
Oral presentation | Higher order methods

Higher order methods-III

Tue. Jul 16, 2024 2:00 PM - 4:00 PM Room C

[5-C-01] A Novel Energy-based Artificial Viscosity for Suppressing Numerical Oscillations in Discontinuous Galerkin and Flux Reconstruction Schemes

*Weicheng Pei¹, Yu-Xin Ren¹ (1. Tsinghua University)

Keywords: shock capturing, artificial viscosity, energy dissipation, discontinuous Galerkin, flux reconstruction

A Novel Energy-based Artificial Viscosity for Suppressing Numerical Oscillations in Discontinuous Galerkin and Flux Reconstruction Schemes

Weicheng Pei* and Yu-Xin Ren*

Corresponding author: weicheng.pei@icloud.com

* Department of Engineering Mechanics, Tsinghua University, Beijing 100084, China

1 Introduction

To construct high-order schemes on unstructured meshes, the discontinuous Galerkin (DG) method [1, 2] and the flux reconstruction (FR) method [3, 4] are two of the most popular choices in the community of computational fluid dynamics (CFD). Besides the inherent compactness, these schemes are much easier to implement p -refinement than their finite difference and finite volume counterparts. However, like other high-order schemes, numerical oscillations would appear near discontinuities or large gradients in the solution given by a DG or FR scheme, if no shock fitting or shock capturing mechanism were incorporated into it.

Various shock capturing techniques, such as limiters [5, 6, 7, 8, 9], filters [10, 11] and artificial viscosities [12, 13, 14], have been developed for DG and FR schemes in the past two decades. However, most of these methods are based on orthogonal (modal) expansions, which are less efficient than pure Lagrange (nodal) expansions. Some of them are even built upon extrapolations, i.e. evaluating an approximating polynomial outside the element it belongs to, which impairs the compactness of DG and FR schemes.

In this paper, a novel artificial viscosity based on an energy measure of oscillation and its damping rate on a DG or FR element is developed. The oscillation energy, which measures the amplitude of numerical oscillations on a given element, is obtained by evaluating the L_2 -norm of the difference between the numerical solutions on the element and its neighbors. The damping rate of this energy on an element can be derived under the assumptions of linear flux–gradient relation and constant viscosity distribution. The value of viscosity for suppressing numerical oscillations is obtained by taking the ratio of the oscillation energy with respect to the product of its damping rate and prescribed time step. Such element-wise constant viscosity distribution could optionally be reconstructed to be C_0 continuous on element interfaces.

2 Methodology

2.1 Element-wise Polynomial Approximations

To solve the two-dimensional conservation law

$$\partial_t u + \partial_{\vec{r}} \cdot \vec{f} = 0, \quad \partial_{\vec{r}} \cdot \vec{f} = \partial_x f^x + \partial_y f^y,$$

using a DG or FR scheme, one may first introduce a map from the physical coordinates to the parametric coordinates on each element:

$$\underbrace{(x, y)}_{\vec{r}} \mapsto \underbrace{(\xi, \eta)}_{\vec{\rho}} \implies \begin{bmatrix} \partial_\xi \phi \\ \partial_\eta \phi \end{bmatrix} = \underbrace{\begin{bmatrix} \partial_\xi x & \partial_\xi y \\ \partial_\eta x & \partial_\eta y \end{bmatrix}}_J \begin{bmatrix} \partial_x \phi \\ \partial_y \phi \end{bmatrix} = \begin{bmatrix} \partial_\xi \vec{r} \\ \partial_\eta \vec{r} \end{bmatrix} \cdot \partial_{\vec{r}} \phi,$$

in which J is the Jacobian matrix of the coordinate map. The conservation law in physical coordinates is then (optionally) transformed into the form in parametric coordinates:

$$\partial_t U + \partial_{\vec{\rho}} \cdot \vec{F} = 0, \quad \partial_{\vec{\rho}} \cdot \vec{F} = \partial_\xi F^\xi + \partial_\eta F^\eta,$$

in which

$$U = u \underbrace{\det(\underline{J})}_J, \quad \begin{bmatrix} F^\xi \\ F^\eta \end{bmatrix} = J \underbrace{\begin{bmatrix} \partial_x \xi & \partial_y \xi \\ \partial_x \eta & \partial_y \eta \end{bmatrix}}_{J^{-1}} \begin{bmatrix} f^x \\ f^y \end{bmatrix} = J \begin{bmatrix} \partial_{\vec{r}} \xi \\ \partial_{\vec{r}} \eta \end{bmatrix} \cdot \vec{f}.$$

To obtain an element-wise polynomial approximation of the solution, an orthonormal (modal) expansion or a Lagrange interpolation has to be introduced for u or $U \equiv Ju$ on the j th element:

$$u(\vec{r}, t) \approx u_j^h(\vec{r}, t) = \begin{cases} \sum_{n=1}^N \hat{u}_{j,n}(t) \phi_{j,n}(\vec{\rho}), & \text{Lagrange interpolation,} \\ \sum_{m=1}^M \tilde{u}_{j,m}(t) \psi_{j,m}(\vec{r}), & \text{orthonormal expansion,} \end{cases}$$

in which $\phi_{j,n}$ is the Lagrange basis associated with the n th node (a.k.a. solution point) on the j th element, which satisfies the Kronecker delta property

$$\phi_{j,m}(\vec{\rho}_{j,n}) = \delta_{mn}, \quad \forall (m, n) \in \{1, \dots, N\}^2,$$

while $\psi_{j,m}$ is the m th orthonormal basis on the j th element, which satisfies the innerproduct delta property

$$\langle \psi_{j,m} | \psi_{j,n} \rangle = \delta_{mn}, \quad \forall (m, n) \in \{1, \dots, M\}^2.$$

