A comparison of artificial viscosity, limiters, and filters, for high order discontinuous Galerkin solutions in nonlinear settings

C. Michoski[ϒ] , C. Dawson,
Institute for Computational Engineering and Sciences (ICES)
Computational Hydraulics Group (CHG)
University of Texas at Austin, Austin, TX, 78712

E. J. Kubatko,
Department of Civil and Environmental Enineering and Geodetic Science
The Ohio State University, Columbus, OH, 43210

D. Wirasaet, S. Brus, J. J. Westerink
Computational Hydraulics Laboratory,
Department of Civil Engineering and Geological Sciences
University of Notre Dame, Notre Dame, IN, 46556

Abstract

Nonlinear systems of equations demonstrate complicated regularity features that are often obfuscated by overly diffuse numerical methods. Using a discontinuous Galerkin finite element method, we study a nonlinear system of advection-diffusion-reaction equations and aspects of its regularity. For numerical regularization, we present a family of solutions consisting of: 1) a sharp, computationally efficient slope limiter, known as the BDS limiter, 2) a standard spectral filter, and 3) a novel artificial diffusion algorithm with a solution-dependent entropy sensor. We analyze these three numerical regularization methods on a classical test in order to test the strengths and weaknesses of each, and then benchmark the methods against a large application model.

Keywords: Discontinuous Galerkin, nonlinear system, high order, regularization, slope limiting, spectral filters, artificial diffusion, artificial viscosity, advection, diffusion, reaction.

[ϒ]Corresponding author, *michoski@ices.utexas.edu*

Contents

§1 Introduction

High order methods are appealing in a large number of scientific contexts, being well-suited for maximizing performance to accuracy ratios in specific applications [18, 22, 23, 48]. As is frequently the case in nonlinear models, the nonlinear partial differential equations that govern the solution behavior, usually as presented in the weak formulation, are well-known mathematically to exhibit discontinuous solutions and singular behaviors. Since these behaviors can be driven by the underlying properties in the coupled mathematical operators, it can be important to preserve these singular and discontinuous aspects of the solution to: 1) determine how frequently and at what spatial density within the solution domain they occur, 2) determine if they are being generated spuriously, or affixed, by some aspect of the numerical approximation method, and 3) determine whether they are physical in nature or are simply model artifacts of the physics-based or empirical mathematical model. Identifying and analyzing all of these features of a solution can indeed be a tall order, as ideally doing so requires careful mathematical analysis, numerical analysis, and exhaustive physics-based validation and study.

Nevertheless, the presence of real, mathematical, and numerical discontinuities in solutions can be viewed as motivation for methods that can locally support and exactly represent discontinuities in their local solution spaces. A popular approach that is capable of representing such singular behavior is the discontinuous Galerkin (DG) finite element method (also frequently referred to as the local spectral finite element method). In the DG method, locally piecewise discontinuous polynomials are used over finite elements to represent solutions that can exactly represent jump

discontinuities along element faces. Though many methods are able to achieve this (such as finite difference and finite volume methods), DG methods can maintain exact piecewise discontinuous projections to arbitrarily high order (e.g. the degree p of the polynomial basis) at the resolution of h along element faces.

The appeal of DG methods in this regard is clear, though as with any method the pros can only be viewed relative to the methods cons. In the case of DG methods, the arguments against them are often aimed at the increased degrees of freedom relative to standard finite element methods. However, careful study has shown that these arguments are, at least at present, largely misleading, as DG methods are unusually well-suited for modern supercomputer architectures that emphasize arithmetic intensity over memory access [24], and also since modal enrichment in p is known to scale more efficiently than spatial refinement in h [48]. In fact, among those who use DG methods, the most concerning feature of the method is not its computational expense or scaling, but rather the fact that DG methods tend to demonstrate remarkable instabilities, with relatively frequent blow-up (or blow-down) behavior.

In linear PDE studies, this stability behavior can be very effectively isolated, and it seems relatively clear that in general the spatial instabilities observed in DG methods are not being caused by the numerical method per se, but rather from the underlying regularity of the PDE itself. That is, a DG solution projects into a larger (or weaker) function space (i.e. L^2_{loc}) than, for example, spectral methods (i.e. L^2) or continuous Galerkin methods (i.e. C^n), and thus are more prone to exhibit features emerging from the rigorous mathematical regularity of the solution. So, for example, if the initial boundary problem for the system of nonlinear equations is well-posed (in the sense of Hadamard) in L^2_{loc}, then the DG solution is more likely to exhibit singular behavior simply by virtue of its larger projection space.

However, even when discontinuities are salient features in the underlying model itself, the impact of these on the solver can be catastrophic and lead to frequent and nonideal blow-up behavior, especially when the discontinuities along the edges migrate into the cell interiors and the modal projection in p is of higher order. In these cases numerical shocks can form. These shocks are simply a consequence of the radius of convergence of the solution curve extending to a larger domain of influence than the chosen hp can explicitly support (as discussed in detail below).

In addition to this spatial irregularity, temporal instabilities are common features that arise in even the simplest of numerical systems, such as when solving scalar hyperbolic conservation laws using explicit timestepping. In this case the Courant-Friedrichs-Lewy (CFL) condition presents a relevant, and often times substantial restriction on the computational well-posedness of the solution, where the temporal discretization dt is (partially) bounded from above by the spatial discretization h: $dt \propto h/p$ as discussed in [47]. More broadly, CFL-type conditions can be viewed as application-dependent restrictions on the timestep dt made to assure that the wavespeed of the system cannot propagate at a rate that outpaces the contiguity of the (localized) numerical fluxes.

With the presence of these forms of potential instability in a solution to a system of nonlinear PDEs, the question of how to manage them in a numerical setting is entirely nontrivial. A number of techniques have been developed in order to regularize DG solutions, or, more informally, make then run to simulation completion without blowing up. The most common method, and the method marketed by some of the classical numerical papers [10–15, 45], is to use slopelimiting to regularize the solution. Slopelimiting can be thought of heuristically as an a posteriori method of mollifying solution gradients, where the system is reinitialized into a new state after every n computations. Slopelimiters can be very effective at reducing system instability, and a substantial amount of work has been done to establish sharper, better, faster, and increasingly more effective slopelimiters [32, 33, 35, 39, 50], many of which are specifically designed with an eye towards preserving important physical properties [34, 51].

Another popular approach, especially when running at high order (e.g. high degree polynomial p) is to implement spectral filters. Spectral filters are remarkably simple to implement, have relatively low computational complexity, tend to be quite stabilizing to the solution, and are analogous to well-known spectral viscosity methods [36]. A number of spectral filters have been developed in the DG setting, including exponential [21] and adaptive [37] filters.

A third regularization technique is in the area of artificial diffusion (and/or artificial viscosity). Though "artificial viscosity" is a term often used at varying levels of formality in the numerics and turbulence communities, in this paper we refer to artificial diffusion (or artificial viscosity) solutions in the context of the discontinuous Galerkin method as those formulated relative to a slightly perturbed form of the original system of PDEs, that formally convergences in the h-vanishing limit to the governing system. In this direction some very promising results have been established [4, 9, 25, 43, 49], where perhaps one of the most promising is the entropy viscosity formulation [52] that is stabilized by the formal mathematical entropy of the governing system of equations.

