



Entropy Stable Discontinuous Galerkin Schemes on Moving Meshes for Hyperbolic Conservation Laws

Gero Schnücke¹ · Nico Krais² · Thomas Bolemann² · Gregor J. Gassner³

Received: 21 December 2018 / Revised: 13 February 2020 / Accepted: 20 February 2020 /
Published online: 3 March 2020
© The Author(s) 2020

Abstract

This work is focused on the entropy analysis of a semi-discrete nodal discontinuous Galerkin spectral element method (DGSEM) on moving meshes for hyperbolic conservation laws. The DGSEM is constructed with a local tensor-product Lagrange-polynomial basis computed from Legendre–Gauss–Lobatto points. Furthermore, the collocation of interpolation and quadrature nodes is used in the spatial discretization. This approach leads to discrete derivative approximations in space that are summation-by-parts (SBP) operators. On a static mesh, the SBP property and suitable two-point flux functions, which satisfy the entropy condition from Tadmor, allow to mimic results from the continuous entropy analysis, if it is ensured that properties such as positivity preservation (of the water height, density or pressure) are satisfied on the discrete level. In this paper, Tadmor’s condition is extended to the moving mesh framework. We show that the volume terms in the semi-discrete moving mesh DGSEM do not contribute to the discrete entropy evolution when a two-point flux function that satisfies the moving mesh entropy condition is applied in the split form DG framework. The discrete entropy behavior then depends solely on the interface contributions and on the domain boundary contribution. The interface contributions are directly controlled by proper choice of the numerical element interface fluxes. If an entropy conserving two-point flux is chosen, the interface contributions vanish. To increase the robustness of the discretization we use so-called entropy stable two-point fluxes at the interfaces that are guaranteed entropy dissipative and thus give a bound on the interface contributions in the discrete entropy balance. The remaining boundary contributions depend on the type of the considered boundary condition. E.g. for periodic boundary conditions that are of entropy conserving type, our methodology with the entropy conserving interface fluxes is fully entropy conservative and with the entropy stable interface fluxes is guaranteed entropy stable. The presented proof does not require any exactness of quadrature in the spatial integrals of the variational forms. As it is the case for static meshes, these results rely on the assumption that additional properties like positivity preservation are satisfied on the discrete level. Besides the entropy stability, the time discretization of the moving mesh DGSEM will be investigated and it will be proven that the moving mesh DGSEM satisfies the free stream preservation property for an arbitrary s -stage Runge–Kutta method, when periodic boundary conditions are used. The theoretical

✉ Gero Schnücke
gschnuec@math.uni-koeln.de

Extended author information available on the last page of the article

properties of the moving mesh DGSEM will be validated by numerical experiments for the compressible Euler equations with periodic boundary conditions.

Keywords Discontinuous Galerkin · Summation-by-parts · Moving meshes · Entropy stability · Free stream preservation

1 Introduction

A lot of applications in engineering and physics require the approximation of conservation laws on time-dependent domains, e.g. domains with moving boundaries. For instance, moving mesh discontinuous Galerkin (DG) methods have been investigated in [5,43,52,54]. In particular, moving mesh discontinuous Galerkin spectral element methods (DGSEM) have been constructed and analyzed in [37,48,64]. In the literature, there are also moving mesh methods with the capability to change the connectivity of the mesh, e.g. with finite volume (FV) methods [44,57] and with a DG method [60]. Moving mesh finite difference methods were constructed in [1,51], in [66] a moving mesh collocation method was constructed and in [27] a moving mesh continuous finite element method was constructed. In general, moving mesh methods are well suited to preserve motion related properties like the Galilean-invariance. These properties are necessary to describe physical processes like the formation of disc galaxies [45].

A common way to approximate conservation laws on time-dependent domains is to use the Arbitrary Lagrangian–Eulerian (ALE) approach [17]. In this approach the conservation law is transformed from the time-dependent domain onto a time-independent reference domain. The motion of the mesh on the physical domain is part of the transformation. Thus, the grid velocity field appears as a new quantity in the equation on the reference domain. On the one hand the ALE transformation simplifies the discretization, since a static mesh can be used in the reference domain. On the other hand, the new quantities in the equation on the reference domain complicate the discrete stability analysis, even in the linear case [37].

In this work, moving mesh DGSEM to solve non-linear, symmetrizable and hyperbolic systems of conservation laws are investigated. It is well known that symmetrizable systems are equipped with an entropy/entropy flux pair [26,49]. For scalar conservation laws, entropy admissibility criteria provide the unique physically relevant weak solution [15,39]. In general, entropy admissibility criteria are not enough to ensure well-posedness for systems of conservation laws [11]. Nevertheless, the entropy is an essential quantity to analyze systems of conservation laws. In particular, for gas dynamics a possible mathematical entropy is the scaled negative thermodynamic entropy which shows that the mathematical model correctly captures the second law of thermodynamics [2]. The entropy is conserved for smooth solutions of a conservation law and decays for discontinuous solutions [29,59].

It is reasonable to construct numerical schemes for conservation laws which reflect the properties of the entropy on the discrete level. Tadmor [58] developed a discrete entropy criterion to construct a specific class of two-point flux functions for low-order finite difference (FD) and FV methods. FD/FV methods with these class of two-point flux functions preserve entropy on the discrete level. Moreover, these FV methods can be modified by adding dissipation to the numerical fluxes such that the entropy is decreasing for all times. Therefore, two-point fluxes with Tadmor's discrete entropy condition are called entropy conservative fluxes. LeFloch et al. [40] gave a framework to construct high-order entropy conservative schemes in periodic domains. Fisher and Carpenter [20] combined this approach

with summation-by-parts (SBP) operators and proved that two-point entropy conservative fluxes can be used to construct high-order schemes when the derivative approximations in space are SBP operators. A SBP operator provides a discrete analogue of the integration-by-parts formula [19,22,38]. It is worth to mention that the derivative matrix in the DGSEM provides a SBP operator, if the tensor-product Lagrange-polynomial basis is computed from Legendre–Gauss–Lobatto (LGL) points and interpolation and quadrature are collocated. Gassner et al. [23,24] showed that split forms of the partial differential equations can be discretely recovered when specific choices of numerical volume fluxes in the flux form volume integral of Fisher and Carpenter are chosen. Thus, an entropy stable DGSEM can be constructed by the following building blocks:

- (1) The derivative matrix satisfies the SBP property.
- (2) There are two-point flux functions with Tadmor’s discrete entropy condition that can be extended to high-order in a split form DG framework.

This methodology has been used in the construction of high-order entropy stable DGSEM on quadrilateral/hexahedral elements, e.g. [4,23,61], or on triangular/tetrahedral elements, e.g. [6,10,13]. All these methods are provably entropy stable and the semi-discrete entropy analysis for them is based merely on the properties of the SBP operators and the assumptions that the time integration is exact. Additionally, properties like positivity preservation (of the water height, density or pressure) must be satisfied on the discrete level. The exactness of quadrature in the spatial integrals of the variational form is not necessary. Available entropy stable moving mesh methods are for instance the low order continuous finite element method by Guermond et al. [27] and a spectral collocation based approach published during the extended review process of the current paper by Yamaleev et al. [66].

The remainder of the paper is organized as follows: The ALE transformation and continuous entropy analysis is presented in the Sects. 2.1, 2.2 and 2.3. The framework for the spectral element discretization with the SBP operator is given in the Sect. 2.4 and the DG split form framework is presented in the Sect. 2.5. The moving mesh DGSEM is finally presented in the Sect. 2.5. A discrete entropy analysis for the moving mesh DGSEM is given in the Sects. 2.6 and 2.7. Furthermore, in Sect. 2.8 it is proven that the moving mesh DGSEM satisfies the free stream preservation property. In Sect. 4, numerical examples with the compressible Euler equations are presented to validate our theoretical findings.

2 Entropy Stable DGSEM on Moving Meshes

The main goal of this work is the construction of an entropy stable moving mesh DGSEM. On static meshes, it is possible to construct high-order entropy stable DGSEM, if the derivative matrix is an SBP operator and entropy conservative two-point flux functions are available. This methodology has been used in the construction of high-order entropy stable DGSEM on quadrilateral/hexahedral elements, e.g. [4,23,61]. In this section, it will be shown that similar ideas can be used to construct high-order entropy stable moving mesh DGSEM. The construction of the entropy stable moving mesh DGSEM will be presented for an arbitrary symmetrizable and hyperbolic system of conservation laws

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{i=1}^3 \frac{\partial \mathbf{f}_i}{\partial x_i} = \mathbf{0}, \quad (2.1)$$

on a time-dependent domain $\Omega(t) \subseteq \mathbb{R}^3$. The vector of conservative variables is \mathbf{u} and \mathbf{f}_i , $i = 1, 2, 3$, are the physical flux vectors. The state vectors are of size p depending on the number of equations in the system under consideration and the conservation law is subjected to appropriate initial and boundary conditions (see the comment below Eq. (2.44) in Sect. 2.3).

The block vector nomenclature in [24] simplifies the analysis of the system (2.1) on curved elements. Thus, we translate the conservation law (2.1) in block vector notation. A block vector is highlighted by the double arrow

$$\overleftrightarrow{\mathbf{f}} := \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \end{bmatrix}. \tag{2.2}$$

The dot product of two block vectors is given by

$$\overleftrightarrow{\mathbf{f}} \cdot \overleftrightarrow{\mathbf{g}} := \sum_{i=1}^3 \mathbf{f}_i^T \mathbf{g}_i. \tag{2.3}$$

Furthermore, the dot product of a vector \vec{v} in the three dimensional space and a block vector is defined by

$$\vec{v} \cdot \overleftrightarrow{\mathbf{f}} := \sum_{i=1}^3 v_i \mathbf{f}_i. \tag{2.4}$$

We note that the dot product (2.3) is a scalar quantity and the dot product (2.4) is a vector in a p dimensional space, where the number p corresponds to the number of conservative variables in the conservation law (2.1). The interaction between a vector \vec{v} and the conservative variables is defined as the block vector

$$\vec{v} \mathbf{u} := \begin{bmatrix} v_1 \mathbf{u} \\ v_2 \mathbf{u} \\ v_3 \mathbf{u} \end{bmatrix}. \tag{2.5}$$

Thus, in particular, the spatial gradient of the conservative variables is defined by

$$\vec{\nabla}_x \mathbf{u} := \begin{bmatrix} \frac{\partial \mathbf{u}}{\partial x_1} \\ \frac{\partial \mathbf{u}}{\partial x_2} \\ \frac{\partial \mathbf{u}}{\partial x_3} \end{bmatrix}. \tag{2.6}$$

The gradient of a vector valued function $\vec{g} = [g_1, g_2, g_3]^T$ is a second order tensor, written in matrix form as

$$\left[\vec{\nabla}_x \otimes \vec{g} \right]^T = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \frac{\partial g_1}{\partial x_3} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \frac{\partial g_2}{\partial x_3} \\ \frac{\partial g_3}{\partial x_1} & \frac{\partial g_3}{\partial x_2} & \frac{\partial g_3}{\partial x_3} \end{bmatrix}, \tag{2.7}$$

where \otimes is the outer product of two vectors in a three dimensional space. The dot product (2.3) and the spatial gradient (2.6) are used to define the divergence of a block vector flux as

$$\vec{\nabla}_x \cdot \overleftrightarrow{\mathbf{f}} := \sum_{i=1}^3 \frac{\partial \mathbf{f}_i}{\partial x_i}. \tag{2.8}$$

Moreover, for a vector valued function \vec{g} and the conservative variables, we have the product rule

$$\vec{\nabla}_x \cdot (\vec{g} \mathbf{u}) = (\vec{\nabla}_x \cdot \vec{g}) \mathbf{u} + \vec{g} \cdot (\vec{\nabla}_x \mathbf{u}) \tag{2.9}$$

with respect to the dot products (2.3) and (2.4). These notations allow to write the conservation law (2.1) in the compact form

$$\frac{\partial \mathbf{u}}{\partial t} + \vec{\nabla}_x \cdot \vec{\mathbf{f}} = \mathbf{0}. \tag{2.10}$$

2.1 Building Blocks of the ALE Transformation for Hexahedral Curved Meshes

In order to set up the moving mesh DGSEM in the Sect. 2.5, we make for all $t \in [0, T]$ the assumptions:

- (A1) For a fixed number $K \in \mathbb{N}$ the physical domain $\Omega(t)$ can be subdivided into K time-dependent, non-overlapping and conforming hexahedral elements, $e_\kappa(t)$, $\kappa = 1, \dots, K$. These elements can have curved faces.
- (A2) The time-dependent elements $e_\kappa(t)$ are mapped into the spatial computational domain $E = [-1, 1]^3$ with a bijective isoparametric transfinite mapping

$$e_\kappa(t) \ni \vec{x}(t) = \vec{\chi}(\vec{\xi}, \tau), \quad \vec{\xi} \in E, \quad \tau \in [0, T]. \tag{2.11}$$

Winters constructed in his PHD thesis [65] a mapping for this set up. Like in [65], it is assumed that the curved faces satisfy for all $\tau \in [0, T]$

$$\begin{aligned} \vec{\Gamma}_1(-1, \xi^3, \tau) &= \vec{\Gamma}_6(-1, \xi^3, \tau), \quad \vec{\Gamma}_2(-1, \xi^3, \tau) = \vec{\Gamma}_6(1, \xi^3, \tau), \\ \vec{\Gamma}_3(-1, \xi^2, \tau) &= \vec{\Gamma}_6(\xi^2, -1, \tau), \\ \vec{\Gamma}_1(1, \xi^3, \tau) &= \vec{\Gamma}_4(-1, \xi^3, \tau), \quad \vec{\Gamma}_2(1, \xi^3, \tau) = \vec{\Gamma}_4(1, \xi^3, \tau), \\ \vec{\Gamma}_3(1, \xi^2, \tau) &= \vec{\Gamma}_4(\xi^2, -1, \tau), \\ \vec{\Gamma}_1(\xi^1, -1, \tau) &= \vec{\Gamma}_3(\xi^1, -1, \tau), \quad \vec{\Gamma}_2(\xi^1, -1, \tau) = \vec{\Gamma}_3(\xi^1, 1, \tau), \\ \vec{\Gamma}_5(-1, \xi^2, \tau) &= \vec{\Gamma}_6(\xi^2, 1, \tau), \\ \vec{\Gamma}_1(\xi^1, 1, \tau) &= \vec{\Gamma}_5(\xi^1, -1, \tau), \quad \vec{\Gamma}_2(\xi^1, 1, \tau) = \vec{\Gamma}_5(\xi^1, 1, \tau), \\ \vec{\Gamma}_5(1, \xi^2, \tau) &= \vec{\Gamma}_4(\xi^2, 1, \tau). \end{aligned} \tag{2.12}$$

The location of the curved faces is sketched in Fig. 1. The curved faces of an element $e_\kappa(t)$ are approximated as interpolation polynomials up to degree N such that

$$\mathbb{I}^N(\vec{\Gamma}_i)(\eta, \zeta, \tau) := \sum_{j,k=0}^N \vec{\Gamma}_i(\eta_j, \zeta_k, \tau) \ell_j(\eta) \ell_k(\zeta), \quad i = 1, 2, 3, 4, 5, 6, \tag{2.13}$$

where $\{\ell_j\}_{j=0}^N, \{\ell_k\}_{k=0}^N$ are the Lagrange polynomials associated with the interpolation points $\{\eta_j\}_{j=0}^N$ and $\{\zeta_k\}_{k=0}^N$.

- (A3) The determinant J of the Jacobian matrix $\left[\vec{\nabla}_{\vec{\xi}} \otimes \vec{\chi} \right]^T$ satisfies

$$J := \det \left(\vec{\nabla}_{\vec{\xi}} \otimes \vec{\chi} \right) > 0, \quad \forall \tau \in [0, T]. \tag{2.14}$$

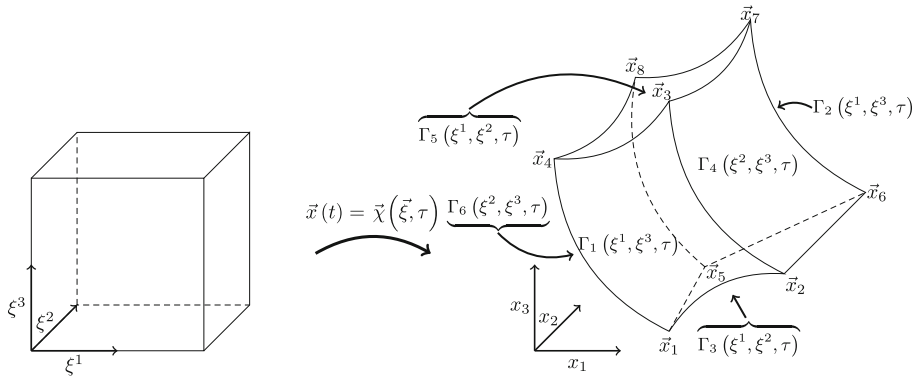


Fig. 1 Left the reference element $E = [-1, 1]^3$ and on the right a general hexahedral element $e_\kappa(t)$ with the curved faces $\bar{\Gamma}_1(\xi^1, \xi^3, \tau)$, $\bar{\Gamma}_2(\xi^1, \xi^3, \tau)$, $\bar{\Gamma}_3(\xi^1, \xi^2, \tau)$, $\bar{\Gamma}_4(\xi^2, \xi^3, \tau)$, $\bar{\Gamma}_5(\xi^1, \xi^2, \tau)$, and $\bar{\Gamma}_6(\xi^2, \xi^3, \tau)$. The mapping $\vec{x}(t) = \vec{\chi}(\vec{\xi}, \tau)$ connects E and $e_\kappa(t)$

Mesh curving techniques are discussed by Hindenlang et al. [30] and methodologies to construct a moving mesh with the properties (A1)–(A3) are given in the literature e.g. the book of Huang and Russell [32, Chapter 6, Chapter 7]. In many situations the moving mesh methodology depends on the underlying problem, e.g. [45].

The mapping provides the grid velocity field

$$\vec{v} = [v_1, v_2, v_3]^T := \left[\frac{\partial \chi_1}{\partial \tau}, \frac{\partial \chi_2}{\partial \tau}, \frac{\partial \chi_3}{\partial \tau} \right]^T = \frac{\partial \vec{\chi}}{\partial \tau}. \tag{2.15}$$

It is desirable that the grid velocity is continuous, since the mesh should be conforming and watertight at each time level. The next statement provides conditions on the element boundaries to guarantee that the grid velocity becomes continuous.

Lemma 2.1 *Let $e_1(t)$ and $e_2(t)$ be two neighboring elements which share one of the faces*

$$\bar{\Gamma}_1^1 = \bar{\Gamma}_2^2, \quad \bar{\Gamma}_3^1 = \bar{\Gamma}_5^2, \quad \bar{\Gamma}_4^1 = \bar{\Gamma}_6^2, \quad \bar{\Gamma}_1^2 = \bar{\Gamma}_2^1, \quad \bar{\Gamma}_3^2 = \bar{\Gamma}_5^1, \quad \bar{\Gamma}_4^2 = \bar{\Gamma}_6^1, \tag{2.16}$$

where $\bar{\Gamma}_i^l$, $l = 1, 2$, and $i = 1, 2, 3, 4, 5, 6$, are the faces of the element $e_l(t)$. Furthermore, suppose that the faces $\bar{\Gamma}_i^l(\cdot, \cdot, \tau)$ are continuously differentiable in the time interval $[0, T]$. Then the grid velocity field is continuous in the points which belong to the face that the elements share.

In Appendix A the Lemma 2.1 is proven in two dimensions. The three dimensional proof can be done by the same argumentation.

