

Discretely entropy stable weight-adjusted discontinuous Galerkin methods: curvilinear meshes and GPU acceleration

Jesse Chan, Lucas Wilcox

Abstract

Things to include: entropy conservation and stability on curvilinear meshes, choosing geometric factors to ensure constant state preservation, high order accuracy. Computational considerations for curvilinear meshes.

Contents

1	Introduction	1
2	Systems of nonlinear conservation laws	2
2.1	Standard DG formulations for nonlinear conservation laws	2
2.1.1	Mathematical notation	3
2.1.2	Discontinuous Galerkin formulations and the L^2 projection	3
3	Discretely entropy stable DG methods on curved meshes	4
3.1	Basis and quadrature rules	4
3.2	Interpolation, differentiation, projection, and lifting matrices	5
3.3	A discretely entropy stable DG formulation on curved meshes	6
3.4	Free stream preservation	6
4	Discretely stable and low storage DG methods on curved meshes	6
4.1	A weight-adjusted approximation to the curvilinear mass matrix	7
4.2	Weight-adjusted projection	7
5	Limiting	9
6	Numerical experiments	9
7	Computational results	9

1 Introduction

Introducing WADG [1, 2].

Matrix WADG: GPU efficiency of WADG vs storing weighted inverses [3].

2 Systems of nonlinear conservation laws

This work addresses is high order schemes for the following system of n nonlinear conservation laws in d dimensions

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{j=1}^d \frac{\partial \mathbf{f}_j(\mathbf{u})}{\partial x_j} = 0, \quad \mathbf{u} \in \mathbb{R}^n, \quad \mathbf{f}_j : \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad (1)$$

where $\mathbf{u}(\mathbf{x}, t)$ denote the *conservative variables* for this system. We will specify boundary conditions in a later section.

We are interested in nonlinear conservation laws for which an entropy function $U(\mathbf{u})$ exists, where $U(\mathbf{u})$ is convex with respect to the conservative variables \mathbf{u} . If this function exists, then it is possible to define *entropy variables* $\mathbf{v}(\mathbf{u}) = \frac{\partial U}{\partial \mathbf{u}}$. These functions symmetrize the system of nonlinear conservation laws (1) [4].

It can be shown (see, for example, [5]) that symmetrization is equivalent to the existence of an entropy flux function $F(\mathbf{u})$ and entropy potential ψ such that

$$\mathbf{v}^T \frac{\partial \mathbf{f}_j}{\partial \mathbf{u}} = \frac{\partial F_j(\mathbf{u})}{\partial \mathbf{u}}^T, \quad \psi_j(\mathbf{v}) = \mathbf{v}^T \mathbf{f}_j(\mathbf{u}(\mathbf{v})) - F_j(\mathbf{u}(\mathbf{v})), \quad \psi'_j(\mathbf{v}) = \mathbf{f}_j(\mathbf{u}(\mathbf{v})).$$

Smooth solutions of (1) can be shown to satisfy a conservation of entropy by multiplying (1) by $\mathbf{v}(\mathbf{u})$. Using the definition of the entropy variables, entropy flux, and the chain rule yields

$$\mathbf{v}^T \frac{\partial \mathbf{f}_j(\mathbf{u})}{\partial x_j} = \frac{\partial U(\mathbf{u})}{\partial \mathbf{u}}^T \frac{\partial \mathbf{f}_j(\mathbf{u})}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial x_j} = \frac{\partial F_j(\mathbf{u})}{\partial x_j}, \quad (2)$$

and that

$$\frac{\partial U(\mathbf{u})}{\partial t} + \sum_{j=1}^d \frac{\partial F_j(\mathbf{u})}{\partial x_j} = 0.$$

Let $\Omega \subset \mathbb{R}^d$ now be a closed domain with boundary $\partial\Omega$. Integrating over Ω and using Gauss' theorem on the spatial derivative yields

$$\int_{\Omega} \frac{\partial U(\mathbf{u})}{\partial t} dx + \int_{\partial\Omega} \sum_{j=1}^d (\mathbf{v}(\mathbf{u})^T \mathbf{f}_j(\mathbf{u}) - \psi_j(\mathbf{v}(\mathbf{u}))) n_j dx = 0, \quad (3)$$

where $\mathbf{n} = (n_1, \dots, n_d)^T$ denotes the unit outward normal vector on $\partial\Omega$.

General solutions (including non-smooth solutions such as shocks) satisfy an entropy *inequality*

$$\int_{\Omega} \frac{\partial U(\mathbf{u})}{\partial t} dx + \int_{\partial\Omega} \sum_{j=1}^d (\mathbf{v}(\mathbf{u})^T \mathbf{f}_j(\mathbf{u}) - \psi_j(\mathbf{v}(\mathbf{u}))) n_j dx \leq 0, \quad (4)$$

which results from considering solutions of an appropriate viscous form of the equations (1) and taking the limit as viscosity vanishes. In this work, schemes which satisfy a discrete form of (4) will be constructed by first enforcing a discrete version of entropy conservation (3), then adding an appropriate numerical dissipation which will enforce the entropy inequality (4).

