

Explicit Jacobian matrix formulas for entropy stable summation-by-parts schemes

Jesse Chan, Christina Taylor

Abstract

Entropy stable schemes replicate an entropy inequality at the semi-discrete level. These schemes rely on an algebraic summation-by-parts (SBP) structure and a technique referred to as flux differencing, which has been explored and optimized in the context of matrix-free time-explicit solvers. In this work, we provide simple and explicit formulas for Jacobian matrices for the semi-discrete systems of ODEs produced by entropy stable discretizations. These formulas are derived based on the structure of flux differencing and automatic differentiation (AD). Numerical results demonstrate the efficiency and utility of these Jacobian formulas, and apply them in the context of two-derivative explicit time-stepping schemes and implicit time-stepping.

1 Introduction

This paper is concerned with the numerical solution of systems of nonlinear conservation laws. Let Ω denote some domain with boundary $\partial\Omega$. Nonlinear conservation laws are expressed as a system of nonlinear partial differential equations (PDEs)

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{i=1}^d \frac{\partial \mathbf{f}_i(\mathbf{u})}{\partial x_i} = 0, \quad S(\mathbf{u}) \text{ convex}, \quad \mathbf{v}(\mathbf{u}) = \frac{\partial S}{\partial \mathbf{u}}, \quad (1)$$

where $\mathbf{u} \in \mathbb{R}^n$ are the conservative variables, \mathbf{f}_i are nonlinear fluxes, and $\mathbf{v}(\mathbf{u})$ are the *entropy variables* with respect to the entropy $S(\mathbf{u})$. By multiplying (1) by the entropy variables, vanishing viscosity solutions [1] of many fluid systems [2, 3] can be shown to satisfy the following entropy inequality

$$\int_{\Omega} \frac{\partial S(\mathbf{u})}{\partial t} d\mathbf{x} + \sum_{i=1}^d \int_{\partial\Omega} (\mathbf{v}^T \mathbf{f}_i(\mathbf{u}) - \psi_i(\mathbf{u})) n_i \leq 0, \quad (2)$$

where n_i denotes the i th component of the outward normal vector. The entropy inequality (2) is a statement of stability for nonlinear conservation laws [4, 5].

High order entropy stable schemes (see for example [6, 7, 3, 8, 9]) reproduce this entropy inequality at the semi-discrete level. The resulting methods display

significantly improved robustness while retaining high order accuracy [10, 11]. These schemes are based on entropy conservative finite volume fluxes [12], which are extended to high order discretizations through a procedure referred to as flux differencing. These methods have mainly been tested in the context of explicit time-stepping. However, recent works have applied entropy stable methods to both the space-time and implicit settings [13, 14].

Both space-time and implicit time discretizations require the solution of a system of nonlinear equations. This can be done using Newton’s method, which involves the Jacobian matrix of the nonlinear equations. While it is possible to compute the solution to the nonlinear system without explicitly computing the Jacobian matrix using Jacobian-free Newton-Krylov methods [15, 16], the Jacobian matrix is commonly used to construct preconditioners [17]. In this work, we present explicit formulas for Jacobian matrices of systems resulting from entropy stable formulations. We also show that computing the Jacobian matrix is not significantly more expensive than evaluating the residual of the nonlinear system. Finally, we apply the new Jacobian formulas to both explicit two-derivative and implicit time-stepping schemes.

2 Explicit Jacobian formulas

For clarity of presentation, we consider first a scalar nonlinear conservation law in one spatial dimension

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0. \quad (3)$$

We assume without loss of generality periodic boundary conditions, which will simplify the presentation of the main results. The generalization to systems of nonlinear conservation laws and higher dimensions is postponed until Section 4.

Let $f_S(x, y)$ denote a bivariate scalar flux function which is symmetric and consistent. Suppose \mathbf{u} is a vector of nodal values of the solution. Define the vector $\mathbf{f} = \mathbf{f}(\mathbf{u})$ approximating the flux derivative $\frac{\partial f(u)}{\partial x}$ as

$$\mathbf{f}(\mathbf{u}) = 2(\mathbf{Q} \circ \mathbf{F}) \mathbf{1}, \quad \mathbf{F}_{ij} = f_S(\mathbf{u}_i, \mathbf{u}_j), \quad (4)$$

where \mathbf{Q} is a discretization matrix to be specified later. The simplest entropy stable numerical schemes based on flux differencing discretize (3) via the system of ODEs

$$\mathbf{M} \frac{d\mathbf{u}}{dt} + \mathbf{f}(\mathbf{u}) = \mathbf{0}.$$

where \mathbf{M} is a diagonal mass (norm) matrix with positive entries. If $f_S(x, y)$ is entropy conservative and \mathbf{Q} is skew-symmetric, then the resulting scheme is also discretely entropy conservative. An entropy stable scheme can be constructed from an entropy conservative scheme by adding appropriate terms which dissipate entropy [3, 18, 14].

We are interested in computing the Jacobian matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$. Let $\text{diag}(\mathbf{x})$ denote the diagonal matrix with the vector \mathbf{x} on the diagonal, and let $\text{diag}(\mathbf{A})$ denote the vector diagonal of \mathbf{A} . Then, we have the following lemma:

Lemma 2.1. *Suppose that $\mathbf{Q} = \pm \mathbf{Q}^T$. Then, the Jacobian matrix of (4) can be expressed as either*

$$\begin{aligned}\frac{\partial \mathbf{f}}{\partial \mathbf{u}} &= 2(\mathbf{Q} \circ \mathbf{F}_y) \pm \text{diag}(\mathbf{1}^T (2\mathbf{Q} \circ \mathbf{F}_y)) \\ \frac{\partial \mathbf{f}}{\partial \mathbf{u}} &= 2(\mathbf{Q} \circ \mathbf{F}_x^T) \pm \text{diag}((2\mathbf{Q} \circ \mathbf{F}_x) \mathbf{1})\end{aligned}$$

where the matrices $\mathbf{F}_x, \mathbf{F}_y$ is

$$(\mathbf{F}_x)_{ij} = \left. \frac{\partial f_S}{\partial x} \right|_{\mathbf{u}_i, \mathbf{u}_j}, \quad (\mathbf{F}_y)_{ij} = \left. \frac{\partial f_S}{\partial y} \right|_{\mathbf{u}_i, \mathbf{u}_j}.$$