2.2 Transformation of the Conservation Law onto a Reference Element

In the following, we show that the system (2.10) can be transformed from a time-dependent element $e_\kappa(t)$ on the reference element E . The mapping (A.1) provides the covariant basis vectors

$$\vec{a}_i := \frac{\partial \vec{\chi}}{\partial \xi^i}, \quad i = 1, 2, 3, \tag{2.17}$$

and the volume weighted contravariant vectors

$$J\vec{a}^i = \vec{a}_j \times \vec{a}_k, \quad (i, j, k) \text{ cyclic.} \tag{2.18}$$

The quantity $\vec{\xi} = (\xi^1, \xi^2, \xi^3)^T$ is a vector in the reference element $E = [-1, 1]^3$. The covariant and the volume weighted contravariant vectors represent the Jacobian matrix $[\vec{\nabla}_{\vec{\xi}} \otimes \vec{\chi}]^T$ and its adjoint matrix

$$[\vec{\nabla}_{\vec{\xi}} \otimes \vec{\chi}]^T = [\vec{a}_1 \ \vec{a}_2 \ \vec{a}_3], \quad \text{adj}([\vec{\nabla}_{\vec{\xi}} \otimes \vec{\chi}]^T) = \begin{bmatrix} (J\vec{a}^1)^T \\ (J\vec{a}^2)^T \\ (J\vec{a}^3)^T \end{bmatrix}. \tag{2.19}$$

Furthermore, the contravariant vectors satisfy the metric identities

$$\sum_{i=1}^3 \frac{\partial J\vec{a}^i}{\partial \xi^i} = 0. \tag{2.20}$$

In particular, the covariant and the contravariant vectors allow to transform differential operators on the time-independent reference element E . On the reference element the gradient of a function f is given by

$$\vec{\nabla}_x f = \frac{1}{J} \left(\sum_{i=1}^3 J\vec{a}^i \frac{\partial f}{\partial \xi^i} \right) = \frac{1}{J} [\text{adj}(\vec{\nabla}_{\vec{\xi}} \otimes \vec{\chi})] \vec{\nabla}_{\vec{\xi}} f \tag{2.21}$$

and the divergence of a vector valued function \vec{g} is given by

$$\vec{\nabla}_x \cdot \vec{g} = \frac{1}{J} \sum_{i=1}^3 \frac{\partial}{\partial \xi^i} (J\vec{a}^i \cdot \vec{g}) = \frac{1}{J} \vec{\nabla}_{\vec{\xi}} \cdot \vec{g}, \tag{2.22}$$

where we used the contravariant flux

$$\vec{g} := \begin{bmatrix} J\vec{a}^1 \cdot \vec{g} \\ J\vec{a}^2 \cdot \vec{g} \\ J\vec{a}^3 \cdot \vec{g} \end{bmatrix} = [\text{adj}(\vec{\nabla}_{\vec{\xi}} \otimes \vec{\chi})]^T \vec{g}. \tag{2.23}$$

In [24], the following block matrix has been introduced to combine the transformations (2.21) and (2.22) with the block vector notation

$$\mathcal{M} = \begin{bmatrix} Ja_1^1 \mathcal{I}_p & Ja_1^2 \mathcal{I}_p & Ja_1^3 \mathcal{I}_p \\ Ja_2^1 \mathcal{I}_p & Ja_2^2 \mathcal{I}_p & Ja_2^3 \mathcal{I}_p \\ Ja_3^1 \mathcal{I}_p & Ja_3^2 \mathcal{I}_p & Ja_3^3 \mathcal{I}_p \end{bmatrix}, \tag{2.24}$$

where the matrix \mathcal{I}_p is the $p \times p$ identity matrix and Ja_j^i is the component of $J\vec{a}^i$ in the j -th Cartesian coordinate direction. The transformation of the gradient becomes

$$\vec{\nabla}_x \mathbf{u} = \frac{1}{J} \mathcal{M} \vec{\nabla}_{\vec{\xi}} \mathbf{u}. \tag{2.25}$$

We note that for a vector valued function \vec{g} the following identity holds

$$\vec{g} \cdot \vec{\nabla}_x \mathbf{u} = \frac{1}{J} \vec{g} \cdot \mathcal{M} \vec{\nabla}_{\vec{\xi}} \mathbf{u} = \frac{1}{J} \vec{g} \cdot \vec{\nabla}_{\vec{\xi}} \mathbf{u}. \tag{2.26}$$

Moreover, by applying the metric identities (2.20), the transformation of the divergence can be written as

$$\vec{\nabla}_x \cdot \vec{\mathbf{f}} = \frac{1}{J} \vec{\nabla}_\xi \cdot \mathcal{M}^T \vec{\mathbf{f}}. \tag{2.27}$$

Hence, the contravariant block vector flux is given by

$$\vec{\mathbf{f}} := \begin{bmatrix} J\vec{a}^1 \cdot \vec{\mathbf{f}} \\ J\vec{a}^2 \cdot \vec{\mathbf{f}} \\ J\vec{a}^3 \cdot \vec{\mathbf{f}} \end{bmatrix} = \mathcal{M}^T \vec{\mathbf{f}}. \tag{2.28}$$

Since the elements $\{e_k(t)\}_{k=1}^K$ are time-dependent, the time evolution of the quantity J needs to be analyzed. Thus, we apply Jacobi’s formula (cf. e.g. Bellman [3]) and obtain by (2.15), (2.19)

$$\frac{\partial J}{\partial \tau} = \text{tr} \left[\left[\text{adj} \left(\vec{\nabla}_\xi \otimes \vec{\chi} \right) \right]^T \frac{\partial}{\partial \tau} \left[\vec{\nabla} \otimes \vec{\chi} \right]^T \right] = \sum_{i=1}^3 J\vec{a}^i \cdot \left(\frac{\partial \vec{a}^i}{\partial \tau} \right) = \sum_{i=1}^3 J\vec{a}^i \cdot \left(\frac{\partial \vec{v}}{\partial \xi^i} \right), \tag{2.29}$$

where $\text{tr}[\cdot]$ denotes the trace of a matrix. The metric identities (2.20) allow to write the Eq.(2.29) in conservation form

$$\frac{\partial J}{\partial \tau} = \sum_{i=1}^3 \frac{\partial}{\partial \xi^i} \left(J\vec{a}^i \cdot \vec{v} \right) = \vec{\nabla}_\xi \cdot \vec{v}. \tag{2.30}$$

The chain rule formula and the identity (2.26) provide

$$\frac{\partial \mathbf{u}}{\partial \tau} = \frac{\partial \mathbf{u}}{\partial t} + \frac{1}{J} \vec{v} \cdot \vec{\nabla}_\xi \mathbf{u}. \tag{2.31}$$

Next, we plug (2.30) into Eq.(2.31), apply the product rule (2.9) and rearrange. This provides the equation

$$J \frac{\partial \mathbf{u}}{\partial t} = \frac{\partial (J\mathbf{u})}{\partial \tau} - \vec{\nabla}_\xi \cdot (\vec{v}\mathbf{u}). \tag{2.32}$$

Finally, we combine the identities (2.27) and (2.32) to write the the conservation law (2.10) in the following form

$$\frac{\partial (J\mathbf{u})}{\partial \tau} + \vec{\nabla}_\xi \cdot \vec{\mathbf{g}} = \mathbf{0}, \tag{2.33}$$

where

$$\vec{\mathbf{g}} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{g}_3 \end{bmatrix} := \begin{bmatrix} \mathbf{f}_1 - \nu_1 \mathbf{u} \\ \mathbf{f}_2 - \nu_2 \mathbf{u} \\ \mathbf{f}_3 - \nu_3 \mathbf{u} \end{bmatrix} = \vec{\mathbf{f}} - \vec{v}\mathbf{u}. \tag{2.34}$$

The formulation (2.33) is the representation of the system (2.10) on the time-independent reference element E for a time-dependent element $e_\kappa(t)$.

Remark 2.2 The metric identities (2.20) and the Eq.(2.30) provide the geometric conservation law (GCL) [18,28,41,42,46]. A numerical method to solve (2.10) on moving and deforming grids needs to satisfy both equations, otherwise the conservation properties of the conservation law (2.10) are not preserved. Farhat et al. [18,28,41] proved that the absence of

these equations has a critical effect on the accuracy and stability of a moving mesh method. In particular, the preservation of constant states is no longer guaranteed, if the GCL is not satisfied on the discrete level.

2.3 Entropy Analysis in Three Dimensions

The system (2.1) is assumed to be symmetrizable. Thus, in particular, it is equipped with entropy/entropy flux pairs (s, f_i^s) , $i = 1, 2, 3$, (cf. Godunov [26] and Mock [49]). The strictly convex function s is the entropy function. The entropy function s provides the entropy variables

$$\mathbf{w} := \frac{\partial s}{\partial \mathbf{u}}, \tag{2.35}$$

and it follows by the chain rule

$$\frac{\partial s}{\partial t} = \mathbf{w}^T \frac{\partial \mathbf{u}}{\partial t}, \quad \frac{\partial s}{\partial x_i} = \mathbf{w}^T \frac{\partial \mathbf{u}}{\partial x_i}, \quad i = 1, 2, 3. \tag{2.36}$$

The entropy flux functions and the flux functions in the conservation law are related and satisfy

$$\mathbf{w}^T \frac{\partial \mathbf{f}_i}{\partial x_i} = \frac{\partial f_i^s}{\partial x_i}, \quad i = 1, 2, 3. \tag{2.37}$$

The identities (2.26) and (2.36) give

$$\mathbf{w}^T (\vec{v} \cdot \vec{\nabla}_\xi \mathbf{u}) = J \mathbf{w}^T (\vec{v} \cdot \vec{\nabla}_x \mathbf{u}) = J (\vec{v} \cdot \vec{\nabla}_x s) = \vec{v} \cdot \vec{\nabla}_\xi s. \tag{2.38}$$

Hence, we obtain with the identity (2.31) and the chain rule

$$J \mathbf{w}^T \frac{\partial \mathbf{u}}{\partial \tau} = J \frac{\partial s}{\partial t} + \vec{v} \cdot \vec{\nabla}_\xi s = J \frac{\partial s}{\partial \tau}. \tag{2.39}$$

Therefore, the product rule provides the identity

$$\begin{aligned} \mathbf{w}^T \frac{\partial (J \mathbf{u})}{\partial \tau} &= J \frac{\partial s}{\partial \tau} + \left(\frac{\partial J}{\partial \tau} \right) \mathbf{w}^T \mathbf{u} \\ &= \frac{\partial (J s)}{\partial \tau} + \left(\frac{\partial J}{\partial \tau} \right) (\mathbf{w}^T \mathbf{u} - s) \\ &= \frac{\partial (J s)}{\partial \tau} + (\vec{\nabla}_\xi \cdot \vec{v}) (\mathbf{w}^T \mathbf{u} - s), \end{aligned} \tag{2.40}$$

where we used the GCL (2.30) in the last step. Next, we apply the relation (2.37) for the entropy flux functions and obtain

$$\mathbf{w}^T \frac{\partial \mathbf{g}_i}{\partial x_i} = \frac{\partial}{\partial x} (f_i^s - v_i s) - \left(\frac{\partial v_i}{\partial x_i} \right) (\mathbf{w}^T \mathbf{u} - s), \quad i = 1, 2, 3. \tag{2.41}$$

Next, we apply the vector notation $\vec{f}^s := [f_1^s, f_2^s, f_3^s]^T$. Then (2.41) and the transformation formulas for the gradient and divergence in the Sect. 2.2 give

$$\mathbf{w}^T \left(\vec{\nabla}_\xi \cdot \vec{\mathbf{g}} \right) = \vec{\nabla}_\xi \cdot (\vec{f}^s - \vec{v} s) - (\vec{\nabla}_\xi \cdot \vec{v}) (\mathbf{w}^T \mathbf{u} - s). \tag{2.42}$$

Finally, the identities (2.40) and (2.42) provide the balance law

$$0 = \mathbf{w}^T \left(\frac{\partial (J\mathbf{u})}{\partial \tau} + \vec{\nabla}_\xi \cdot \vec{\mathbf{g}} \right) = \frac{\partial (Js)}{\partial \tau} + \vec{\nabla}_\xi \cdot (\vec{f}^s - \vec{v}_s). \tag{2.43}$$

We integrate the Eq.(2.43) over the domain $E \times [0, T]$ and obtain

$$\frac{\partial}{\partial \tau} \int \int_E Js d\vec{\xi} d\tau = - \int_0^T \int_{\partial E} (\vec{f}^s - \vec{v}_s)^T \hat{n} dS d\tau. \tag{2.44}$$

Boundary conditions then need to be specified so that the bound on the entropy depends only on the boundary data. We will assume here that boundary data is given in a way that the right hand side in Eq.(2.44) is non-positive so that the entropy will not increase in time.

For discontinuous solutions the Eq.(2.44) is not satisfied, but under further assumptions it is possible to proof that a weak solution of (2.33) satisfies the inequality

$$\frac{\partial (Js)}{\partial \tau} + \vec{\nabla}_\xi \cdot (\vec{f}^s - \vec{v}_s) \leq 0 \tag{2.45}$$

in the sense of distributions on $E \times (0, T)$ (see Godlewski and Raviart [25, Chapter 1, Theorem 3.3]). The inequality (2.45) means that it holds the inequality

$$\int_0^T \int_E Js \frac{\partial \phi}{\partial \tau} d\vec{\xi} d\tau \geq - \int_0^T \int_E (\vec{f}^s - \vec{v}_s)^T \vec{\nabla}_\xi \phi d\vec{\xi} d\tau, \quad \forall \phi \in C_0^\infty(E \times (0, T)), \quad \phi \geq 0. \tag{2.46}$$

2.4 Building Blocks for the Spectral Element Approximation

A nodal approach is used for the spectral element approximation. The Lagrange basis functions are given by

$$\ell_j(\xi) := \prod_{i=0, i \neq j}^N \frac{\xi - \xi_i}{\xi_j - \xi_i}, \quad j = 0, \dots, N, \tag{2.47}$$

where the nodal points $\{\xi_i\}_{i=0}^N$ are the LGL points. We note that $\xi_0 = -1$ and $\xi_N = 1$. The Lagrange basis functions satisfy the cardinal property

$$\ell_i(\xi_j) = \delta_{ji}, \tag{2.48}$$

where δ_{ji} is the Kronecker delta. On the reference element $E = [-1, 1]^3$ the solution and fluxes of the system (2.33) are approximated by tensor product Lagrange polynomials of degree N , e.g.

$$\mathbf{u}(\xi^1, \xi^2, \xi^3, t) \approx \mathbf{U}(\xi^1, \xi^2, \xi^3, t) := \sum_{i,j,k=0}^N \mathbf{U}_{ijk}(t) \ell_i(\xi^1) \ell_j(\xi^2) \ell_k(\xi^3). \tag{2.49}$$

In the following, polynomial approximations are highlighted by capital letters, e.g. \mathbf{U} is an approximation for the state vector \mathbf{u} and $\mathbf{F}_l, l = 1, 2, 3$, are approximations for the fluxes $\mathbf{f}_l, l = 1, 2, 3$. The determinant J of the Jacobian matrix $\vec{\nabla}_{\vec{\xi}} \vec{\chi}$ is also approximated by tensor product Lagrange polynomials

$$J(\xi^1, \xi^2, \xi^3, t) \approx \mathbf{J}(\xi^1, \xi^2, \xi^3, t) := \sum_{i,j,k=0}^N \mathbf{J}_{ijk}(t) \ell_i(\xi^1) \ell_j(\xi^2) \ell_k(\xi^3). \tag{2.50}$$

In particular, the interpolation operator for a function \mathbf{g} is given by

$$\mathbb{I}^N(\mathbf{g})(\xi^1, \xi^2, \xi^3) = \sum_{i,j,k=0}^N \mathbf{g}_{ijk} \ell_i(\xi^1) \ell_j(\xi^2) \ell_k(\xi^3), \tag{2.51}$$

where $\mathbf{g}_{ijk} := \mathbf{g}(\xi_i^1, \xi_j^2, \xi_k^3)$ and $\{\xi_i^1\}_{i=0}^N, \{\xi_i^2\}_{i=0}^N, \{\xi_i^3\}_{i=0}^N$ are sets of LGL points. Derivatives are approximated by exact differentiation of the polynomial interpolants. In general we have $(\mathbb{I}^N(g))' \neq \mathbb{I}^{N-1}(g')$ (cf. e.g. [7,36]), as differentiation and interpolation only commute if there are no discretization errors. However, the contravariant coordinate vectors need to be discretized in such a way that the metric identities (2.20) are satisfied on the discrete level, too. Kopriva [35] introduced the conservative curl form that computes

$$\mathbb{J}a_\beta^\alpha := -\hat{x}_\alpha \cdot \vec{\nabla}_\xi \times \left(\mathbb{I}^N(\chi_\gamma \vec{\nabla}_\xi \chi_\delta) \right), \quad \alpha = 1, 2, 3, \quad \beta = 1, 2, 3, \quad (\beta, \gamma, \delta) \text{ cyclic}, \tag{2.52}$$

to approximate the metric terms. Here $\vec{\chi} = [\chi_1, \chi_2, \chi_3]^T$ represents the mapping from the element to the reference element and \hat{x}_i is the unit vector in the i -th Cartesian coordinate direction. The representation (2.52) ensures that

$$\sum_{\alpha=1}^3 \frac{\partial \mathbb{I}^N(\mathbb{J}a_\beta^\alpha)}{\partial \xi^\alpha} = 0, \quad \beta = 1, 2, 3. \tag{2.53}$$

From now on, the discrete contravariant coordinate vectors are denoted by $\mathbb{J}a_\beta^\alpha$, when the curl form (2.52) has been used to compute these quantities.

Integrals are approximated by a tensor product extension of a $2N - 1$ accurate LGL quadrature formula. Hence, interpolation and quadrature nodes are collocated. In one spatial dimension the LGL quadrature formula is given by

$$\int_{-1}^1 g(\xi) d\xi \approx \sum_{i=0}^N \omega_i g(\xi_i) = \sum_{i=0}^N \omega_i g_i, \tag{2.54}$$

where $\omega_i, i = 0, \dots, N$, are the quadrature weights and $\xi_i, i = 0, \dots, N$, are the LGL quadrature points. The formula (2.54) motivates the definition of the discrete quantity

$$\langle \mathbf{f}, \mathbf{g} \rangle_N := \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^N \omega_i \omega_j \omega_k \mathbf{f}_{ijk}^T \mathbf{g}_{ijk} = \sum_{i,j,k=0}^N \omega_{ijk} \mathbf{f}_{ijk}^T \mathbf{g}_{ijk} \tag{2.55}$$

for two functions \mathbf{f} and \mathbf{g} . We note that (2.55) satisfies

$$\langle \mathbb{I}^N(\mathbf{g}), \boldsymbol{\varphi} \rangle_N = \langle \mathbf{g}, \boldsymbol{\varphi} \rangle_N, \quad \forall \boldsymbol{\varphi} \in \mathbb{P}^N(E, \mathbb{R}^p). \tag{2.56}$$

Furthermore, for a block vector $\vec{\mathbf{f}}$ and test functions $\boldsymbol{\varphi} \in \mathbb{P}^N(E, \mathbb{R}^p)$, we define the discrete surface integral

$$\begin{aligned}
 \int_{\partial E, N} \boldsymbol{\varphi}^T \left\{ \overleftrightarrow{\mathbf{f}} \cdot \hat{\mathbf{n}} \right\} dS &:= \sum_{j,k=0}^N \omega_j \omega_k \left(\boldsymbol{\varphi}_{Njk}^T(\mathbf{F}_1)_{Njk} - \boldsymbol{\varphi}_{0jk}^T(\mathbf{F}_1)_{0jk} \right) \\
 &+ \sum_{i,k=0}^N \omega_i \omega_k \left(\boldsymbol{\varphi}_{iNk}^T(\mathbf{F}_2)_{iNk} - \boldsymbol{\varphi}_{i0k}^T(\mathbf{F}_2)_{i0k} \right) \\
 &+ \sum_{i,j=0}^N \omega_i \omega_j \left(\boldsymbol{\varphi}_{ijN}^T(\mathbf{F}_3)_{ijN} - \boldsymbol{\varphi}_{ij0}^T(\mathbf{F}_3)_{ij0} \right),
 \end{aligned} \tag{2.57}$$

where $\hat{\mathbf{n}}$ is the unit outward normal at the faces of the reference element E .

The spectral element approximation with LGL points for interpolation and quadrature provides a SBP operator $\mathcal{Q} = \mathcal{M}\mathcal{D}$ with the mass matrix \mathcal{M} and the derivative matrix \mathcal{D} . The mass matrix and the derivative matrix are given by

$$\mathcal{M}_{ij} = \omega_i \delta_{ij}, \quad \mathcal{D}_{ij} = \ell'_j(\xi_i) \quad i, j = 0, \dots, N. \tag{2.58}$$

The important characteristic of this SBP operator is the property

$$\mathcal{Q} + \mathcal{Q}^T = \mathcal{B}, \tag{2.59}$$

where $\mathcal{B} = \text{diag}(-1, 0, \dots, 0, 1)$. A SBP operator provides a discrete analogue of the integration-by-parts formula [19,22,38].

Finally, we note that in the LGL points $\xi_i^1, \xi_j^2, \xi_k^3, i, j, k = 0, \dots, N$, the Eq. (2.53) gives

$$\sum_{m=0}^N \left(\mathcal{D}_{im} \left(\mathbb{J}a_\beta^1 \right)_{mjk} + \mathcal{D}_{jm} \left(\mathbb{J}a_\beta^2 \right)_{imk} + \mathcal{D}_{km} \left(\mathbb{J}a_\beta^3 \right)_{ijm} \right) = 0, \quad \beta = 1, 2, 3. \tag{2.60}$$

2.5 The Semi-discrete Discontinuous Galerkin Method

Now, we apply the notation introduced in Sect. 2.4 and construct a moving mesh DGSEM. We discretize the Eqs. (2.30) and (2.33) simultaneously. In this way, it is ensured that the Eq. (2.30) is satisfied on the discrete level [37,48,64]. First, we replace the solution \mathbf{u} by (2.49), the Jacobian J by (2.50) and approximate the fluxes by the interpolation operator (2.51). Next, we multiply the GCL (2.30) by test functions $\varphi \in \mathbb{P}^N(E)$, the Eq. (2.33) with $\boldsymbol{\varphi} \in \mathbb{P}^N(E, \mathbb{R}^p)$, integrate the resulting equations and use integration-by-parts to separate boundary and volume contributions. The volume integrals in the variational form are approximated with the LGL quadrature. Then, we insert numerical surface fluxes \vec{v}^* and $\overleftrightarrow{\mathbf{G}}^*$ at the spatial element interfaces. Afterwards, we use the SBP property (2.59) for the volume contribution to get the standard DGSEM in strong form:

$$\left\langle \frac{\partial \mathbb{J}}{\partial \tau}, \boldsymbol{\varphi} \right\rangle_N = \left\langle \vec{\nabla}_\xi \cdot \mathbb{I}^N(\vec{v}), \boldsymbol{\varphi} \right\rangle_N + \int_{\partial E, N} \varphi (\tilde{v}_n^* - \tilde{v}_n) dS, \quad \forall \varphi \in \mathbb{P}^N(E), \tag{2.61a}$$

$$\left\langle \frac{\partial (\mathbb{J}\mathbf{U})}{\partial \tau}, \boldsymbol{\varphi} \right\rangle_N = - \left\langle \vec{\nabla}_\xi \cdot \mathbb{I}^N(\overleftrightarrow{\mathbf{g}}), \boldsymbol{\varphi} \right\rangle_N - \int_{\partial E, N} \boldsymbol{\varphi}^T (\tilde{\mathbf{G}}_n^* - \tilde{\mathbf{G}}_n) dS, \quad \forall \boldsymbol{\varphi} \in \mathbb{P}^N(E, \mathbb{R}^p), \tag{2.61b}$$

where we used the notation (2.55) and the notation (2.57) for the discrete surface integral.