2.1 Standard DG formulations for nonlinear conservation laws

We begin by reviewing the construction of standard high order DG formulations for (1).

2.1.1 Mathematical notation

Let the domain $\Omega \subset \mathbb{R}^d$ be decomposed into elements (subdomains) D^k , and let \hat{D} denote a d -dimensional reference element with boundary $\partial\hat{D}$. Let $\hat{\mathbf{x}}$ denote coordinates on \hat{D} , and let \hat{n}_i denote the i th component of the unit normal vector on $\partial\hat{D}$. We assume that \hat{n}_i is constant; i.e., that the faces of the reference element are planar (this assumption holds for all commonly used reference elements [6]).

We will assume that each physical element D^k is the image of \hat{D} under some smoothly differentiable mapping $\Phi_k(\hat{\mathbf{x}})$ such that

$$\mathbf{x} = \Phi_k(\hat{\mathbf{x}}), \quad \mathbf{x} \in D^k.$$

This also implies that integrals over physical elements can be mapped back to the reference element as follows

$$\int_{D^k} u \, d\mathbf{x} = \int_{\hat{D}} u J \, d\hat{\mathbf{x}},$$

where J denotes the determinant of the Jacobian of Φ_k . Integrals over physical faces of D^k can similarly be mapped back to reference faces.

We define an approximation space using degree N polynomials on the reference element. For example, on a d -dimensional reference simplex, the natural polynomial space are total degree N polynomials

$$P^N(\hat{D}) = \left\{ \hat{x}_1^{i_1} \dots \hat{x}_d^{i_d}, \quad \hat{\mathbf{x}} \in \hat{D}, \quad 0 \leq \sum_{k=1}^d i_k \leq N \right\}.$$

Other element types possess different natural polynomial spaces [6], but typically contain the space of total degree N polynomials. This work is directly applicable to other elements and spaces as well. We denote the dimension of the approximation space P^N as $N_p = \dim(P^N(\hat{D}))$.

Finally, we define the L^2 norm and inner products over the reference element \hat{D} and the surface of the reference element $\partial\hat{D}$

$$(\mathbf{u}, \mathbf{v})_{\hat{D}} = \int_{\hat{D}} \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x} = \int_{\hat{D}} \mathbf{u} \cdot \mathbf{v} J^k \, d\hat{\mathbf{x}}, \quad \|\mathbf{u}\|_{\hat{D}}^2 = (\mathbf{u}, \mathbf{u})_{\hat{D}}, \quad \langle \mathbf{u}, \mathbf{v} \rangle_{\partial\hat{D}} = \int_{\partial\hat{D}} \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x},$$

Finally, we introduce the continuous L^2 projection operator Π_N and lifting operator L . For $u \in L^2(\hat{D})$, the L^2 projection $\Pi_N u$ is defined through

$$\int_{\hat{D}} \Pi_N u v \, d\hat{\mathbf{x}} = \int_{\hat{D}} u v \, d\hat{\mathbf{x}}, \quad \forall v \in P^N(\hat{D}). \quad (5)$$

Likewise, for a boundary function $u \in L^2(\partial\hat{D})$, the lifting operator L [19, 20] is defined through

$$(Lu, v)_{\hat{D}} = \langle u, v \rangle_{\partial\hat{D}}, \quad \forall v \in P^N. \quad (6)$$

2.1.2 Discontinuous Galerkin formulations and the L^2 projection

Discontinuous Galerkin methods have been very widely applied to systems of nonlinear conservation laws (1) [7, 8, 9, 10]. The development of new discontinuous Galerkin methods for nonlinear conservation laws have focused heavily on the choice of numerical flux [11] or the development of slope limiters [12, 13, 14] and artificial viscosity strategies [15, 16, 17]. However, the treatment of the underlying volume discretization remains relatively unchanged between each of these approaches.

Ignoring terms involving filters, limiters, or artificial viscosity, a semi-discrete “weak” DG formulation for (1) can be given locally over an element D^k : find $\mathbf{u} \in (P^N(D^k))^n$ such that

$$\int_{D^k} \left(\frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} - \sum_{j=1}^d \mathbf{f}_j(\mathbf{u}) \cdot \frac{\partial \mathbf{v}}{\partial x_j} \right) d\mathbf{x} + \sum_{j=1}^d \int_{\partial D^k} (\mathbf{f}_j^*(\mathbf{u}^+, \mathbf{u})) \cdot \mathbf{v} n_j \, d\mathbf{x} = 0, \quad \forall \mathbf{v} \in (P^N(D^k))^n, \quad (7)$$

where the numerical flux \mathbf{f}^* is a function of the solution \mathbf{u} on both D^k and neighboring elements.