The coefficient $\hat{u}_{j,n}$ is the nodal value of the approximate solution, i.e. $\hat{u}_{j,n} = u_j^h(\vec{r}(\vec{\rho}_n), t)$, while the coefficient $\tilde{u}_{j,m}$ is the projection of the approximate solution on the m th basis, i.e. $\tilde{u}_{j,m} = \langle u_j^h | \psi_{j,m} \rangle$

The number of terms N or M in the summations, i.e. the dimension of the polynomial space, depends on the degree of solution P and the dimension of physical space D . For tensor-product elements, e.g. quadrangles, hexahedra, the dimension of a Lagrange basis is equal to the product of the number of solution points in each dimension, which leads to

$$N(P, D) = (P + 1)^D,$$

For building the orthonormal basis (in physical coordinates), one usually apply the Gram–Schmidt process to the basis formed by monomials whose degrees are less than or equal to P , which leads to

$$M(P, D) = \binom{P + D}{D}.$$

In general, the dimension of a nodal basis is much larger than that of the modal basis with the same (P, D) .

2.2 Semi-discretized Systems from DG and FR

An ordinary differential equation (ODE) system can then be derived from either the DG method or the FR method.

The DG method requires the conservation law to be satisfied in a weaker sense

$$\int_{E_j} \phi_n \frac{\partial u}{\partial t} = \int_{E_j} \vec{f} \cdot \nabla \phi_n - \oint_{\partial E_j} \vec{v} \cdot \vec{f} \phi_n, \quad \forall n \in \{1, \dots, N\},$$

in which the surface integral comes from integration-by-part. The solution u and the flux \vec{f} in the volume integrals should be replaced by one of the polynomial expansion discussed in the previous section, while the normal component of the flux in the surface integral should be replaced by a common flux function f^I . This procedure could also be applied to the conservation law in parametric coordinates, which gives

$$\int_{\mathcal{E}_j} \phi_n \frac{\partial U_j^h}{\partial t} = \int_{\mathcal{E}_j} \vec{F}_j^D \cdot \nabla \phi_n - \oint_{\partial \mathcal{E}_j} F^I \phi_n, \quad \forall n \in \{1, \dots, N\},$$

where \mathcal{E}_j represents the *standard* element defined in the parametric space. The superscript D in \vec{F}_j^D emphasizes that the approximated flux are discontinuous on element interfaces. Either of the two weak forms gives a semi-discretized ordinary differential equation (ODE) system.

The FR method, on the other hand, gives the ODE system by first introducing a $(P + 1)$ -degree correction function to the discontinuities flux \vec{F}_j^D , which then becomes C_0 continuous on element interfaces.

For one-dimensional problems, the correction procedure is given by

$$F_j^R(\xi) = F_j^D(\xi) + [F^I - F_j^D]_{\xi=-1} g_{-1}(\xi) + [F^I - F_j^D]_{\xi=+1} g_{+1}(\xi),$$

where the $(P + 1)$ -degree g 's must satisfy

$$g_{+1}(+1) = 1, \quad g_{+1}(-1) = 0, \quad g_{-1}(-\xi) = g_{+1}(+\xi), \quad \forall \xi \in [-1, 1],$$

and approximate 0 in some appropriate sense. Possible choices of g are plotted in Figure 1.

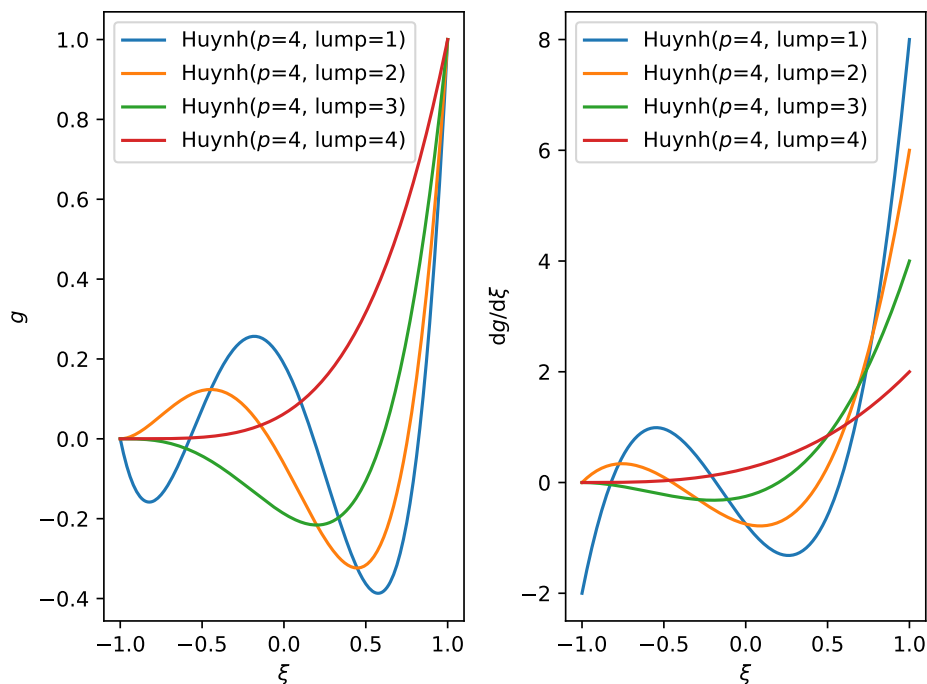


Figure 1: Candidates of the correction function g_{+1} .