The approximation of \vec{v} and the nonlinear flux \vec{g} by the interpolation operator (2.51) causes aliasing errors in the standard strong form. The aliasing errors cannot be bounded and the errors are independent of the choice of the numerical surface flux. In Gassner [22] a detailed explanation and analysis of the aliasing problem is given. Furthermore, a specific reformulation of the volume integrals by using the skew-symmetry strategy has been developed to fix the aliasing problem. This approach has been enhanced and generalized by Gassner et al. in [23,24] with a technique developed for high-order FD schemes (LeFloch et al. [40] or Fisher and Carpenter [20]). The generalized approach is called split form DG framework. Here, we proceed similar as in [24] and replace the interpolation operators in the discrete volume integrals by derivative projection operators. The interpolation operator in the discrete equation for the GCL (2.30) is replaced by

$$\begin{aligned} \mathbb{D}^N \cdot \vec{v}_{ijk} &:= \sum_{m=0}^N 2 \mathcal{D}_{im} \{ \{ \vec{v} \} \}_{(i,m)jk} \cdot \{ \{ \mathbb{J} \vec{a}^1 \} \}_{(i,m)jk} \\ &+ 2 \mathcal{D}_{jm} \{ \{ \vec{v} \} \}_{i(j,m)k} \cdot \{ \{ \mathbb{J} \vec{a}^2 \} \}_{i(j,m)k} \\ &+ 2 \mathcal{D}_{km} \{ \{ \vec{v} \} \}_{ij(k,m)} \cdot \{ \{ \mathbb{J} \vec{a}^3 \} \}_{ij(k,m)} \end{aligned} \tag{2.62}$$

with the volume averages of the metric terms, e.g.

$$\{ \{ \cdot \} \}_{(i,m)jk} := \frac{1}{2} [(\cdot)_{ijk} + (\cdot)_{mjk}]. \tag{2.63}$$

The derivative projection operator in the discrete equation for (2.33) is computed as in [24]. Thus, the operator is given by

$$\begin{aligned} \mathbb{D}^N \cdot \vec{G}_{ijk}^{\text{EC}} &:= \sum_{m=0}^N 2 \mathcal{D}_{im} \left(\vec{G}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk}) \cdot \{ \{ \mathbb{J} \vec{a}^1 \} \}_{(i,m)jk} \right) \\ &+ 2 \mathcal{D}_{jm} \left(\vec{G}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{imk}, \mathbf{U}_{ijk}, \mathbf{U}_{imk}) \cdot \{ \{ \mathbb{J} \vec{a}^2 \} \}_{i(j,m)k} \right) \\ &+ 2 \mathcal{D}_{km} \left(\vec{G}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{ijm}, \mathbf{U}_{ijk}, \mathbf{U}_{ijm}) \cdot \{ \{ \mathbb{J} \vec{a}^3 \} \}_{ij(k,m)} \right). \end{aligned} \tag{2.64}$$

We note that the discrete volume weighted contravariant vectors $\mathbb{J} \vec{a}^l$, $l = 1, 2, 3$, in the derivative projection operator (2.62) and (2.64) are computed by the conservative curl form (2.52). The flux \vec{G}^{EC} is consistent and symmetric such that, e.g.

$$\vec{G}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}, \mathbf{U}) = \vec{F} (\mathbf{U}) - \{ \{ \vec{v} \} \}_{(i,m)jk} \mathbf{U}, \tag{2.65}$$

and

$$\vec{G}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk}) = \vec{G}^{\text{EC}} (\vec{v}_{mjk}, \vec{v}_{ijk}, \mathbf{U}_{mjk}, \mathbf{U}_{ijk}), \tag{2.66}$$

for $i, j, k, m = 0, \dots, N$. Furthermore, the flux functions \mathbf{G}_l^{EC} , $l = 1, 2, 3$, satisfy for $i, j, k, m = 0, \dots, N$, the following discrete entropy conditions

$$\begin{aligned} \llbracket \mathbf{W} \rrbracket_{(i,m)jk}^T \mathbf{G}_l^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk}) &= \llbracket \Psi_l \rrbracket_{(i,m)jk} - \{ \{ v_l \} \}_{(i,m)jk} \llbracket \Phi \rrbracket_{(i,m)jk}, \\ \llbracket \mathbf{W} \rrbracket_{i(j,m)k}^T \mathbf{G}_l^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{imk}, \mathbf{U}_{ijk}, \mathbf{U}_{imk}) &= \llbracket \Psi_l \rrbracket_{i(j,m)k} - \{ \{ v_l \} \}_{i(j,m)k} \llbracket \Phi \rrbracket_{i(j,m)k}, \\ \llbracket \mathbf{W} \rrbracket_{ij(k,m)}^T \mathbf{G}_l^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{ijm}, \mathbf{U}_{ijk}, \mathbf{U}_{ijm}) &= \llbracket \Psi_l \rrbracket_{ij(k,m)} - \{ \{ v_l \} \}_{ij(k,m)} \llbracket \Phi \rrbracket_{ij(k,m)}. \end{aligned} \tag{2.67}$$

The quantities Φ and Ψ_l are polynomial approximations which satisfy in the LGL points

$$\Phi_{ijk} = \mathbf{W}_{ijk}^T \mathbf{U}_{ijk} - S_{ijk}, \quad (\Psi_l)_{ijk} := \mathbf{W}_{ijk}^T (\mathbf{F}_l)_{ijk} - (F_l^s)_{ijk}, \quad l = 1, 2, 3, \quad (2.68)$$

where \mathbf{W}_{ijk} , S_{ijk} and $(F_l^s)_{ijk}$ are the nodal values of the polynomials

$$\mathbf{W} := \mathbb{I}^N(\mathbf{w}), \quad S := \mathbb{I}^N(s), \quad F_l^s := \mathbb{I}^N(f_l^s), \quad l = 1, 2, 3. \quad (2.69)$$

Here, s represents an entropy for the system (2.1) with the corresponding entropy flux functions f_l^s , $l = 1, 2, 3$, and entropy variables \mathbf{w} . Furthermore, the volume jumps in (2.67) are, e.g.

$$\llbracket \cdot \rrbracket_{(i,m)jk} := (\cdot)_{ijk} - (\cdot)_{mjk}. \quad (2.70)$$

In Appendix B, flux functions with these properties are presented for the Euler equations.

Finally, for each element $e_\kappa(t)$ the semi-discrete moving mesh DGSEM can be written in the following form:

$$\left\langle \frac{\partial \mathbf{J}}{\partial \tau}, \boldsymbol{\varphi} \right\rangle_N = \left\langle \mathbb{D}^N \cdot \vec{v}, \boldsymbol{\varphi} \right\rangle_N + \int_{\partial E, N} \varphi (\tilde{v}_n^* - \tilde{v}_n) dS, \quad \forall \varphi \in \mathbb{P}^N(E), \quad (2.71a)$$

$$\left\langle \frac{\partial (\mathbf{J}\mathbf{U})}{\partial \tau}, \boldsymbol{\varphi} \right\rangle_N = - \left\langle \mathbb{D}^N \cdot \overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}}, \boldsymbol{\varphi} \right\rangle_N - \int_{\partial E, N} \boldsymbol{\varphi}^T (\tilde{\mathbf{G}}_n^* - \tilde{\mathbf{G}}_n) dS, \quad \forall \boldsymbol{\varphi} \in \mathbb{P}^N(E, \mathbb{R}^p). \quad (2.71b)$$

The unit outward facing normal vector and surface element on the element side are constructed from the element metrics by

$$\vec{n} := \frac{1}{\hat{s}} \sum_{l=1}^3 (\mathbb{J}\vec{a}^l) \hat{n}^l, \quad \hat{s} := \left| \sum_{l=1}^3 (\mathbb{J}\vec{a}^l) \hat{n}^l \right|. \quad (2.72)$$

Thus, the quantity \tilde{v}_n in (2.71a) and the flux $\tilde{\mathbf{G}}_n$ in (2.71b) are defined by

$$\tilde{v}_n = (\hat{s}\vec{n}) \cdot \vec{v} = \sum_{l=1}^3 \hat{n}^l (\mathbb{J}a_1^l v_1 + \mathbb{J}a_2^l v_2 + \mathbb{J}a_3^l v_3), \quad (2.73)$$

$$\tilde{\mathbf{G}}_n = (\hat{s}\vec{n}) \cdot \overset{\leftrightarrow}{\mathbf{G}} = \sum_{l=1}^3 \hat{n}^l (\mathbb{J}a_1^l \mathbf{G}_1 + \mathbb{J}a_2^l \mathbf{G}_2 + \mathbb{J}a_3^l \mathbf{G}_3) = \left\{ \mathcal{M} \overset{\leftrightarrow}{\mathbf{G}} \right\} \cdot \hat{n}. \quad (2.74)$$

To define the numerical surface fluxes in (2.71a) and (2.71b), we introduce notation for states at the LGL nodes along an interface between two spatial elements to be a primary “−” and complement the notation with a secondary “+” to denote the value at the LGL nodes on the opposite side. Then the orientated jump and the arithmetic mean at the interfaces are defined by

$$\llbracket \cdot \rrbracket := (\cdot)^+ - (\cdot)^-, \quad \{\{ \cdot \} \} := \frac{1}{2} [(\cdot)^+ + (\cdot)^-]. \quad (2.75)$$

When applied to vectors, the average and jump operators are evaluated separately for each vector component. Then the normal vector \vec{n} is defined unique to point from the “−” to the “+” side. This notation allows to compute the contravariant surface numerical fluxes in (2.71a) as

$$\tilde{v}_n^* = \hat{s} (n_1 \{\{ v_1 \} \} + n_2 \{\{ v_2 \} \} + n_3 \{\{ v_3 \} \}). \quad (2.76)$$

We note that due to the assumptions made in Sect. 2.1, the mesh velocity is a continuous function and the averages reduce to the uniquely defined values on the surface. The contravariant surface numerical fluxes in (2.71b) are given by

$$\tilde{\mathbf{G}}_n^* = \hat{s} (n_1 \mathbf{G}_1^{\text{EC}} + n_2 \mathbf{G}_2^{\text{EC}} + n_3 \mathbf{G}_3^{\text{EC}}), \tag{2.77}$$

where the Cartesian fluxes $\mathbf{G}_l^{\text{EC}}, l = 1, 2, 3$, satisfy (2.65), (2.66), (2.67). We note that these fluxes are the baseline choices without interface dissipation, to get a baseline scheme that is entropy conservative.

- Remark 2.3** (i) The discrete volume weighted contravariant vectors $\mathbb{J}\vec{a}^\alpha, \alpha = 1, 2, 3$, do not depend on the solution J of (2.71a), since these vectors are computed by the conservative curl form (2.52). Thus, the discrete metric identities (2.53) are satisfied and the normal computation (2.72) is watertight. This means the normal vector and the surface element are continuous across element interfaces.
- (ii) Since the discrete volume weighted contravariant vectors $\mathbb{J}\vec{a}^\alpha, \alpha = 1, 2, 3$, are computed by the conservative curl form (2.52), the Eq.(2.20) is satisfied on the discrete level.
- (iii) The Eqs.(2.71a) and (2.71a) ensure that the Eq.(2.30) is satisfied on the discrete level.

2.6 Semi-discrete Entropy Conservation

The spatial integral of the entropy is bounded in time on the continuous level. Thus, it is desirable that a numerical method is stable in the sense that a discrete version of this integral is bounded in time, too. In the context of the moving mesh semi-discrete DGSEM (2.71), we are interested to find an upper bound for the quantity

$$\bar{S}(\tau) := \sum_{k=1}^K \langle S(\tau), J(\tau) \rangle_N, \quad \forall \tau \in [0, T], \tag{2.78}$$

where $S = \mathbb{I}^N(s)$ is a polynomial approximation for the entropy s . Next, we prove the following statement for the semi-discrete moving mesh DGSEM.

Theorem 2.4 *Suppose the flux functions $\overleftrightarrow{\mathbf{G}}^{\text{EC}}$ in the derivative projection operator (2.64), the numerical surface fluxes $\tilde{\mathbf{G}}_n^*$ are computed by Cartesian fluxes $\mathbf{G}_l^{\text{EC}}, l = 1, 2, 3$, with the properties (2.65), (2.66), (2.67) and periodic boundary conditions are used. Then the semi-discrete moving mesh DGSEM (2.71) satisfies the discrete entropy equation*

$$\bar{S}(\tau) = \bar{S}(0), \quad \forall \tau \in [0, T], \tag{2.79}$$

where $\bar{S}(\tau)$ is given by (2.78).

Proof We proceed similar as in the continuous entropy analysis, use the polynomial approximation $\varphi = \mathbb{I}^N(\mathbf{w}) = \mathbf{W}$ as test function in the Eq. (2.71b) and obtain

$$\left\langle \frac{\partial (\mathbf{J}\mathbf{U})}{\partial \tau}, \mathbf{W} \right\rangle_N = - \left\langle \overleftrightarrow{\mathbb{D}}^N \cdot \overleftrightarrow{\mathbf{G}}^{\text{EC}}, \mathbf{W} \right\rangle_N - \int_{\partial E, N} \mathbf{W}^T (\tilde{\mathbf{G}}_n^* - \tilde{\mathbf{G}}_n) dS. \tag{2.80}$$

First, we consider the left hand side in the Eq.(2.80). Since interpolation and quadrature nodes are collocated, the nodal values can be analyzed by the same arguments as in the continuous computations (2.38) and (2.39). Hence, we obtain for all $i, j, k = 0, \dots, N$

$$\begin{aligned}
 J_{ijk} \mathbf{W}_{ijk}^T \left(\frac{\partial}{\partial \tau} \mathbf{U}_{ijk} \right) &= J_{ijk} \mathbf{W}_{ijk}^T \frac{\partial \mathbf{U}_{ijk}}{\partial t} + \mathbf{W}_{ijk}^T \left(\vec{v}_{ijk} \cdot \vec{\nabla}_\xi \mathbf{U}_{ijk} \right) \\
 &= J_{ijk} \frac{\partial S_{ijk}}{\partial t} + \vec{v} \cdot \vec{\nabla}_\xi S_{ijk} = J_{ijk} \left(\frac{\partial}{\partial \tau} S_{ijk} \right).
 \end{aligned}
 \tag{2.81}$$

Next we multiply the Eq. (2.81) by ω_{ijk} and compute the sum over all LGL nodes. This gives

$$\left\langle \mathbf{J} \frac{\partial \mathbf{U}}{\partial \tau}, \mathbf{W} \right\rangle_N = \left\langle \frac{\partial S}{\partial \tau}, \mathbf{J} \right\rangle_N.
 \tag{2.82}$$

Since we assume time continuity for our semi-discrete analysis, we apply the product rule in time and obtain by (2.82)

$$\begin{aligned}
 \left\langle \frac{\partial (\mathbf{J}\mathbf{U})}{\partial \tau}, \mathbf{W} \right\rangle_N &= \left\langle \frac{\partial S}{\partial \tau}, \mathbf{J} \right\rangle_N + \left\langle \frac{\partial \mathbf{J}}{\partial \tau}, \mathbf{W}^T \mathbf{U} \right\rangle_N \\
 &= \frac{\partial}{\partial \tau} \langle S, \mathbf{J} \rangle_N + \left\langle \frac{\partial \mathbf{J}}{\partial \tau}, \Phi \right\rangle_N \\
 &= \frac{\partial}{\partial \tau} \langle S, \mathbf{J} \rangle_N + \left\langle \vec{\mathbb{D}}^N \cdot \vec{v}, \Phi \right\rangle_N + \int_{\partial E, N} (\vec{v}_n^* - \tilde{v}_n) \Phi \, dS,
 \end{aligned}
 \tag{2.83}$$

where we used in the last step the Eq. (2.71a) with the test function $\varphi = \Phi$. We note that the quantity Φ is defined as a polynomial with the nodal values (2.68). In the Appendix C.1, the following equation is proven

$$\left\langle \vec{\mathbb{D}}^N \cdot \vec{\mathbf{G}}^{\text{EC}}, \mathbf{W} \right\rangle_N = \int_{\partial E, N} (\tilde{F}_n^s - \tilde{v}_n^s S) \, dS - \left\langle \vec{\mathbb{D}}^N \cdot \vec{v}, \Phi \right\rangle_N,
 \tag{2.84}$$

where $\tilde{F}_n^s = (\hat{s}\vec{n}) \cdot \vec{F}^s$ with $\vec{F}^s = [F_1^s, F_2^s, F_3^s]^T$. Here the polynomials $F_l^s, l = 1, 2, 3$, are given by (2.69). Moreover, we obtain by (2.73) and (2.74)

$$\begin{aligned}
 &- \mathbf{W}^T \left(\tilde{\mathbf{G}}_n^* - \tilde{\mathbf{G}}_n \right) - \left(\tilde{F}_n^s - \tilde{v}_n^s S \right) \\
 &= \sum_{l=1}^3 \left\{ \hat{s}n_l \left(\mathbf{W}^T \mathbf{F}_l - F_l^s \right) - \hat{s}n_l \left(\mathbf{W}^T \mathbf{U} - S \right) \right\} - \mathbf{W}^T \tilde{\mathbf{G}}_n^* \\
 &= \tilde{\Psi}_n - \tilde{v}_n^* \Phi - \mathbf{W}^T \tilde{\mathbf{G}}_n^*,
 \end{aligned}
 \tag{2.85}$$

where Φ as well as $\Psi_l, l = 1, 2, 3$, are polynomials with nodal values (2.68) and $\tilde{\Psi}_n := (\hat{s}\vec{n}) \cdot \vec{\Psi}$ with $\vec{\Psi} = [\Psi_1, \Psi_2, \Psi_3]^T$. Next, we plug the Eqs. (2.83), (2.84), (2.85) in (2.80) and rearrange. This results in the equation

$$\begin{aligned}
 \frac{\partial}{\partial \tau} \langle S, \mathbf{J} \rangle_N &= - \int_{\partial E, N} \left\{ \mathbf{W}^T \left(\tilde{\mathbf{G}}_n^* - \tilde{\mathbf{G}}_n \right) + \left(\tilde{F}_n^s - \tilde{v}_n^s S \right) - \left(\tilde{v}_n^* - \tilde{v}_n \right) \Phi \right\} \, dS \\
 &= \int_{\partial E, N} \left(\tilde{\Psi}_n - \tilde{v}_n^* \Phi - \mathbf{W}^T \tilde{\mathbf{G}}_n^* \right) \, dS.
 \end{aligned}
 \tag{2.86}$$

Then, we sum the Eq. (2.86) over all elements and use that the normal computation (2.72) is watertight. This provides the equation

$$\begin{aligned} \frac{\partial}{\partial \tau} \bar{S}(\tau) &= \sum_{\substack{\text{Boundary} \\ \text{faces}}} \int_{\partial E, N} \left(\tilde{\Psi}_{\hat{n}} - v_{\hat{n}}^* \Phi - \mathbf{W}^T \tilde{\mathbf{G}}_{\hat{n}}^* \right) dS \\ &\quad - \sum_{\substack{\text{Interior} \\ \text{faces}}} \int_{\partial E, N} \left(\llbracket \tilde{\Psi}_{\hat{n}} \rrbracket - \{\{ \tilde{v}_{\hat{n}} \} \} \llbracket \Phi \rrbracket - \llbracket \mathbf{W} \rrbracket^T \tilde{\mathbf{G}}_{\hat{n}}^* \right) dS. \end{aligned} \tag{2.87}$$

Since the numerical surface fluxes $\tilde{\mathbf{G}}_{\hat{n}}^*$ are computed by Cartesian fluxes \mathbf{G}_l^{EC} , $l = 1, 2, 3$, with the properties (2.67), it follows

$$\llbracket \tilde{\Psi}_{\hat{n}} \rrbracket - \{\{ \tilde{v}_{\hat{n}} \} \} \llbracket \Phi \rrbracket - \llbracket \mathbf{W} \rrbracket^T \tilde{\mathbf{G}}_{\hat{n}}^* = \sum_{l=1}^3 \hat{\sigma} n_l \left(\llbracket \Psi_l \rrbracket - \{\{ v_l \} \} \llbracket \Phi \rrbracket - \llbracket \mathbf{W} \rrbracket^T \mathbf{G}_l^* \right) = 0. \tag{2.88}$$

Hence, we obtain the equation

$$\frac{\partial}{\partial \tau} \bar{S}(\tau) = \sum_{\substack{\text{Boundary} \\ \text{faces}}} \int_{\partial E, N} \left(\tilde{\Psi}_{\hat{n}} - \tilde{v}_{\hat{n}}^* \Phi - \mathbf{W}^T \tilde{\mathbf{G}}_{\hat{n}}^* \right) dS. \tag{2.89}$$

Since the method is investigated with periodic boundary conditions, we obtain the desired entropy equation by integrating the Eq. (2.89) over the time interval $[0, T]$. This completes the proof of Theorem 2.4. \square

Remark 2.5 The proof of Theorem 2.4 requires the assumptions that the time integration is exact and that properties like positivity preservation (of the water height, density or pressure) are satisfied on the discrete level.