Proof. We will prove the first formula involving \mathbf{F}_y . The second formula follows via symmetry and similar steps. By the chain rule,

$$\begin{aligned}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)_{ij} &= \frac{\partial f_i}{\partial \mathbf{u}_j} = \sum_k 2\mathbf{Q}_{ik} \frac{\partial}{\partial \mathbf{u}_j} f_S(\mathbf{u}_i, \mathbf{u}_k) \\ &= \sum_k 2\mathbf{Q}_{ik} \left(\left. \frac{\partial f_S}{\partial x} \right|_{\mathbf{u}_i, \mathbf{u}_k} \frac{\partial \mathbf{u}_i}{\partial \mathbf{u}_j} + \left. \frac{\partial f_S}{\partial y} \right|_{\mathbf{u}_i, \mathbf{u}_k} \frac{\partial \mathbf{u}_k}{\partial \mathbf{u}_j} \right)\end{aligned}$$

If $i \neq j$, then $\frac{\partial \mathbf{u}_k}{\partial \mathbf{u}_j} = \delta_{jk}$ and

$$\frac{\partial f_i}{\partial \mathbf{u}_j} = 2\mathbf{Q}_{ij} \left. \frac{\partial f_S}{\partial y} \right|_{\mathbf{u}_i, \mathbf{u}_j}.$$

When $i = j$, $\frac{\partial \mathbf{u}_i}{\partial \mathbf{u}_j} = \frac{\partial \mathbf{u}_i}{\partial \mathbf{u}_i} = 1$, and

$$\frac{\partial f_i}{\partial \mathbf{u}_i} = \left(\sum_k 2\mathbf{Q}_{ik} \left. \frac{\partial f_S}{\partial x} \right|_{\mathbf{u}_i, \mathbf{u}_k} \right) + 2\mathbf{Q}_{ii} \left. \frac{\partial f_S}{\partial y} \right|_{\mathbf{u}_i, \mathbf{u}_i}.$$

The term $2\mathbf{Q}_{ii} \left. \frac{\partial f_S}{\partial y} \right|_{\mathbf{u}_i, \mathbf{u}_i}$ is the diagonal of the matrix $2(\mathbf{Q} \circ \mathbf{F}_y)$, and we can simplify the first summation term. By the symmetry of $f_S(x, y)$, we have that

$$\left. \frac{\partial f_S}{\partial y} \right|_{x, y} = \left. \frac{\partial f_S}{\partial x} \right|_{y, x}$$

Thus, by $\mathbf{Q} = \pm \mathbf{Q}^T$,

$$\begin{aligned}\sum_k 2\mathbf{Q}_{ik} \left. \frac{\partial f_S}{\partial x} \right|_{\mathbf{u}_i, \mathbf{u}_k} &= \sum_k 2\mathbf{Q}_{ik} \left. \frac{\partial f_S}{\partial y} \right|_{\mathbf{u}_k, \mathbf{u}_i} \\ &= ((2\mathbf{Q} \circ \mathbf{F}_y^T) \mathbf{1})_i = (\pm \mathbf{1}^T (2\mathbf{Q} \circ \mathbf{F}_y))_i.\end{aligned}$$

□

While we consider only symmetric and skew-symmetric matrices \mathbf{Q} in this work, one can use this lemma to compute the Jacobian $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ for arbitrary matrices \mathbf{Q} since any real matrix can be decomposed into symmetric and skew parts

$$\mathbf{Q} = \frac{1}{2}(\mathbf{Q} + \mathbf{Q}^T) + \frac{1}{2}(\mathbf{Q} - \mathbf{Q}^T).$$

Two applications of Lemma 2.1 then provide a formula for the Jacobian of (4).

2.1 Computing derivatives of bivariate flux functions

The aforementioned proofs require partial derivatives of flux functions $f_S(u_L, u_R)$ with respect to at least one argument. While this can be done by hand for simple fluxes, it rapidly becomes cumbersome for complex or piecewise-defined flux functions such as the logarithmic mean [19, 20]. This can be avoided by using Automatic Differentiation (AD) [21]. AD is distinct from both symbolic differentiation and finite difference approximations in that it does not return an explicit expression for derivatives, but is still able to evaluate derivatives accurately up to machine precision. Given a computational function, AD provides a black-box function which evaluates the derivative.

In this work, we utilize the Julia implementation of forward-mode automatic differentiation provided by `ForwardDiff.jl` [22]. The procedure is remarkably simple: given some flux function $\mathbf{f}(\mathbf{x}, \mathbf{y})$, `ForwardDiff.jl` returns the derivative with respect to either x or y as another function. For example, defining the function $\left. \frac{\partial f}{\partial y} \right|_{x,y}$ is a one-line operation:

```
dfdy(x,y) = ForwardDiff.derivative(y->f(x,y),y)
```

The computation of Jacobians for vector-valued flux functions can be performed similarly using `ForwardDiff.jacobian`. This simple API is made possible by the flexible Julia type system [23].

In principle, AD can be directly applied to $\mathbf{f}(\mathbf{u})$ to compute the Jacobian matrix. However, because AD scales with the number of inputs and outputs, the cost of applying AD directly to $\mathbf{f}(\mathbf{u})$ increases as the discretization resolution increases. In contrast, using the approach in this paper, AD is applied only to the flux function, which has a small number of inputs and outputs which are fixed independently of the discretization resolution. As a result, the cost of evaluating derivatives of the flux function is roughly the same cost as evaluating the flux function, and entries of the Jacobian matrix can be computed for roughly the same cost as a single evaluation of the nonlinear flux $\mathbf{f}(\mathbf{u})$. Moreover, explicit Jacobian formulas enable implementations to take advantage of sparsity in \mathbf{Q} when assembling the Jacobian matrix.

3 Examples of matrices which appear in entropy stable numerical schemes

In this section, we give some examples of matrices \mathbf{Q} which appear in entropy stable numerical discretizations. We assume periodicity, which ensures a skew-symmetric structure. The treatment of non-periodic

3.1 Finite volume methods

The spatial discretization for a finite volume scheme can be reformulated in terms of (4) [24]. Suppose that the 1D interval $[-1, 1]$ is decomposed into K non-overlapping elements of size h . An entropy conservative finite volume scheme is given as

$$\begin{aligned} \frac{du_1}{dt} + \frac{f_S(u_2, u_1) - f_S(u_1, u_K)}{h} &= 0 \\ \frac{du_i}{dt} + \frac{f_S(u_{i+1}, u_i) - f_S(u_i, u_{i-1})}{h} &= 0, \quad i = 2, \dots, K-1, \\ \frac{du_K}{dt} + \frac{f_S(u_1, u_K) - f_S(u_K, u_{K-1})}{h} &= 0, \end{aligned}$$

where u_i denotes the average value of the solution on each element and f_S is an entropy conservative flux. Let $\mathbf{M} = h\mathbf{I}$ and let \mathbf{Q} be the periodic second-order central difference matrix

$$\mathbf{Q} = \frac{1}{2} \begin{bmatrix} 0 & 1 & & \dots & -1 \\ -1 & 0 & 1 & & \\ & -1 & 0 & 1 & \\ & & & \ddots & \\ 1 & & \dots & -1 & 0 \end{bmatrix}.$$

Then, an entropy finite volume scheme is equivalent to

$$\frac{d\mathbf{u}}{dt} + 2(\mathbf{Q} \circ \mathbf{F}) \mathbf{1} = \mathbf{0}, \quad \mathbf{F}_{ij} = f_S(u_i, u_j)$$

where $\mathbf{u} = [u_1, \dots, u_K]^T$ is the vector of solution values.