The effect of the correction procedure for the linear convection flux $f(u) = u$ is demonstrated in Figure 2.

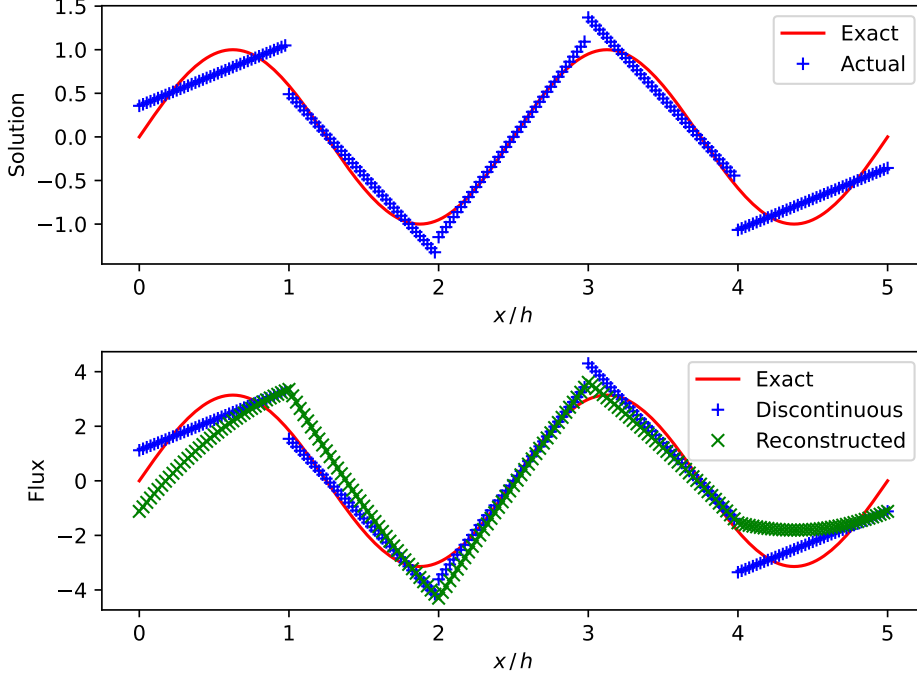


Figure 2: Discontinuous flux v. reconstructed continuous flux.

By forcing the conservation law to be satisfied on all solution points, and replace U and F by U_j^h and F_j^R :

$$\frac{d\hat{U}_{j,n}(t)}{dt} + \frac{\partial F_j^R(U^h(\xi_n, t))}{\partial \xi} = 0, \quad \forall n \in \{1, \dots, N\}.$$

For two-dimensional problems, if the element E_j is a quadrangle, the correction procedure could be applied in a dimension-by-dimension way, which gives

$$\begin{aligned} \frac{d\hat{U}_{j,m,n}}{dt} = & \frac{\partial F_j^{\xi,D}(\xi_m, \eta_m)}{\partial \xi} + \sum_{a=\pm 1} [F^I - F_j^{\xi,D}]_{\xi=a, \eta_m} \frac{dg_a(\xi_m)}{d\xi} \\ & + \frac{\partial F_j^{\eta,D}(\xi_m, \eta_m)}{\partial \eta} + \sum_{b=\pm 1} [F^I - F_j^{\eta,D}]_{\xi_m, \eta=b} \frac{dg_b(\eta_m)}{d\eta}, \end{aligned}$$

2.3 Direct Viscous Fluxes on Element Interfaces

Once an element-wise polynomial approximation is applied to the solution u , the flux $f(u, \nabla u)$ is generally discontinuous on the interfaces of adjacent elements. For the convection term in a conservation law, it is now a common practice to invoke an exact or an approximate Riemann solver [15]. However, methods for getting the common diffusion flux are still actively been developed.

For solving problems involving diffusion terms, people in the field of DG usually introduce a new variable $\vec{q} \equiv \nabla u$ and transform the second-order partial differential equation (PDE) into a first-order PDE system. Schemes for getting the common value of \vec{q} on element interfaces are then designed. The famous Bassi–Rebay (BR) methods [16, 17] and the more general local DG (LDG) method [18] all apply this *indirect* procedure. Recently, a method called direct DG (DDG) [19, 20, 21] is becoming more and more popular, due to the fact that it gives the common value of ∇u *directly* without introducing the auxillary variable \vec{q} . The common gradients are obtained by adding the penalty on jumps of value and second-order derivatives to the averaged gradient:

$$\begin{bmatrix} \partial_x u \\ \partial_y u \end{bmatrix}_{\partial E} = \beta_0 \Delta^{-1} \begin{bmatrix} n_x u \\ n_y u \end{bmatrix}_{R-L} + \begin{bmatrix} \partial_x u \\ \partial_y u \end{bmatrix}_{(R+L)/2} + \beta_1 \Delta \begin{bmatrix} n_x \partial_{xx}^2 u + n_y \partial_{xy}^2 u \\ n_x \partial_{yx}^2 u + n_y \partial_{yy}^2 u \end{bmatrix}_{R-L},$$

in which Δ is the characteristic length of the element, and the β 's are the penalty weights.

