Physics*, 252:518–557, 2013.

[2] Mark H Carpenter, Travis C Fisher, Eric J Nielsen, and Steven H Frankel. Entropy Stable Spectral Collocation Schemes for the Navier–Stokes Equations: Discontinuous Interfaces. *SIAM Journal on Scientific Computing*, 36(5):B835–B867, 2014.

[3] Gregor J Gassner, Andrew R Winters, and David A Kopriva. Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations. *Journal of Computational Physics*, 327:39–66, 2016.

[4] Gregor J Gassner. A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods. *SIAM Journal on Scientific Computing*, 35(3):A1233–A1253, 2013.

[5] Zhijian J Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, Hung T Huynh, et al. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[6] Constantine M Dafermos. *Hyperbolic conservation laws in continuum physics*. Springer, 2005.

[7] Gregor J Gassner, Andrew R Winters, Florian J Hindenlang, and David A Kopriva. The BR1 scheme is stable for the compressible Navier–Stokes equations. *Journal of Scientific Computing*, pages 1–47, 2017.

[8] Eitan Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Mathematics of Computation*, 49(179):91–103, 1987.

[9] Tianheng Chen and Chi-Wang Shu. Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws. *Journal of Computational Physics*, 345:427–461, 2017.

[10] Jared Crean, Jason E Hicken, David C Del Rey Fernández, David W Zingg, and Mark H Carpenter. Entropy-stable summation-by-parts discretization of the Euler equations on general curved elements. *Journal of Computational Physics*, 356:410–438, 2018.

[11] Jesse Chan. On discretely entropy conservative and entropy stable discontinuous Galerkin methods. *Journal of Computational Physics*, 362:346 – 374, 2018.

[12] Jesse Chan and Lucas C Wilcox. Discretely entropy stable weight-adjusted discontinuous Galerkin methods on curvilinear meshes. *arXiv preprint arXiv:1805.10934*, 2018.

[13] Jared Crean, Jason E Hicken, David C Del Rey Fernández, David W Zingg, and Mark H Carpenter. High-Order, Entropy-Stable Discretizations of the Euler Equations for Complex Geometries. In *23rd AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, 2017.

[14] Matteo Parsani, Mark H Carpenter, Travis C Fisher, and Eric J Nielsen. Entropy Stable Staggered Grid Discontinuous Spectral Collocation Methods of any Order for the Compressible Navier–Stokes Equations. *SIAM Journal on Scientific Computing*, 38(5):A3129–A3162, 2016.

[15] David A Kopriva and Gregor Gassner. On the quadrature and weak form choices in collocation type discontinuous Galerkin spectral element methods. *Journal of Scientific Computing*, 44(2):136–155, 2010.

[16] Florian Hindenlang, Gregor J Gassner, Christoph Altmann, Andrea Beck, Marc Staudenmaier, and Claus-Dieter Munz. Explicit discontinuous Galerkin methods for unsteady problems. *Computers & Fluids*, 61:86–93, 2012.

[17] Jesse Chan, Zheng Wang, Axel Modave, Jean-Francois Remacle, and T Warburton. GPU-accelerated discontinuous Galerkin methods on hybrid meshes. *Journal of Computational Physics*, 318:142–168, 2016.

[18] David C Del Rey Fernández, Jason E Hicken, and David W Zingg. Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. *Computers & Fluids*, 95:171–196, 2014.

[19] David C Del Rey Fernández, Jason E Hicken, and David W Zingg. Simultaneous approximation terms for multi-dimensional summation-by-parts operators. *Journal of Scientific Computing*, 75(1):83–110, 2018.

[20] David C Del Rey Fernández, Pieter D Boom, and David W Zingg. A generalized framework for nodal first derivative summation-by-parts operators. *Journal of Computational Physics*, 266:214–239, 2014.

[21] Sigrun Ortleb. Kinetic energy preserving DG schemes based on summation-by-parts operators on interior node distributions. *PAMM*, 16(1):857–858, 2016.

[22] Sigrun Ortleb. A Kinetic Energy Preserving DG Scheme Based on Gauss–Legendre Points. *Journal of Scientific Computing*, 71(3):1135–1168, 2017.

[23] Hendrik Ranocha. Generalised summation-by-parts operators and variable coefficients. *Journal of Computational Physics*, 362:20 – 48, 2018.
[24] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.
[25] Andrew R Winters, Dominik Derigs, Gregor J Gassner, and Stefanie Walch. A uniquely defined entropy stable matrix dissipation operator for high Mach number ideal MHD and compressible Euler simulations. *Journal of Computational Physics*, 332:274–289, 2017.
[26] Jan Nordström. Conservative finite difference formulations, variable coefficients, energy estimates and artificial dissipation. *Journal of Scientific Computing*, 29(3):375–404, 2006.
[27] David A Kopriva. Metric identities and the discontinuous spectral element method on curvilinear meshes. *Journal of Scientific Computing*, 26(3):301–327, 2006.
[28] Amiram Harten. On the symmetric form of systems of conservation laws with entropy. *Journal of computational physics*, 49(1):151–164, 1983.
[29] Thomas JR Hughes, LP Franca, and M Mallet. A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Computer Methods in Applied Mechanics and Engineering*, 54(2):223–234, 1986.
[30] Praveen Chandrashekar. Kinetic energy preserving and entropy stable finite volume schemes for compressible Euler and Navier-Stokes equations. *Communications in Computational Physics*, 14(5):1252–1286, 2013.
[31] Chi-Wang Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *Advanced numerical approximation of nonlinear hyperbolic equations*, pages 325–432. Springer, 1998.
[32] Alfio Quarteroni and Alberto Valli. *Numerical Approximation of Partial Differential Equations*. Springer, 1994.
[33] Mo Lenoir. Optimal isoparametric finite elements and error estimates for domains involving curved boundaries. *SIAM Journal on Numerical Analysis*, 23(3):562–580, 1986.
[34] Christine Bernardi. Optimal finite-element interpolation on curved domains. *SIAM Journal on Numerical Analysis*, 26(5):1212–1240, 1989.
[35] T Warburton. A low-storage curvilinear discontinuous Galerkin method for wave problems. *SIAM Journal on Scientific Computing*, 35(4):A1987–A2012, 2013.
[36] David M Williams and Antony Jameson. Nodal points and the nonlinear stability of high-order methods for unsteady flow problems on tetrahedral meshes. In *21st AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2013.
[37] Geoffrey I Taylor and Albert E Green. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 158(895):499–521, February 1937.
[38] James DeBonis. Solutions of the Taylor-Green vortex problem using high-resolution explicit finite difference methods. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, January 2013.