3.2 Multi-block summation-by-parts finite differences

We next consider a multi-element summation-by-parts (SBP) finite element discretization [25, 26]. Suppose again that the domain $[-1, 1]$ is decomposed into K non-overlapping elements D^k of size h . Let $\mathbf{M}, \mathbf{Q} \in \mathbb{R}^{N_p \times N_p}$ denote diagonal mass (norm) and nodal differentiation matrices, such that $\mathbf{M}^{-1}\mathbf{Q}$ approximates the first derivative on a reference interval and is exact for polynomials up to

degree N . The operators \mathbf{M}, \mathbf{Q} satisfy an SBP property if

$$\mathbf{Q} + \mathbf{Q}^T = \mathbf{B}, \quad \mathbf{B} = \begin{bmatrix} -1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}. \quad (5)$$

We note that nodal discontinuous Galerkin spectral element (DG-SEM) discretizations [27] also fall into a SBP framework [28], and are thus also included in this framework.

These matrices can be used to construct entropy conservative high order discretizations. Let $J = h/2$ is the Jacobian of the mapping from the reference element $[-1, 1]$ to a physical interval of size h and let $\mathbf{F}_{ij}^k = f_S(u_{i,k}, u_{j,k})$ denote the matrix of flux interactions between different nodes on the element D^k . A local formulation on the element D^k is given by

$$J_k \mathbf{M} \frac{d\mathbf{u}_k}{dt} + 2(\mathbf{Q} \circ \mathbf{F}^k) \mathbf{1} + \mathbf{B}(\mathbf{f}^* - f(\mathbf{u}_k)) = \mathbf{0},$$

$$\mathbf{f}^* = \begin{bmatrix} f_S(u_{1,k}^+, u_{1,k}) \\ 0 \\ \vdots \\ 0 \\ f_S(u_{N_p,k}^+, u_{N_p,k}) \end{bmatrix}. \quad (6)$$

where $u_{1,k}^+, u_{N_p,k}^+$ denote the exterior values of $u_{1,k}, u_{N_p,k}$ on neighboring elements. Assuming that the elements are ordered from left to right in ascending order, for interior element indices $1 < k < K$, these are given by

$$u_{1,k}^+ = u_{N_p,k-1}, \quad u_{N_p,k}^+ = u_{1,k+1}.$$

In other words, the first node on D^k is connected to the last node on the previous element, and the last node on D^k is connected to the first node on the next element.

For periodic boundary conditions, this local formulation can be understood as inducing a global skew-symmetric matrix. To show this, we first use the SBP property to rewrite (7) in a skew-symmetric form [29]

$$J_k \mathbf{M} \frac{d\mathbf{u}_k}{dt} + ((\mathbf{Q} - \mathbf{Q}^T) \circ \mathbf{F}^k) \mathbf{1} + \mathbf{B} \mathbf{f}^* = \mathbf{0}. \quad (7)$$

We now define a global vector $\mathbf{u}_\Omega = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K]^T$. Let the global flux matrix be defined as

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{11} & \dots & \mathbf{F}_{1K} \\ \vdots & \ddots & \vdots \\ \mathbf{F}_{K1} & \dots & \mathbf{F}_{KK} \end{bmatrix}, \quad (\mathbf{F}_{k_1, k_2})_{ij} = f_S(\mathbf{u}_{k_1, i}, \mathbf{u}_{k_2, j}).$$

The blocks of the matrix \mathbf{F} capture flux interactions between solution values at different nodes and elements. The local formulations can now be concatenated

into a single skew-symmetric matrix

$$\mathbf{M}_\Omega \frac{d\mathbf{u}_\Omega}{dt} + 2(\mathbf{Q}_\Omega \circ \mathbf{F}) \mathbf{1} = \mathbf{0},$$

where \mathbf{M}_Ω is the block-diagonal matrix with blocks $J_k \mathbf{M}$, and

$$\mathbf{Q}_\Omega = \frac{1}{2} \begin{bmatrix} \mathbf{S} & \mathbf{B}_R & & -\mathbf{B}_L \\ -\mathbf{B}_L & \mathbf{S} & \mathbf{B}_R & \\ & -\mathbf{B}_L & \ddots & \mathbf{B}_R \\ \mathbf{B}_R & & -\mathbf{B}_L & \mathbf{S} \end{bmatrix}, \quad \mathbf{S} = (\mathbf{Q} - \mathbf{Q}^T), \quad (8)$$

where the matrices $\mathbf{B}_L, \mathbf{B}_R$ are zeros except for a single entry

$$\mathbf{B}_L = \begin{bmatrix} & & 1 \\ & \ddots & \\ 0 & & \end{bmatrix}, \quad \mathbf{B}_R = \mathbf{B}_L^T = \begin{bmatrix} & & 0 \\ & \ddots & \\ 1 & & \end{bmatrix}$$

The matrix \mathbf{Q}_Ω can be considered a high order generalization of the finite volume matrix (5). Similar “global SBP operator” approaches were used to construct simultaneous approximation (SBP-SAT) interface coupling terms in [8, 30, 31].

The generalization to higher dimensional domains and curved geometric mappings is straightforward, but notationally much more complicated. The construction of skew matrices \mathbf{Q}_Ω follows from approaches detailed in [8, 9, 32, 30, 29, 14]. We omit it for conciseness in this work, but note that the efficient assembly of DG matrices is discussed in detail elsewhere [17, 33, 34].