For non-periodic boundary conditions, a proper choice of discrete boundary condition is necessary to bound the term

$$\sum_{\substack{\text{Boundary} \\ \text{faces}}} \int_{\partial E, N} \left(\tilde{\Psi}_{\hat{n}} - \tilde{v}_{\hat{n}}^* \Phi - \mathbf{W}^T \tilde{\mathbf{G}}_{\hat{n}}^* \right) dS \tag{2.90}$$

in Eq. (2.89) such that the method becomes entropy conservative/stable. Thus, we obtain from Theorem 2.4 the following Corollary.

Corollary 2.6 Suppose the flux functions $\overset{\leftrightarrow}{\mathbf{G}}^{EC}$ in the derivative projection operator (2.64) and the numerical surface fluxes $\tilde{\mathbf{G}}_{\hat{n}}^*$ (2.77) are computed by Cartesian fluxes \mathbf{G}_l^{EC} , $l = 1, 2, 3$, with the properties (2.65), (2.66), (2.67) and a proper dissipative boundary condition, e.g., [14,31,53], is applied. Then the semi-discrete moving mesh DGSEM (2.71) satisfies the discrete entropy inequality

$$\bar{S}(\tau) \leq \bar{S}(0), \quad \forall \tau \in [0, T]. \tag{2.91}$$

2.7 Semi-discrete Entropy Stability

Entropy conservation can be merely expected when a reversible process is described by a system of PDEs. In general, conservation laws are describing irreversible processes with

discontinuous solutions. Hence, it cannot be expected that the entropy conservative moving mesh DGSEM provides a physical meaningful discretization for the system (2.1). However, the entropy conservative flux at the element interfaces can be augmented by an artificial dissipation term.

In the literature, there are different strategies to add dissipation to an entropy conservative flux. Here, dissipation is added via a matrix operator. This approach, for instance, has been used in the context of gas dynamics by Chandrashekar [8] or Winters et al. [63].

The conservative variables \mathbf{u} can be written in dependence of the entropy variables \mathbf{w} . Differentiation of the conservative variables $\mathbf{u} = \mathbf{u}(\mathbf{w})$ provides the symmetric positive definite matrix $\frac{\partial \mathbf{u}}{\partial \mathbf{w}}$, since the system (2.1) is assumed to be symmetrizable (cf. e.g. [29]). Thus, it follows by a Taylor expansion up to first order

$$[[\mathbf{u}]] = \left(\frac{\partial \mathbf{u}}{\partial \mathbf{w}} \right) [[\mathbf{w}]] + \mathcal{O} (||[\mathbf{w}]||^2), \tag{2.92}$$

where the jump operator is defined by (2.75) at the interfaces. Furthermore, the system (2.1) is hyperbolic. Thus, the flux Jacobian matrices $\frac{\partial \mathbf{f}_l}{\partial \mathbf{u}}, l = 1, 2, 3$, are diagonalizable and have real eigenvalues $\{\lambda_i^l(\mathbf{u})\}_{i=1}^p \subseteq \mathbb{R}$. The corresponding right eigenvector matrices are \mathcal{R}_l . For the Euler equations Merriam [47] has shown that there are block diagonal scaling matrices such that the Hessian matrix of the entropy can be represented by scaled right eigenvector matrices. This result has been generalized and is known as the eigenvector scaling theorem (cf. Barth [2, Theorem 4]). Hence, according to the eigenvector scaling theorem, there are symmetric block diagonal scaling matrices \mathcal{T}_l with

$$\frac{\partial \mathbf{f}_l}{\partial \mathbf{u}} = \tilde{\mathcal{R}}_l \Lambda_l(\mathbf{u}) \tilde{\mathcal{R}}_l^{-1}, \quad \frac{\partial \mathbf{u}}{\partial \mathbf{w}} = \tilde{\mathcal{R}}_l \tilde{\mathcal{R}}_l^T, \quad \tilde{\mathcal{R}}_l = \mathcal{R}_l \mathcal{T}_l, \quad l = 1, 2, 3, \tag{2.93}$$

where $\Lambda_l(\mathbf{u}) := \text{diag}(\lambda_1^l(\mathbf{u}), \dots, \lambda_p^l(\mathbf{u}))$. The flux Jacobian matrices $\frac{\partial \mathbf{g}_l}{\partial \mathbf{u}} = \frac{\partial \mathbf{f}_l}{\partial \mathbf{u}} - \nu_l \mathcal{I}_p$ have the real eigenvalues $\{\lambda_i^l(\mathbf{u}) - \nu_l\}_{i=1}^p$ and the same right eigenvectors as the flux Jacobian $\frac{\partial \mathbf{f}_l}{\partial \mathbf{u}}$. We note that \mathcal{I}_p is the $p \times p$ identity matrix. Hence, it follows

$$\frac{\partial \mathbf{g}_l}{\partial \mathbf{u}} = \tilde{\mathcal{R}}_l \Lambda_l(\nu, \mathbf{u}) \tilde{\mathcal{R}}_l^{-1}, \quad \Lambda_l(\nu, \mathbf{u}) := \text{diag}(\lambda_1^l(\mathbf{u}) - \nu_l, \dots, \lambda_p^l(\mathbf{u}) - \nu_l), \tag{2.94}$$

$l = 1, 2, 3.$

Furthermore, we obtain by (2.93)

$$\left(\frac{\partial \mathbf{g}_l}{\partial \mathbf{w}} \right) = \left(\frac{\partial \mathbf{g}_l}{\partial \mathbf{u}} \right) \left(\frac{\partial \mathbf{u}}{\partial \mathbf{w}} \right) = \tilde{\mathcal{R}}_l \Lambda_l(\nu, \mathbf{u}) \tilde{\mathcal{R}}_l^T, \quad l = 1, 2, 3. \tag{2.95}$$

The Eq. (2.95) motivates the definition of the following matrix dissipation operators

$$\mathcal{H}_l = \hat{\mathcal{R}}_l |\Lambda_l| \hat{\mathcal{R}}_l^T, \quad \hat{\mathcal{R}}_l = \mathcal{R}_l^* \mathcal{T}_l^*, \quad l = 1, 2, 3. \tag{2.96}$$

where the matrices $\mathcal{R}_l^*, \mathcal{T}_l^*$, depend on some averaged values of the states $\mathbf{U}^-, \mathbf{U}^+$ and they are consistent with the right eigenvector matrix \mathcal{R}_l and the scaling matrix \mathcal{T}_l . The matrix $|\Lambda_l|$ depends on the values $\{\lambda_i^l(\mathbf{U}^-) - \nu_l^-\}_{i=1}^p$ and $\{\lambda_i^l(\mathbf{U}^+) - \nu_l^+\}_{i=1}^p$. The matrix \mathcal{H}_l needs to be a symmetric positive definite matrix. Therefore, the matrix $|\Lambda_l|$ has to be chosen carefully. In Appendix B.3, the matrices to construct the dissipation operator (2.96) for the compressible Euler equations are given. There it can be also seen which average values are used to evaluate the states $\mathbf{U}^-, \mathbf{U}^+$ in the matrices.

The dissipation operator (2.96) is used to modify the Cartesian numerical surface flux at the element interfaces as follows

$$\mathbf{G}_l^{ES} := \mathbf{G}_l^{EC} - \frac{1}{2} \mathcal{H}_l \llbracket \mathbf{W} \rrbracket, \quad l = 1, 2, 3, \tag{2.97}$$

where the Cartesian fluxes $\mathbf{G}_l^{EC}, l = 1, 2, 3$, satisfy (2.65), (2.66), (2.67). The contravariant surface numerical fluxes $\tilde{\mathbf{G}}_n^{EC}$ are computed by (2.77). We note that the dissipation operator (2.96) is not used to modify the entropy conservative fluxes in the derivative projection operator (2.64).

The numerical fluxes $\tilde{\mathbf{G}}_n^{ES}$ do not provide an entropy conservative scheme, but the result in Theorem 2.4 can be used to prove that the moving mesh DGSEM becomes entropy stable, such that the discrete mathematical entropy is bounded at any time by its initial data, when the numerical fluxes $\tilde{\mathbf{G}}_n^{ES}$ are used at the element interfaces and it is assumed that the time integration is exact and that properties like positivity preservation (of the water height, density or pressure) are satisfied on the discrete level.

Corollary 2.7 *Suppose the flux functions $\vec{\mathbf{G}}^{EC}$ in the derivative projection operator (2.64) are computed by Cartesian fluxes $\mathbf{G}_l^{EC}, l = 1, 2, 3$, with the properties (2.65), (2.66), (2.67), the numerical surface fluxes $\tilde{\mathbf{G}}_n^* = \tilde{\mathbf{G}}_n^{ES}$ are computed by the Cartesian fluxes $\mathbf{G}_l^{ES}, l = 1, 2, 3$, given by (2.97) and periodic boundary conditions are used. Then the semi-discrete moving mesh DGSEM (2.71) satisfies the discrete entropy inequality*

$$\bar{S}(\tau) \leq \bar{S}(0), \quad \forall \tau \in [0, T]. \tag{2.98}$$

Furthermore, with proper dissipative boundary conditions, the method satisfies again the inequality (2.98) for non-periodic problems.

Proof We proceed as in the proof of Theorem 2.4 and obtain the equation

$$\begin{aligned} \frac{\partial}{\partial \tau} \bar{S}(\tau) = & \sum_{\substack{\text{Boundary} \\ \text{faces}}} \int_{\partial E, N} \left(\tilde{\Psi}_n - \tilde{v}_n^* \Phi - \mathbf{W}^T \tilde{\mathbf{G}}_n^{ES} \right) dS \\ & - \sum_{\substack{\text{Interior} \\ \text{faces}}} \int_{\partial E, N} \left(\llbracket \tilde{\Psi}_n \rrbracket - \llbracket \tilde{v}_n \rrbracket \llbracket \Phi \rrbracket - \llbracket \mathbf{W} \rrbracket^T \tilde{\mathbf{G}}_n^{ES} \right) dS. \end{aligned} \tag{2.99}$$

Since the numerical surface fluxes $\tilde{\mathbf{G}}_n^{ES}$ are computed by the Cartesian fluxes (2.97) and the fluxes $\mathbf{G}_l^{EC}, l = 1, 2, 3$, satisfy (2.67), it follows

$$\begin{aligned} & \llbracket \tilde{\Psi}_n \rrbracket - \llbracket \Phi \rrbracket \llbracket \tilde{v}_n \rrbracket - \llbracket \mathbf{W} \rrbracket^T \tilde{\mathbf{G}}_n^{ES} \\ & = \sum_{l=1}^3 \hat{s} n_l \left(\llbracket \Psi_l \rrbracket - \llbracket \Phi \rrbracket \llbracket v_l \rrbracket - \llbracket \mathbf{W} \rrbracket^T \mathbf{G}_l^{EC} + \frac{1}{2} \llbracket \mathbf{W} \rrbracket^T \mathcal{H}_l \llbracket \mathbf{W} \rrbracket \right) \\ & = \frac{1}{2} \sum_{l=1}^3 \hat{s} n_l \llbracket \mathbf{W} \rrbracket^T \mathcal{H}_l \llbracket \mathbf{W} \rrbracket. \end{aligned} \tag{2.100}$$

Since the matrices $\mathcal{H}_l, l = 1, 2, 3$, are symmetric positive definite and the outward normal vectors of the curved elements are positive oriented, the Eq. (2.100) provides

$$- \sum_{\substack{\text{Interior} \\ \text{faces}}} \int_{\partial E, N} \left(\llbracket \tilde{\Psi}_n \rrbracket - \llbracket \tilde{v}_n \rrbracket \llbracket \Phi \rrbracket - \llbracket \mathbf{W} \rrbracket^T \tilde{\mathbf{G}}_n^{ES} \right) dS \leq 0. \tag{2.101}$$

Hence, we obtain the inequality

$$\frac{\partial}{\partial \tau} \bar{S}(\tau) \leq \sum_{\substack{\text{Boundary} \\ \text{faces}}} \int_{\partial E, N} \left(\tilde{\Psi}_{\hat{n}} - \tilde{v}_{\hat{n}}^* \Phi - \mathbf{W}^T \tilde{\mathbf{G}}_{\hat{n}}^{\text{ES}} \right) dS. \tag{2.102}$$

The right hand side of the inequality vanishes, since the method is investigated with periodic boundary conditions. Hence, we obtain the the desired inequality (2.98) by integrating (2.102) over the temporal interval $[0, T]$. \square

2.8 Free Stream Preservation for the Moving Mesh DGSEM

In this section, we check the discretization of the geometric and metric terms in time. Since DG methods with the forward Euler discretization are unstable [9, 16], we investigate directly the discretization by an explicit RK method with $s \geq 2$ stages and the characteristic coefficients $\{a_{\varrho\sigma}\}_{\varrho, \sigma=1}^s, \{b_{\sigma}\}_{\sigma=1}^s, \{c_{\sigma}\}_{\sigma=1}^s$. It is worth to mention that a Courant–Friedrichs–Lewy (CFL) restriction is necessary when an explicit s -stage RK method is used in the DG framework. In order to present the RK discretization of the semi-discrete DGSEM (2.71), it is beneficial to write the method in the equivalent nodal representation. This representation is for all $i, j, k = 0, \dots, N$, given by

$$\frac{\partial \mathbf{J}_{ijk}}{\partial \tau} = \mathbf{V}((\tilde{\mathbf{v}})_{ijk}), \tag{2.103a}$$

$$\frac{\partial (\mathbf{J}_{ijk} \mathbf{U}_{ijk})}{\partial \tau} = \mathbf{G}((\tilde{\mathbf{v}})_{ijk}, \mathbf{U}_{ijk}), \tag{2.103b}$$

where the right hand sides are given by

$$\mathbf{V}((\tilde{\mathbf{v}})_{ijk}) := \bar{\mathbb{D}}^N \cdot \tilde{\mathbf{v}}_{ijk} + \frac{1}{\omega_i \omega_j \omega_k} \int_{\partial E, N} \ell_i \ell_j \ell_k (\tilde{v}_{\hat{n}}^* - \tilde{v}_{\hat{n}}) dS, \tag{2.104}$$

$$\mathbf{G}((\tilde{\mathbf{v}})_{ijk}, \mathbf{U}_{ijk}) := -\bar{\mathbb{D}}^N \cdot \overset{\leftrightarrow}{\tilde{\mathbf{G}}}_{ijk}^{\text{EC}} - \frac{1}{\omega_i \omega_j \omega_k} \int_{\partial E, N} \ell_i \ell_j \ell_k (\tilde{\mathbf{G}}_{\hat{n}}^* - \tilde{\mathbf{G}}_{\hat{n}}) dS \tag{2.105}$$

with the tensorial Lagrange polynomials $\ell_i \ell_j \ell_k$ given by (2.47).

It should be noted that the solutions $\mathbf{J}_{ijk}, i, j, k = 0, \dots, N$, of the ordinary differential equations (ODEs) (2.103a) need to be positive. We note that the solutions $\mathbf{J}_{ijk}, i, j, k = 0, \dots, N$ of the ODEs (2.103a) are not used to compute the volume weighted contravariant coordinate vectors $\mathbb{J} \vec{a}^l, l = 1, 2, 3$, in the right hand sides (2.104) and (2.105). These vectors are computed from the mapping by the conservative curl form (2.52). Hence, the right hand sides (2.104) are independent of $\mathbf{J}_{ijk}, i, j, k = 0, \dots, N$. Therefore, the solutions of the ODEs (2.103a) are positive, if the grid velocity does not cause too much distortion in the mesh which is ensured when the assumptions (A1)-(A3) are satisfied.

Next, the interval $[0, T]$ is divided in time points t^n . The step size of the time discretization is Δt . The DGSEM solutions, the fluxes and the grid velocity field are approximated in the time points t^n , e.g. $\mathbf{U}(t^n) \approx \mathbf{U}^n$. Then, the RK discretization of the semi-discrete DGSEM is given by

$$\text{for } \varrho = 1, \dots, s :$$

$$J_{ijk}^{(\varrho)} = J_{ijk}^n + \Delta t \sum_{\sigma=1}^{\varrho-1} a_{\varrho\sigma} V \left((\vec{v})_{ijk}^{n+\sigma} \right), \tag{2.106a}$$

$$U_{ijk}^{(\varrho)} = U_{ijk}^n + \frac{\Delta t}{J_{ijk}^{(\varrho)}} \sum_{\sigma=1}^{\varrho-1} a_{\varrho\sigma} \left(\mathbf{G} \left((\vec{v})_{ijk}^{n+\sigma}, U_{ijk}^{(\sigma)} \right) - V \left((\vec{v})_{ijk}^{n+\sigma} \right) U_{ij}^n \right), \tag{2.106b}$$

$$J_{ijk}^{n+1} = J_{ijk}^n + \Delta t \sum_{\sigma=1}^s b_{\sigma} V \left((\vec{v})_{ijk}^{n+\sigma} \right), \tag{2.106c}$$

$$U_{ijk}^{n+1} = U_{ijk}^n + \frac{\Delta t}{J_{ijk}^{(n+1)}} \sum_{\sigma=1}^s b_{\sigma} \left(\mathbf{G} \left((\vec{v})_{ijk}^{n+\sigma}, U_{ijk}^{(\sigma)} \right) - V \left((\vec{v})_{ijk}^{n+\sigma} \right) U_{ijk}^n \right), \tag{2.106d}$$

where $(\vec{v})_{ijk}^{n+\sigma} := \vec{v} \left(\xi_i^1, \xi_j^2, \xi_k^3, t^n + c_{\sigma} \Delta t \right)$ and $\{\xi_i^1\}_{i=0}^N, \{\xi_j^2\}_{i=0}^N, \{\xi_k^3\}_{i=0}^N$ are sets of LGL points. Next, we prove that the fully-discrete split form RK-DGSEM (2.106) satisfies the free stream preservation property.