It is straight forward to apply the DDG method to polynomials given by orthonormal expansions defined in physical coordinates (i.e. $\vec{r} \equiv (x, y)$). However, it is more expensive and error-prone to be applied to solutions using Lagrange interpolation in parametric coordinates (i.e. $\vec{\rho} \equiv (\xi, \eta)$). If, the interpolation is applied to u , the second-order derivatives are

$$\begin{aligned} \begin{bmatrix} \partial_{xx}^2 u & \partial_{xy}^2 u \\ \partial_{yx}^2 u & \partial_{yy}^2 u \end{bmatrix} &= \underbrace{\underline{J}^{-1} \begin{bmatrix} \partial_\xi \\ \partial_\eta \end{bmatrix}}_{\begin{bmatrix} \partial_x \\ \partial_y \end{bmatrix}} \underbrace{\left(\begin{bmatrix} \partial_\xi u & \partial_\eta u \end{bmatrix} \underline{J}^{-T} \right)}_{\begin{bmatrix} \partial_x u & \partial_y u \end{bmatrix}} \\ &= \underline{J}^{-1} \begin{bmatrix} \partial_\xi & \partial_\eta \end{bmatrix} \begin{bmatrix} \partial_\xi u & \partial_\eta u \\ \partial_\xi u & \partial_\eta u \end{bmatrix} \underline{J}^{-T} + \underline{J}^{-1} \begin{bmatrix} \partial_\xi u & \partial_\eta u \\ \partial_\xi u & \partial_\eta u \end{bmatrix} \begin{bmatrix} \partial_\xi \underline{J}^{-T} \\ \partial_\eta \underline{J}^{-T} \end{bmatrix}, \end{aligned}$$

in which the derivatives of the Jacobian matrix

$$\frac{\partial \underline{J}^{-T}}{\partial \xi} = \left(-\underline{J}^{-1} \frac{\partial \underline{J}}{\partial \xi} \underline{J}^{-1} \right)^T, \quad \frac{\partial \underline{J}^{-T}}{\partial \eta} = \left(-\underline{J}^{-1} \frac{\partial \underline{J}}{\partial \eta} \underline{J}^{-1} \right)^T,$$

should be precomputed and cached on each flux point. If the interpolation is applied to $U \equiv Ju$, the second-order derivatives are more complex:

$$\begin{aligned} \begin{bmatrix} \partial_{xx}^2 u & \partial_{xy}^2 u \\ \partial_{yx}^2 u & \partial_{yy}^2 u \end{bmatrix} &= \underline{J}^{-1} \underbrace{\begin{bmatrix} \partial_\xi \\ \partial_\eta \end{bmatrix}}_{\begin{bmatrix} \partial_x \\ \partial_y \end{bmatrix}} \underbrace{\left(\begin{bmatrix} \partial_\xi U & \partial_\eta U \end{bmatrix} \underline{J}^{-T} \underline{J}^{-1} - \begin{bmatrix} \partial_\xi J & \partial_\eta J \end{bmatrix} \underline{J}^{-T} \underline{J}^{-2} U \right)}_{\begin{bmatrix} \partial_x u & \partial_y u \end{bmatrix}} \\ &= \underline{J}^{-1} \begin{bmatrix} \partial_\xi \partial_\xi U & \partial_\xi \partial_\eta U \\ \partial_\eta \partial_\xi U & \partial_\eta \partial_\eta U \end{bmatrix} \underline{J}^{-T} \underline{J}^{-1} + \dots, \\ \partial_\xi \left(\begin{bmatrix} \partial_\xi U & \partial_\eta U \end{bmatrix} \underline{J}^{-T} \underline{J}^{-1} \right) &= \begin{bmatrix} \partial_\xi \partial_\xi U & \partial_\xi \partial_\eta U \end{bmatrix} \underline{J}^{-T} \underline{J}^{-1} \\ &\quad + \begin{bmatrix} \partial_\xi U & \partial_\eta U \end{bmatrix} \left(\frac{\partial \underline{J}^{-T}}{\partial \xi} \underline{J}^{-1} + \underline{J}^{-T} (\partial_\xi \underline{J}^{-1}) \right), \\ \partial_\xi \left(\begin{bmatrix} \partial_\xi J & \partial_\eta J \end{bmatrix} \underline{J}^{-T} \underline{J}^{-2} U \right) &= \begin{bmatrix} \partial_\xi \partial_\xi J & \partial_\xi \partial_\eta J \end{bmatrix} \underline{J}^{-T} \underline{J}^{-2} U \\ &\quad + \begin{bmatrix} \partial_\xi J & \partial_\eta J \end{bmatrix} \left(\frac{\partial \underline{J}^{-T}}{\partial \xi} \underline{J}^{-2} U + \underline{J}^{-T} \frac{\partial \underline{J}^{-2}}{\partial \xi} U + \underline{J}^{-T} \underline{J}^{-2} \frac{\partial U}{\partial \xi} \right), \end{aligned}$$

in which, the derivatives of Jacobian determinant

$$\begin{aligned} \frac{\partial \underline{J}^{-1}}{\partial \xi} &= -\underline{J}^{-2} \frac{\partial \underline{J}}{\partial \xi}, \quad \frac{\partial \underline{J}^{-2}}{\partial \xi} = -2\underline{J}^{-3} \frac{\partial \underline{J}}{\partial \xi}, \\ \frac{\partial \underline{J}}{\partial \xi} &= \frac{\partial \det(\underline{J})}{\partial \xi} = \det(\underline{J}) \operatorname{tr} \left(\underline{J}^{-1} \frac{\partial \underline{J}}{\partial \xi} \right) = \underline{J} \operatorname{tr} \left(\underline{J}^{-1} \frac{\partial \underline{J}}{\partial \xi} \right), \end{aligned}$$

should also be cached.