4 Systems of conservation laws

In this section, we extend the explicit Jacobian formulas from scalar nonlinear conservation laws to an $n \times n$ system of conservation laws. Let $\mathbf{f}_S(\mathbf{u}_L, \mathbf{u}_R) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ denote an entropy conservative flux function for a 1D system of conservation laws. We first formulate a system of ODEs by modifying the definition of arrays and matrices in (8).

Let \mathbf{u}_Ω denote a vector of vectors

$$\mathbf{u}_\Omega = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{bmatrix}, \quad \mathbf{u}_i = \begin{bmatrix} \mathbf{u}_{i,1} \\ \mathbf{u}_{i,2} \\ \vdots \\ \mathbf{u}_{i,K} \end{bmatrix}, \quad \mathbf{u}_{i,k} = \begin{bmatrix} \mathbf{u}_{i,k,1} \\ \mathbf{u}_{i,k,2} \\ \vdots \\ \mathbf{u}_{i,k,N_p} \end{bmatrix}$$

for $\ell = 1, \dots, n$, $k = 1, \dots, K$ and $j = 1, \dots, N_p$. Here, $\mathbf{u}_{\ell,k,j}$ denotes the j th degree of freedom for the ℓ th component of the solution on the k th element. Let (k_1, j_1) and (k_2, j_2) be multi-indices which correspond to row and columns indices of a matrix, respectively. We now define the flux matrix \mathbf{F} as the $n \times n$ block diagonal matrix containing flux interactions between different nodes

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & & \\ & \ddots & \\ & & \mathbf{F}_n \end{bmatrix}, \quad (\mathbf{F}_\ell)_{(k_1, j_1), (k_2, j_2)} = (\mathbf{f}_S(\mathbf{u}_{:,k_1,j_1}, \mathbf{u}_{:,k_2,j_2}))_\ell. \quad (9)$$

where $\mathbf{u}_{:,k,j}$ denotes the vector containing all solution components at the k th element and j th node, and each entry of the block \mathbf{F}_ℓ for $\ell = 1, \dots, n$ corresponds to the ℓ th component of the vector-valued flux evaluated at solution states $\mathbf{u}_{:,k_1,j_1}, \mathbf{u}_{:,k_2,j_2}$.

Let $\mathbf{M}_\Omega, \mathbf{Q}_\Omega$ denote the global mass and differentiation matrices in (8). Then, an entropy conservative scheme is given by

$$(\mathbf{I}_n \otimes \mathbf{M}_\Omega) \frac{d\mathbf{u}_\Omega}{dt} + 2((\mathbf{I}_n \otimes \mathbf{Q}_\Omega) \circ \mathbf{F}) \mathbf{1} = \mathbf{0}.$$

where $\mathbf{e}_n \in \mathbb{R}^n$ is the vector of all ones, and $\mathbf{e}_n \mathbf{e}_n^T$ is the $n \times n$ matrix of all ones.

We now construct explicit Jacobian matrix formulas for systems of nonlinear conservation laws. The right hand side function $\mathbf{f}(\mathbf{u})$ for systems can be rewritten as

$$\mathbf{f}(\mathbf{u}) = 2((\mathbf{I}_n \otimes \mathbf{Q}_\Omega) \circ \mathbf{F}) \mathbf{1} = 2 \begin{bmatrix} (\mathbf{Q}_\Omega \circ \mathbf{F}_1) \\ \vdots \\ (\mathbf{Q}_\Omega \circ \mathbf{F}_n) \end{bmatrix} \mathbf{1}$$

Then, the Jacobian matrix is

$$\frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \begin{bmatrix} \partial \mathbf{F}_{1,\mathbf{u}_1} & \cdots & \partial \mathbf{F}_{1,\mathbf{u}_n} \\ \vdots & \ddots & \vdots \\ \partial \mathbf{F}_{n,\mathbf{u}_1} & \cdots & \partial \mathbf{F}_{n,\mathbf{u}_n} \end{bmatrix} \quad (10)$$

where the Jacobian block $\partial \mathbf{F}_{i,\mathbf{u}_j}$ is evaluated as in Lemma 2.1

$$\partial \mathbf{F}_{i,\mathbf{u}_j} = 2(\mathbf{Q}_\Omega \circ \mathbf{F}_{i,\mathbf{u}_j}) \pm \text{diag}(\mathbf{1}^T (2\mathbf{Q}_\Omega \circ \mathbf{F}_{i,\mathbf{u}_j})).$$

for $\mathbf{Q}_\Omega = \pm \mathbf{Q}_\Omega^T$. Here, the flux matrix $\mathbf{F}_{i,\mathbf{u}_j}$ is evaluated via

$$(\mathbf{F}_{i,\mathbf{u}_j})_{j_1 k_1, j_2 k_2} = \left. \frac{\partial (\mathbf{f}_S)_i}{\partial \mathbf{u}_{R,j}} \right|_{\mathbf{u}_{:,k_1,j_1}, \mathbf{u}_{:,k_2,j_2}},$$

where $\frac{\partial (\mathbf{f}_S)_i}{\partial \mathbf{u}_{R,j}}$ denotes the derivative of the i th component of the flux $\mathbf{f}_S(\mathbf{u}_L, \mathbf{u}_R)$ with respect to the j th solution component of \mathbf{u}_R . Thus, each entry of the block $\mathbf{F}_{i,\mathbf{u}_j}$ corresponds to an entry of the Jacobian (with respect to \mathbf{u}_R) of $\mathbf{f}_S(\mathbf{u}_L, \mathbf{u}_R)$ and an entry of the global differentiation matrix \mathbf{Q}_Ω .

Remark. An alternative ordering of degrees of freedom [8] yields a Jacobian matrix with a more compact bandwidth. A smaller matrix bandwidth can also be achieved by precomputing bandwidth-minimizing row and column permutations using the reverse Cuthill-McKee algorithm [35].

5 Non-collocated schemes: hybridized SBP operators, entropy projection, over-integration

Most entropy stable schemes rely on “collocated” SBP operators (where the mass matrix \mathbf{M}_Ω is diagonal) built on nodal sets which include boundary nodes

[3, 8]. However, in certain cases, energy and entropy stable SBP schemes constructed using non-diagonal mass matrices [9, 24] and more general nodal sets [36, 37, 38, 30] achieve higher accuracy than SBP schemes built on nodal sets which include boundary nodes. We discuss how to extend explicit Jacobian formulas to such “modal” settings.