Theorem 2.8 *Suppose the fully-discrete split form RK-DGSEM (2.106) is investigated with periodic boundary conditions and the solution of the scheme is given by $U_{ijk}^n = \mathbf{C} := (c_1, \dots, c_p)^T \in \mathbb{R}^p$ for all elements $e_{\kappa}(t^n), \kappa = 1, \dots, K$, and the numerical fluxes satisfy (2.65). Then, the constant states $c_l, l = 1, \dots, p$, are preserved in each Runge–Kutta stage (2.106b). In particular, the solution of the fully-discrete DGSEM method at time level t^{n+1} is $U_{ijk}^{n+1} = \mathbf{C}$.*

Proof Let $\varrho \in \{1, \dots, s\}$ be an arbitrary fixed index. We are interested to investigate the ϱ -th RK stage. Hence, without loss of generality, we can assume that $U^{(\sigma)} = \mathbf{C}$ for all $\sigma = 0, \dots, \varrho - 1$. Then, since the flux $\vec{\mathbf{G}}^{\text{EC}}$ satisfies (2.65), it follows

$$\begin{aligned} \mathbb{D}^N \cdot \vec{\mathbf{G}}_{ijk}^{\text{EC}} &= 2 \sum_{m=0}^N \mathcal{D}_{im} \{ \mathbb{J} \vec{a}^1 \}_{(i,m)jk} \cdot \vec{\mathbf{F}}(\mathbf{C}) \\ &\quad + 2 \sum_{m=0}^N \mathcal{D}_{jm} \{ \mathbb{J} \vec{a}^2 \}_{i(j,m)k} \cdot \vec{\mathbf{F}}(\mathbf{C}) \\ &\quad + 2 \sum_{m=0}^N \mathcal{D}_{km} \{ \mathbb{J} \vec{a}^3 \}_{ij(k,m)} \cdot \vec{\mathbf{F}}(\mathbf{C}) - \mathbb{D}^N \cdot \left(\vec{v} \right)_{ijk}^{n+\sigma} \mathbf{C}. \end{aligned} \tag{2.107}$$

Furthermore, since the metric terms are computed by the conservative curl form (2.52), we obtain

$$\begin{aligned} &2 \sum_{m=0}^N \left(\mathcal{D}_{im} \{ \mathbb{J} \vec{a}^1 \}_{(i,m)jk} + \mathcal{D}_{jm} \{ \mathbb{J} \vec{a}^2 \}_{i(j,m)k} + \mathcal{D}_{km} \{ \mathbb{J} \vec{a}^3 \}_{ij(k,m)} \right) \\ &= \sum_{m=0}^N \left(\mathcal{D}_{im} (\mathbb{J} \vec{a}^1)_{mjk} + \mathcal{D}_{jm} (\mathbb{J} \vec{a}^2)_{imk} + \mathcal{D}_{km} (\mathbb{J} \vec{a}^3)_{ijm} \right) = 0. \end{aligned} \tag{2.108}$$

Here we used the split form Lemma from Gassner et al. [23, Lemma 1] in the first step and in the second step we used the identity (2.60) for the discrete metric identities. Thus, it follows that

$$\vec{\mathbb{D}}^N \cdot \vec{\mathbf{G}}_{ijk}^{EC} = -\vec{\mathbb{D}}^N \cdot \left(\vec{v} \right)_{ijk}^{n+\sigma} \mathbf{C}. \tag{2.109}$$

Similar, since the flux $\vec{\mathbf{G}}^*$ satisfies (2.65), follows

$$\begin{aligned} \vec{\mathbf{G}}_{\hat{n}}^* - \vec{\mathbf{G}}_{\hat{n}} &= \sum_{l=1}^3 \hat{s}n_l (\mathbf{F}_l(\mathbf{C}) - \{v_l^{n+\sigma}\} \mathbf{C}) - (\hat{s}\vec{n}) \cdot \left(\vec{\mathbf{F}}(\mathbf{C}) - (\vec{v})^{n+\sigma} \mathbf{C} \right) \\ &= -(\hat{s}\vec{n}) \cdot (\{(\vec{v})^{n+\sigma}\} - (\vec{v})^{n+\sigma}) \mathbf{C} = -(\vec{v}_{\hat{n}}^{*,n+\sigma} - \vec{v}_{\hat{n}}^{n+\sigma}) \mathbf{C}. \end{aligned} \tag{2.110}$$

Thus, the Eqs. (2.109) and (2.110) give

$$\begin{aligned} \mathbf{G} \left((\vec{v})_{ijk}^{n+\sigma}, \mathbf{C} \right) &= \left(\vec{\mathbb{D}}^N \cdot \left(\vec{v} \right)_{ijk}^{n+\sigma} + \frac{1}{\omega_i \omega_j \omega_k} \int_{\partial E, N} \ell_i \ell_j \ell_k \left(\vec{v}_{\hat{n}}^{*,n+\sigma} - \vec{v}_{\hat{n}}^{n+\sigma} \right) dS \right) \mathbf{C} \\ &= \mathbf{V} \left((\vec{v})_{ijk}^{n+\sigma} \right) \mathbf{C}. \end{aligned} \tag{2.111}$$

Hence, the solution of the RK stage (2.106b) is given by

$$\mathbf{U}_{ijk}^{(\varrho)} = \mathbf{C} + \frac{\Delta t}{\mathbf{J}_{ijk}^{(\varrho)}} \sum_{\sigma=1}^{\varrho-1} a_{\varrho\sigma} \left(\mathbf{G} \left((\vec{v})_{ijk}^{n+\sigma}, \mathbf{C} \right) - \mathbf{V} \left((\vec{v})_{ijk}^{n+\sigma} \right) \mathbf{C} \right) = \mathbf{C}. \tag{2.112}$$

Since the parameter ϱ was arbitrary chosen, it follows $\mathbf{U}_{ijk}^{(\varrho)}$ for all $\varrho = 1, \dots, s$. In particular, it follows

$$\mathbf{U}_{ijk}^{n+1} = \mathbf{C} + \frac{\Delta t}{\mathbf{J}_{ijk}^{n+1}} \sum_{\sigma=1}^s b_{\sigma} \left(\mathbf{G} \left((\vec{v})_{ijk}^{n+\sigma}, \mathbf{C} \right) - \mathbf{V} \left((\vec{v})_{ijk}^{n+\sigma} \right) \mathbf{C} \right) = \mathbf{C}. \tag{2.113}$$

This completes the proof of Theorem 2.8. □

3 Numerical Results

The numerical computations in this section are performed with the open source code FLEXI¹ and the three-dimensional high-order meshes for the simulations are generated with the open source tool HOPR.²

We present tests on three dimensional moving hexahedral curved meshes for the compressible Euler equations. Based on these tests we will evaluate the theoretical findings of the previous sections. The three dimensional compressible Euler equations are given by

$$\frac{\partial \mathbf{u}}{\partial t} + \vec{\nabla} \cdot \vec{\mathbf{f}} = \mathbf{0}. \tag{3.1}$$

¹ www.flexi-project.org.

² www.hopr-project.org.

The state vector and the components of the block vector flux, $\overset{\leftrightarrow}{\mathbf{f}}$, are given by

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ E \end{bmatrix}, \quad \mathbf{f}_1 = \begin{bmatrix} \rho u_1 \\ \rho u_1^2 + p \\ \rho u_1 u_2 \\ \rho u_1 u_3 \\ (E + p) u_1 \end{bmatrix}, \quad \mathbf{f}_2 = \begin{bmatrix} \rho u_2 \\ \rho u_1 u_2 \\ \rho u_2^2 + p \\ \rho u_2 u_3 \\ (E + p) u_2 \end{bmatrix}, \quad \mathbf{f}_3 = \begin{bmatrix} \rho u_3 \\ \rho u_1 u_3 \\ \rho u_2 u_3 \\ \rho u_3^2 + p \\ (E + p) u_3 \end{bmatrix} \quad (3.2)$$

where the conservative variables are the density ρ , the momentum $\rho \vec{u} = [\rho u_1, \rho u_2, \rho u_3]^T$ and the total energy E . In order to close the system, we assume an ideal gas such that the pressure is defined as

$$p = (\gamma - 1) \left(E - \frac{\rho}{2} |\vec{u}|^2 \right), \quad (3.3)$$

where γ is the adiabatic exponent. We choose $\gamma = 1.4$ in the following experiments. The system (3.1) is investigated in the domain $\Omega = [x_{\min}, x_{\max}]^3$. At initial time $t = 0$ the domain is divided in K non-overlapping and conforming cartesian hexahedral elements $e_\kappa(0), \kappa = 1, \dots, K$. For each element $e_\kappa(0), \kappa = 1, \dots, K$, the temporal distribution of a grid point

$$\vec{x}_\kappa(0) = (x_1^\kappa(0), x_2^\kappa(0), x_3^\kappa(0))^T \in e_\kappa(0) \quad (3.4)$$

is given by

$$\vec{x}_\kappa(t) = \vec{x}_\kappa(0) + 0.05L \sin(2\pi t) \sin\left(\frac{2\pi}{L} x_1^\kappa(0)\right) \sin\left(\frac{2\pi}{L} x_2^\kappa(0)\right) \sin\left(\frac{2\pi}{L} x_3^\kappa(0)\right), \quad (3.5)$$

where $L := x_{\max} - x_{\min}$. In Fig. 2, we show a slice through a three dimensional mesh with $K = 16^3$ elements at initial time and at its maximal distortion. The mesh velocity is calculated by exact differentiation of Eq. (3.5). Note that the formula (3.5) is a common way to construct a deforming domain. For instance, similar formulas were used for the DG methods in [37,64] and the collocation method in [66]. Furthermore, the five stage fourth order low-storage two-register explicit RK method (RK4(3)5[2R+]) from Kennedy, Carpenter and Lewis [34, Section 3.4.] is used for the time-integration in the numerical experiments. The CFL restriction is computed as in [12]

$$\frac{\Delta t}{\min_{1 \leq \kappa \leq K} |h_\kappa(t^n)|} \leq \frac{C_{\text{CFL}}}{(2N + 1) \lambda_{\max}}, \quad (3.6)$$

where $h_\kappa(t^n)$ is the minimum element size of $e_\kappa(t^n)$, $C_{\text{CFL}} \in (0, 1]$ and λ_{\max} is the largest advective wave speed at the current time level traveling in either the x_1, x_2, x_3 -direction.

3.1 Experimental Convergence Rates

In this section, we verify the high-order approximation of the moving DGSEM (2.71). For this purpose, we investigate the domain $\Omega = [-1, 1]^3$ and apply the method of manufactured solutions. Thus, we assume a solution of the form

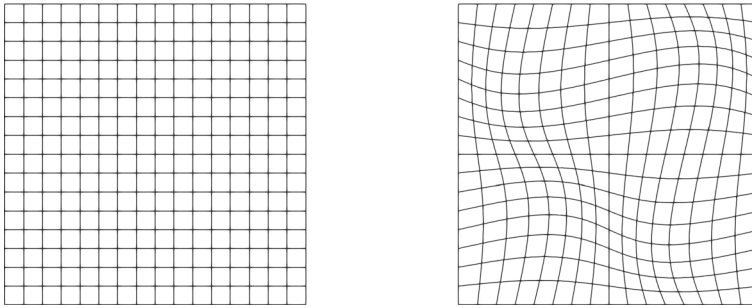


Fig. 2 A slice through a three dimensional mesh with $K = 16^3$ elements at initial time (left) and at its maximal distortion (right)

$$\mathbf{U}(\vec{x}, t) = \begin{bmatrix} \rho(\vec{x}, t) \\ \rho u_1(\vec{x}, t) \\ \rho u_2(\vec{x}, t) \\ \rho u_3(\vec{x}, t) \\ E(\vec{x}, t) \end{bmatrix} = \begin{bmatrix} 2 + 0.1 \sin(\pi(x_1 + x_2 + x_3 - 2 \cdot 0.3t)) \\ 2 + 0.1 \sin(\pi(x_1 + x_2 + x_3 - 2 \cdot 0.3t)) \\ 2 + 0.1 \sin(\pi(x_1 + x_2 + x_3 - 2 \cdot 0.3t)) \\ 2 + 0.1 \sin(\pi(x_1 + x_2 + x_3 - 2 \cdot 0.3t)) \\ [2 + 0.1 \sin(\pi(x_1 + x_2 + x_3 - 2 \cdot 0.3t))]^2 \end{bmatrix}. \tag{3.7}$$

We plug solution (3.7) into the Euler system and compute the residual using a computer algebra system. This term is used as a source term in our convergence tests. We note that this term is handled and discretized as a solution independent part in the numerical computation.

We run the convergence test with periodic boundary conditions. Furthermore, the moving mesh DGSEM (2.71) is applied with the flux function in Appendix B.1 as volume and surface flux. In addition, the surface flux is stabilized by the dissipation operator in Appendix B.3. Besides using the grid point distribution given in (3.5), we also compute static reference solutions, by setting the grid velocity to zero. In this case, the moving mesh DGSEM (2.71) degenerates to the split form DGSEM for static meshes [23,24].

In Table 1, we list the experimental order of convergence (EOC) and L^2 errors for the conservative variables that we obtain for polynomials with odd degree $N = 3$ on a static mesh (top) and on a moving mesh (bottom). To calculate the L^2 norm, we interpolate the polynomial solution to a higher degree (at least twice the degree of the solution) and perform integration on that higher degree. The convergence rates on the moving mesh are not as good as on a static mesh, which can be justified by the high distortion in the mesh from the grid point distribution formula (3.5). However, with an increasing number of elements the same convergence rates as on a static mesh are almost reached. Moreover, the experimental order of convergence (EOC) and L^2 errors for the conservative variables that we obtain for polynomials with even degree $N = 4$ are listed in the Table 2. We observe a similar behavior as for the odd degree $N = 3$. This indicate the high-order approximation properties of the moving mesh DGSEM.

3.2 Entropy Analysis Validation

The three dimensional Euler equations (3.1) are equipped with the entropy/entropy flux pairs

$$s = -\frac{\rho S}{\gamma - 1}, \quad f_l^s = -\frac{\rho S u_l}{\gamma - 1}, \quad l = 1, 2, 3, \tag{3.8}$$

Table 1 Experimental order of convergence (EOC) and L^2 errors at time $T = 5$ for the Euler manufactured solution test (3.7)

K	$L^2(\rho)$	$EOC(\rho)$	$L^2(\rho_{u1})$	$EOC(\rho_{u1})$	$L^2(\rho_{u2})$	$EOC(\rho_{u2})$	$L^2(\rho_{u3})$	$EOC(\rho_{u3})$	$L^2(E)$	$EOC(E)$
2^3	2.84E-02	-	2.74E-02	-	2.74E-02	-	2.74E-02	-	5.47E-02	-
4^3	5.54e-03	2.36	5.43E-03	2.34	5.43E-03	2.34	5.43E-03	2.34	1.03E-03	2.40
8^3	4.35E-05	6.99	4.28E-05	6.99	4.28E-05	6.99	4.28E-05	6.99	1.06E-04	6.61
16^3	2.10E-06	4.37	2.07E-06	4.37	2.08E-06	4.37	2.07E-06	4.37	5.33E-06	4.31
32^3	1.26E-07	4.06	1.24E-07	4.06	1.24E-07	4.06	1.24E-07	4.06	3.19E-07	4.06
64^3	7.82E-09	4.01	7.67E-09	4.01	7.67E-09	4.01	7.67E-09	4.01	1.97E-08	4.01
2^3	4.16E-02	-	3.73E-02	-	3.73E-02	-	3.73E-02	-	5.61E-02	-
4^3	3.77E-03	3.46	3.52E-03	3.41	3.52E-03	3.41	3.52E-03	3.41	6.06E-03	3.21
8^3	1.99E-04	4.25	1.75E-04	4.33	1.75E-04	4.33	1.75E-04	4.33	3.24E-04	4.23
16^3	5.37E-06	5.21	4.91E-06	5.16	4.91E-06	5.16	4.91E-06	5.16	1.20E-05	4.75
32^3	2.18E-07	4.62	2.07E-07	4.57	2.07E-07	4.57	2.07E-07	4.57	5.83E-07	4.36
64^3	1.45E-08	3.92	1.34E-08	3.95	1.34E-08	3.95	1.34E-08	3.95	3.95E-08	3.88

The moving mesh DGSEM is used with $N = 3$ on a static mesh (top) and on a moving mesh (bottom) with the grid point distribution (3.5)

Table 2 Experimental order of convergence (EOC) and L^2 errors at time $T = 5$ for the Euler manufactured solution test (3.7)

K	$L^2(\rho)$	$EOC(\rho)$	$L^2(\rho_{u1})$	$EOC(\rho_{u1})$	$L^2(\rho_{u2})$	$EOC(\rho_{u2})$	$L^2(\rho_{u3})$	$EOC(\rho_{u3})$	$L^2(E)$	$EOC(E)$
2^3	6.99E-03	-	6.64E-03	-	6.64E-03	-	6.64E-03	-	1.16E-02	-
4^3	4.02E-04	4.12	3.97E-04	4.06	3.97E-04	4.06	3.97E-04	4.06	7.96E-04	3.87
8^3	4.50E-06	6.48	4.50E-06	6.47	4.50E-06	6.47	4.50E-06	6.47	1.16E-05	6.10
16^3	1.37E-07	5.04	1.38E-07	5.02	1.38E-07	5.02	1.38E-07	5.02	3.66E-07	4.98
32^3	4.33E-09	4.98	4.40E-09	4.97	4.40E-09	4.97	4.40E-09	4.97	1.16E-08	4.97
64^3	1.36E-10	4.99	1.38E-10	4.99	1.38E-10	4.99	1.38E-10	4.99	3.66E-10	4.99
2^3	1.02E-02	-	9.06E-03	-	9.06E-03	-	9.06E-03	-	1.45E-02	-
4^3	4.53E-04	4.50	4.13E-04	4.46	4.13E-04	4.46	4.13E-04	4.46	7.18E-04	4.33
8^3	1.10E-05	5.37	1.02E-05	5.35	1.02E-05	5.35	1.02E-05	5.35	1.86E-05	5.27
16^3	1.91E-07	5.85	1.72E-07	5.88	1.72E-07	5.88	1.72E-07	5.88	3.81E-07	5.61
32^3	7.28E-09	4.71	6.33E-09	4.77	6.33E-09	4.77	6.33E-09	4.77	1.38E-08	4.78
64^3	2.79E-10	4.71	2.38E-10	4.74	2.38E-10	4.74	2.38E-10	4.74	5.40E-10	4.68

The moving mesh DGSEM is used with $N = 4$ on a static mesh (top) and on a moving mesh (bottom) with the grid point distribution (3.5)

where $\varsigma = \log(p\rho^{-\gamma})$. We are interested in the behavior of the discrete entropy conservation error

$$\Delta_S(T) = \bar{S}(T) - \bar{S}(0), \tag{3.9}$$

where $\bar{S}(\cdot)$ is computed by (2.78). Therefore, we investigate the inviscid Taylor-Green vortex (TGV) test case [56] in the domain $\Omega = [0, 2\pi]^3$. The inviscid TGV can be a challenging test case regarding the robustness of a numerical scheme, partly because the dynamics produce arbitrarily small scales. The flow field is thus by design under-resolved, which makes it a suitable test case to investigate the entropy conservation properties of the scheme. The TGV evolves from the initial data

$$\begin{aligned} \rho &= 1, \\ \vec{u} &= [\sin(x_1) \cos(x_2) \cos(x_3), -\cos(x_1) \sin(x_2) \cos(x_3), 0]^T, \\ p &= p_0 + \frac{1}{16} (\cos(2x_1) + \cos(2x_2)) (\cos(2x_3) + 2). \end{aligned} \tag{3.10}$$

To render the simulation close to incompressible, the Mach number $M_0 = \frac{1}{\sqrt{\gamma p_0}}$ is set to 0.1 by adjusting the pressure correspondingly. We run the simulation with $K = 16^3$ elements and periodic boundary conditions. The final time is chosen to be $T = 13$. Furthermore, we apply the flux function in Appendix B.1 to compute the derivative projection operator (2.64). To calculate the discrete integral entropy, the SBP mass matrices are used directly. In Fig. 3 we present a log-log plot of the entropy conservation error for $N = 3, 4$. We note that the flux in Appendix B.1 is used as surface flux without a dissipation term in these computations, rendering the semi-discrete discretization fully entropy conserving. As expected, we observe the reduction of the remaining entropy conservation error according to the order of the RK method for decreasing CFL numbers. In Fig. 4 the temporal evolution of the entropy conservation errors $\Delta_S(T)$ is given. The CFL number is set to $C_{\text{CFL}} = 0.125$ and polynomial degrees $N = 3$ and $N = 4$ are used. We observe that the entropy conservation error $\Delta_S(T)$ is constant in time (dashed line) when the flux in Appendix B.1 is used without a dissipation term. This indicates the entropy conservation in the TVG test case. On the other hand the entropy conservation error $\Delta_S(T)$ is decreasing in time (solid line) when the surface flux is stabilized by the dissipation term in Appendix B.3. Thus, the moving mesh DGSEM is an entropy stable scheme in this test case. These observations agree with the results in Theorem 2.4 and Corollary 2.6.

3.3 Robustness Test

As has been stated in Sect. 3.2 and noted in literature [50,62], the inviscid TGV is a notoriously challenging test case for the stability of the numerical scheme. While for lower polynomial degrees calculations may be possible, high-order simulations are known to crash even if aliasing-reducing methods like polynomial dealiasing are used [50]. Thus, we use the TGV test case (3.10) to demonstrate the increased robustness of the entropy stable moving mesh DGSEM. To do so, we run the simulation up to $T = 13$ using a polynomial degree of $N = 7$ on three different meshes employing $K_1 = 14^3$, $K_2 = 19^3$ and $K_3 = 26^3$ elements. These cases correspond to the most restrictive simulations from [50]. Again, the point distribution given in (3.5) is used. We use the flux function in Appendix B.1 as volume and surface flux and stabilize the surface flux by the dissipation operator in Appendix B.3.

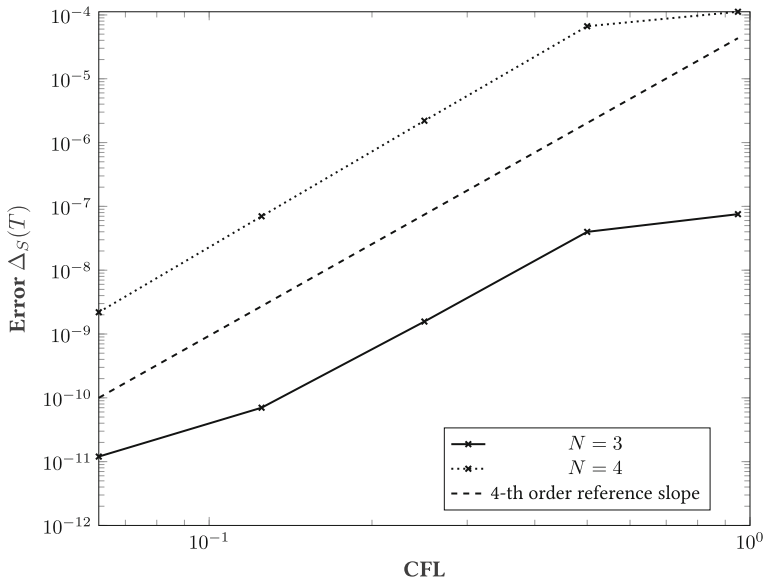


Fig. 3 Log-log plot of the entropy conservation errors $\Delta_S(T)$ for the Euler equations with initial data (3.10). The errors are given at time $T = 13$ for polynomials with degree $N = 3$ (solid line) and $N = 4$ (dotted line) on a curved moving mesh with $K = 16^3$ elements

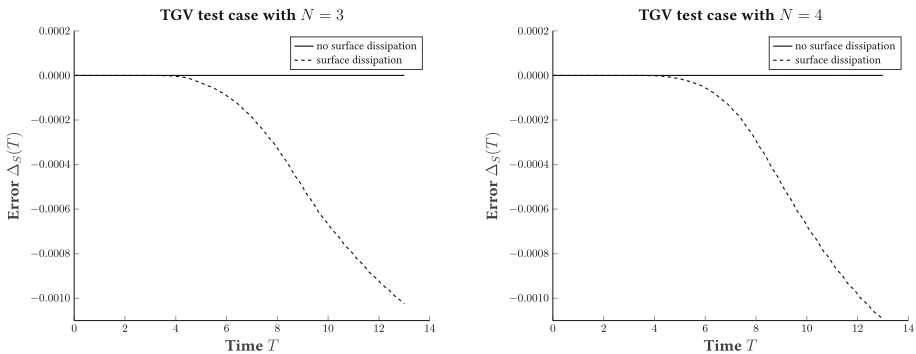


Fig. 4 Temporal evolution of the entropy conservation errors $\Delta_S(T)$ for the Euler equations with initial data (3.10). The flux in Appendix B.1 is used as surface flux without dissipation (solid line) and with the dissipation term in Appendix B.3 (dashed line)

Using the entropy stable moving DGSEM, we were able to run all simulations until final time. This shows that the consistent dissipation operators in combination with the entropy conservative volume fluxes can lead to superior stability properties. We note that simulations without dissipative surface fluxes crash before reaching the final time for higher-order simulations ($N \geq 3$). This highlights the role that entropy conservation plays in the stabilization of challenging numerical problems.