2.4 Quasi-linear Artificial Viscosity

In this work, numerical oscillations are suppressed augmenting the original conservation law (system) with a quasi-linear viscous term. For a one-dimensional scalar problem, the augmented equation becomes

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = \frac{\partial}{\partial x} \left(\nu(u) \frac{\partial u}{\partial x} \right), \quad \nu(u) \geq 0.$$

Assume the viscosity value is a constant across the entire domain, and recall the fact that the artificial viscosity model and the numerical scheme for it on element interfaces are both linear, the ODE system

given by the DG or FR scheme can be written into the following form

$$\frac{d}{dt} |\hat{u}_j\rangle = (\text{inviscid terms}) + \nu (\underline{D}_j |\hat{u}_j\rangle + \underline{E}_j |\hat{u}_{j-1}\rangle + \underline{F}_j |\hat{u}_{j+1}\rangle),$$

where $|\hat{u}_j\rangle, |\hat{u}_{j-1}\rangle, |\hat{u}_{j+1}\rangle$ are column matrices formed by nodal or modal coefficients on E_j, E_{j-1}, E_{j+1} respectively, and $\underline{D}_j, \underline{E}_j, \underline{F}_j$ are constant square matrices given the spatial discretization and the numerical flux on element interfaces. Without loss of generality, we hereforth only consider the nodal interpolation in which solution points are also Gaussian quadrature points, which is a common practice in the DG spectral element method (SEM) [22, 9].

One may now define the *kinetic energy* on E_j to be

$$K_j \equiv \int_{E_j} \frac{1}{2} (u_j^h)^2 \approx \frac{1}{2} \sum_{n=1}^N w_{j,n} \hat{u}_{j,n} \hat{u}_{j,n} = \frac{1}{2} \underbrace{\begin{bmatrix} \hat{u}_{j,1} & \cdots & \hat{u}_{j,N} \end{bmatrix}}_{\langle \hat{u}_j |} \underbrace{\begin{bmatrix} w_{j,1} & & \\ & \ddots & \\ & & w_{j,N} \end{bmatrix}}_{\underline{W}_j} \underbrace{\begin{bmatrix} \hat{u}_{j,1} \\ \vdots \\ \hat{u}_{j,N} \end{bmatrix}}_{|\hat{u}_j\rangle},$$

where \underline{W}_j is the diagonal matrix form by Gaussian quadrature weights. The dissipation rate of K_j can easily be derived, which is

$$\frac{dK_j}{dt} = \langle \hat{u}_j | \underline{W}_j \frac{d}{dt} |\hat{u}_j\rangle = (\text{inviscid terms}) + \nu \langle \hat{u}_j | \underline{W}_j \underline{D}_j |\hat{u}_j\rangle + (\text{inter-cell viscous terms}),$$

The viscosity value on E_j is then determined by ignoring the inviscid terms and inter-cell viscous terms in the dissipation rate, which leads to

$$\nu_j = \frac{\Delta K_j}{-G_j \tau} \iff \frac{d}{dt} K_j = (\text{inviscid terms}) + \nu \underbrace{\langle \hat{u}_j | \underline{W}_j \underline{D}_j |\hat{u}_j\rangle}_{G_j} + (\text{inter-cell viscous terms}),$$

which means the oscillation energy ΔK_j is dissipated by the viscosity within a prescribed time range τ .

For a one-dimensional system, the previous procedure could be applied to either each conservative variable or each characteristic variable. For a multi-dimensional problem, one may just apply the proposed procedure to each conservative variable, i.e.

$$\frac{\partial}{\partial t} \begin{bmatrix} u_1 \\ \vdots \\ u_K \end{bmatrix} + \nabla \cdot \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_K \end{bmatrix} = \nabla \cdot \begin{bmatrix} \nu_1 \nabla u_1 \\ \vdots \\ \nu_K \nabla u_K \end{bmatrix},$$

2.5 Oscillation Energy

The last question in the current method is how to evaluate ΔK_j , which quantitatively measures the numerical oscillation on E_j .

In one-dimensional cases, we define the oscillation energy on the j th element to be

$$\Delta K_j = \int_{x_{j-1/2}}^{x_j} (u_j^h - u_{j-1}^h)^2 dx + \int_{x_j}^{x_{j+1/2}} (u_j^h - u_{j+1}^h)^2 dx,$$

which is the square of the L_2 -norm of difference between the solution on E_j and those on its immediate neighbors. The integrals are as cheap as weighted summations of nodal values if solution points are also Gaussian quadrature points. The extrapolations are also cheap, since the coordinate map in this case is usually linear and therefore trivial.