5.1 Entropy conservative schemes and hybridized SBP operators

We now assume that the solution is approximated by

$$u(\mathbf{x}, t) \approx \sum_{j=1}^{N_p} \widehat{\mathbf{u}}_{k,i}(t) \phi_i(\mathbf{x}),$$

where $\widehat{\mathbf{u}}_{k,i}$ denotes the coefficients of the solution on an element D^k . We assume two sets of quadrature points: volume quadrature points and weights $\{w_i, \mathbf{x}_{q,i}\}_{i=1}^{N_q}$ and surface quadrature points $\{w_{f,i}, \mathbf{x}_{f,i}\}_{i=1}^{N_f}$. We assume both quadrature rules are exact for certain classes of integrands as detailed in [29, 24].

Evaluating $\mathbf{u}(\mathbf{x}, t)$ at quadrature points is equivalent to multiplication by an interpolation matrix \mathbf{V}

$$\begin{aligned} \mathbf{V}_{ij} &= \phi_j(\mathbf{x}_i), & i &= 1, \dots, N_q, & j &= 1, \dots, N_p \\ (\mathbf{V}_f)_{ij} &= \phi_j(\mathbf{x}_{f,i}), & i &= 1, \dots, N_f, & j &= 1, \dots, N_p. \end{aligned}$$

We can similarly define mass and projection matrices \mathbf{M}, \mathbf{P}

$$\mathbf{M} = \mathbf{V}^T \mathbf{W} \mathbf{V}, \quad \mathbf{P} = \mathbf{M}^{-1} \mathbf{V}^T \mathbf{W},$$

where \mathbf{W} is a diagonal matrix whose entries are the quadrature weights w_i . We also define a face interpolation matrix

$$\mathbf{E} = \mathbf{V}_f \mathbf{P}$$

which evaluates the solution at face quadrature points. Finally, we define the matrix \mathbf{V}_h as the mapping between local coefficients $\widehat{\mathbf{u}}_k$ and combined volume and surface quadrature points

$$\mathbf{V}_h = \begin{bmatrix} \mathbf{V} \\ \mathbf{V}_f \end{bmatrix}.$$

These matrices are involved in the application of hybridized SBP operators (originally referred to as decoupled SBP operators) [9, 39]. We present the main ideas in a 1D setting, and refer the reader to [8, 9, 29] for details on multi-dimensional settings.

Given some modal weak differentiation matrix $\widehat{\mathbf{Q}}$ which acts on the basis coefficients $\widehat{\mathbf{u}}_k$, we define a nodal differentiation matrix $\mathbf{Q} = \mathbf{P}^T \widehat{\mathbf{Q}} \mathbf{P}$. Then, we can define a hybridized SBP operator as

$$\mathbf{Q}_h = \frac{1}{2} \begin{bmatrix} \mathbf{Q} - \mathbf{Q}^T & \mathbf{E}^T \mathbf{B} \\ \mathbf{B} \mathbf{E} & \mathbf{B} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -1 & \\ & 1 \end{bmatrix}.$$

The operator \mathbf{Q}_h can be used to approximate coefficients of the derivative in the basis $\phi_i(\mathbf{x})$. Let $f(u)$ denote some function of $u(\mathbf{x})$, and let $\widehat{\mathbf{u}}$ denote the basis coefficients of $u(\mathbf{x})$. Then,

$$\frac{\partial f(u)}{\partial x} \approx \sum_j \widehat{f}_j \phi_j(\mathbf{x}), \quad \widehat{\mathbf{f}} = \mathbf{M}^{-1} \mathbf{V}_h^T \mathbf{Q}_h f(\mathbf{V}_h \widehat{\mathbf{u}})$$

We now construct global matrices for the multi-element (periodic) case. We begin by concatenating the local coefficients $\widehat{\mathbf{u}}_{k,i}$ into a global coefficient vector $\widehat{\mathbf{u}}_\Omega$. We also introduce boundary matrices $\mathbf{B}, \mathbf{B}_L, \mathbf{B}_R$ which enforce coupling between different elements and are defined as

$$\mathbf{B}_L = \begin{bmatrix} & 1 \\ 0 & \end{bmatrix}, \quad \mathbf{B}_R = \begin{bmatrix} & 0 \\ 1 & \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -1 & \\ & 1 \end{bmatrix}.$$

In the multi-dimensional case, the entries of $\mathbf{B}_L, \mathbf{B}_R$ correspond instead to outward normals [8, 9].

We can also use \mathbf{Q}_h to construct a globally skew-symmetric differentiation matrix (see also [30]). Define the matrix $\mathbf{S} = \mathbf{Q} - \mathbf{Q}^T$, and define \mathbf{Q}_Ω as the global block matrix

$$\mathbf{Q}_\Omega = \frac{1}{2} \begin{bmatrix} \mathbf{S} & \mathbf{E}^T \mathbf{B} & & & \\ -\mathbf{B}\mathbf{E} & & \mathbf{B}_R & & \\ & & & & -\mathbf{B}_L \\ & -\mathbf{B}_L & \mathbf{S} & \mathbf{E}^T \mathbf{B} & \\ & & -\mathbf{B}_R & & \\ & & & \ddots & \ddots \\ & & & \ddots & \ddots \\ & & & & \ddots \\ \mathbf{B}_R & & & & \mathbf{S} & \mathbf{E}^T \mathbf{B} \\ & & & -\mathbf{B}_L & -\mathbf{B}\mathbf{E} \end{bmatrix},$$

We also abuse notation and redefine $\mathbf{V}, \mathbf{E}, \mathbf{P}, \mathbf{V}_h$ as *global* interpolation, projection, and extrapolation matrices

$$\begin{aligned} \mathbf{V} &\longrightarrow \mathbf{I}_K \otimes \mathbf{V}, & \mathbf{P} &\longrightarrow \mathbf{I}_K \otimes \mathbf{P} \\ \mathbf{E} &\longrightarrow \mathbf{I}_K \otimes \mathbf{E}, & \mathbf{V}_h &\longrightarrow \mathbf{I}_K \otimes \mathbf{V}_h \end{aligned}$$

Finally, we assume that the global solution $\mathbf{u}(\mathbf{x}, t) \in \mathbb{R}^n$ is vector-valued, and order the solution coefficients as in Section 4.