In this paper we restrict to a manageable system of nonlinear advection-diffusion-reaction equations. This system is a good example of a relatively simple nonlinear model that displays singular behaviors and supports multiple types of irregularity in its admissible function spaces. We provide several explicit examples of this generalized formalism below, notably in the setting of the discontinuous Galerkin shallow water equation model (DGSWEM) [7, 26–29]. All computations are run in parallel on the Texas Advanced Computing Center's (TACC's) 10 PetaFLOPS Dell Linux Cluster based Stampede system, on between 16 and 1024 processors.

An outline of the papers is as follows. In §2 we present the initial boundary value problem under consideration. We then discuss the spatial discretization in the discontinuous Galerkin framework, and restrict to the modal discontinuous Galerkin method. The system of equations is recast into the weak form, spatially discretized, integrated locally on elements, and then temporally discretized using a stability Runge-

Kutta method. In §3 we discuss the relevant regularization methods. First we discuss some of the salient regularity features of a nonlinear system of advection-diffusion-reaction equations. We then present the Bell-Dawson-Shubin (BDS) slopelimiter, and discuss the details of its implementation. Next we review the foundations of spectral filters and provide implementational details. Finally we present a generalized form of artificial diffusion, with a novel entropy sensor. In §4 we first restrict to a classical scalar conservation problem for studying shock propagation. We provide error behavior, graphs, and analysis of the computational system in order to develop a better understanding of how these regularization methods differ, and which one is best suited for different types of nonlinear applications. We finish the paper with a validation study of tidal constituent behavior in a benchmark problem over a large unstructured North Atlantic grid.

§2 Nonlinear advection-diffusion-reaction

Consider the following model system in divergence form

$$\iota_t + \nabla \cdot (\boldsymbol{u}\iota) - \nabla \cdot (f(\iota)\nabla\iota) = g, \tag{2.1}$$

given initial-boundary data

$$\iota_{|t=0} = \iota_0, \qquad \iota_{|\partial\Omega} = \iota_b. \tag{2.2}$$

For the sake of this work, our domain will be restricted to two spatial dimensions $\boldsymbol{x} = (x, y) \in \Omega$, with static velocity field $\boldsymbol{u} = (u, v)$ and n-vector $\iota = \iota(x, y) = \iota(\iota_1, \iota_2, \ldots, \iota_n)$. The derived function f will be of some interest to us here, and generally will take the form $f = \epsilon(\iota) + \eta$, where $\eta \in \mathbb{R}^+$ is a nonnegative constant representative of the systems native diffusivity, while $\epsilon = \epsilon(x, y, t)$ will represent the so-called artificial diffusion discussed below. The function g is a generalized source-reaction term, where $g = g(x, y, t)$ or $g = g(\iota)$.

As motivated by [2, 3] for families of elliptic subsystems, we introduce $\boldsymbol{\sigma}$ and cast (2.1) into the first order system,

$$\iota_t + \nabla \cdot (\boldsymbol{u}\iota) - \nabla \cdot (f(\iota)\boldsymbol{\sigma}) = g, \quad \text{and} \quad \boldsymbol{\sigma} = \nabla\iota. \tag{2.3}$$

This system is discretized using a discontinous Galerkin finite element method as discussed in detail in [16, 26, 29, 31, 38–40]. The discretization in time uses a family of SSP (strong stability preserving) Runge-Kutta schemes as discussed in [28, 30, 44, 45].

For notational completeness we adopt the following discretization scheme motivated by [20, 38]. Take an open $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$, given $T > 0$ such that $\mathcal{Q}_T = ((0, T) \times \Omega)$. Let $\mathscr{T}_h$ denote the partition of the closure of the triangulation of Ω, which we denote Ω_h, into a finite number of triangular elements denoted Ω_e, such that $\mathscr{T}_h = \{\Omega_{e_1}, \Omega_{e_2}, \ldots, \Omega_{e_{ne}}\}$, for $ne \in \mathbb{N}$ the number of elements in Ω_h. In this work we define the mesh diameter h to satisfy $h = \min_{i,j}(d_{ij})$ for the distance function

$d_{ij} = d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and element vertices $\boldsymbol{x}_i, \boldsymbol{x}_j \in \partial\Omega_e$ when the mesh is structured and regular.

Let Γ_{ij} denote the edge shared by two neighboring elements Ω_{e_i} and Ω_{e_j}, and for $i \in I \subset \mathbb{Z}^+ = \{1, 2, \ldots\}$ define the indexing set $r(i) = \{j \in I : \Omega_{e_j} \text{ is a neighbor of } \Omega_{e_i}\}$. Let us denote all $\Omega_{e_i} \cap \partial\Omega_h$ by S_j and letting $I_B \subset \mathbb{Z}^- = \{-1, -2, \ldots\}$ define $s(i) = \{j \in I_B : S_j \text{ is an edge of } \Omega_{e_i}\}$ such that $\Gamma_{ij} = S_j$ for $\Omega_{e_i} \in \Omega_h$ when $S_j \in \partial\Omega_{e_i}$, $j \in I_B$. Then for $\Xi_i = r(i) \cup s(i)$, we have

$$\partial\Omega_{e_i} = \bigcup_{j \in \Xi(i)} \Gamma_{ij}, \quad \text{and} \quad \partial\Omega_{e_i} \cap \partial\Omega_h = \bigcup_{j \in s(i)} \Gamma_{ij}.$$

We are interested in obtaining an approximate solution to ι at time t on the finite dimensional space of discontinuous piecewise polynomial functions over Ω restricted to $\mathscr{T}_h$, given as

$$S_h^p(\Omega_h, \mathscr{T}_h) = \{v : v_{|\Omega_{e_i}} \in \mathscr{P}^p(\Omega_{e_i}) \ \ \forall \Omega_{e_i} \in \mathscr{T}_h\}$$

for $\mathscr{P}^p(\Omega_{e_i})$ the space of degree $\leq p$ polynomials over Ω_{e_i}.

Choose a set of basis functions N_l for $l = 0, \ldots, n_p$ the corresponding degrees of freedom indexed by the (spatial) spectral order p of the local element, such that we can denote the modal solution over a finite element cell Ω_{e_i} as

$$\iota_{hp}^i(t, \boldsymbol{x}) = \sum_{l=0}^{n_p} \iota_l^i(t) N_l^i(\boldsymbol{x}), \tag{2.4}$$

where the N_l^i's are the finite element shape functions. The j arbitrary finite dimensional test functions φ_j are then characterized by

$$\varphi_j^i(\boldsymbol{x}) = \sum_{l=0}^{n_p} \varphi_l^i N_l^i(\boldsymbol{x}), \quad \forall\ \boldsymbol{x} \in \Omega_{e_i},$$

where φ_j^i are the modal coordinates in each Ω_{e_i}. These functions are chosen as members of the broken Sobolev space over the partition $\mathscr{T}_h$ defined by

$$W^{k,q}(\Omega_h, \mathscr{T}_h) = \{\omega : \omega_{|\Omega_{e_i}} \in W^{k,q}(\Omega_{e_i}) \ \ \forall \Omega_{e_i} \in \mathscr{T}_h\}.$$

In particular, when $q = 2$ and $k = 0$ we have the standard L^2-norm, such that $\varphi_j \in L^2_{loc}$ with piecewise polynomials representations $\varphi_j^i \in S_h^p(\Omega_h, \mathscr{T}_h)$. For the d-dimensional vector basis (in our case the spatial dimension dim $= d = 2$) we will denote the j arbitrary finite dimensional test functions $\boldsymbol{\zeta}_j$ characterized as the direct product space, such that in the usual sense, when $\varphi_j \in L^2$ with piecewise polynomials $\varphi_j^i \in S_h^p(\Omega_h, \mathscr{T}_h)$, this implies that $\boldsymbol{\zeta}_j \in [L^2]^2$ with piecewise polynomials $\zeta_j^i \in [S_h^p(\Omega_h, \mathscr{T}_h)]^2$, and so on.