3.4 Free Stream Preservation Validation

We consider the Euler equations (3.1) on the domain $\Omega = [0, 2\pi]^3$ with the initial data

Table 3 Free stream preservation test for $N = 3$ (top) and $N = 4$ (bottom)

C_{CFL}	$L^\infty(\rho)$	$L^\infty(\rho u_1)$	$L^\infty(\rho u_2)$	$L^\infty(\rho u_3)$	$L^\infty(E)$
0.95	2.47E-14	1.40E-12	4.46E-12	4.48E-12	1.33E-12
0.5	2.47E-14	1.40E-12	4.46E-12	4.48E-12	1.33E-12
0.25	2.70E-14	1.40E-12	4.46E-12	4.48E-12	1.36E-12
0.125	3.10E-14	1.40E-12	4.46E-12	4.48E-12	1.43E-12
0.0625	3.78E-14	1.40E-12	4.46E-12	4.48E-12	1.56E-12
0.95	2.07E-14	1.24E-12	5.28E-12	5.21E-12	1.12E-12
0.5	2.49E-14	1.24E-12	5.28E-12	5.21E-12	1.30E-12
0.25	2.81E-14	1.24E-12	5.28E-12	5.21E-12	1.34E-12
0.125	3.32E-14	1.24E-12	5.28E-12	5.21E-12	1.40E-12
0.0625	4.24E-14	1.24E-12	5.28E-12	5.21E-12	1.59E-12

The L^∞ errors measure the difference between the initial data (3.11) and the numerical solution at time $T = 20$ for different constants C_{CFL} .

$$\mathbf{U}(x, t) = \begin{bmatrix} \rho(x, t) \\ \rho u_1(x, t) \\ \rho u_2(x, t) \\ \rho u_3(x, t) \\ E(x, t) \end{bmatrix} = \begin{bmatrix} 1 \\ 0.3 \\ 0 \\ 0 \\ 17 \end{bmatrix}. \tag{3.11}$$

The entropy stable DGSEM is applied with the flux function in Appendix B.1 as volume and surface flux as well as the dissipation operator in Appendix B.3 to stabilize the surface flux. We apply $K = 16^3$ elements, the formula (3.5) to describe the displacement of the mesh points and periodic boundary conditions are used in the simulation. Furthermore, the final time is set to $T = 20$. In Table 3, we present the L^∞ errors between the initial data (3.11) and the numerical solution at time $T = 20$ for polynomials of degree $N = 3$ (top), $N = 4$ (bottom) and different CFL numbers C_{CFL} . The errors are computed by super sampling the polynomial solution, at least doubling the amount of nodes per direction from the underlying solution. We observe that the errors are close to zero and vary slightly for the different CFL numbers. These results indicate the compliance of the free stream preservation property.

4 Conclusions

In this work a moving mesh DGSEM to solve non-linear conservation laws has been constructed and analyzed. The semi-discrete method is provably entropy stable and the free stream preservation property is satisfied for each explicit s -stage Runge–Kutta method.

The moving mesh DGSEM has been presented for three dimensional conservation laws. The derivatives in space are approximated with high-order derivative matrices which are SBP operators. Furthermore, the split form DG framework [23,24] has been used to avoid aliasing in the discretization of the volume integrals. In addition, two-point flux functions with the generalized entropy condition (2.67) are used in the split form DG framework. These modules in the spatial discretization are the basis to prove that the moving mesh DGSEM is an entropy stable scheme, when periodic boundary conditions are used. Non-periodic boundary conditions require the construction of suitable dissipative boundary conditions to enforce the entropy stability. The discrete entropy analysis requires the assumptions that the

time derivatives can be evaluated exactly and that properties like positivity preservation (of the water height, density or pressure) are satisfied on the discrete level. Operations like the integration-by-parts formula are mimicked by the SPB operators on the discrete level.

The three dimensional Euler equations have been considered to verify the proven properties of the moving mesh DGSEM in our numerical experiments. We presented convergence tests for smooth test problems to verify that the split form DG framework provides also on a moving mesh a high-order accurate approximation. Furthermore, the numerical robustness tests in the Sect. 3.3 emphasize the relevance of the entropy stable DGSEM, since the method was able to run the challenging inviscid TGV test case until final time.

Acknowledgements Open Access funding provided by Projekt DEAL. Gero Schnücke and Gregor Gassner are supported by the European Research Council (ERC) under the European Union’s Eight Framework Program Horizon 2020 with the research project Extreme, ERC Grant Agreement No. 714487. The authors gratefully acknowledge the support and the computing time on “Hazel Hen” provided by the HLRS through the project “hpcdg”.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

A Proof for Lemma 2.1

In this section, we prove Lemma 2.1. For the sake of simplicity we present merely the proof in two dimensions. The three dimensional proof can be done by the same argumentation.

A way to compute the two dimensional bijective isoparametric transfinite mapping is given in the book of Kopriva [36, Chapter 6, equation (6.18)]. The mapping is for all $[\xi^1, \xi^2]^T \in E$ and $\tau \in [0, T]$ given by

$$\begin{aligned}
 e_\kappa(t) \ni \vec{x}(t) = \vec{\chi}(\xi^1, \xi^2, \tau) = & \frac{1}{2} \left[(1 - \xi^1) \mathbb{I}^N(\vec{\Gamma}_4)(\xi^2, \tau) + (1 + \xi^1) \mathbb{I}^N(\vec{\Gamma}_2)(\xi^2, \tau) \right] \\
 & + \frac{1}{2} \left[(1 - \xi^2) \mathbb{I}^N(\vec{\Gamma}_1)(\xi^1, \tau) + (1 + \xi^2) \mathbb{I}^N(\vec{\Gamma}_3)(\xi^1, \tau) \right] \\
 & - \frac{1}{4} \left[(1 + \xi^1) \left\{ (1 - \xi^2) \mathbb{I}^N(\vec{\Gamma}_1)(1, \tau) + (1 + \xi^2) \mathbb{I}^N(\vec{\Gamma}_3)(1, \tau) \right\} \right. \\
 & \left. - \frac{1}{4} \left[(1 - \xi^1) \left\{ (1 - \xi^2) \mathbb{I}^N(\vec{\Gamma}_1)(-1, \tau) + (1 + \xi^2) \mathbb{I}^N(\vec{\Gamma}_3)(-1, \tau) \right\} \right] \right].
 \end{aligned}
 \tag{A.1}$$

It is worth to mention that the mapping $\vec{\chi}(\xi^1, \xi^2, \tau)$ matches with the boundary faces in the interpolation points. The location of the curved faces $\vec{\Gamma}_1(\xi^1, \tau)$, $\vec{\Gamma}_2(\xi^2, \tau)$, $\vec{\Gamma}_3(\xi^1, \tau)$ and $\vec{\Gamma}_4(\xi^2, \tau)$ is sketched in Fig. 5.

In the following, $e_1(t)$ and $e_2(t)$ are two neighboring elements which share the same boundary face. Without loss of generality the elements share the face $\vec{\Gamma}_3^1 = \vec{\Gamma}_2^2$ as it is illustrated in Fig. 6. Then, for the elements $e_l(t)$, $l = 1, 2$, the grid velocity field is given by

$$\vec{v}^l(\xi^1, \xi^2, \tau) = \frac{1}{2} \left[(1 - \xi^1) \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \vec{\Gamma}_4^l \right) (\xi^2, \tau) + (1 + \xi^1) \mathbb{I}^N \left(\frac{d}{dt} \vec{\Gamma}_2^l \right) (\xi^2, \tau) \right]$$

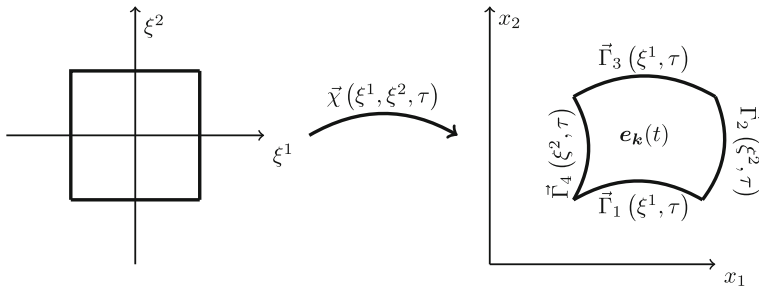
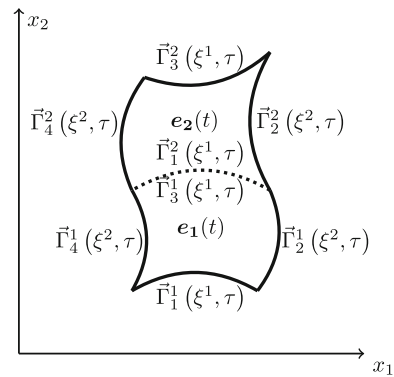


Fig. 5 Left the reference element $E = [-1, 1]^2$ and on the right a general quadrilateral element $e_k(t)$ with the curved faces $\bar{\Gamma}_1(\xi^1, \tau)$, $\bar{\Gamma}_2(\xi^2, \tau)$, $\bar{\Gamma}_3(\xi^1, \tau)$ and $\bar{\Gamma}_4(\xi^2, \tau)$. The mapping $\bar{\chi}(\xi^1, \xi^2, \tau)$ connects E and $e_k(t)$

Fig. 6 Two elements $e_1(t)$ and $e_2(t)$ of a conforming mesh sharing the same curved boundary (dotted curve)



$$\begin{aligned}
 & + \frac{1}{2} \left[(1 - \xi^2) \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \bar{\Gamma}_1^l \right) (\xi^1, \tau) + (1 + \xi^2) \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \bar{\Gamma}_3^l \right) (\xi^1, \tau) \right] \\
 & - \frac{1}{4} \left[(1 + \xi^1) \left\{ (1 - \xi^2) \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \bar{\Gamma}_1^l \right) (1, \tau) \right. \right. \\
 & \quad \left. \left. + (1 + \xi^2) \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \bar{\Gamma}_3^l \right) (1, \tau) \right\} \right] \\
 & - \frac{1}{4} \left[(1 - \xi^1) \left\{ (1 - \xi^2) \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \bar{\Gamma}_1^l \right) (-1, \tau) \right. \right. \\
 & \quad \left. \left. + (1 + \xi^2) \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \bar{\Gamma}_3^l \right) (-1, \tau) \right\} \right], \tag{A.2}
 \end{aligned}$$

since it holds the identity

$$\frac{\partial}{\partial \tau} \mathbb{I}^N(\bar{\Gamma}_i^l) = \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \bar{\Gamma}_i^l \right), \quad l = 1, 2, \quad \text{and} \quad i = 1, 2, 3, 4. \tag{A.3}$$

Furthermore, since for $l = 1, 2$, and $i = 1, 2, 3, 4$, the faces $\bar{\Gamma}_i^l(\cdot, \cdot, t)$ are continuously differentiable in the time interval $[0, T]$, it holds

$$\mathbb{I}^N \left(\frac{\partial}{\partial \tau} \bar{\Gamma}_4^1 \right) (1, \tau) = \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \bar{\Gamma}_3^1 \right) (-1, \tau), \quad \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \bar{\Gamma}_2^1 \right) (1, \tau) = \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \bar{\Gamma}_3^1 \right) (1, \tau), \tag{A.4}$$

$$\mathbb{I}^N \left(\frac{\partial}{\partial \tau} \tilde{\Gamma}_4^2 \right) (-1, \tau) = \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \tilde{\Gamma}_1^2 \right) (-1, \tau), \quad \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \tilde{\Gamma}_2^2 \right) (-1, \tau) = \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \tilde{\Gamma}_1^2 \right) (1, \tau), \tag{A.5}$$

$$\frac{\partial}{\partial \tau} \tilde{\Gamma}_3^1 (\zeta_j, \tau) = \frac{d}{dt} \tilde{\Gamma}_1^2 (\zeta_j, \tau), \quad j = 0, \dots, N, \tag{A.6}$$

where $\{\zeta_j\}_{j=0}^N$ are interpolation points.

For the element $e_1(t)$ the points along the interface with $e_2(t)$ are mapped in the set $\{(\xi, 1) : \xi \in [-1, 1]\}$. Hence, the grid velocity becomes

$$\bar{v}^1 (\xi, 1, \tau) = \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \tilde{\Gamma}_3^1 \right) (\xi, \tau), \quad \forall \xi \in [-1, 1] \tag{A.7}$$

by (A.4). On the opposite, for the element $e_2(t)$ the points along the interface with $e_1(t)$ are mapped in the set $\{(\xi, -1) : \xi \in [-1, 1]\}$ and we obtain

$$\bar{v}^2 (\xi, -1, \tau) = \mathbb{I}^N \left(\frac{\partial}{\partial \tau} \tilde{\Gamma}_1^2 \right) (\xi, \tau), \quad \forall \xi \in [-1, 1], \tag{A.8}$$

by (A.5). Thus, we obtain $\bar{v}^1 (\cdot, 1, \tau) = \bar{v}^2 (\cdot, -1, \tau)$ by (A.6). This proves that the grid velocity is continuous in the interface points of the two neighboring elements.

B Entropy Stable Moving Mesh Euler Fluxes

We present entropy stable Cartesian fluxes $\mathbf{G}_l^{\text{EC}}, l = 1, 2, 3$, for the compressible Euler equations (3.1) equipped with the entropy/entropy flux pairs (3.8). Then the entropy variables are given by

$$\mathbf{w} = \left[\frac{\gamma - \xi}{\gamma - 1} - \beta |\bar{u}|^2, 2\beta u_1, 2\beta u_2, 2\beta u_3, -2\beta \right]^T, \quad \text{with } \beta := \frac{\rho}{2p} \tag{B.1}$$

and the entropy functionals are given by

$$\phi = \mathbf{w}^T \mathbf{u} - s = \rho, \quad \psi_l = \mathbf{w}^T \mathbf{f}_l - f_l^s = \rho u_l, \quad l = 1, 2, 3. \tag{B.2}$$

B.1 Entropy Conservative Euler Flux Based on the Flux in [8]

In the following the logarithmic mean $\{\{\cdot\}\}^{\log}$ will be used. For two positive states a^- and a^+ , the logarithmic mean is defined by

$$\{\{a\}\}^{\log} := \begin{cases} \frac{\llbracket a \rrbracket}{\llbracket \log(a) \rrbracket}, & \text{if } a^- \neq a^+, \\ a^-, & \text{if } a^- = a^+. \end{cases} \tag{B.3}$$

A numerically stable procedure to compute the logarithmic mean (B.3) is provided by Ismail and Roe [33, Appendix B]. Friedrich et al. [21, Theorem 3] constructed the following state function

$$\mathbf{U}^\# = \begin{bmatrix} \{\{\rho\}\}^{\log} \\ \{\{\rho\}\}^{\log} \{\{u_1\}\} \\ \{\{\rho\}\}^{\log} \{\{u_2\}\} \\ \{\{\rho\}\}^{\log} \{\{u_3\}\} \\ \frac{\{\{\rho\}\}^{\log}}{2(\gamma-1)\{\{\beta\}\}^{\log}} + \frac{1}{2} \{\{\rho\}\}^{\log} \overline{|\bar{u}|^2} \end{bmatrix}, \tag{B.4}$$

where

$$|\vec{u}|^2 = 2 (\{u_1\}^2 + \{u_2\}^2 + \{u_3\}^2) - (\{u_1^2\} + \{u_2^2\} + \{u_3^2\}). \tag{B.5}$$

The state function (B.4) is consistent, symmetric and it holds

$$\{\mathbf{w}\}^T \mathbf{U}^\# = \{\rho\} = \{\phi\}. \tag{B.6}$$

In [40, Eg. (3.4) p. 1974] LeFloch et al. gave a general formula to compute entropy conservative numerical state flux-functions. Furthermore, Chandrashekar constructed in [8] a kinetic energy preserving and entropy conservative (KEPEC) numerical flux function for the compressible Euler equations. In the x_1 -direction Chandrashekar’s KEPEC flux is given by

$$\mathbf{F}_1^{\text{EC-CH}} = \begin{bmatrix} \{\rho\}^{\log\{u_1\}} \\ \{\rho\}^{\log\{u_1\}} \{u_1\}^2 + \frac{\{\rho\}}{2\{\beta\}} \\ \{\rho\}^{\log\{u_1\}} \{u_2\} \\ \{\rho\}^{\log\{u_1\}} \{u_3\} \\ \{\rho\}^{\log\{u_1\}} \left(\frac{1}{2(\gamma-1)\{\beta\}^{\log}} + \frac{1}{2} |\vec{u}|^2 \right) + \frac{\{\rho\}\{u_1\}}{2\{\beta\}} \end{bmatrix}. \tag{B.7}$$

The flux (B.7) is consistent with \mathbf{f}_1 and symmetric. In particular, Chandrashekar proved that

$$\{\mathbf{w}\}^T \mathbf{F}_1^{\text{EC-CH}} = \{\rho u_1\} = \{\psi_1\}. \tag{B.8}$$

The state function (B.4) and the flux function (B.7) are used to construct the flux

$$\mathbf{G}_1^{\text{EC-CH}} = \mathbf{F}_1^{\text{EC-CH}} - \{v_1\} \mathbf{U}^\# = \begin{bmatrix} \{\rho\}^{\log\{u_1 - v_1\}} \\ \{\rho\}^{\log\{u_1 - v_1\}} \{u_1\} + \frac{\{\rho\}}{2\{\beta\}} \\ \{\rho\}^{\log\{u_1 - v_1\}} \{u_2\} \\ \{\rho\}^{\log\{u_1 - v_1\}} \{u_3\} \\ \{\rho\}^{\log\{u_1 - v_1\}} \left(\frac{1}{2(\gamma-1)\{\beta\}^{\log}} + \frac{1}{2} |\vec{u}|^2 \right) + \frac{\{\rho\}\{u_1\}}{2\{\beta\}} \end{bmatrix}. \tag{B.9}$$

The flux (B.9) is consistent with $\mathbf{g}_1 = \mathbf{f}_1 - v_1 \mathbf{u}$, symmetric and it follows

$$\{\mathbf{w}\}^T \mathbf{G}_1^{\text{EC-CH}} = \{\mathbf{w}\}^T \mathbf{F}_1^{\text{EC-CH}} - \{v_1\} \{\mathbf{w}\}^T \mathbf{U}^\# = \{\psi_1\} - \{v_1\} \{\phi\} \tag{B.10}$$

by (B.6) and (B.8). In the same way, the x_2 -direction and the x_3 -direction of Chandrashekar’s KEPEC flux can be used to construct

$$\mathbf{G}_2^{\text{EC-CH}} = \begin{bmatrix} \{\rho\}^{\log\{u_2 - v_2\}} \\ \{\rho\}^{\log\{u_1\}} \{u_2 - v_2\} \\ \{\rho\}^{\log\{u_2\}} \{u_2 - v_2\} + \frac{\{\rho\}}{2\{\beta\}} \\ \{\rho\}^{\log\{u_2 - v_2\}} \{u_3\} \\ \{\rho\}^{\log\{u_2 - v_2\}} \left(\frac{1}{2(\gamma-1)\{\beta\}^{\log}} + \frac{1}{2} |\vec{u}|^2 \right) + \frac{\{\rho\}\{u_2\}}{2\{\beta\}} \end{bmatrix} \tag{B.11}$$

and

$$\mathbf{G}_3^{\text{EC-CH}} = \begin{bmatrix} \{\rho\}^{\log\{u_3 - v_3\}} \\ \{\rho\}^{\log\{u_1\}} \{u_3 - v_3\} \\ \{\rho\}^{\log\{u_2\}} \{u_3 - v_3\} \\ \{\rho\}^{\log\{u_3\}} \{u_3 - v_3\} + \frac{\{\rho\}}{2\{\beta\}} \\ \{\rho\}^{\log\{u_3 - v_3\}} \left(\frac{1}{2(\gamma-1)\{\beta\}^{\log}} + \frac{1}{2} |\vec{u}|^2 \right) + \frac{\{\rho\}\{u_3\}}{2\{\beta\}} \end{bmatrix}. \tag{B.12}$$