It was shown in [40, 9] that, when either the mass matrix is non-diagonal or the nodal set does not contain appropriate boundary points, it is necessary to perform an entropy projection (or extrapolation [39]) step to ensure discrete entropy stability. Let $\mathbf{v}(\mathbf{u})$ denote the entropy variables as a function of the conservative variables, and let $\mathbf{u}(\mathbf{v})$ denote the inverse mapping. We define the entropy projected variables $\widetilde{\mathbf{u}}_\Omega$ as

$$\widetilde{\mathbf{u}}_\Omega = \mathbf{u}(\mathbf{V}_h \mathbf{P} \mathbf{v}(\mathbf{V} \widehat{\mathbf{u}}_\Omega)). \quad (11)$$

Let \mathbf{F} again denote the block-diagonal flux matrix in (9). We evaluate each flux block \mathbf{F}_ℓ using the entropy projected variables

$$(\mathbf{F}_\ell)_{(k_1, j_1), (k_2, j_2)} = (\mathbf{f}_S(\tilde{\mathbf{u}}_{:, k_1, j_1}, \tilde{\mathbf{u}}_{:, k_2, j_2}))_\ell. \quad (12)$$

Then, an entropy conservative method is given by

$$(\mathbf{I}_n \otimes \mathbf{M}_\Omega) \frac{d\mathbf{u}_\Omega}{dt} + 2(\mathbf{I}_n \otimes \mathbf{V}_h)^T ((\mathbf{e}_n \mathbf{e}_n^T \otimes \mathbf{Q}_\Omega) \circ \mathbf{F}) \mathbf{1} = \mathbf{0}.$$

where \mathbf{I}_n is the $n \times n$ identity matrix.

5.2 Jacobian matrices for hybridized SBP schemes

We redefine the nonlinear term as

$$\mathbf{f}(\tilde{\mathbf{u}}) = 2(\mathbf{I}_n \otimes \mathbf{V}_h)^T ((\mathbf{e}_n \mathbf{e}_n^T \otimes \mathbf{Q}_\Omega) \circ \mathbf{F}) \mathbf{1}.$$

where the flux matrix \mathbf{F} is computed using the entropy projected conservative variables (11) via (12). Let $\frac{\partial \mathbf{u}}{\partial \mathbf{v}}$ and $\frac{\partial \mathbf{v}}{\partial \mathbf{u}}$ denote Jacobians of the conservative variables with respect to the entropy variables and vice versa. These have been explicitly derived for several equations (for example, the Jacobians for the compressible Navier-Stokes equations are given in [2]).

We can compute the Jacobian of $\mathbf{f}(\tilde{\mathbf{u}})$ via the chain rule. We assume a scalar equation $n = 1$ for simplicity, and motivate our approach by considering an entry $i \neq j$ of the Jacobian

$$\left(\frac{\partial \mathbf{f}}{\partial \tilde{\mathbf{u}}_\Omega} \right)_{ij} = 2\mathbf{V}_h^T \frac{\partial}{\partial (\tilde{\mathbf{u}}_\Omega)_j} ((\mathbf{Q}_\Omega \circ \mathbf{F}) \mathbf{1})_i.$$

We focus on the latter term $\frac{\partial}{\partial \tilde{\mathbf{u}}_\Omega} (\mathbf{Q}_\Omega \circ \mathbf{F}) \mathbf{1}$

$$\begin{aligned} \left(\frac{\partial}{\partial \tilde{\mathbf{u}}_\Omega} (\mathbf{Q}_\Omega \circ \mathbf{F}) \mathbf{1} \right)_{ij} &= \frac{\partial}{\partial \tilde{\mathbf{u}}_{\Omega, j}} \sum_k (\mathbf{Q}_\Omega)_{ik} \mathbf{f}_S(\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_k) \\ &= \sum_k (\mathbf{Q}_\Omega)_{ik} \left. \frac{\partial \mathbf{f}_S}{\partial y} \right|_{\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_k} \frac{\partial \tilde{\mathbf{u}}_i}{\partial \tilde{\mathbf{u}}_{\Omega, j}} \end{aligned}$$

We observe that the term $\frac{\partial \tilde{\mathbf{u}}_i}{\partial \tilde{\mathbf{u}}_{\Omega, j}}$ does not disappear as it did in the proof of Lemma 2.1. We thus treat the Jacobian matrix in two parts. First, we define the matrix $\frac{\partial \tilde{\mathbf{f}}}{\partial \tilde{\mathbf{u}}}$ as

$$\left(\frac{\partial \tilde{\mathbf{f}}}{\partial \tilde{\mathbf{u}}} \right)_{ij} = (\mathbf{Q}_\Omega)_{ij} \left. \frac{\partial \mathbf{f}_S}{\partial y} \right|_{\tilde{\mathbf{u}}_i, \tilde{\mathbf{u}}_j} = (\mathbf{Q}_\Omega \circ \mathbf{F}_y)_{ij}.$$

The construction of $\frac{\partial \tilde{\mathbf{f}}}{\partial \tilde{\mathbf{u}}}$ for systems ($n > 1$) is similar to the procedure described in Section 4. Now, let $\tilde{\mathbf{v}}$ denote the projected entropy variables at evaluated at volume quadrature points

$$\tilde{\mathbf{v}} = \mathbf{V}_h \mathbf{P} \mathbf{v}(\mathbf{V} \tilde{\mathbf{u}}_\Omega).$$

The vector $\frac{\partial \tilde{\mathbf{u}}}{\partial \tilde{\mathbf{u}}_\Omega}$ can be further expanded as

$$\frac{\partial \tilde{\mathbf{u}}}{\partial \tilde{\mathbf{u}}_\Omega} = \frac{\partial \mathbf{u}}{\partial \mathbf{v}} \Big|_{\tilde{\mathbf{v}}} \mathbf{V}_h \mathbf{P} \frac{\partial \mathbf{v}}{\partial \mathbf{u}} \Big|_{\mathbf{v}_{\tilde{\mathbf{u}}_\Omega}} \mathbf{V}.$$

where the Jacobian matrices for the maps between conservative and entropy variables are block diagonal matrices given by

$$\frac{\partial \mathbf{u}}{\partial \mathbf{v}} \Big|_{\tilde{\mathbf{u}}} = \begin{bmatrix} \frac{\partial \mathbf{u}}{\partial \mathbf{v}} \Big|_{\tilde{\mathbf{u}}_1} & & \\ & \ddots & \\ & & \frac{\partial \mathbf{u}}{\partial \mathbf{v}} \Big|_{\tilde{\mathbf{u}}_K} \end{bmatrix}, \quad \frac{\partial \mathbf{u}}{\partial \mathbf{v}} \Big|_{\tilde{\mathbf{u}}_k} = \begin{bmatrix} \frac{\partial \mathbf{u}}{\partial \mathbf{v}} \Big|_{\tilde{\mathbf{u}}_{1,k}} & & \\ & \ddots & \\ & & \frac{\partial \mathbf{u}}{\partial \mathbf{v}} \Big|_{\tilde{\mathbf{u}}_{N_p,k}} \end{bmatrix},$$

and the local block $\frac{\partial \mathbf{u}}{\partial \mathbf{v}} \Big|_{\tilde{\mathbf{u}}_{j,k}}$ is the Jacobian matrix $\frac{\partial \mathbf{u}}{\partial \mathbf{v}}$ evaluated at the j th nodal solution value $\tilde{\mathbf{u}}_{j,k}$ on the k th element.