We can equivalently recast the modal discontinuous Galerkin solution into the *nodal basis* (i.e. the nodal discontinuous Galerkin solution), by choosing $\tilde{n}_p$ support points in each finite element cell,

$$\iota_{hp}^i(t,\boldsymbol{x}) = \sum_{l=0}^{n_p} \iota_l^i(t)\varphi_l^i(\boldsymbol{x}) = \sum_{\ell=0}^{\tilde{n}_p} \iota_\ell^i(\boldsymbol{x}_\ell^i,t)\phi_\ell^i(\boldsymbol{x}), \quad \forall\ \boldsymbol{x}\in\Omega_{e_i},$$

for nodal basis ϕ_ℓ^i.

It turns out that this distinction between the modal and nodal bases is one of occasional importance in the development of regularization methods, as we will discuss in detail below. We will primarily restrict in this paper to the case of the modal DG basis, and constrast the implementational features with that of the nodal basis construction.

Then for ι a weak solution to (2.3), multiplying by φ_h and integrating elementwise by parts yields the system:

$$\begin{aligned}
&\frac{d}{dt}\int_{\Omega_{e_i}} \iota\varphi_h^i dxdy + \int_{\Omega_{e_i}} \nabla\cdot(\boldsymbol{u}\iota\varphi_h^i)dxdy - \int_{\Omega_{e_i}} \boldsymbol{u}\iota\cdot\nabla\varphi_h^i dxdy \\
&\qquad - \int_{\Omega_{e_i}} \nabla\cdot(f(\iota)\boldsymbol{\sigma}\varphi_h^i)dxdy + \int_{\Omega_{e_i}} f(\iota)\boldsymbol{\sigma}\cdot\nabla\varphi_h^i dxdy = \int_{\Omega_{e_i}} g\varphi_h^i dxdy, \qquad (2.5)\\
&\int_{\Omega_{e_i}} \boldsymbol{\sigma}\cdot\zeta_h^i dxdy - \int_{\Omega_{e_i}} \nabla\cdot(\iota\zeta_h)dxdy + \int_{\Omega_{e_i}} \iota\nabla\cdot\zeta_h^i dxdy = 0.
\end{aligned}$$

Replacing ι with ι_{hp}, and applying standard numerical fluxes to (2.5) for the convective term, along with those asscoiated by way of the unified framework [3] for the auxiliary equation and the diffusive term, then we arrive with the standard semidiscrete formulation of the disconintuous Galerkin method.

The discretization in time follows now directly from (2.5), where we employ a family of SSP (strong stability preserving) Runge-Kutta schemes as discussed in [44, 45]. That is, for the generalized SSP Runge-Kutta scheme we rewrite the evolution equation in (2.5) in the form, $\mathbf{M}(d\iota/dt) = \mathbf{R}$, where $\mathbf{R} = \mathbf{R}(\boldsymbol{u},\iota,\boldsymbol{\sigma})$ is the advection–diffusion contribution along with the source term, and where $\mathbf{M}$ is the usual mass matrix. Then the generalized s stage of order γ SSP Runge-Kutta method (denoted SSPRK(s,γ)) may be written to satisfy:

$$\begin{aligned}
\iota^{(0)} &= \iota^n,\\
\iota^{(i)} &= \sum_{r=0}^{i-1}\left(\alpha_{ir}\iota^r + dt\beta_{ir}\mathbf{M}^{-1}\mathbf{R}^r\right), \quad \text{for}\ \ i=1,\dots,s \qquad (2.6)\\
\iota^{n+1} &= \iota^{(s)},
\end{aligned}$$

where $\mathbf{R}^r = \mathbf{R}(\boldsymbol{u},\iota^r,\boldsymbol{\sigma}^r)$, and the solution at the n–th timestep is given as $\iota^n = \iota_{|t=t^n}$ and at the n–th plus first timestep by $\iota^{n+1} = \iota_{|t=t^{n+1}}$, with $t^{n+1} = t^n + dt$.

It is often possible to optimize the generalized SSP schemes of 2.6 by restricting to an optimization class of stage exceeding order SSP Runge–Kutta time discretizations of [26, 29] as long as $p \leq 3$. This class of SSP Runge–Kutta schemes has the advantage of optimizing the polynomial order p of the approximate solution $\boldsymbol{U}_h$ with respect to the r stage of the SSP Runge–Kutta scheme (incidentally satisfying $\mathrm{SSP}(r, p+1)$) in order to minimize the effect of the rigid constraint introduced by the CFL condition on the timestep dt. The limitation on p (*i.e.* $p \leq 3$) is generally more restrictive than we encounter here, and thus, as will become apparent below, in the context of dynamic p-enriched slope limited solutions we are generally unable to exploit these optimization schemes directly.

§3 Methods of regularization

Regularization procedures are frequently studied in the numerical setting with regards to first order hyperbolic subsystems. Doing this from a theoretical point of view is often convenient as the characteristics in the standard scalar tranport setting, for example, are well-defined, and the numerical fluxes can be constructed such that they depend only upon local information from neighboring cells. Moreoso, the following definition provides a particularly simple setting in this limiting case.

Definition 3.1. *A first order system is hyperbolic if the eigenvalues of its Jacobian matrix are real and distinct, or if its eigenvalues are real and the Jacobian matrix of the system is non-defective.*

However, in general physics-based models, it is far more common to run into systems such as (2.1) that are at best convection-dominated, and in general have both parabolic and reactive subsystems adjoined to them. This can lead to substantial complications. First, the presence of a parabolic subsystem not only perturbs the system away from the purely hyperbolic system, but can do so in a way that stiffens the timestepping restriction, globalizes the flux response, and fundamentally changes the eigenstructure of the operator space. Formally we have that:

Definition 3.2. *A first order linear system is parabolic if its eigenvalues are real and the Jacobian matrix of the system is defective, i.e. the determinant of the Jacobian matrix (or Gram matrix) is zero.*

This means, strictly speaking, that the eigenmodes of the parabolic subsystem do not have a complete linearly independent basis. In other words, in contrast to the complete basis in the hyperbolic subsystem, the eigenmodes in the parabolic subsystem can have eigendirections (i.e. eigenvectors) with multiplicity that obey a differing sign valence than the hyperbolic subsystem.

As if this were not complicated enough, the deceptively simple reaction term g adjoins additional subtlety to the system. Now the reactive subsystem comprises a (potentially) nonlinear dynamical subsystem, where the eigenstructure of the Jacobian

matrix of the linearized subsystem can be used broadly to determine local stability features, e.g. whether a fixed point is an attractor, repeller, saddle point, etc. While this reactive Jacobian matrix does not explicitly impact numerical fluxes, it does strongly impact subcellular dynamics within the finite element, and can thus have a large effect on the regularity features of the solution [41].