Unfortunately, solutions to (7) do not (in general) obey a discrete version of the entropy inequality (4). Since (4) is a generalized statement of energy stability, the lack of a discrete entropy inequality implies that the discrete solution can blow up in finite time. The reason for this is due the fact that, in practice, the integrals in (7) are not computed exactly and are instead approximated using polynomially exact quadratures. This is compounded by the fact that the nonlinear flux function $\mathbf{f}_j(\mathbf{u})$ is often rational and impossible to integrate exactly using polynomial quadratures. To account for the inexactness of quadrature, we rewrite (7) in strong form using a discrete quadrature-based L^2 projection.

For polynomial approximation spaces, $\frac{\partial \mathbf{v}}{\partial x_i}$ is polynomial. Then, mapping (7) back to the reference element \hat{D} and using the L^2 projection and (5), we have that

$$\int_{D^k} \mathbf{f}_j(\mathbf{u}) \cdot \frac{\partial \mathbf{v}}{\partial x_i} d\mathbf{x} = \int_{\hat{D}} \Pi_N \mathbf{f}_j(\mathbf{u}) \cdot \frac{\partial \mathbf{v}}{\partial x_i} J d\mathbf{x}.$$

Thus, integrating by parts (7) recovers a “strong” DG formulation involving the projection operator

$$\begin{aligned} & \int_{D^k} \left(\frac{\partial \mathbf{u}}{\partial t} - \sum_{j=1}^d \frac{\partial \Pi_N \mathbf{f}_j(\mathbf{u})}{\partial x_j} \right) \cdot \mathbf{v} d\mathbf{x} \\ & + \sum_{j=1}^d \int_{\partial D^k} (\mathbf{f}_j^*(\mathbf{u}^+, \mathbf{u}) - \Pi_N \mathbf{f}_j(\mathbf{u})) \cdot \mathbf{v} n_j d\mathbf{x} = 0, \quad \forall \mathbf{v} \in (P^N(D^k))^n. \end{aligned} \quad (8)$$

From this, we see that our discrete scheme does not differentiate the nonlinear flux function $\mathbf{f}_j(\mathbf{u})$ exactly, but instead differentiates the projection of $\Pi_N \mathbf{f}_j(\mathbf{u})$ onto polynomials of degree N . Because the L^2 projection operator is introduced, the chain rule no longer holds at the discrete level and step (2) of the proof of entropy conservation is no longer valid. Thus, ensuring discrete entropy stability will require a discrete formulation of the system of nonlinear conservation laws (1) from which we can prove a discrete entropy inequality without relying on the chain rule.

3 Discretely entropy stable DG methods on curved meshes

We will first show how to construct discretely entropy stable high order DG methods on curvilinear meshes, but will present this using a matrix formulation as opposed to a continuous formulation. This is to ensure that the effects of discretization, nonlinear, and quadrature are accounted for in the proof of semi-discrete entropy stability. We first introduce quadrature-based matrices, which we will then use to construct discretely entropy stable DG formulations.

3.1 Basis and quadrature rules

We now introduce quadrature-based matrices for the d -dimensional reference element \hat{D} , which we will use to construct matrix-vector formulations of DG methods. Assuming $u(\mathbf{x}) \in P^N(\hat{D})$, it can be represented in terms of the vector of coefficients \mathbf{u} using some polynomial basis ϕ_i of degree N and dimension N_p

$$u(\mathbf{x}) = \sum_{j=1}^{N_p} \mathbf{u}_j \phi_j(\hat{\mathbf{x}}), \quad P^N(\hat{D}) = \text{span} \{ \phi_i(\hat{\mathbf{x}}) \}_{i=1}^{N_p}.$$

We construct quadrature-based based on ϕ_i and appropriate volume and surface quadrature rules. The volume and surface quadrature rules are given by points and positive weights $\{(\mathbf{x}_i, w_i)\}_{i=1}^{N_q}$ and $\{(\mathbf{x}_i^f, w_i^f)\}_{i=1}^{N_q^f}$, respectively. We make the following assumptions on the strength of these quadratures:

Assumption 1. The volume quadrature rule $\{(\mathbf{x}_i, w_i)\}_{i=1}^{N_q}$ is exact for polynomials of degree $2N - 1$. Additionally, for any $u, v \in P^N(\hat{D})$, integration by parts

$$\left(\frac{\partial u}{\partial x_i}, v\right)_{\hat{D}} = \langle u, v n_i \rangle_{\partial \hat{D}} - \left(u, \frac{\partial v}{\partial x_i}\right)_{\hat{D}}$$

holds when volume and surface integrals are approximated using quadrature.