However, when it is generalized to multi-dimensional cases, extrapolations using nodal expansions in parametric coordinates are generally expensive, since it incurs solving a non-linear algebraic equation

$$\begin{bmatrix} x \\ y \end{bmatrix}_{\text{query}} = \begin{bmatrix} x^h(\xi, \eta) \\ y^h(\xi, \eta) \end{bmatrix},$$

for the value of (ξ, η) from the given (x, y) on a neighboring cell. Even worse, the solution of this equation might not exist, even for a 4-node quadrangular element. Figure 3 demonstrates a case of such failure,

in which the \times sign is the query point while the $+$ signs are the initial and intermediate states of the iteration for solving the non-linear (x, y) -to- (ξ, η) map.

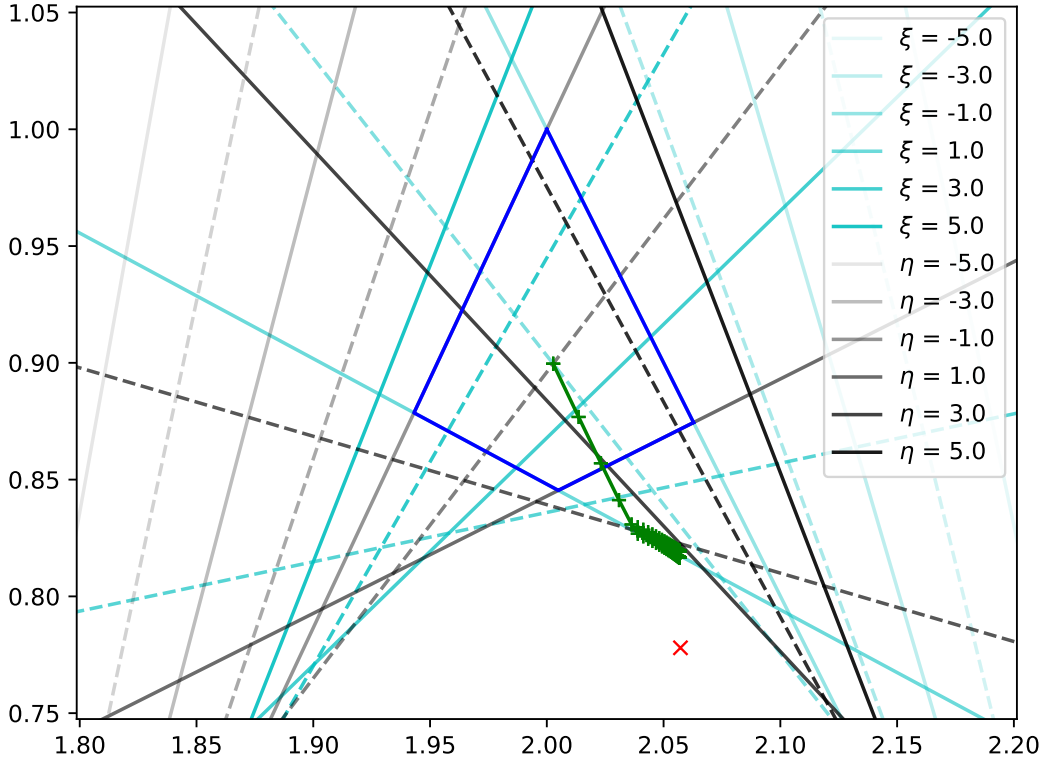


Figure 3: A failed case of the (x, y) -to- (ξ, η) map which might occur in extrapolations.

It is an open problem to design an oscillation measure requiring no extrapolations. In current work, we tried the following surface integral of value jumps

$$\oint_{\partial E_j} (u_j^h - u_{j'}^h)^2, \quad j' \in \text{neighbors of } E_j,$$

which gives reasonable results in ordinary tests. Further tests on more challenging problems are still in progress.

3 Results

In this section, the effectiveness of the artificial viscosity proposed in the previous section is demonstrated by some standard test cases. For one-dimensional problems, the computational domain is divided into 100 elements uniformly.

A fifth-order ($P = 4$) FR scheme using the g_2 correction function from [3], which is equivalent to a DG SEM scheme of the same order of accuracy, is used for spatial discretization. The resulting ODE system is solved by an explicit three-stage third-order strong stability preserving Runge–Kutta method [23].

3.1 Shock Tube Problems

Riemann problems, as well as their exact and approximate solvers, play a pivot role in the development of CFD schemes [15]. Among the infinite number of Riemann problems of the 1D Euler system, the Sod

problem with the initial condition

$$[\rho \quad u \quad p]_{t=0} = \begin{cases} \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}, & x < 0, \\ \begin{bmatrix} 0.125 & 0 & 0.1 \end{bmatrix}, & x > 0, \end{cases} \quad (1)$$

and the Lax problem in with the initial condition

$$[\rho \quad u \quad p]_{t=0} = \begin{cases} \begin{bmatrix} 0.445 & 0.698 & 3.528 \end{bmatrix}, & x < 0, \\ \begin{bmatrix} 0.5 & 0 & 0.571 \end{bmatrix}, & x > 0, \end{cases} \quad (2)$$

are two of the most famous ones that are frequently used for testing shock capturing techniques.

Figure 4 and 5 gives the solution (the *Actual* curves) of these two problems at the moment ($t = 0.5$ for Sod, $t = 0.3$ for Lax) and the viscosity distributions for each characteristic variables at all time steps. The *Expect* curves in these figures are the exact solutions obtained from analytical procedures.