As a consequence the spatial dynamics of (2.1) can broadly be viewed as being dictated by three spatial operators, of hyperbolic, parabolic, and reactive type, each consisting of uniquely identifiable eigenstructures, stability features, and regularity characteristics. Of course, it is the coupling between these operators that ultimately defines the system itself. The likelihood, in such a broad class, of the solution to (2.1) happening across a numerical shock front becomes nontrivial, and cannot, of course, be calculated exactly without understanding what function spaces the solution to the full system of PDEs in the weakened form lies within.

In this paper we will define the right hand side operator $\mathscr{L}$ as decomposing into three peices, $\mathscr{L} = \mathscr{C} + \mathscr{D} + \mathscr{R}$ — a convective operator $\mathscr{C}$, a diffusive operator $\mathscr{D}$, and a reactive operator $\mathscr{R}$. Moreover we will define a numerical shock as any numerical solution to (2.1) that, when expanded into an approximate local Taylor expansion of order p at any $(x, y) \in \Omega_h$, has a radius of convergence ϱ such that $\varrho < h_{\text{effective}}$ which can of course be explicitely approximated pointwise using h and p. When shocks are present some form of numerical regularization must be applied in order to mollify the solution back into the space of numerically admissible solutions. Below we discuss three such techniques.

3.1 **Slopelimiting using the sharp BDS limiter**

Slopelimiting offers a very robust solution for regularizing solutions that have become numerically unstable and/or develop numerical shocks. Generally, slopelimiters are developed over subdomains of the solution, wherein minmod-type functions are applied to – above all — preserve local maximum/minimum type principles, e.g. over some subdomain (or stencil) $\mathfrak{X}_i \subset \Omega_h$ the solution ι at $\boldsymbol{x}_i$ is bounded from above by $\max_{\mathfrak{X}_i} \iota(\boldsymbol{x})$ and below by $\min_{\mathfrak{X}_i} \iota(\boldsymbol{x})$. These methods can even be adapted to only operate in very spatially or temporally localized regions, where for example, entropy sensors can be used to sense nonlinearities and/or high local gradients [34, 39], and limiting is only performed in such areas.

This kind of approach makes a lot of sense when the right hand side is a hyperbolic operator or the problem is strongly, or completely, convection dominated. Additionally, these methods can have a lot of appeal when conservation properties or global valence (e.g. positivity preservation) of the solution must be preserved. For example, when the physics of the system does not support negative concentrations then undershoots along numerical shock fronts are indicative of nonphysical solutions, and slopelimiters can be devised that assure preservation of such features [34, 51].

In [39] we performed a relatively thorough numerical study on a large family of slopelimiters of a hierarchical type. Broadly, hierarchical limiters can be defined as

slope limiters that utilize hierarchical-type bases to reconstruct solutions over local stencils by applying the local maximum principle in the form of stencilized (or localized) minmod-type function evaluations. There we studied a number of different limiters, including the Durlofsky-Engquist-Osher limiter [19], Barth-Jespersen type limiters [5], vertex type limiters [32], MUSCL type limiters [42], ENO type limiters [1, 35], etc.

Here we briefly review the generalized form of one of the most effective and robust limiters we have analyzed to date: the Bell-Dawson-Shubin (BDS) limiter outlined in [6] and extended in [39] to include higher order terms. For the sake of simplicity and accessibility, let us first present a simplified version of the algorithm in one spatial dimension. Take the vertices x_j of a one dimensional finite element grid, defined as those points that share more than one element. In this case, any vertex x_j shares two elements, which we will denote by Ω_j and Ω_i (Note that Ω_i in one dimension is either Ω_{j-1} or Ω_{j+1}, for example).

Restricting to the linear modes for any degree p polynomial, and setting $\tilde{\iota}_i$ as the cell average, the maximum $\iota_j^{\max}$ and minimum $\iota_j^{\min}$ at any point x_j is taken as the extremum over this vertex-localized stencil, i.e.

$$\iota_j^{\max} = \max(\tilde{\iota}_i, \tilde{\iota}_j) \quad \text{and} \quad \iota_j^{\min} = \min(\tilde{\iota}_i, \tilde{\iota}_j). \tag{3.1}$$

The local linear solution is then evaluated at both vertices of the cell; the left $\iota^L_{j,\text{linear}}$ and the right $\iota^R_{j,\text{linear}}$ vertex, repectively. The following simple minmod function ϱ_j^L on the left vertex, for example, is then used:

$$\varrho_j^L = \max\Big\{ \min\big\{\iota^L_{j,\text{linear}}, \iota_j^{\max,L}\big\}, \iota_j^{\min,L}\Big\}, \tag{3.2}$$

where the superscript L indicates the left vertex, and the same process is subsequently performed at the right vertex. The vertex value on the cell is then reset/limited to $\iota^L_{j,\text{linear}} := \varrho_j^L$.

These limited values $\iota^L_{j,\text{linear}}$ are then averaged over the cell, $\text{Avg}(\iota_{j,\text{linear}}) = \frac{1}{2}(\iota^L_{j,\text{linear}} + \iota^R_{j,\text{linear}})$, where this Average is then used to compute a vertex weighting

$$W_j = 2(\text{Avg}(\iota_{j,\text{linear}}) - \tilde{\iota}_j),$$

which is basically a measure of how much of the excess slope is to be redistributed across the cell. This W_j is used to define a function at *every vertex*, given by

$$\mathfrak{D}_j = \big(\iota^L_{j,\text{linear}} - \tilde{\iota}_j\big)\,\text{sgn}W_j \tag{3.3}$$

where $\text{sgn}(\cdot)$ is the usual signum function, except we set $\text{sgn}(0) := 1$. When $\mathfrak{D}_j$ is positive, it means that either both the average and the approximate solution at the vertex are larger than the cell average $\tilde{\iota}_j$, or they are both smaller than the cell average $\tilde{\iota}_\ell$. Whenever this is the case, it means the local solution has failed to satisfy

a local maximum type principle, in this case simply meaning that overshoots and/or undershoots are present in the solution. To quantify this failure, we sum the $\mathfrak{D}_j$ over each of the cells vertices, and if both evaluations are positive we set a counter $\mathcal{D} = 2$, else $\mathcal{D} = 1$.

This allows us now to generate a vertex-wise redistribution factor $\mathscr{R}_j$ over each vertex, defined simply by setting

$$\mathscr{R}_j = \begin{cases} (W_j \mathrm{sgn} W_j)/\mathcal{D}, & \text{if } \mathfrak{D}_j > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{3.4}$$

where the maximum allowed value $\mathscr{R}_j^{\max}$ is determined by:

$$\mathscr{R}_j^{\max} = \begin{cases} \iota_{j,\mathrm{linear}}^L - \iota_j^{\min} & \text{if} \quad \mathrm{sgn} W_j > 0, \\ \iota_j^{\max} - \iota_{j,\mathrm{linear}}^L & \text{otherwise.} \end{cases} \tag{3.5}$$

The approximate values at the vertices are then updated, where we make sure the maximum redistribution amount is not exceeded, $\mathscr{R}_j = \min(\mathscr{R}_j, \mathscr{R}_j^{\max})$. The redistributed vertex value is updated explicitly to satisfy:

$$\iota_{j,\mathrm{linear}}^L := \iota_{j,\mathrm{linear}}^L - \mathscr{R}_j \mathrm{sgn} W_j. \tag{3.6}$$

The difference is then reweighted for the computation at the right vertex by determining the amount of "slope" still available to redistribute by computing: $W_j := (W_j - \mathscr{R}_j \mathrm{sgn} W_j)$. After the right vertex is redistributed, the two points at the vertices uniquely determine a line, and thus the linear solution on the current cell is determined.