Assumption (1) holds, for example, for any surface quadrature rule which is exact for degree $2N$ polynomials on the boundary of the reference element $\partial \hat{D}$.

3.2 Interpolation, differentiation, projection, and lifting matrices

Let \mathbf{W}, \mathbf{W}_f denote diagonal matrices whose entries are volume and surface quadrature weights, respectively. We define the volume and surface quadrature interpolation matrices \mathbf{V}_q and \mathbf{V}_f

$$\begin{aligned} (\mathbf{V}_q)_{ij} &= \phi_j(\hat{\mathbf{x}}_i), & 1 \leq j \leq N_p, & \quad 1 \leq i \leq N_q, \\ (\mathbf{V}_f)_{ij} &= \phi_j(\hat{\mathbf{x}}_i^f), & 1 \leq j \leq N_p, & \quad 1 \leq i \leq N_q^f, \end{aligned} \quad (9)$$

which map coefficients \mathbf{u} to evaluations of u at volume and surface quadrature points.

Next, let \mathbf{D}_i denote the differentiation matrix with respect to the i th coordinate, defined implicitly through the relation

$$u(\hat{\mathbf{x}}) = \sum_{j=1}^{N_p} \mathbf{u}_j \phi_j(\hat{\mathbf{x}}), \quad \frac{\partial u}{\partial \hat{x}_i} = \sum_{j=1}^{N_p} (\mathbf{D}_i \mathbf{u})_j \phi_j(\hat{\mathbf{x}}).$$

The matrix \mathbf{D}_i maps basis coefficients of some polynomial $u \in P^N$ to coefficients of its i th derivative with respect to the reference coordinate $\hat{\mathbf{x}}$, and is sometimes referred to as a “modal” differentiation matrix (with respect to a general non-nodal “modal” basis [18]).

Using the volume quadrature interpolation matrix \mathbf{V}_q , we can compute a quadrature-based mass matrix \mathbf{M} by evaluating L^2 inner products of different basis functions using quadrature

$$\mathbf{M} = \mathbf{V}_q^T \mathbf{W} \mathbf{V}_q, \quad \mathbf{M}_{ij} = \sum_{k=1}^{N_q} w_k \phi_j(\mathbf{x}_k) \phi_i(\mathbf{x}_k) \approx \int_{\hat{D}} \phi_j \phi_i d\hat{\mathbf{x}} = (\phi_j, \phi_i)_{\hat{D}}.$$

The approximation in the formula for the mass matrix becomes an equality if the volume quadrature rule is exact for polynomials of degree $2N$. The mass matrix is symmetric and positive definite under Assumption 1; however, we do not make any distinctions between diagonal and dense (lumped) mass matrices in this work.

The mass matrix appears in the discretization of L^2 projection (5) and lift operators (6) using quadrature. The result are quadrature-based L^2 projection and lift operators $\mathbf{P}_q, \mathbf{L}_q$,

$$\mathbf{P}_q = \mathbf{M}^{-1} \mathbf{V}_q^T \mathbf{W}, \quad \mathbf{L}_q = \mathbf{M}^{-1} \mathbf{V}_f^T \mathbf{W}_f, \quad (10)$$

which are discretizations of the continuous L^2 projection operator Π_N and continuous lift operator L . The matrix \mathbf{P}_q maps a function (in terms of its evaluation at quadrature points) to coefficients of the L^2 projection in the basis $\phi_j(x)$, while the matrix \mathbf{L}_q “lifts” a function (evaluated at surface quadrature points) from the boundary of an element to coefficients of a basis defined in the interior of the element.

Finally, we define a curved mass matrix over each element by weighting the discrete L^2 norm with values of J at quadrature points

$$\mathbf{M}_k = \mathbf{V}_q^T \mathbf{W}_J \mathbf{V}_q, \quad (\mathbf{W}_J)_{ij} = \delta_{ij} w_i J(\mathbf{x}_i).$$

These matrices are distinct from element to element. [finish](#)

3.3 A discretely entropy stable DG formulation on curved meshes

3.4 Free stream preservation

We seek conditions for which free-stream preservation

$$\frac{\partial \mathbf{u}}{\partial t} = \nabla_h \cdot \mathbf{u} = 0$$

is satisfied if \mathbf{u} constant. Free stream preservation is not always maintained at the discrete level in 3D due to the fact that geometric factors are higher degree polynomials than the corresponding discrete space [21, 22]. For curvilinear meshes, $\widehat{\nabla} \cdot J\mathbf{G}^T \neq 0$ due to polynomial aliasing of geometric factors.

This can be remedied by using an interpolation of the curl-conservative form of the geometric factors, which ensures that $\widehat{\nabla} \cdot (J\mathbf{G}^T) = 0$ locally [23, 21]. However, because the geometric factors are computed by applying the curl, the geometric factors are approximated as degree $(N - 1)$ polynomials rather than degree N polynomials, which can reduce accuracy. We take a different approach, based on a strategy described in [24, 25].