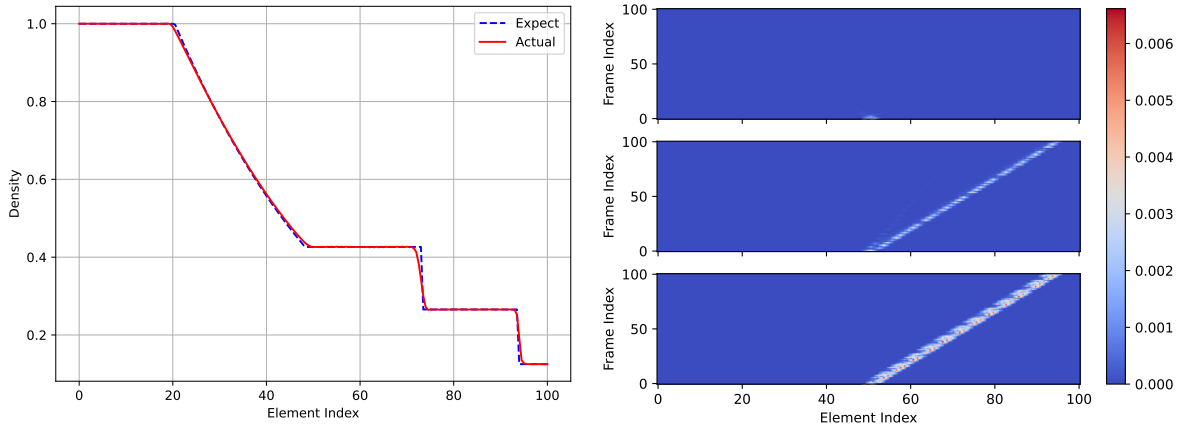


Figure 4: Solution and viscosity distribution of the Sod problem.

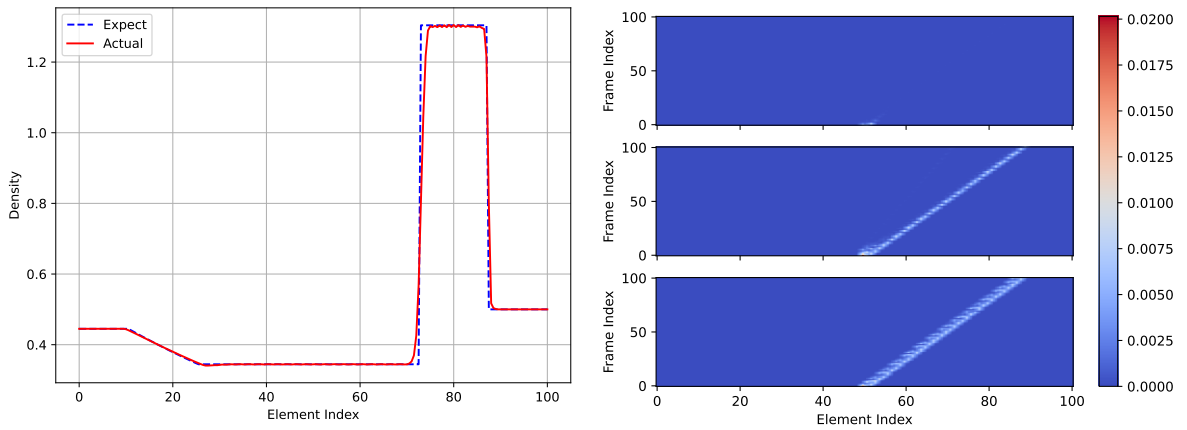


Figure 5: Solution and viscosity distribution of the Lax problem.

3.2 The Shu–Osher Problem

The Shu–Osher problem is designed to mimic the interaction of a running shock with a standing isentropic wave [24].

The computational domain is $x \in [0, 10]$ with two no-reflection conditions applied at the left and

right boundaries. The time range of interest is $t \in [0, 1.8]$ with the initial condition set to be

$$[\rho \quad u \quad p]_{t=0} = \begin{cases} \begin{bmatrix} 3.857143 & 2.629369 & 10.33333 \end{bmatrix}, & x \in [0, 1); \\ \begin{bmatrix} 1 + 0.2 \sin(5x) & 0 & 1 \end{bmatrix}, & x \in (1, 10]. \end{cases} \quad (3)$$

Figure 6 gives the solution (the *Actual* curve) of this problem at the final ($t = 1.8$) moment and the viscosity distribution for each characteristic variables at all time steps. The *Expect* curve in this figure is the approximate solution given by the same FR scheme with a p -weighted limiter [9] on a finer mesh (200 elements).