In two dimension the formulation becomes more complicated. Here we define the neighborhood of a point $\boldsymbol{x}_j$ as that object comprised of elements that share a common vertex $\boldsymbol{x}_i$, indexed with respect to the vertices of each cell Ω_{e_j}. The *focal neighborhood* $\Omega_f = \{\Omega_{e_j}\}_i$ (in the sense of the foci of geometric optics, as shown in figure 1) is the collection of elements such that $\boldsymbol{x}_i \in \Omega_{e_j}$, where $\{\Omega_{e_j}\}_i$ is the collection of vertices of neighboring elements that share vertices with a base element Ω_{e_j}, such that $i = 1, 2, 3$ for triangular elements.

Restricting to the sub-quadratic terms of the hierarchical modal basis φ_{ij} for any degree p polynomial, implies we want $i + j \leq 1$. Then setting $\iota_i^{e_j}$ as the constant term of the basis in ι_h of the base element containing coordinates $\boldsymbol{x}_i = (\xi_i, \eta_i)$ in the master element representation, we define the maximum $\iota_i^{\max}$ and minimum $\iota_i^{\min}$ values for each unknown at every $\boldsymbol{x}_i \in \hat{\Omega}_{e_j}$ over the chosen stencil $\hat{\Omega}_{\mathfrak{X}_i}$ (where this can be an edge or focal stencil, or a variation, etc.) as

$$\iota_i^{\max} = \max_{\forall \hat{\Omega}_{e_j} \in \hat{\Omega}_{\mathfrak{X}_i}} \left\{ \iota_i^{e_j} \right\} \quad \text{and} \quad \iota_i^{\min} = \min_{\forall \hat{\Omega}_{e_j} \in \hat{\Omega}_{\mathfrak{X}_i}} \left\{ \iota_i^{e_j} \right\}. \tag{3.7}$$

Next we take the full approximate solution restricted to its sub-quadratic part and evaluated at the three vertices of the cell, denoted by the three values $\iota(\boldsymbol{x}_\ell)|_{i+j \leq 1}$ for

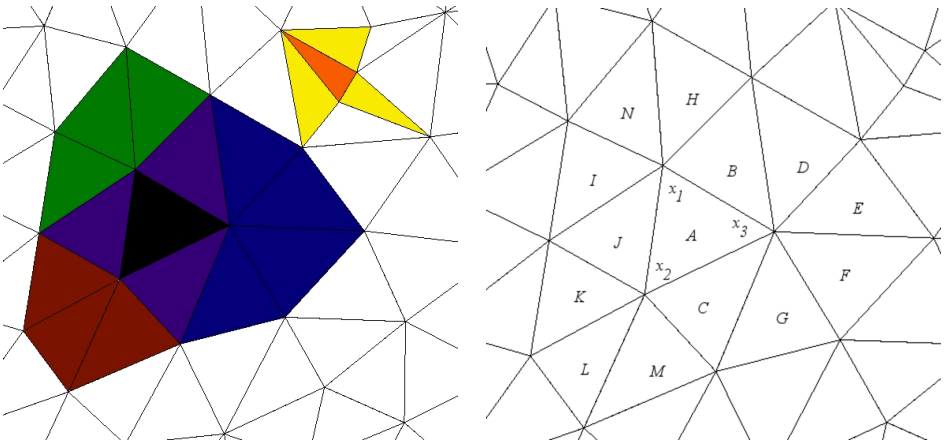

Figure 1: On the left, the *focal neighborhood* Ω_f of a base element Ω_{e_i} filled in black. Green, red and blue are the three *focal neighborhood* groups based at vertices $\boldsymbol{x}_i$ of the black base cell, while purple are cells contained in more than one of the two *focal neighbor stencils* (incidentally comprising the *edge neighborhood* of Ω_{e_i}). In a contrasting geometric locale, the orange base cell's *edge neighbors* Ω_{E_j} are each filled in yellow, comprising the *edge neighborhood* Ω_E. On the right the *focal stencil* Ω_{f_i} and the *edge stencil* Ω_{E_i} of a base element $\Omega_{e_l} = A$. The stencils are defined with respect to the base elements vertices $\boldsymbol{x}_i$ for $i = 1, 2, 3$, such that the *focal stencil* at $\boldsymbol{x}_1$ is $\Omega_{f_1} = \{J, I, N, H, B, A\}$, and likewise $\Omega_{f_2} = \{J, K, L, M, C, A\}$ and $\Omega_{f_3} = \{C, G, F, E, D, B, A\}$. Similarly the *edge stencils* are given by: $\Omega_{E_1} = \{J, B, A\}$, $\Omega_{E_2} = \{J, C, A\}$ and $\Omega_{E_3} = \{B, C, A\}$. Notice that the union of sets recovers the *focal neighborhood* ($\Omega_f = \cup_i \Omega_{f_i}$) and the *edge neighborhood* ($\Omega_E = \cup_i \Omega_{E_i}$), while the restriction of the symmetric difference of sets defines the *focal neighborhood group* ($\ominus_i \Omega_{f_i}|_{\Omega_{f_j}}$) and *edge neighborhood group* ($\ominus_i \Omega_{E_i}|_{\Omega_{E_j}}$) for any vertex j.

$\ell = 1, 2, 3$ corresponding to the vertices, while $i + j$ corresponds to the monomial degree. At each vertex we employ the following minmod function $\varrho_{\boldsymbol{x}_\ell} = \varrho_{\boldsymbol{x}_\ell}(\iota(\boldsymbol{x}_\ell)|_{i+j\leq 1})$:

$$\varrho_{\boldsymbol{x}_\ell} = \max\left\{ \min\left\{ (\iota(\boldsymbol{x}_\ell)|_{i+j\leq 1}, \iota_\ell^{\max} \right\}, \iota_\ell^{\min} \right\}, \tag{3.8}$$

where we subsequently reset the vertex value to $\iota(\boldsymbol{x}_\ell)|_{i+j\leq 1} := \varrho_{\boldsymbol{x}_\ell}(\iota(\boldsymbol{x}_\ell)|_{i+j\leq 1})$.