Integrating by parts the DG formulation yields, for any constant \mathbf{u}

$$\begin{aligned} (\nabla_h \cdot \mathbf{u}, v)_\Omega &= \sum_k \left(\widehat{\nabla} \cdot (J\mathbf{G}^T \mathbf{u}), v \right)_{\widehat{D}} + \langle J\mathbf{G}^T \mathbf{u} - \Pi_N (J\mathbf{G}^T \mathbf{u}), v \widehat{\mathbf{n}} \rangle_{\partial \widehat{D}} \\ &= \sum_k \left(-J\mathbf{G}^T \mathbf{u}, \widehat{\nabla} v \right)_{\widehat{D}} + \langle (J\mathbf{G} \widehat{\mathbf{n}})^T \mathbf{u}, v \rangle_{\partial \widehat{D}} \\ &= \sum_k \left(-J\mathbf{G}^T \mathbf{u}, \widehat{\nabla} v \right)_{\widehat{D}} + \langle \mathbf{n} \cdot \mathbf{u}, v \rangle_{\partial \widehat{D}}. \end{aligned}$$

for any $v \in V_h$. Thus, to ensure that this sums to zero, we modify the geometric factors by seeking $J\mathbf{G}$ which minimizes the L^2 error for a degree N polynomial approximation to the true geometric factors, subject to a weakly divergence-free constraint

$$\begin{aligned} \min_{J\mathbf{G}_i \in P^N} \quad & \frac{1}{2} \left\| \tilde{J}\mathbf{G}_i - J\mathbf{G}_i \right\|_{L^2(\widehat{D})}^2, \\ \text{s.t.} \quad & \langle \mathbf{n}_i, v \rangle_{\partial \widehat{D}} - \left(\tilde{J}\mathbf{G}_i, \nabla v \right)_{\widehat{D}} = 0, \quad \forall v \in P^N. \end{aligned}$$

We note that the constraint corresponding to $v = 1$ yields that $\langle \mathbf{n}_i, v \rangle_{\partial \widehat{D}} = 0$. Thus, in order to guarantee a solution to this problem, we require that $\langle \mathbf{n}_i, v \rangle_{\partial \widehat{D}} = 0$ for consistency of the right hand side. [Clarify when this is satisfied!!](#).

This strategy was first introduced in the context of SBP-SAT terms in [24]. We approach its implementation slightly differently, and use the fact that the quadratic program can be solved explicitly using the null space method, which is computationally feasible since the null space of the constraint matrix is computed only once on the reference element. This null space corresponds to a discretely divergence-free basis, which we extract using the SVD.

The resulting free-stream preserving geometric factors result in separate approximations of the volume and surface geometric factors. The surface geometric factors are constructed at surface quadrature points, which guarantees that neighboring surface normal terms cancel for watertight meshes. The volume geometric factors are constructed at quadrature points and projected onto a polynomial basis of degree N with weakly divergence-free constraints involving the surface normals.

4 Discretely stable and low storage DG methods on curved meshes

While the

4.1 A weight-adjusted approximation to the curvilinear mass matrix

The L^2 projection on a curvilinear domain Π_N^k is defined as

$$(\Pi_N^k u, v)_{D^k} = (J \Pi_N^k u, v)_{\hat{D}} = (uJ, v)_{\hat{D}} = (u, v)_{D^k}, \quad \forall v \in V_h.$$

To define a weight-adjusted approximation to the curvilinear L^2 inner product, we first define the operator T_w^{-1} as follows

$$(w T_w^{-1} u, v)_{\hat{D}} = (u, v)_{\hat{D}}, \quad \forall v \in V_h.$$

Taking $w = 1/J$ then provides an approximation of the curvilinear L^2 inner product

$$(Ju, v)_{\hat{D}} \approx (T_{1/J}^{-1} u, v)_{\hat{D}}, \quad \forall u, v \in V_h.$$

This forms the basis of the weight-adjusted approximation of weighted L^2 inner products.