The fluxes (B.11), (B.12) are consistent with $\mathbf{g}_2 = \mathbf{f}_2 - \nu_2 \mathbf{u}$, $\mathbf{g}_3 = \mathbf{f}_3 - \nu_3 \mathbf{u}$, symmetric and satisfy

$$\llbracket \mathbf{w} \rrbracket^T \mathbf{G}_2^{\text{EC-CH}} = \llbracket \mathbf{w} \rrbracket^T \mathbf{F}_2^{\text{EC-CH}} - \{\nu_2\} \llbracket \mathbf{w} \rrbracket^T \mathbf{U}^\# = \llbracket \psi_2 \rrbracket - \{\nu_2\} \llbracket \phi \rrbracket, \tag{B.13}$$

$$\llbracket \mathbf{w} \rrbracket^T \mathbf{G}_3^{\text{EC-CH}} = \llbracket \mathbf{w} \rrbracket^T \mathbf{F}_3^{\text{EC-CH}} - \{\nu_3\} \llbracket \mathbf{w} \rrbracket^T \mathbf{U}^\# = \llbracket \psi_3 \rrbracket - \{\nu_3\} \llbracket \phi \rrbracket. \tag{B.14}$$

B.2 Entropy Conservative Euler Flux Based on the Flux in [55]

Ranocha constructed in his PHD thesis [55] another KEPEC numerical flux for the compressible Euler equations. We proceed as in the Appendix B.1 and use the state (B.4) and Ranocha’s flux to construct the following two-point flux functions

$$\mathbf{G}_1^{\text{EC-R}} = \begin{bmatrix} \{\rho\}^{\log\{u_1 - v_1\}} \\ \{\rho\}^{\log\{u_1 - v_1\}} \{u_1\} + \{p\} \\ \{\rho\}^{\log\{u_1 - v_1\}} \{u_2\} \\ \{\rho\}^{\log\{u_1 - v_1\}} \{u_3\} \\ \{\rho\}^{\log\{u_1 - v_1\}} \left(\frac{1}{2(\gamma-1)\{\beta\}^{\log}} + \frac{1}{2} |\bar{u}|^2 \right) + 2\{p\} \{u_1\} - \{p u_1\} \end{bmatrix}, \tag{B.15a}$$

$$\mathbf{G}_2^{\text{EC-R}} = \begin{bmatrix} \{\rho\}^{\log\{u_2 - v_2\}} \\ \{\rho\}^{\log\{u_1\}} \{u_2 - v_2\} \\ \{\rho\}^{\log\{u_2\}} \{u_2 - v_2\} + \{p\} \\ \{\rho\}^{\log\{u_2 - v_2\}} \{u_3\} \\ \{\rho\}^{\log\{u_2 - v_2\}} \left(\frac{1}{2(\gamma-1)\{\beta\}^{\log}} + \frac{1}{2} |\bar{u}|^2 \right) + 2\{p\} \{u_2\} - \{p u_2\} \end{bmatrix}, \tag{B.15b}$$

and

$$\mathbf{G}_3^{\text{EC-R}} = \begin{bmatrix} \{\rho\}^{\log\{u_3 - v_3\}} \\ \{\rho\}^{\log\{u_1\}} \{u_3 - v_3\} \\ \{\rho\}^{\log\{u_2\}} \{u_3 - v_3\} \\ \{\rho\}^{\log\{u_3\}} \{u_3 - v_3\} + \{p\} \\ \{\rho\}^{\log\{u_3 - v_3\}} \left(\frac{1}{2(\gamma-1)\{\beta\}^{\log}} + \frac{1}{2} |\bar{u}|^2 \right) + 2\{p\} \{u_3\} - \{p u_3\} \end{bmatrix}. \tag{B.15c}$$

The flux functions (B.15) are consistent with \mathbf{g}_1 , \mathbf{g}_2 , \mathbf{g}_3 , symmetric and satisfy

$$\llbracket \mathbf{w} \rrbracket^T \mathbf{G}_l^{\text{EC-R}} = \llbracket \psi_l \rrbracket - \{\nu_l\} \llbracket \phi \rrbracket, \quad l = 1, 2, 3. \tag{B.16}$$

B.3 Matrix Dissipation Term for the Euler Flux

We will use the Euler fluxes from the previous Appendices B.1 and B.2 with the dissipation operators from Winters et al. [63]. In the following, the matrices to construct the entropy stable dissipation operators (2.96) are listed. The average components of the dissipation term in the x_1 -direction are given by

$$\mathcal{R}_1^* = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ \{u_1\} - \bar{c} & \{u_1\} & 0 & 0 & \{u_1\} + \bar{c} \\ \{u_2\} & \{u_2\} & 1 & 0 & \{u_2\} \\ \{u_3\} & \{u_3\} & 0 & 1 & \{u_3\} \\ \bar{h} - \{u_1\} \bar{c} & \frac{1}{2} |\bar{u}|^2 & \{u_2\} & \{u_3\} & \bar{h} + \{u_1\} \bar{c} \end{bmatrix}, \tag{B.17}$$

$$\mathcal{T}_1^* = \text{diag} \left(\sqrt{\frac{\{\{\rho\}\}^{\log}}{2\gamma}}, \sqrt{\frac{(\gamma - 1)}{\gamma} \{\{\rho\}\}^{\log}}, \sqrt{\frac{\{\{\rho\}\}}{2\{\{\beta\}\}}}, \sqrt{\frac{\{\{\rho\}\}}{2\{\{\beta\}\}}}, \sqrt{\frac{\{\{\rho\}\}^{\log}}{2\gamma}} \right), \tag{B.18}$$

$$\Lambda_1 = \text{diag} (|\{\{u_1 - v_1\}\} - \bar{c}|, |\{\{u_1 - v_1\}\}|, |\{\{u_1 - v_1\}\}|, |\{\{u_1 - v_1\}\}|, |\{\{u_1 - v_1\}\} + \bar{c}|), \tag{B.19}$$

where

$$\bar{c} := \sqrt{\frac{\gamma \{\{\rho\}\}}{2\{\{\rho\}\}^{\log} \{\{\beta\}\}}}, \quad \bar{h} := \frac{\gamma}{2(\gamma - 1) \{\{\beta\}\}^{\log}} + \frac{1}{2} |\bar{u}|^2. \tag{B.20}$$

In the x_2 -direction the components are given by

$$\mathcal{R}_2^* = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ \{\{u_1\}\} & 1 & \{\{u_1\}\} & 0 & \{\{u_1\}\} \\ \{\{u_2\}\} - \bar{c} & 0 & \{\{u_2\}\} & 0 & \{\{u_2\}\} + \bar{c} \\ \{\{u_3\}\} & 0 & \{\{u_3\}\} & 1 & \{\{u_3\}\} \\ \bar{h} - \{\{u_2\}\} \bar{c} & \{\{u_1\}\} & \frac{1}{2} |\bar{u}|^2 & \{\{u_3\}\} & \bar{h} + \{\{u_2\}\} \bar{c} \end{bmatrix}, \tag{B.21}$$

$$\mathcal{T}_2^* = \text{diag} \left(\sqrt{\frac{\{\{\rho\}\}^{\log}}{2\gamma}}, \sqrt{\frac{\{\{\rho\}\}}{2\{\{\beta\}\}}}, \sqrt{\frac{(\gamma - 1)}{\gamma} \{\{\rho\}\}^{\log}}, \sqrt{\frac{\{\{\rho\}\}}{2\{\{\beta\}\}}}, \sqrt{\frac{\{\{\rho\}\}^{\log}}{2\gamma}} \right), \tag{B.22}$$

$$\Lambda_2 = \text{diag} (|\{\{u_2 - v_2\}\} - \bar{c}|, |\{\{u_2 - v_2\}\}|, |\{\{u_2 - v_2\}\}|, |\{\{u_2 - v_2\}\}|, |\{\{u_2 - v_2\}\} + \bar{c}|), \tag{B.23}$$

and in the x_3 -direction the components are given by

$$\mathcal{R}_3^* = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ \{\{u_1\}\} & 1 & 0 & \{\{u_1\}\} & \{\{u_1\}\} \\ \{\{u_2\}\} & 0 & 1 & \{\{u_2\}\} & \{\{u_2\}\} \\ \{\{u_3\}\} - \bar{c} & 0 & 0 & \{\{u_3\}\} & \{\{u_3\}\} + \bar{c} \\ \bar{h} - \{\{u_3\}\} \bar{c} & \{\{u_1\}\} & \{\{u_2\}\} & \frac{1}{2} |\bar{u}|^2 & \bar{h} + \{\{u_3\}\} \bar{c} \end{bmatrix}, \tag{B.24}$$

$$\mathcal{T}_3^* = \text{diag} \left(\sqrt{\frac{\{\{\rho\}\}^{\log}}{2\gamma}}, \sqrt{\frac{\{\{\rho\}\}}{2\{\{\beta\}\}}}, \sqrt{\frac{\{\{\rho\}\}}{2\{\{\beta\}\}}}, \sqrt{\frac{(\gamma - 1)}{\gamma} \{\{\rho\}\}^{\log}}, \sqrt{\frac{\{\{\rho\}\}^{\log}}{2\gamma}} \right), \tag{B.25}$$

$$\Lambda_3 = \text{diag} (|\{\{u_3 - v_3\}\} - \bar{c}|, |\{\{u_3 - v_3\}\}|, |\{\{u_3 - v_3\}\}|, |\{\{u_3 - v_3\}\}|, |\{\{u_3 - v_3\}\} + \bar{c}|). \tag{B.26}$$

C Proofs of Entropy Conservation for Advection Terms

In this section, we apply the following identities which result from the properties of the SBP operator \mathcal{Q}

$$\sum_{i,j=0}^N \mathcal{Q}_{ij} [a]_{(i,j)} [b]_{(i,j)} = \sum_{i,j=0}^N \mathcal{Q}_{ij} a_i b_j - (a_N b_N - a_0 b_0), \tag{C.1}$$

$$\sum_{i,j=0}^N \mathcal{Q}_{ij} [a]_{(i,j)} [b]_{(i,j)} [c]_{(i,j)} = \sum_{i,j=0}^N 2\mathcal{Q}_{ij} a_i [b]_{(i,j)} [c]_{(i,j)} - (a_N b_N c_N - a_0 b_0 c_0), \tag{C.2}$$

where $\{a\}_{i=0}^N$, $\{b\}_{i=0}^N$ and $\{c\}_{i=0}^N$ are generic nodal values. These identities can be proven in a similar way as the discrete split forms in Lemma 1 in [23]. Thus, we skip a proof in this paper.

C.1 Proof for Eq. (2.84)

The flux $\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}}$ satisfies the symmetry property (2.66) and the SBP property (2.59) provides

$$2\omega_i \mathcal{D}_{im} = 2\mathcal{Q}_{im} = \mathcal{Q}_{im} - \mathcal{Q}_{mi} + \mathcal{B}_{im}, \quad i, m = 0, \dots, N. \tag{C.3}$$

Thus, we obtain

$$\begin{aligned} & \left\langle \overset{\leftrightarrow}{\mathbb{D}}^N \cdot \overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}}, \mathbf{w} \right\rangle_N \\ &= \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m,k=0}^N \mathcal{Q}_{im} \mathbb{W}_{(i,m)j}^T \left(\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk}) \cdot \{\{\mathbb{J}\bar{a}^1\}\}_{(i,m)jk} \right) \\ &+ \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m=0}^N \mathcal{B}_{im} \mathbf{W}_{ijk}^T \left(\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk}) \cdot \{\{\mathbb{J}\bar{a}^1\}\}_{(i,m)jk} \right) \\ &+ \sum_{i,k=0}^N \omega_i \omega_k \sum_{j,m=0}^N \mathcal{Q}_{jm} \mathbb{W}_{i(j,m),k}^T \left(\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{imk}, \mathbf{U}_{ijk}, \mathbf{U}_{imk}) \cdot \{\{\mathbb{J}\bar{a}^2\}\}_{i(j,m)k} \right) \\ &+ \sum_{i,k=0}^N \omega_i \omega_k \sum_{j,m=0}^N \mathcal{B}_{jm} \mathbf{W}_{ijk}^T \left(\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{imk}, \mathbf{U}_{ijk}, \mathbf{U}_{imk}) \cdot \{\{\mathbb{J}\bar{a}^2\}\}_{i(j,m)k} \right) \\ &+ \sum_{i,j=0}^N \omega_i \omega_j \sum_{j,m=0}^N \mathcal{Q}_{km} \mathbb{W}_{ij(k,m)}^T \left(\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{ijm}, \mathbf{U}_{ijk}, \mathbf{U}_{ijm}) \cdot \{\{\mathbb{J}\bar{a}^3\}\}_{ij(k,m)} \right) \\ &+ \sum_{i,j=0}^N \omega_i \omega_j \sum_{j,m=0}^N \mathcal{B}_{km} \mathbf{W}_{ijk}^T \left(\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{ijm}, \mathbf{U}_{ijk}, \mathbf{U}_{ijm}) \cdot \{\{\mathbb{J}\bar{a}^3\}\}_{ij(k,m)} \right) \end{aligned} \tag{C.4}$$

by the same calculation as in [21, Appendix C.1., Equations (C.4) and (C.5)] or [24, Appendix B.3., Equation (B.31)]. To evaluate the fluxes $\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}}$ at the element interfaces, we apply the consistence condition (2.65), such that e.g.

$$\begin{aligned} \overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{Njk}, \vec{v}_{Njk}, \mathbf{U}_{Njk}, \mathbf{U}_{Njk}) &= \overset{\leftrightarrow}{\mathbf{F}} (\mathbf{U}_{Njk}) - \{\{\vec{v}\}\}_{(N,N)jk} \mathbf{U}_{Njk} = \overset{\leftrightarrow}{\mathbf{G}}_{Njk}, \\ j, k &= 0, \dots, N. \end{aligned} \tag{C.5}$$

This provides the identity

$$\begin{aligned} & \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m=0}^N \mathcal{B}_{im} \mathbf{W}_{ijk}^T \left(\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk}) \cdot \{\{\mathbb{J}\bar{a}^1\}\}_{(i,m)jk} \right) \\ &+ \sum_{i,k=0}^N \omega_i \omega_k \sum_{j,m=0}^N \mathcal{B}_{jm} \mathbf{W}_{ijk}^T \left(\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{imk}, \mathbf{U}_{ijk}, \mathbf{U}_{imk}) \cdot \{\{\mathbb{J}\bar{a}^2\}\}_{i(j,m)k} \right) \end{aligned}$$

$$\begin{aligned}
 & + \sum_{i,j=0}^N \omega_i \omega_j \sum_{k,m=0}^N \mathcal{B}_{km} \mathbf{W}_{ijk}^T \left(\overleftrightarrow{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{ijm}, \mathbf{U}_{ijk}, \mathbf{U}_{ijm}) \cdot \{\{\mathbb{J}\vec{a}^3\}\}_{ij(k,m)} \right) \\
 & = \sum_{j,k=0}^N \omega_j \omega_k \left[\mathbf{W}_{Njk}^T \left(\overleftrightarrow{\mathbf{G}}_{Njk} \cdot (\mathbb{J}\vec{a}^1)_{Njk} \right) - \mathbf{W}_{0jk}^T \left(\overleftrightarrow{\mathbf{G}}_{0jk} \cdot (\mathbb{J}\vec{a}^1)_{0jk} \right) \right] \\
 & + \sum_{i,k=0}^N \omega_i \omega_k \left[\mathbf{W}_{iNk}^T \left(\overleftrightarrow{\mathbf{G}}_{iNk} \cdot (\mathbb{J}\vec{a}^2)_{iNk} \right) - \mathbf{W}_{i0k}^T \left(\overleftrightarrow{\mathbf{G}}_{i0k} \cdot (\mathbb{J}\vec{a}^2)_{i0k} \right) \right] \\
 & + \sum_{i,j=0}^N \omega_i \omega_j \left[\mathbf{W}_{ijN}^T \left(\overleftrightarrow{\mathbf{G}}_{ijN} \cdot (\mathbb{J}\vec{a}^3)_{ijN} \right) - \mathbf{W}_{ij0}^T \left(\overleftrightarrow{\mathbf{G}}_{ij0} \cdot (\mathbb{J}\vec{a}^3)_{ij0} \right) \right] \\
 & = \int_{\partial E, N} \mathbf{W}^T \left\{ \overleftrightarrow{\mathbf{G}} \cdot \hat{n} \right\} dS. \tag{C.6}
 \end{aligned}$$

Next, we investigate the first sum on the right hand side in the Eq.(C.4). Since the fluxes $\mathbf{G}_l^{\text{EC}}, l = 1, 2, 3$, satisfy the entropy condition (2.67), follows

$$\begin{aligned}
 & \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m=0}^N \mathcal{Q}_{im} \llbracket \mathbf{W} \rrbracket_{(i,m)jk}^T \left(\overleftrightarrow{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk}) \cdot \{\{\mathbb{J}\vec{a}^1\}\}_{(i,m)jk} \right) \\
 & = \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m=0}^N \mathcal{Q}_{im} \llbracket \mathbf{W} \rrbracket_{(i,m)jk}^T \mathbf{G}_1^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk}) \{\{\mathbb{J}a_1^1\}\}_{(i,m)jk} \\
 & + \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m=0}^N \mathcal{Q}_{im} \llbracket \mathbf{W} \rrbracket_{(i,m)jk}^T \mathbf{G}_2^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk}) \{\{\mathbb{J}a_2^1\}\}_{(i,m)jk} \\
 & + \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m=0}^N \mathcal{Q}_{im} \llbracket \mathbf{W} \rrbracket_{(i,m)jk}^T \mathbf{G}_3^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk}) \{\{\mathbb{J}a_3^1\}\}_{(i,m)jk} \tag{C.7} \\
 & = \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m=0}^N \mathcal{Q}_{im} (\llbracket \Psi_1 \rrbracket_{(i,m)jk} - \{v_1\}_{(i,m)jk} \llbracket \Phi \rrbracket_{(i,m)jk}) \{\{\mathbb{J}a_1^1\}\}_{(i,m)jk} \\
 & + \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m=0}^N \mathcal{Q}_{im} (\llbracket \Psi_2 \rrbracket_{(i,m)jk} - \{v_2\}_{(i,m)jk} \llbracket \Phi \rrbracket_{(i,m)jk}) \{\{\mathbb{J}a_2^1\}\}_{(i,m)jk} \\
 & + \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m=0}^N \mathcal{Q}_{im} (\llbracket \Psi_3 \rrbracket_{(i,m)jk} - \{v_3\}_{(i,m)jk} \llbracket \Phi \rrbracket_{(i,m)jk}) \{\{\mathbb{J}a_3^1\}\}_{(i,m)jk}.
 \end{aligned}$$

For $l = 1, 2, 3$, the SPB properties (C.1) and (C.2) provide

$$\begin{aligned}
 & \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m=0}^N \mathcal{Q}_{im} (\llbracket \Psi_l \rrbracket_{(i,m)jk} - \{v_l\}_{(i,m)jk} \llbracket \Phi \rrbracket_{(i,m)jk}) \{\{\mathbb{J}a_l^1\}\}_{(i,m)jk} \\
 & = - \sum_{j,k=0}^N \omega_j \omega_k \left[((\Psi_l)_{Njk} - (v_l)_{Njk} \Phi_{Njk}) (\mathbb{J}a_l^1)_{Njk} \right]
 \end{aligned}$$

$$\begin{aligned}
 & - \left((\Psi_l)_{0jk} - (v_l)_{0jk} \Phi_{0jk} \right) (\mathbb{J}a_l^1)_{0jk} \Big] \\
 & + \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k (\Psi_l)_{ijk} \sum_{m=0}^N \mathcal{D}_{im} (\mathbb{J}a_l^1)_{mjk} \\
 & - \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k \Phi_{ijk} \sum_{m=0}^N 2 \mathcal{D}_{im} \{ \{v_l\} \}_{(i,m)jk} \{ \{ \mathbb{J}a_l^1 \} \}_{(i,m)jk} .
 \end{aligned} \tag{C.8}$$

Hence, we obtain the identity

$$\begin{aligned}
 & \sum_{j,k=0}^N \omega_j \omega_k \sum_{i,m=0}^N \mathcal{Q}_{im} \llbracket \mathbf{W} \rrbracket_{(i,m)jk}^T \left(\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{mjk}, \mathbf{U}_{ijk}, \mathbf{U}_{mjk}) \cdot \{ \{ \mathbb{J}a^1 \} \}_{(i,m)jk} \right) \\
 & = - \sum_{j,k=0}^N \omega_j \omega_k \left[\left((\tilde{\Psi})_{Njk} - (\vec{v})_{Njk} \Phi_{Njk} \right) \cdot (\mathbb{J}a^1)_{Njk} - \left((\tilde{\Psi})_{0jk} \right. \right. \\
 & \quad \left. \left. - (\vec{v})_{0jk} \Phi_{0jk} \right) (\mathbb{J}a^1)_{0jk} \right] \\
 & + \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k (\tilde{\Psi})_{ijk} \cdot \left(\sum_{m=0}^N \mathcal{D}_{im} (\mathbb{J}a^1)_{mjk} \right) \\
 & - \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k \Phi_{ijk} \sum_{m=0}^N 2 \mathcal{D}_{im} \{ \{ \vec{v} \} \}_{(i,m)jk} \cdot \{ \{ \mathbb{J}a^1 \} \}_{(i,m)jk} .
 \end{aligned} \tag{C.9}$$