Proceeding, we estimate the average vertex value over the stencil to its value on the minmod'ed neighborhood by computing, $\mathrm{Avg}_\ell(\iota(\boldsymbol{x}_\ell)|_{i+j\leq 1}) = \frac{1}{3}\sum_\ell \iota(\boldsymbol{x}_\ell)|_{i+j\leq 1}$, and then we calculate a vertex-weighted difference between this average and the cell average $\iota_\ell^{e_j}$,

$$W_\ell = 3\left(\mathrm{Avg}_\ell(\iota(\boldsymbol{x}_\ell)|_{i+j\leq 1}) - \iota_\ell^{e_j}\right). \tag{3.9}$$

The restricted difference functions $\mathfrak{D}_\ell$ are then given with respect to each vertex $\boldsymbol{x}_\ell$,

$$\mathfrak{D}_\ell = \left(\iota(\boldsymbol{x}_\ell)|_{i+j\leq 1} - \iota_\ell^{e_j}\right) \mathrm{sgn} W_\ell \tag{3.10}$$

A schematic of the dynamic adaptive linear restriction method

- (ξ_ℓ, η_ℓ) – vertices of Ω_{e_j}
- $\iota_i^{\max}, \iota_i^{\min}$ – the extrema over the stencil $\Omega_{\mathfrak{X}_i}$
- $\varrho_{\boldsymbol{x}_\ell}$ – the minmod function at (ξ_ℓ, η_ℓ)
- W_ℓ – the vertex-weighted difference of averages
- $\mathscr{R}_\ell$ – the redistribution factor
- $\iota(\boldsymbol{x}_\ell)|_{i+j\leq 1}$ – the updated solution at (ξ_ℓ, η_ℓ)
- $\iota_{ij}|_{i+j\leq 1}$ – the updated monomial coefficents

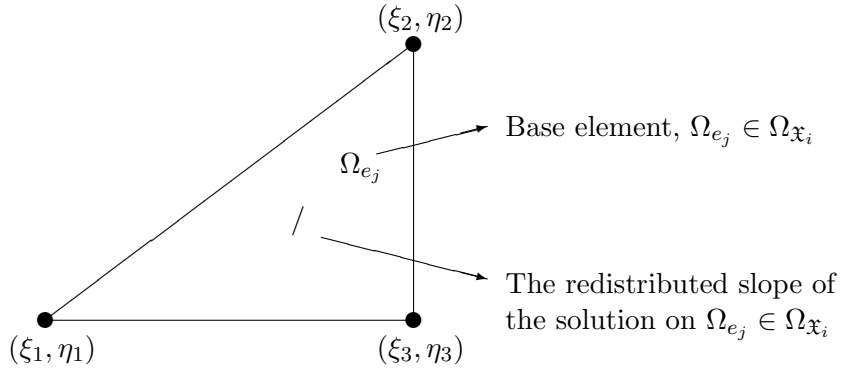

Figure 2: A key for the BDS limiter. As dependent on the stencil $\Omega_{\mathfrak{X}_i}$, the linear part of the full solution of order p, and on a redistribution strategy that can be thought of heuristically as "a consistent redistribution of linear gradients."

where $\mathrm{sgn}(\cdot)$ is the usual signum function except that $\mathrm{sgn}(0) := 1$. Then, if $\mathfrak{D}_\ell$ is positive, which means that either both the average and the approximate solution at the vertex are each larger than $\iota_\ell^{e_j}$, or similarly that they are both smaller than $\iota_\ell^{e_j}$, then we set:

$$\mathcal{D} = \max\left(1, \sum_{m=0}^{I} 1\right), \quad \text{where } I = \sum_{\ell} \mathrm{sgn}\mathfrak{D}_\ell, \quad \text{for each } \boldsymbol{x}_\ell \text{ restricted such that } \mathfrak{D}_\ell > 0. \tag{3.11}$$

This allows us now to generate a vertex-wise redistribution factor $\mathscr{R}_\ell$ over each element, defined simply by setting

$$\mathscr{R}_\ell = \begin{cases} (W_\ell \mathrm{sgn} W_\ell)/\mathcal{D}, & \text{if } \mathfrak{D}_\ell > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{3.12}$$

where the maximum allowed value $\mathscr{R}_\ell^{\max}$ is determined by:

$$\mathscr{R}_\ell^{\max} = \begin{cases} \left(\iota(\boldsymbol{x}_\ell)|_{i+j\leq 1} - \iota_\ell^{\min}\right) & \text{if} \quad \mathrm{sgn} W_\ell > 0, \\ \left(\iota_\ell^{\max} - \iota(\boldsymbol{x}_\ell)|_{i+j\leq 1}\right) & \text{otherwise.} \end{cases} \tag{3.13}$$

The approximate values at the vertices are then updated, where we make sure the maximum redistribution amount is not exceeded, $\mathscr{R}_\ell = \min(\mathscr{R}_\ell, \mathscr{R}_\ell^{\max})$. The redistributed vertex value is updated explicitly to satisfy:

$$\iota(\boldsymbol{x}_\ell)|_{i+j\leq 1} := \iota(\boldsymbol{x}_\ell)|_{i+j\leq 1} - \mathscr{R}_\ell \text{sgn} W_\ell. \tag{3.14}$$

Finally we make sure that the difference is properly reweighted for the next computation at the elements next vertex (if one exists) by determining the amount available to redistribute by computing: $W_\ell := (W_\ell - \mathscr{R}_\ell \text{sgn} W_\ell)$. This proceeds until no vertices are left to evaluate in the cell.

Thus we arrive with the sub-quadratic approximate solution, but what we need are the coefficients on φ_{10} and φ_{01} in the basis. To get these we must simply invert the following local constant matrix:

$$\begin{pmatrix} \varphi_{00}(\boldsymbol{x}_1) & \varphi_{10}(\boldsymbol{x}_2) & \varphi_{01}(\boldsymbol{x}_3) \\ \varphi_{00}(\boldsymbol{x}_1) & \varphi_{10}(\boldsymbol{x}_2) & \varphi_{01}(\boldsymbol{x}_3) \\ \varphi_{00}(\boldsymbol{x}_1) & \varphi_{10}(\boldsymbol{x}_2) & \varphi_{01}(\boldsymbol{x}_3) \end{pmatrix} \begin{pmatrix} \iota_{00} \\ \iota_{10} \\ \iota_{01} \end{pmatrix} = \begin{pmatrix} \iota(\boldsymbol{x}_1)|_{i+j\leq 1} \\ \iota(\boldsymbol{x}_2)|_{i+j\leq 1} \\ \iota(\boldsymbol{x}_3)|_{i+j\leq 1} \end{pmatrix}, \tag{3.15}$$

which provides the unknowns.

The extension to higher order terms is relatively straightforward and is discussed in detail in [39]. The basic idea is that linear recombinations are evaluated about linearizations of the higher order terms, and limited from the top degree down to the sub-quadratic degree. It should be noted that our previous studies suggest that while the higher order extensions can improve stability features, they also, in general, tend to reduce the convergence order of the solution, or, at least, cannot be guaranteed not to reduce this order. Because of this aspect of the higher order extension, along with the substantial computational cost incurred from higher order limiters, we recommend using the first order form of the BDS limiter for nonlinear application models.

It is also important to note here that implementing the BDS limiter for nodal DG methods is slightly different than its implementation for its modal counterpart. When performing the limiting in the nodal basis a couple of things must be kept in mind. First, the nodal basis must be interpolated down to the lower order linear basis in order to perform the reconstruction. Once in the lower degree representation, the basic outline of the limiting procedure can then be followed as long as the support points include the vertices of the cell. If not, then the value at the vertices must be recovered from the solution at the support points, which can be an additional step in the limiting procedure. Finally, after the limiting has been performed, the solution must be interpolated back up into the proper degree basis in order to perform the remainder of the calculation.