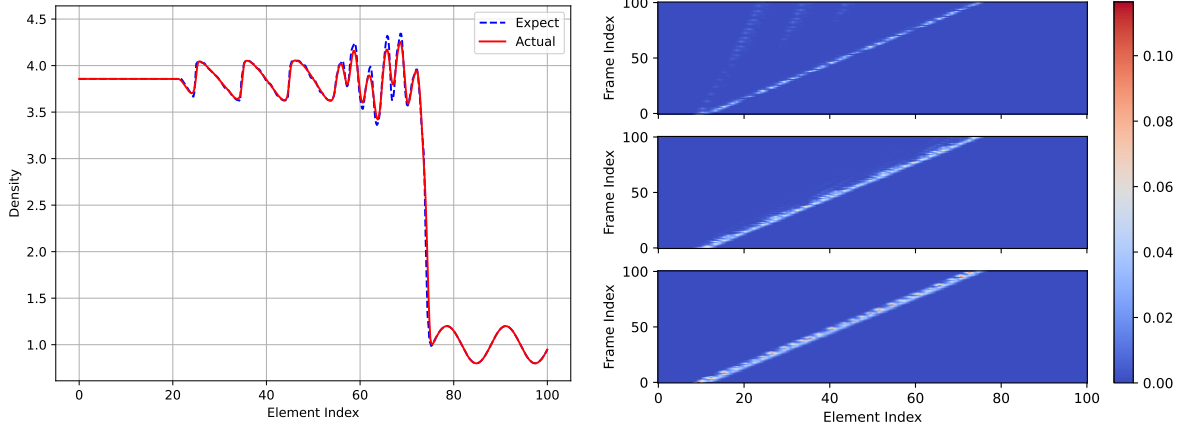


Figure 6: Solution and viscosity distribution of the Shu–Osher problem.

4 Conclusions

The standard cases for testing shock capturing methods show that the proposed artificial viscosity is sufficiently large for suppressing numerical oscillations near physical discontinuities, such as shocks and contacts, while keeps negligible in other regions for maintaining the high-order accuracy of the DG or FR solution.

References

- [1] Bernardo Cockburn and Chi-Wang Shu. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.
- [2] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods*. Springer, New York, 2008.
- [3] H. T. Huynh. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In *18th AIAA Computational Fluid Dynamics Conference*, Miami, Florida, USA, jun 2007. American Institute of Aeronautics and Astronautics.
- [4] H. T. Huynh, Z. J. Wang, and P. E. Vincent. High-order methods for CFD: A brief review of compact differential formulations on unstructured grids. *Computers & Fluids*, 98:209–220, jul 2014.
- [5] Xinghui Zhong and Chi-Wang Shu. A simple weighted essentially non-oscillatory limiter for Runge–Kutta discontinuous Galerkin methods. *Journal of Computational Physics*, 232(1):397–415, jan 2013.
- [6] Jun Zhu, Xinghui Zhong, Chi-Wang Shu, and Jianxian Qiu. Runge–Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes. *Journal of Computational Physics*, 248:200–220, sep 2013.
- [7] Hongqiang Zhu, Jianxian Qiu, and Jun Zhu. A simple, high-order and compact WENO limiter for RKDG method. *Computers and Mathematics with Applications*, 79(2):317–336, jan 2020.
- [8] Jun Zhu, Chi-Wang Shu, and Jianxian Qiu. RKDG methods with multi-resolution WENO limiters for solving steady-state problems on triangular meshes. *Communications on Applied Mathematics and Computation*, may 2023.

- [9] Wanai Li, Qian Wang, and Yu-Xin Ren. A p -weighted limiter for the DG method on one-dimensional and two-dimensional triangular grids. *Journal of Computational Physics*, 407:109246, apr 2020.
- [10] Konstantinos T. Panourgias and John A. Ekaterinaris. A nonlinear filter for high order discontinuous Galerkin discretizations with discontinuity resolution within the cell. *Journal of Computational Physics*, 326:234–257, December 2016.
- [11] T. Dzanic and F.D. Witherden. Positivity-preserving entropy-based adaptive filtering for discontinuous spectral element methods. *Journal of Computational Physics*, 468:111501, nov 2022.
- [12] Per-Olof Persson and Jaime Peraire. Sub-cell shock capturing for discontinuous galerkin methods. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, USA, January 2006. American Institute of Aeronautics and Astronautics.
- [13] A. Klöckner, T. Warburton, and J. S. Hesthaven. Viscous shock capturing in a time-explicit discontinuous galerkin method. *Mathematical Modelling of Natural Phenomena*, 6(3):57–83, 2011.
- [14] Niccolò Discacciati, Jan S. Hesthaven, and Deep Ray. Controlling oscillations in high-order discontinuous galerkin schemes using artificial viscosity tuned by neural networks. *Journal of Computational Physics*, 409:109304, may 2020.
- [15] Eleuterio F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer-Verlag, Berlin Heidelberg, 3 edition, 2009.
- [16] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 131(2):267–279, mar 1997.
- [17] Francesco Bassi, Andrea Crivellini, Stefano Rebay, and Marco Savini. Discontinuous galerkin solution of the reynolds-averaged navier–stokes and k - ω turbulence model equations. *Computers & Fluids*, 34(4-5):507–540, may 2005.
- [18] Bernardo Cockburn and Chi-Wang Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, dec 1998.
- [19] Hailiang Liu and Jue Yan. The direct discontinuous Galerkin (DDG) methods for diffusion problems. *SIAM Journal on Numerical Analysis*, 47(1):675–698, jan 2009.
- [20] Jian Cheng, Xiaoquan Yang, Xiaodong Liu, Tiegang Liu, and Hong Luo. A direct discontinuous Galerkin method for the compressible Navier–Stokes equations on arbitrary grids. *Journal of Computational Physics*, 327:484–502, dec 2016.
- [21] Xiaoquan Yang, Jian Cheng, Hong Luo, and Qijun Zhao. Robust implicit direct discontinuous Galerkin method for simulating the compressible turbulent flows. *AIAA Journal*, 57(3):1113–1132, mar 2019.
- [22] David A. Kopriva. *Implementing Spectral Methods for Partial Differential Equations*. Springer Netherlands, 2009.
- [23] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):89–112, jan 2001.
- [24] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83(1):32–78, jul 1989.