By the same computation, the third sum on the right hand side in the Eq. (C.4) becomes

$$\begin{aligned}
 & \sum_{i,k=0}^N \omega_i \omega_k \sum_{j,m=0}^N \mathcal{Q}_{jm} \llbracket \mathbf{W} \rrbracket_{i(j,m)k}^T \left(\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{imk}, \mathbf{U}_{ijk}, \mathbf{U}_{imk}) \cdot \{ \{ \mathbb{J}a^2 \} \}_{i(j,m)k} \right) \\
 & = - \sum_{i,k=0}^N \omega_i \omega_k \left[\left((\tilde{\Psi})_{iNk} - (\vec{v})_{iNk} \Phi_{iNk} \right) \cdot (\mathbb{J}a^2)_{iNk} - \left((\tilde{\Psi})_{i0k} \right. \right. \\
 & \quad \left. \left. - (\vec{v})_{i0k} \Phi_{i0k} \right) (\mathbb{J}a^2)_{i0k} \right] \\
 & + \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k (\tilde{\Psi})_{ijk} \cdot \left(\sum_{m=0}^N \mathcal{D}_{jm} (\mathbb{J}a^2)_{imk} \right) \\
 & - \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k \Phi_{ijk} \sum_{m=0}^N 2 \mathcal{D}_{jm} \{ \{ \vec{v} \} \}_{i(j,m)k} \cdot \{ \{ \mathbb{J}a^2 \} \}_{i(j,m)k}
 \end{aligned} \tag{C.10}$$

and the sum next-to-last on the right hand side in the Eq. (C.4) becomes

$$\begin{aligned}
 & \sum_{i,j=0}^N \omega_i \omega_j \sum_{k,m=0}^N \mathcal{Q}_{km} \llbracket \mathbf{W} \rrbracket_{ij(k,m)}^T \left(\overset{\leftrightarrow}{\mathbf{G}}^{\text{EC}} (\vec{v}_{ijk}, \vec{v}_{ijm}, \mathbf{U}_{ijk}, \mathbf{U}_{ijm}) \cdot \{ \{ \mathbb{J}a^3 \} \}_{ij(k,m)} \right) \\
 & = - \sum_{i,j=0}^N \omega_i \omega_j \left[\left((\tilde{\Psi})_{ijN} - (\vec{v})_{ijN} \Phi_{ijN} \right) \cdot (\mathbb{J}a^3)_{ijN} - \left((\tilde{\Psi})_{ij0} \right. \right. \\
 & \quad \left. \left. - (\vec{v})_{ij0} \Phi_{ij0} \right) (\mathbb{J}a^3)_{ij0} \right]
 \end{aligned}$$

$$\begin{aligned}
 & + \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k (\vec{\Psi})_{ijk} \cdot \left(\sum_{m=0}^N \mathcal{D}_{km} (\mathbb{J}\vec{a}^3)_{ijm} \right) \\
 & - \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k \Phi_{ijk} \sum_{m=0}^N 2 \mathcal{D}_{km} \{\{\vec{v}\}\}_{ij(k,m)} \cdot \{\{\mathbb{J}\vec{a}^3\}\}_{ij(k,m)}. \tag{C.11}
 \end{aligned}$$

The definition of the derivative projection operator (2.62) in the Eq.(2.71a) provides

$$\begin{aligned}
 & - \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k \Phi_{ijk} \sum_{m=0}^N 2 \mathcal{D}_{im} \{\{\vec{v}\}\}_{(i,m)jk} \cdot \{\{\mathbb{J}\vec{a}^1\}\}_{(i,m)jk} \\
 & - \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k \Phi_{ijk} \sum_{m=0}^N 2 \mathcal{D}_{jm} \{\{\vec{v}\}\}_{i(j,m)k} \cdot \{\{\mathbb{J}\vec{a}^2\}\}_{i(j,m)k} \\
 & - \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k \Phi_{ijk} \sum_{m=0}^N 2 \mathcal{D}_{km} \{\{\vec{v}\}\}_{ij(k,m)} \cdot \{\{\mathbb{J}\vec{a}^3\}\}_{ij(k,m)} \\
 & = - \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k \Phi_{ijk} (\mathbb{D}^N \cdot \vec{v}_{ijk}) = - \left\langle \mathbb{D}^N \cdot \vec{v}, \Phi \right\rangle_N. \tag{C.12}
 \end{aligned}$$

Next, we plug the Eqs.(C.6), (C.9), (C.10), (C.11) in the Eq.(C.4) and apply the identity (C.12). This results in the identity

$$\begin{aligned}
 \left\langle \mathbb{D}^N \cdot \vec{\mathbf{G}}^{\text{EC}}, \mathbf{W} \right\rangle_N & = \int_{\partial E, N} \left[\mathbf{W}^T \left\{ \vec{\mathbf{G}} \cdot \hat{n} \right\} - (\vec{\Psi} - \vec{v}\Phi) \cdot \hat{n} \right] dS - \left\langle \mathbb{D}^N \cdot \vec{v}, \Phi \right\rangle_N \\
 & + \sum_{i,j,k=0}^N \omega_i \omega_j \omega_k (\vec{\Psi})_{ijk} \cdot \left(\sum_{m=0}^N (\mathcal{D}_{im} (\mathbb{J}\vec{a}^1)_{mjk} + \mathcal{D}_{jm} (\mathbb{J}\vec{a}^2)_{imk} \right. \\
 & \left. + \mathcal{D}_{km} (\mathbb{J}\vec{a}^3)_{ijm} \right). \tag{C.13}
 \end{aligned}$$

The definition of the contravariant vector flux functions (2.23) and contravariant block vector flux functions (2.28) provide the equality

$$\begin{aligned}
 \mathbf{W}^T \left\{ \vec{\mathbf{G}} \cdot \hat{n} \right\} - (\vec{\Psi} - \vec{v}\Phi) \cdot \hat{n} & = \sum_{l,r=1}^3 \mathbb{J}a_l^r (\mathbf{W}^T \mathbf{G}_l - \mathbf{g}_l + v_l \Phi) \hat{n}^r \\
 & = \sum_{l,r=1}^3 \mathbb{J}a_l^r (F_l^s - v_l S) \hat{n}^r \\
 & = (\hat{s}\vec{n}) \cdot (\vec{F}^s - \vec{v}S) = \tilde{F}_{\hat{n}}^s - \tilde{v}_{\hat{n}}S. \tag{C.14}
 \end{aligned}$$

Therefore, the Eq.(C.13) simplifies to

$$\left\langle \mathbb{D}^N \cdot \vec{\mathbf{G}}^{\text{EC}}, \mathbf{W} \right\rangle_N = \int_{\partial E, N} (\tilde{F}_{\hat{n}}^s - \tilde{v}_{\hat{n}}S) dS - \left\langle \mathbb{D}^N \cdot \vec{v}, \Phi \right\rangle_N, \tag{C.15}$$

since the discrete volume weighted contravariant vectors $\mathbb{J}\vec{a}^\alpha$, $\alpha = 1, 2, 3$, are computed by the conservative curl form (2.52) and the discrete metric identities (2.60) are satisfied in the LGL points.

References

1. Abe, Y., Nonomura, T., Iizuka, N., Fujii, K.: Geometric interpretations and spatial symmetry property of metrics in the conservative form for high-order finite-difference schemes on moving and deforming grids. *J. Comput. Phys.* **260**, 163–203 (2014)
2. Barth, T.J.: Numerical methods for gasdynamic systems on unstructured meshes. In: Kröner, D., Ohlberger, M., Rohde, C. (eds.) *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*, pp. 195–285. Springer, Berlin (1999)
3. Bellman, R.: *Introduction to Matrix Analysis*, volume 19 of *Classics in Applied Mathematics*, 2nd edn. SIAM, Philadelphia (1987)
4. Böhm, M., Winters, A.R., Gassner, G.J., Derigs, D., Hindenlang, F., Saur, J.: An entropy stable nodal discontinuous Galerkin method for the resistive MHD equations. Part I: theory and numerical verification. *J. Comput. Phys.* (2018). <https://doi.org/10.1016/j.jcp.2018.06.027>
5. Boscheri, W., Dumbser, M.: Arbitrary-Lagrangian–Eulerian discontinuous Galerkin schemes with a posteriori subcell finite volume limiting on moving unstructured meshes. *J. Comput. Phys.* **346**, 449–479 (2017)
6. Chan, J.: On discretely entropy conservative and entropy stable discontinuous Galerkin methods. *J. Comput. Phys.* **362**, 346–374 (2018)
7. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: *Spectral Methods: Fundamentals in Single Domains*. Springer, Berlin (2006)
8. Chandrashekar, P.: Kinetic energy preserving and entropy stable finite volume schemes for compressible Euler and Navier–Stokes equations. *Commun. Comput. Phys.* **14**, 1252–1286 (2013)
9. Chavent, G., Cockburn, B.: The local projection $P^0 P^1$ -discontinuous Galerkin finite element method for scalar conservation laws. *ESAIM Math. Model. Num. (M²AN)* **23**, 565–592 (1989)
10. Chen, T., Shu, C.-W.: Entropy stable high order discontinuous Galerkin methods with suitable quadrature rules for hyperbolic conservation laws. *J. Comput. Phys.* **345**, 427–461 (2017)
11. Chiodaroli, E., De Lellis, C., Kreml, O.: Global Ill-posedness of the isentropic system of gas dynamics. *Commun. Pure Appl. Math.* **68**, 1157–1190 (2015)
12. Cockburn, B., Shu, C.-W.: Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.* **16**, 173–261 (2001)
13. Crean, J., Hicken, J.E., Fernández, D.C.D.R., Zingg, D.W., Carpenter, M.H.: Entropy-stable summation-by-parts discretization of the Euler equations on general curved elements. *J. Comput. Phys.* **356**, 410–438 (2018)
14. Dalcin, L., Rojas, D., Zampini, S., Fernández, D.C.D.R., Carpenter, M.H., Parsani, M.: Conservative and entropy stable solid wall boundary conditions for the compressible Navier–Stokes equations: adiabatic wall and heat entropy transfer. *J. Comput. Phys.* **397**, 108775 (2019)
15. Di Perna, R.J.: Convergence of approximate solutions to conservation laws. *Arch. Ration. Mech. Anal.* **82**, 27–70 (1983)
16. Di Pietro, D.A., Ern, A.: *Mathematical Aspects of Discontinuous Galerkin Methods*, vol. 69. Springer, Berlin (2011)
17. Donea, J., Huerta, A., Ponthot, J.P., Rodríguez-Ferran, A.: *Arbitrary Lagrangian–Eulerian Methods*. Encyclopedia of Computational Mechanics, 2nd edn, pp. 1–23. Wiley, Hoboken (2017)
18. Farhat, C., Geuzaine, P., Grandmont, C.: The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *J. Comput. Phys.* **174**, 669–694 (2001)
19. Fernández, D.C.D.R., Hicken, J.E., Zingg, D.W.: Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. *Comput. Fluids* **95**, 171–196 (2014)
20. Fisher, T.C., Carpenter, M.H.: High order entropy stable finite difference schemes for nonlinear conservation laws: finite domains. *J. Comput. Phys.* **252**, 518–557 (2013)
21. Friedrich, L., Schnücke, G., Winters, A.R., Fernández, D.C.D.R., Gassner, G.J., Carpenter, M.H.: Entropy stable space-time discontinuous Galerkin schemes with summation-by-parts property for hyperbolic conservation laws. *J. Sci. Comput.* **80**, 175–222 (2019)
22. Gassner, G.J.: A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods. *SIAM J. Sci. Comput.* **35**, A1233–A1253 (2013)
23. Gassner, G.J., Winters, A.R., Kopriva, D.A.: Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations. *J. Comput. Phys.* **327**, 39–66 (2016)
24. Gassner, G.J., Winters, A.R., Hindenlang, F.J., Kopriva, D.A.: The BR1 scheme is stable for the compressible Navier–Stokes equations. *J. Sci. Comput.* **77**, 1–47 (2018)
25. Godlewski, E., Raviart, P.-A.: *Hyperbolic Systems of Conservation Laws*. Ellipses, Paris (1991)

26. Godunov, S.K.: An interesting class of quasilinear systems. *Dokl. Acad. Nauk SSSR* **139**, 521–523 (1961)
27. Guermond, J.L., Popov, B., Saavedra, L., Yang, Y.: Invariant domains preserving arbitrary Lagrangian Eulerian approximation of hyperbolic systems with continuous finite elements. *SIAM J. Sci. Comput.* **39**(2), A385–A414 (2017)
28. Guillard, H., Farhat, C.: On the significance of the geometric conservation law for flow computations on moving meshes. *Comput. Method. Appl. Mech. Eng.* **190**, 1467–1482 (2000)
29. Harten, A.: On the symmetric form of systems of conservation laws with entropy. *J. Comput. Phys.* **49**, 151–164 (1983)
30. Hindenlang, F.J., Bolemann, T., Munz, C.-D.: Mesh curving techniques for high order discontinuous Galerkin simulations. In: Kroll, N., Hirsch, C., Bassi, F., Johnston, C., Hillewaert, K. (eds.) *IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach*, pp. 133–152. Springer, Cham (2015)
31. Hindenlang, F.J., Gassner, G.J., Kopriva, D.A.: Stability of wall boundary condition procedures for discontinuous Galerkin spectral element approximations of the compressible Euler equations (2019). arXiv preprint [arXiv:1901.04924](https://arxiv.org/abs/1901.04924)
32. Huang, W., Russell, R.D.: *Adaptive Moving Mesh Methods*, vol. 174. Springer, Berlin (2010)
33. Ismail, F., Roe, P.L.: Affordable, entropy-consistent Euler flux functions II: entropy production at shocks. *J. Comput. Phys.* **228**, 5410–5436 (2009)
34. Kennedy, C.A., Carpenter, M.H., Lewis, R.M.: Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations. *Appl. Numer. Math.* **35**, 177–219 (2000)
35. Kopriva, D.A.: Metric identities and the discontinuous spectral element method on curvilinear meshes. *J. Sci. Comput.* **26**, 301–327 (2006)
36. Kopriva, D.A.: *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. Springer, Berlin (2009)
37. Kopriva, D.A., Winters, A.R., Bohm, M., Gassner, G.J.: A provably stable discontinuous Galerkin spectral element approximation for moving hexahedral meshes. *Comput. Fluids* **139**, 148–160 (2016)
38. Kreiss, H.-O., Olinger, J.: Comparison of accurate methods for the integration of hyperbolic equations. *Tellus* **24**, 199–215 (1972)
39. Kružkov, S.N.: First order quasilinear equations in several independent variables. *Math. USSR Sbornik* **10**, 217–243 (1970)
40. LeFloch, P.G., Mercier, J.-M., Rohde, C.: Fully discrete, entropy conservative schemes of arbitrary order. *SIAM J. Numer. Anal.* **40**, 1968–1992 (2002)
41. Lesoinne, M., Farhat, C.: Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Comput. Method. Appl. Mech. Eng.* **134**, 71–90 (1996)
42. Lombard, C.K., Thomas, P.D.: Geometric conservation law and its application to flow computations on moving grids. *AIAA J.* **17**, 1030–1037 (1979)
43. Lomtev, I., Kirby, R.M., Karniadakis, G.E.: A discontinuous Galerkin ALE method for compressible viscous flows in moving domains. *J. Comput. Phys.* **155**, 128–159 (1999)
44. Loubère, R., Maire, P.H., Shashkov, M., Breil, J., Galera, S.: ReALE: a reconnection-based arbitrary-Lagrangian–Eulerian method. *J. Comput. Phys.* **229**, 4724–4761 (2010)
45. Marinacci, F., Pakmor, R., Springel, V.: The formation of disc galaxies in high-resolution moving-mesh cosmological simulations. *Mon. Not. R. Astron. Soc.* **437**, 1750–1775 (2013)
46. Mavriplis, D.J., Yang, Z.: Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes. *J. Comput. Phys.* **213**, 557–573 (2006)
47. Merriam, M.L.: Towards a rigorous approach to artificial dissipation. In: *Computational Fluid Dynamics*. No. AIAA-Paper-89-0471; CONF-890123. National Aeronautics and Space Administration, Moffett Field, CA, USA. Ames Research Center (1989)
48. Minoli, C.A.A., Kopriva, D.A.: Discontinuous Galerkin spectral element approximations on moving meshes. *J. Comput. Phys.* **230**, 1876–1902 (2011)
49. Mock, M.S.: Systems of conservation laws of mixed type. *J. Differ. Equ.* **37**, 70–88 (1980)
50. Moura, R.C., Mengaldo, G., Peiró, J., Sherwin, S.J.: On the eddy-resolving capability of high-order discontinuous Galerkin approaches to implicit LES/under-resolved DNS of Euler turbulence. *J. Comput. Phys.* **330**, 615–623 (2017)
51. Nikkar, S., Nordström, J.: Fully discrete energy stable high order finite difference methods for hyperbolic problems in deforming domains. *J. Comput. Phys.* **291**, 82–98 (2015)
52. Nguyen, V.T.: An arbitrary Lagrangian–Eulerian discontinuous Galerkin method for simulations of flows over variable geometries. *J. Fluid. Struct.* **26**, 312–329 (2010)
53. Parsani, M., Carpenter, M.H., Nielsen, E.J.: Entropy stable wall boundary conditions for the three-dimensional compressible Navier–Stokes equations. *J. Comput. Phys.* **292**, 88–113 (2015)

54. Persson, P.O., Bonet, J., Peraire, J.: Discontinuous Galerkin solution of the Navier–Stokes equations on deformable domains. *Comput. Methods Appl. Mech. Eng.* **198**, 1585–1595 (2009)
55. Ranocha, H.: Generalised Summation-by-Parts Operators and Entropy Stability of Numerical Methods for Hyperbolic Balance Laws. Cuvillier, Göttingen (2018)
56. Shu, C.W., Don, W.S., Gottlieb, D., Schilling, O., Jameson, L.: Numerical convergence study of nearly incompressible, inviscid Taylor–Green vortex flow. *J. Sci. Comput.* **24**, 1–27 (2005)
57. Springel, V.: E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Mon. Not. R. Astron. Soc.* **401**, 791–851 (2010)
58. Tadmor, Eitan: The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Math. Comput.* **49**, 91–103 (1987)
59. Tadmor, E.: Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. *Acta Numer.* **12**, 451–512 (2003)
60. Wang, L., Persson, P.O.: High-order discontinuous Galerkin simulations on moving domains using ALE formulations and local remeshing and projections. In: 53rd AIAA Aerospace Sciences Meeting. AIAA SciTech Forum (AIAA 2015-0820). <https://doi.org/10.2514/6.2015-0820>. Cited 20 Feb 2020
61. Wintermeyer, N., Winters, A.R., Gassner, G.J., Kopriva, D.A.: An entropy stable nodal discontinuous Galerkin method for the two dimensional shallow water equations on unstructured curvilinear meshes with discontinuous bathymetry. *J. Comput. Phys.* **340**, 200–242 (2017)
62. Winters, A.R., Moura, R.C., Mengaldo, G., Gassner, G.J., Walch, S., Peiro, J., Sherwin, S.J.: A comparative study on polynomial dealiasing and split form discontinuous Galerkin schemes for under-resolved turbulence computations. *J. Comput. Phys.* **372**, 1–21 (2018)
63. Winters, A.R., Derigs, D., Gassner, G.J., Walch, S.: A uniquely defined entropy stable matrix dissipation operator for high Mach number ideal MHD and compressible Euler simulations. *J. Comput. Phys.* **332**, 274–289 (2017)
64. Winters, A.R., Kopriva, D.A.: ALE-DGSEM approximation of wave reflection and transmission from a moving medium. *J. Comput. Phys.* **263**, 233–267 (2014)
65. Winters, A.R.: Discontinuous Galerkin spectral element approximations for the reflection and transmission of waves from moving material interfaces. Dissertation, The Florida State University (2014)
66. Yamaleev, N.K., Fernandez, D.C.D.R., Lou, J., Carpenter, M.H.: Entropy stable spectral collocation schemes for the 3-D Navier–Stokes equations on dynamic unstructured grids. *J. Comput. Phys.* **399**, 108897 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Gero Schnücke¹  · Nico Krais² · Thomas Bolemann² · Gregor J. Gassner³

Nico Krais
krais@iag.uni-stuttgart.de

Thomas Bolemann
bolemann@iag.uni-stuttgart.de

Gregor J. Gassner
ggassner@math.uni-koeln.de

¹ Mathematical Institute, University of Cologne, Cologne, Germany

² Institute of Aerodynamics and Gas Dynamics (IAG), University of Stuttgart, Stuttgart, Germany

³ Mathematical Institute, Center for Data and Simulation Science (CDS), University of Cologne, Cologne, Germany