3.2 **Modal filtering**

An alternative to slope limiting, which is based on adding dissipative modal diffusion to the solution, is to regularize using simple spectral filtering. The immediate advantages of spectral filtering are: 1) ease of implementation, 2) relative computational

expense compared to many slope limiters, and 3) relative robustness for high polynomial degree p simulations. For a more comprehensive review of what spectral filters are, and how they are used, we refer the reader to [36], [21], [37]. Here we give a basic review of some salient features of these methods.

A spectral filter applied to (2.1) can be motivated as a restriction to the parabolic subsystem in the sense of timesplitting, or fractional multistepping. That is, if we consider in one coordinate dimension with constant $\delta \in \mathbb{R}^+$,

$$\iota_t - \delta(-1)^{1+\tilde{s}} \frac{\partial}{\partial x}\left(f(\iota)\frac{\partial}{\partial x}\iota\right) = 0, \tag{3.16}$$

and notice that (2.4) provides that up to a change of polynomial basis (i.e. linear transformation), the Sturm-Liouville equation is satisfied for positive $f = m(x)$, such that

$$-\frac{d}{dx}\left(m(x)\frac{d}{dx}N_l^i\right) + l(l+1)N_l^i = 0,$$

where then, for example, using a forward Euler time integration it follows that elementwise

$$\begin{aligned}\iota_h^{n,i} &= \iota_h^{n-1,i} + dt^n \delta(-1)^{\tilde{s}+1} \sum_{l=1}^{n_p+1} (l(l-1))^{\tilde{s}} \iota_l^{n-1,i} N_l^i \\ &\approx \sum_{l=1}^{n_p+1} \sigma\left(\frac{l-1}{n_p+1}\right) \iota_l^{n-1,i} N_l^i, \quad \text{with } \delta = \left(\frac{1}{dt(n_p+1)^{2\tilde{s}}}\right).\end{aligned}$$

This follows for any explicit time integrator, where the choice of δ is easily generalized accordingly. Also note that N_l^i can be chosen as a Legendre polynomial basis, for example, in which case $h = 1 - x^2$ for example.

The filter function $\sigma = \sigma(\varsigma)$ itself, for nonnegative $\varsigma \in \mathbb{R}^+$ is required to satisfy the following conditions:

$$\sigma(\varsigma)\begin{cases} = 1, & \varsigma = 0 \\ \leq 1, & \varsigma \leq 1 \\ = 0, & \varsigma > 1 \end{cases}$$

Many candidate filters σ exist, though in this work we will utilize the so-called exponential-type filter $\sigma = e^{-\alpha\varsigma^{\tilde{s}}}$, where $\tilde{s}$ is a positive integer, $\alpha \approx -\log_{10}(\epsilon_M)$, ϵ_M is machine epsilon, and $\varsigma = \mathfrak{l}(l)/(p+1)$ such that $\mathfrak{l}$ is a basis dependent mapping that gives the monomial degree of the evaluated degree of freedom in the polynomial basis.

3.3 **Artificial diffusion for modal coefficients**

In contrast to slope limiting and spectral filtering, a third regularization method is artificial diffusion, or often referred to as artificial viscosity. In artificial diffusion,

the equation itself is perturbed slightly away from its native form. For example, in the slopelimiting case for a scalar transport equation, $f = \epsilon(\iota) + \eta$ and is set to zero by setting $\epsilon = \eta = 0$ where $g(t, x, y) = 0$, such that pure hyperbolic transport is recovered. In this case the slopelimiter, as discussed above, comes in to preserve a local maximum principle over a restricted subdomain (i.e. stencil). Again, this is essentially an a posteriori step that takes an initial state, "senses small-scale oscillatory behavior," adjusts relative to it, and reconstructs a smoother state at every timestep (or stage, etc.).

In contrast to this, the artificial diffusion can be viewed from a slightly different perspective. In the case of artificial diffusion, the assumption is again that the preasymptotic behavior of the solution is unstable due to solutions demonstrating subcellular nonlinearities, e.g. a convergence radius smaller than that supported by h and p. But now, in order to recover a stable solution, the underlying mathematical system of equations is perturbed away from that of purely hyperbolic transport by setting $f = \epsilon(\iota)$, with $\epsilon(\boldsymbol{x}, t) \neq 0$ for some $\boldsymbol{x} \in \Omega_h$. Notice the effect that this has. Now, in contrast to the spectral filter and the slopelimiter, the regularization here is mediated directly by the elliptic operator $\mathscr{D} = \nabla \cdot (f \nabla \iota)$ present in the the parabolic subsystem, $\partial_t \iota = \mathscr{L}$. Of course, it should be noted that in an indirect way both slopelimiting and spectral filtering (or spectral diffusion) can also be viewed as types of artificial diffusion, but in these cases the artificial diffusion is not a part of the mathematical system, but merely a discrete postprocessing step performed after solving the discrete system. More concretely, in artificial diffusion the right hand side operator (e.g. $\mathscr{L} = \mathscr{C} + \mathscr{D} + \mathscr{R}$) is updated relative to a rescaling of the parabolic contribution $\mathscr{D}$ to the solution, while in slopelimiting and spectral filtering, the entire rand hand side $\mathscr{R}$ is rescaled.

In the DG setting the first effect using a second order operator has, is it sends the regularization operation into the unified framework [3]. The second effect this has, is that the local maximum principle is no longer enforced over a stencil relative to the solution variation in ι_h, meaning that subcellular oscillations are no longer artificially excised from the solution, but rather at most, simply dampened.

That said, the crux of the artificial diffusion methods lies in the construction of the function $\epsilon(\iota)$. In simplified cases, the most efficient and precise approach is to use the systems entropy functional $\mathfrak{E}(\iota)$ to construct ϵ, in which case rigorous numerical analysis can assure convergence behavior and ϵ can be weighted properly in p and h to assure that the asymptotics of the numerical analysis push through. This case, when made fully formal, can be characterized as the entropy viscosity method [52]. However, in the nonlinear setting, which we are focused on here, the entropy method can be highly unwieldy for a number of reasons. First, in nonlinear physically driven systems the form of the reaction/source term $g(\iota)$ can fundamentally change the entropy form of the equation. This is of course also true, though to a lesser extent, with the dissipative coefficient $f(\iota)$. The effect this has then, is that when using the entropy formulation for the regularity sensor, the sensor becomes intrinsically linked to the

exact form of the base equation, making its ability to be generalized over all forms that $g(\iota)$ might take in practice, an intractable problem. As a consequence, in our setting, we are more interested in a robust heuristic construction for artificial diffusion in the context of potentially nonlinear convective-diffusive-reaction equations..

One should also note the differences in the nodal and modal implementations of artificial diffusion. In nodal DG, the artificial diffusion $\epsilon(\iota)$ is naturally parameterized by the position of the solution $\boldsymbol{x}$. This is because the nodal DG solution varies as a function of space at support points, allowing $\epsilon(\boldsymbol{x})$ to vary within a single cell. As a consequence, in nodal DG, pointwise entropy-type sensors are frequently used to test for local (subcellular) regularity. An entropy-type sensor might also be called a regularity sensor, where entropy does not necessarily refer to the exact physical entropy of the system, but rather more heuristically as a measure of the relative local disorder, energy, or variation present in the solution.