Let N_p, N_q, N_f denote the number of total basis functions, quadrature points, and face quadrature points respectively, and define $N_{\text{total}} = N_q + N_f$. The structure and dimensions of matrices involved in constructing the Jacobian matrix are illustrated as follows:

$$\frac{\partial \mathbf{f}}{\partial \tilde{\mathbf{u}}_\Omega} = \begin{array}{c} \boxed{\mathbf{V}_h^T} \\ (N_p \times N_{\text{total}}) \end{array} \begin{array}{c} \boxed{\mathbf{Q}_\Omega \circ \frac{\partial \tilde{\mathbf{f}}}{\partial \tilde{\mathbf{u}}}} \\ (N_{\text{total}} \times N_{\text{total}}) \end{array} \begin{array}{c} \boxed{\frac{\partial \mathbf{u}}{\partial \mathbf{v}} \Big|_{\tilde{\mathbf{v}}}} \\ (N_{\text{total}} \times N_{\text{total}}) \end{array} \begin{array}{c} \boxed{\mathbf{V}_h \mathbf{P}} \\ (N_{\text{total}} \times N_q) \end{array} \begin{array}{c} \boxed{\frac{\partial \mathbf{v}}{\partial \mathbf{u}} \Big|_{\mathbf{v}_{\tilde{\mathbf{u}}_\Omega}}} \\ (N_q \times N_q) \end{array} \begin{array}{c} \boxed{\mathbf{V}} \\ (N_q \times N_p) \end{array}$$

We note that several matrices become simplified under common assumptions for entropy stable DG discretizations. These typically fall into two categories: collocated volume nodes and collocated volume and surface nodes [41]. For collocated volume nodes, the solution is represented using a nodal Lagrange basis constructed using $N_q = N_p$ volume quadrature nodes. Both collocated volume and surface nodes refers to the case when both the volume nodes collocated and the volume quadrature nodes also include surface quadrature nodes as a subset [28, 3]. For example, if only volume nodes are collocated [30], then $\mathbf{V} = \mathbf{I}$. If both volume and surface nodes are collocated, the system reduces to the simplified system described in Section 4 using the fact that $\frac{\partial \mathbf{u}}{\partial \mathbf{v}} = \left(\frac{\partial \mathbf{v}}{\partial \mathbf{u}} \right)^{-1}$.

6 Numerical experiments

In this section, we compare the computational efficiency of the explicit formulas derived in this paper with other methods for computing the Jacobian. Additionally, we apply explicit Jacobian formulas to enable two-derivative time-stepping methods [42] and time-implicit discretizations.

6.1 Comparisons of computational cost

- Compare cost of evaluating scalar flux function vs cost of evaluating AD derivative (Burgers' flux, rational kernel $1/(a+b)$, and logmean).

- Compare cost of computing Jacobian matrix via direct finite differences, `FiniteDiff.jl`, AD applied to $\mathbf{f}(\mathbf{u})$, and the explicit Jacobian formula.

These computations are performed using a single thread. Because the steps are trivially parallelizable with high arithmetic intensity, the use of multi-threading or GPU acceleration is not expected to significantly change the relative costs.

6.2 Two-derivative time-stepping methods

Consider a general system of ODEs

$$\frac{d\mathbf{u}}{dt} + \mathbf{f}(\mathbf{u}) = \mathbf{0}.$$

Two-derivative explicit time-stepping methods are constructed based on the assumption that second derivatives of \mathbf{u} in time are available [42, 43]. The resulting schemes can achieve higher order accuracy with fewer stages and function evaluations compared to standard Runge-Kutta methods.

Let $\mathbf{g}(\mathbf{u})$ denote the second derivative of \mathbf{u} in time

$$\mathbf{g}(\mathbf{u}) = \frac{d^2\mathbf{u}}{dt^2} = \frac{d}{dt}\mathbf{f}(\mathbf{u}) = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \frac{d\mathbf{u}}{dt},$$

where we have used the chain rule in the final step. The simplest two-derivative method is the second order scheme given in [42] as

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \Delta t \mathbf{f}(\mathbf{u}^k) - \frac{\Delta t^2}{2} \mathbf{g}(\mathbf{u}^k),$$

where \mathbf{u}^k denotes the solution at the k th time-step.

6.3 Time-implicit discretizations

Jacobian matrices also appear in time-implicit discretizations of nonlinear ODEs. Consider the implicit midpoint rule

$$\mathbf{u}^{k+1} = \mathbf{u}^k - \Delta t \mathbf{f}\left(\frac{\mathbf{u}^{k+1} + \mathbf{u}^k}{2}\right).$$

This can be rewritten in the following form where $\mathbf{u}^{k+1/2} = \frac{\mathbf{u}^{k+1} + \mathbf{u}^k}{2}$

$$\begin{aligned} \mathbf{u}^{k+1/2} &= \mathbf{u}^k - \frac{\Delta t}{2} \mathbf{f}(\mathbf{u}^{k+1/2}) \\ \mathbf{u}^{k+1} &= 2\mathbf{u}^{k+1/2} - \mathbf{u}^k. \end{aligned}$$

Solving for $\mathbf{u}^{k+1/2}$ is a nonlinear equation and can be done via Newton's method

$$\begin{aligned} \mathbf{u}^{k+1/2, \ell+1} &= \\ \mathbf{u}^{k+1/2, \ell} - \left(\mathbf{I} + \frac{\Delta t}{2} \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \Big|_{\mathbf{u}^{k+1/2, \ell}} \right)^{-1} &\left(\mathbf{u}^{k+1/2, \ell} + \frac{\Delta t}{2} \mathbf{f}(\mathbf{u}^{k+1/2, \ell}) - \mathbf{u}^k \right). \end{aligned}$$

7 Conclusion and acknowledgments

Jesse Chan gratefully acknowledges support from the National Science Foundation under awards DMS-1719818, DMS-1712639, and DMS-CAREER-1943186.