4.2 Weight-adjusted projection

We now define the weight-adjusted projection operator P_N as

$$P_N u = \Pi_N \left(\frac{1}{J} \Pi_N (uJ) \right).$$

Note that P_N is self-adjoint with respect to the J -weighted L^2 inner product

$$(JP_N u, v) = \left(\Pi_N \left(\frac{1}{J} \Pi_N (uJ) \right), vJ \right) = \left(uJ, \Pi_N \left(\frac{1}{J} \Pi_N (vJ) \right) \right) = (uJ, P_N v). \quad (11)$$

Furthermore, using that $T_{1/J}^{-1}$ is self-adjoint for $v \in V_h$ with respect to the L^2 inner product [1], we have that a projection-like property holds for the weight-adjusted L^2 inner product

$$(T_{1/J}^{-1} P_N u, v) = \left(\frac{1}{J} \Pi_N (uJ), T_{1/J}^{-1} v \right) = (\Pi_N (uJ), v) = (uJ, v), \quad \forall v \in V_h. \quad (12)$$

Let $\Pi_N^k u$ be the L^2 projection of u with respect to the weighted (curvilinear) L^2 inner product. We observe in numerical experiments that for a fixed geometric mapping, $\|\Pi_N^k u - P_N u\|_{L^2(\Omega)} = O(h^{N+2})$. Because the difference between the L^2 and WADG projection is superconvergent, the results are indistinguishable for a fixed geometric mapping.

We can prove this bound using results from [1, 2]. The first theorem we need shows that $T_{1/J}^{-1}$ can be used to approximate weighted curvilinear L^2 inner products with order $(2N+2)$ accuracy.

Theorem 1. *Let $u \in W^{N+1,2}(D^k)$ and $v \in V_h(D^k)$. Then,*

$$\begin{aligned} & \left| (u, vJ)_{\hat{D}} - (T_{1/J}^{-1} u, v)_{\hat{D}} \right| \leq \\ & Ch^{2N+2} \|J\|_{L^\infty(D^k)} \left\| \frac{1}{J} \right\|_{L^\infty(D^k)}^2 \|J\|_{W^{N+1,\infty}(D^k)}^2 \|u\|_{W^{N+1,2}(D^k)} \|v\|_{W^{N+1,2}(D^k)}. \end{aligned}$$

Proof. The proof involves straightforward adaptations of Theorem 4, Theorem 5, and Theorem 6 in [1] to the reference element \hat{D} . \square

The next result we need is a generalized inverse inequality.

Lemma 1. Let $v \in P^N(D^k)$, and let $h = \text{diam}(D^k)$. Then,

$$\|v\|_{W^{N+1,2}(D^k)} \leq C_N h^{-N} \|v\|_{L^2(D^k)}.$$

where C_N is independent of h .

Proof. The result is the consequence of a scaling argument and a Rayleigh quotient bound involving the largest eigenvalue of the generalized eigenvalue problem

$$\mathbf{K}_N \mathbf{u} = \lambda \mathbf{M} \mathbf{u},$$

where \mathbf{M} is the L^2 mass matrix over D^k and \mathbf{K}_N is the Gram matrix corresponding to the Sobolev inner product for $W^{N+1,2}(D^k)$. We note that the constant C_N depends on the largest eigenvalue, which in turn depends on the order N and dimension d . \square

We can now prove that $P_N u$ is superconvergent to the curvilinear L^2 projection $\Pi_N^k u$

Theorem 2. Let $u \in W^{N+1,2}(D^k)$. The difference between the L^2 projection $\Pi_N^k u$ and the weight-adjusted projection $P_N u$ is

$$\|\Pi_N^k u - P_N u\|_{L^2(D^k)} \leq C \|J\|_{L^\infty(D^k)} \left\| \frac{1}{J} \right\|_{L^\infty(D^k)}^2 \|J\|_{W^{N+1,\infty}(D^k)}^2 h^{N+2} \|u\|_{W^{N+1,2}(D^k)}.$$

where C is a mesh-independent constant.

Proof. Let $\Pi_N^k u$ be the L^2 projection over the curved element D^k , such that

$$(J \Pi_N^k u, v) = (u J, v), \quad \forall v \in V_h.$$

Using the fact that $P_N u$ satisfies an analogous property (12), we can rewrite the norm of the difference between the weight-adjusted and L^2 projections

$$\|\Pi_N^k u - P_N u\|_{L^2(D^k)}^2 = (\Pi_N^k u - P_N u, v J)_{\hat{D}}, \quad v = \Pi_N^k u - P_N u.$$

Because $v \in P^N(D^k)$, we can also evaluate the squared error as

$$\begin{aligned} \|\Pi_N^k u - P_N u\|_{L^2(D^k)}^2 &= |(\Pi_N^k u, v J)_{\hat{D}} - (P_N u, v J)_{\hat{D}}| \\ &= |(u, v J)_{\hat{D}} - (P_N u, v J)_{\hat{D}}| = \left| \left(T_{1/J}^{-1} P_N u, v \right)_{\hat{D}} - (J P_N u, v)_{\hat{D}} \right|. \end{aligned}$$

Applying Theorem 1 and Lemma 1 then yields that

$$\begin{aligned} \|\Pi_N^k u - P_N u\|_{L^2(D^k)}^2 &\leq C h^{2N+2} C_J \|u\|_{W^{N+1,2}(D^k)} \|v\|_{W^{N+1,2}(\hat{D})} \\ &\leq C h^{N+2} C_J \|u\|_{W^{N+1,2}(D^k)} \|v\|_{L^2(D^k)}, \end{aligned}$$

where C_J is a mesh-independent constant depending on J . Dividing through by

$$\|v\|_{L^2(D^k)} = \|\Pi_N^k u - P_N u\|_{L^2(D^k)}$$

gives the desired result. \square

5 Limiting

The evaluation of $\mathbf{u}_v = \mathbf{u}(\Pi_N v)$ can increase entropy pointwise, such that $S(\mathbf{u}_v) \geq S(\mathbf{u})$. This can manifest as spikes in \mathbf{u}_v . We wish to mollify the effect of such spikes.

The first approach is to limit the conservative variable \mathbf{u}

$$\tilde{\mathbf{u}} = \bar{\mathbf{u}} + \Theta(\mathbf{u} - \bar{\mathbf{u}})$$

where Θ is some diagonal matrix with entries in $[0, 1]$. We want to ensure that $\rho, E - \frac{\rho u^2}{2} > 0$.

The entropy for the compressible Euler equations is

$$U(\mathbf{u}) = -\frac{\rho s}{\gamma - 1},$$

where $s = \log\left(\frac{p}{\rho^\gamma}\right)$ is the physical specific entropy. The entropy variables under this choice of entropy are then

$$v_1 = \frac{\gamma - s}{\gamma - 1} - \frac{\rho u^2}{2p}, \quad v_2 = \frac{\rho u}{\rho e}, \quad v_3 = -\frac{\rho}{\rho e}.$$

where the internal energy $\rho e = E - u^2/2$.

The inverse mapping is given by

$$\rho = -(\rho e)v_3, \quad m = (\rho e)v_2, \quad E = (\rho e)\left(1 - \frac{v_2^2}{2v_3}\right),$$

where ρe and s in terms of the entropy variables are

$$\rho e = \left(\frac{(\gamma - 1)}{(-v_3)^\gamma}\right)^{1/(\gamma-1)} e^{\frac{-s}{\gamma-1}}, \quad s = \gamma - v_1 + \frac{v_2^2}{2v_3}.$$

The mapping is invertible so long as $\rho, E - \frac{\rho u^2}{2} > 0$, which can be ensured using standard limiters. However, we have to ensure also that $\rho(\Pi_N \mathbf{v}) > 0$ (similarly for internal energy). This boils down to ensuring that $\Pi_N v_3(x) < 0$, which guarantees that $(\rho e) > 0$ as well.

It can be helpful to ensure a stronger condition, that $\Pi_N v_3(x) \leq \max_x v_3(x)$. This guarantees a bound constraint on the conservative variables evaluated using the projected entropy variables.

Another approach is to change the time-step size based on the difference between \mathbf{u} and \mathbf{u}_v . This would be similar to local time-stepping (high order interpolation of the numerical fluxes and multiple evaluations of)

- Adaptively choose time-step size?
- Expensive option: bisection or Newton algorithm for finding theta.

6 Numerical experiments

7 Computational results

The semi-discrete evolution equation is as follows

$$\frac{\partial \mathbf{u}}{\partial t} = \begin{bmatrix} \mathbf{P}_q & \mathbf{L}_q \end{bmatrix} \left(\begin{bmatrix} 2\mathbf{D}_q & \mathbf{V}_q \mathbf{L}_q \\ \mathbf{V}_f \mathbf{P}_q & \mathbf{I} \end{bmatrix} \circ \begin{bmatrix} \mathbf{f}_S(\mathbf{u}, \mathbf{u}_f) & \mathbf{f}_S(\mathbf{u}, \mathbf{u}_f) \\ \mathbf{f}_S(\mathbf{u}, \mathbf{u}_f) & \mathbf{f}_S(\mathbf{u}_f^+, \mathbf{u}_f) \end{bmatrix} \mathbf{1} \right).$$

where \mathbf{u}, \mathbf{u}_f are evaluations at volume and surface quadrature points.

The method is outlined as follows: we store the conservation variables \mathbf{u} at quadrature points, and compute projected entropy variables $\mathbf{v}_u = \Pi_N(\mathbf{v}(\mathbf{u}))$

1. Apply volume flux differencing (\mathbf{D}_q and $\mathbf{V}_q \mathbf{L}_q$) and projection ($\mathbf{V}_q \mathbf{P}_q$) (volume kernel)
2. Apply surface flux differencing ($\mathbf{V}_f \mathbf{P}_q$) and lifting ($\mathbf{V}_q \mathbf{L}_q$) (volume kernel)
3. Apply WADG (scale by $1/J$ and apply $\mathbf{V}_q \mathbf{P}_q$ and update \mathbf{u} at volume quad points. Compute curvilinear projected entropy variables $\mathbf{v}_u = P_N \mathbf{v}(\mathbf{u}) = \Pi_N \left(\frac{1}{J} \Pi_N (\mathbf{v}(\mathbf{u})J) \right)$, interpolate to volume quad points, and compute conservative variable volume values (update)
4. Interpolate entropy variables at surface quadrature points and write out conservative variable surface values (face).