References

- [1] Stanislav N Kružkov. First order quasilinear equations in several independent variables. *Mathematics of the USSR-Sbornik*, 10(2):217, 1970.
- [2] 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.
- [3] 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.
- [4] Michael S Mock. Systems of conservation laws of mixed type. *Journal of Differential equations*, 37(1):70–88, 1980.
- [5] Amiram Harten. On the symmetric form of systems of conservation laws with entropy. *Journal of computational physics*, 49(1):151–164, 1983.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] Jesse Chan. On discretely entropy conservative and entropy stable discontinuous Galerkin methods. *Journal of Computational Physics*, 362:346 – 374, 2018.
- [10] Andrew R Winters, Rodrigo C Moura, Gianmarco Mengaldo, Gregor J Gassner, Stefanie Walch, Joaquim Peiro, and Spencer J Sherwin. A comparative study on polynomial dealiasing and split form discontinuous Galerkin

- schemes for under-resolved turbulence computations. *Journal of Computational Physics*, 372:1–21, 2018.
- [11] Diego Rojas, Radouan Boukharfane, Lisandro Dalcin, David C Fernandez, Hendrik Ranocha, David E Keyes, and Matteo Parsani. On the robustness and performance of entropy stable discontinuous collocation methods for the compressible Navier-Stokes equations. *arXiv preprint arXiv:1911.10966*, 2019.
 - [12] Eitan Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Mathematics of Computation*, 49(179):91–103, 1987.
 - [13] Lucas Friedrich, Gero Schnücke, Andrew R Winters, David C Del Rey Fernández, Gregor J Gassner, and Mark H Carpenter. Entropy stable space–time discontinuous Galerkin schemes with summation-by-parts property for hyperbolic conservation laws. *Journal of Scientific Computing*, 80(1):175–222, 2019.
 - [14] Jason E Hicken. Entropy-stable, high-order summation-by-parts discretizations without interface penalties. *Journal of Scientific Computing*, 82(2):50, 2020.
 - [15] Dana A Knoll and David E Keyes. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.
 - [16] Philipp Birken, Gregor J Gassner, and Lea M Versbach. Subcell finite volume multigrid preconditioning for high-order discontinuous Galerkin methods. *International Journal of Computational Fluid Dynamics*, pages 1–9, 2019.
 - [17] P-O Persson and Jaime Peraire. Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier–Stokes equations. *SIAM Journal on Scientific Computing*, 30(6):2709–2733, 2008.
 - [18] Johnathon Upperman and Nail K Yamaleev. Entropy stable artificial dissipation based on Brenner regularization of the Navier-Stokes equations. *Journal of Computational Physics*, 393:74–91, 2019.
 - [19] Farzad Ismail and Philip L Roe. Affordable, entropy-consistent Euler flux functions II: Entropy production at shocks. *Journal of Computational Physics*, 228(15):5410–5436, 2009.
 - [20] Andrew R Winters, Christof Czernik, Moritz B Schily, and Gregor J Gassner. Entropy stable numerical approximations for the isothermal and polytropic Euler equations. *BIT Numerical Mathematics*, pages 1–34, 2019.
 - [21] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*, volume 105. SIAM, 2008.

- [22] J. Revels, M. Lubin, and T. Papamarkou. Forward-Mode Automatic Differentiation in Julia. *arXiv:1607.07892 [cs.MS]*, 2016.
- [23] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.
- [24] Jesse Chan. Entropy stable reduced order modeling of nonlinear conservation laws. *arXiv preprint arXiv:1909.09103*, 2019.
- [25] H-O Kreiss and Godela Scherer. Finite element and finite difference methods for hyperbolic partial differential equations. In *Mathematical aspects of finite elements in partial differential equations*, pages 195–212. Elsevier, 1974.
- [26] Mark H Carpenter, Jan Nordström, and David Gottlieb. A stable and conservative interface treatment of arbitrary spatial accuracy. *Journal of Computational Physics*, 148(2):341–365, 1999.
- [27] David A Kopriva. *Implementing spectral methods for partial differential equations: Algorithms for scientists and engineers*. Springer Science & Business Media, 2009.
- [28] 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.
- [29] Jesse Chan. Skew-Symmetric Entropy Stable Modal Discontinuous Galerkin Formulations. *Journal of Scientific Computing*, 81(1):459–485, Oct 2019.
- [30] Jesse Chan, David C Del Rey Fernández, and Mark H Carpenter. Efficient entropy stable Gauss collocation methods. *SIAM Journal on Scientific Computing*, 41(5):A2938–A2966, 2019.
- [31] David C Fernandez, Mark H Carpenter, Lisandro Dalcin, Stefano Zampini, and Matteo Parsani. Entropy Stable h/p-Nonconforming Discretization with the Summation-by-Parts Property for the Compressible Euler and Navier-Stokes Equations. *arXiv preprint arXiv:1910.02110*, 2019.
- [32] Jesse Chan and Lucas C Wilcox. Discretely entropy stable weight-adjusted discontinuous Galerkin methods on curvilinear meshes. *Journal of Computational Physics*, 378:366 – 393, 2019.
- [33] Koen Hillewaert. *Development of the discontinuous Galerkin method for high-resolution, large scale CFD and acoustics in industrial geometries*. PhD thesis, Université catholique de Louvain, 2013.
- [34] Dominic Kempf, René Heß, Steffen Müthing, and Peter Bastian. Automatic code generation for high-performance discontinuous galerkin methods on modern architectures. *arXiv preprint arXiv:1812.08075*, 2018.

- [35] Elizabeth Cuthill and James McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, pages 157–172, 1969.
- [36] 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.
- [37] Hendrik Ranocha. Generalised summation-by-parts operators and variable coefficients. *Journal of Computational Physics*, 362:20 – 48, 2018.
- [38] 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.
- [39] Tianheng Chen and Chi-Wang Shu. Review of entropy stable discontinuous Galerkin methods for systems of conservation laws on unstructured simplex meshes, 2019. Accessed July 25, 2019.
- [40] 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.
- [41] Siavosh Shadpey and David W Zingg. Energy-and Entropy-Stable Multidimensional Summation-by-Parts Discretizations on Non-Conforming Grids. In *AIAA Aviation 2019 Forum*, page 3204, 2019.
- [42] Robert PK Chan and Angela YJ Tsai. On explicit two-derivative Runge-Kutta methods. *Numerical Algorithms*, 53(2-3):171–194, 2010.
- [43] Andrew J Christlieb, Sigal Gottlieb, Zachary Grant, and David C Seal. Explicit strong stability preserving multistage two-derivative time-stepping schemes. *Journal of Scientific Computing*, 68(3):914–942, 2016.