References

- [1] Jesse Chan, Russell J Hewett, and T Warburton. Weight-adjusted discontinuous Galerkin methods: wave propagation in heterogeneous media. *SIAM Journal on Scientific Computing*, 39(6):A2935–A2961, 2017.
- [2] Jesse Chan, Russell J Hewett, and T Warburton. Weight-adjusted discontinuous Galerkin methods: curvilinear meshes. *SIAM Journal on Scientific Computing*, 39(6):A2395–A2421, 2017.
- [3] Jesse Chan. Weight-adjusted discontinuous Galerkin methods: Matrix-valued weights and elastic wave propagation in heterogeneous media. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a. nme.5720.
- [4] Thomas JR Hughes, LP Franca, and M Mallet. A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Computer Methods in Applied Mechanics and Engineering*, 54(2):223–234, 1986.
- [5] Michael S Mock. Systems of conservation laws of mixed type. *Journal of Differential equations*, 37(1):70–88, 1980.
- [6] Jesse Chan, Zheng Wang, Axel Modave, Jean-Francois Remacle, and T Warburton. GPU-accelerated discontinuous Galerkin methods on hybrid meshes. *Journal of Computational Physics*, 318:142–168, 2016.
- [7] Bernardo Cockburn and Chi-Wang Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Mathematics of computation*, 52(186):411–435, 1989.
- [8] Bernardo Cockburn and Chi-Wang Shu. The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, 1998.
- [9] Bernardo Cockburn. Devising discontinuous Galerkin methods for non-linear hyperbolic conservation laws. *Journal of Computational and Applied Mathematics*, 128(1-2):187–204, 2001.
- [10] George Karniadakis and Spencer Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.
- [11] Jianxian Qiu, Boo Cheong Khoo, and Chi-Wang Shu. A numerical study for the performance of the Runge–Kutta discontinuous Galerkin method based on different numerical fluxes. *Journal of Computational Physics*, 212(2):540–565, 2006.
- [12] Lilia Krivodonova. Limiters for high-order discontinuous Galerkin methods. *Journal of Computational Physics*, 226(1):879–896, 2007.

- [13] Xiangxiong Zhang, Yinhua Xia, and Chi-Wang Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. *Journal of Scientific Computing*, 50(1):29–62, 2012.
- [14] Michael Dumbser, Olindo Zanotti, Raphaël Loubère, and Steven Diot. A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Journal of Computational Physics*, 278:47–75, 2014.
- [15] Per-Olof Persson and Jaime Peraire. Sub-cell shock capturing for discontinuous Galerkin methods. *AIAA*, 112, 2006.
- [16] Garrett E Barter and David L Darmofal. Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation. *Journal of Computational Physics*, 229(5):1810–1827, 2010.
- [17] Andreas Klöckner, Tim Warburton, and Jan S Hesthaven. Viscous shock capturing in a time-explicit discontinuous Galerkin method. *Mathematical Modelling of Natural Phenomena*, 6(3):57–83, 2011.
- [18] Jason E Hicken, David C Del Rey Fernández, and David W Zingg. Multidimensional summation-by-parts operators: General theory and application to simplex elements. *SIAM Journal on Scientific Computing*, 38(4):A1935–A1958, 2016.
- [19] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, volume 54. Springer, 2007.
- [20] Daniele Antonio Di Pietro and Alexandre Ern. *Mathematical aspects of discontinuous Galerkin methods*, volume 69. Springer Science & Business Media, 2011.
- [21] David A Kopriva. Metric identities and the discontinuous spectral element method on curvilinear meshes. *Journal of Scientific Computing*, 26(3):301–327, 2006.
- [22] Amaury Johnen, J.F. Remacle, and Christophe Geuzaine. Geometrical validity of curvilinear finite elements. *Journal of Computational Physics*, 233:359–372, 2013.
- [23] Miguel R Visbal and Datta V Gaitonde. On the use of higher-order finite-difference schemes on curvilinear and deforming meshes. *Journal of Computational Physics*, 181(1):155–185, 2002.
- [24] David C Fernández, Jason E Hicken, and David W Zingg. Simultaneous approximation terms for multi-dimensional summation-by-parts operators. *arXiv preprint arXiv:1605.03214*, 2016.
- [25] Jared Crean, Jason E Hicken, David C Del Rey Fernández, David W Zingg, and Mark H Carpenter. Entropy-stable summation-by-parts discretization of the Euler equations on general curved elements. *Journal of Computational Physics*, 356:410–438, 2018.