In the modal DG setting, however, the situation is different as the solution coefficients $\iota_l^i(t)$ do not vary across the cell, but only across the monomial degrees of freedom l of the basis. In this setting entropy-type sensors must be cell-based rather than subcellular, and the behavior of the artificial diffusion must be uniform over the cell. In this sense, it is appropriate to view modal DG as a spatially spectral method (e.g in contrast to standard spectral methods which are spectral in frequency space) and artificial diffusion as a type of local spectral diffusion/viscosity [46].

The artificial term $\epsilon(\iota)$ is frequently determined using local regularity (or entropy) measures/sensors that are meant to sense the onset and formation of local shock fronts. In fact, in the case of entropy viscosity [52], it is observed that the correct choice of $\epsilon(x, y, t)$ can allow for a remarkably loose CFL condition, that does not always seem to (strongly) require the more restrictive h^2 dependency inferred from the adjoined parabolic subsystem. In some cases [53], the CFL condition for the hyperbolic subsystem can be used as data to solve the coupled entropy system.

In the general setting, these sensors can be easily formed by way of modal truncation or through the use of mathematical entropies. In our case, since we are concerned with a reaction-diffusion-advective flow, we develop a regularity sensor S_{ij} that measures spectral irregularity as a function of the modal growth in the spatial expansion. That is,

$$S_{ij}(t, \boldsymbol{x}) = \begin{cases} \log_{10}\left(\frac{L^2_{loc}(\tilde{\iota}^i_{hp})}{L^2_{loc}(\iota^i_{hp})}\right), & \text{if } \iota^i_{hp}(t, \boldsymbol{x}) > \varepsilon \\ K, & \text{otherwise} \end{cases}$$

for $\Omega_i \in \Omega_h$ and $j = 1, 2, \ldots$ where j runs over the indices of the state vector ι, K is a large negative real number, ι^i_{hp} is the approximate solution, and $\tilde{\iota}^i_{hp}$ is the modal expansion of the solution vector for all but the constant mode, e.g. in the hierarchical basis $\tilde{\iota}^i_{hp}(t, \boldsymbol{x}) = \sum_{l=2}^{n_p} \iota^i_l(t) N^i_l(\boldsymbol{x})$.

This sensor is then used to determine which cells Ω_i are to be "regularized" by way of adding nonzero ϵ_i. These entropy sensors can also be used to make the diffusion proportional to the entropy itself componentwise, $\epsilon_{ij} \propto S_{ij}$. Each component of the

artificial diffusion ϵ_j is calculated cell-wise by setting

$$\epsilon_{ij} = \begin{cases} 0, & \text{if } S_{ij} < S_0 + \kappa \\ \tilde{\epsilon}_{ij} |S_{ij} c_j + \mathscr{B}_j|^{p_i}, & \text{if } S_{ij} \geq S_0 + \kappa \end{cases}$$

where c_j is a nonnegative constant, p_i is the modal degree of the cell Ω_{e_i}, and $\mathscr{B}_j \in \mathbb{R}$. Generic starting points for the tuning parameters are $S_0 \sim \max_{\Omega_h} S(0, \boldsymbol{x})$, $\tilde{\epsilon}_{ij} \sim (h/p)^2$, and $\kappa \sim \mathrm{Avg}_{\Omega_h} S(0, \boldsymbol{x})$. It is of value to note that in order to recover the formal properties of the artificial diffusion in the numerical formalism, such as vanishing in the $h \searrow 0$ limit, $\epsilon_{ij} = \epsilon_{ij}(h_i, p_i)$ must have properly weighted dependencies in h and p such that it vanishes in the limit for the specific system of equations.

§4 Numerical tests

4.1 A classical test case

In order to parse some of the basic features and differences between the regularity precedures from §3, we solve a classical rotating shock solution to a scalar transport equation. This is of course a hyperbolic conservation law, though it serves as a very effective way of teasing out the subtle differences in each of the three regularization methods discussed above.

Formally we are interesting in single component hyperbolic advection,

$$\partial_t \iota + \boldsymbol{u} \cdot \nabla_x \iota = 0, \tag{4.1}$$

with initial-boundary data given by

$$\iota_{|t=0} = \iota_0, \quad \text{and} \quad \iota_b = 0,$$

corresponding to vanishing Dirichlet conditions, given a time-independent velocity vector field $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x})$ with the transported scalar quantity $\iota = \iota(t, \boldsymbol{x})$ in two dimensions, such that $\boldsymbol{x} = (x, y)$ and $\boldsymbol{u} = (u, v)$.

We choose the domain $\Omega = \left[-\frac{1}{2}, \frac{1}{2} \right]^2$ with velocity field $\boldsymbol{u} = (y, -x)$. Letting $\tau_{\mathcal{O}} = \pi/4$ and defining the auxiliary variables

$$\mathcal{O}_x = x \cos \tau_{\mathcal{O}} - y \sin \tau_{\mathcal{O}} \quad \text{and} \quad \mathcal{O}_y = y \cos \tau_{\mathcal{O}} + x \sin \tau_{\mathcal{O}},$$

the initial data satisfies

$$\iota_0 = \begin{cases} 1, & \text{if } A, \\ 1 - Ba^{-1}, & \text{if } B \leq a, \\ \frac{1}{4}(1 + \cos \pi r), & \text{otherwise,} \end{cases} \tag{4.2}$$

where

$$A = (a_0 \leq B \leq a) \wedge (\mathcal{O}_x \leq a_1), \qquad B = \sqrt{\left(\mathcal{O}_x - \frac{1}{4} \right)^2 + \mathcal{O}_y^2},$$

and

$$r = a^{-1} \min\left(a, \sqrt{\mathcal{O}_x^2 + (\mathcal{O}_y + 1/4)^2}\right).$$

Here we choose $a = 0.18$, $a_0 = 0.025$ and $a_1 = -0.23$.

The exact solution may be determined by noticing that for any $F(x, y)$, where $x = x(t)$ and $y = y(t)$,

$$\frac{dF}{dt} = \partial_t F + \begin{pmatrix} x' \\ y' \end{pmatrix} \nabla F = 0.$$

This implies that for

$$\boldsymbol{u} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} y \\ -x \end{pmatrix}, \quad \text{we have the system} \quad x' = y \quad \text{and} \quad y' = -x,$$

such that the solution to the second order ODE, $y'' + y = 0$ can be viewed as a generator of the rotation matrix R about the origin. That is, we obtain the clockwise transformation

$$R = \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix}, \tag{4.3}$$

such that $R\boldsymbol{x}$ yields the exact solution.

Some numerical results are presented in Table 1, with a number of interesting observations. First, it should be noted that one of the substantial advantages of the BDS limiter is that it does not require tuning. In nonlinear application settings, having to tune parameters can add substantial development and research time to any problem, and the importance of not having to do so cannot be understated. However, that notwithsdtanding, and even though the BDS limiter is one of the strongest available limiters [39], its accuracy behavior is not the most effective of the tested regularization methods. The BDS limiter also shows very minimal error reduction in p, which though nonideal, is, of course, also expected. For $p = 1$ the BDS limiter is very competitive with the nonregularized (e.g. best approximate) solution in the L^2-norm. As p increases though the BDS limiter fails to show the behavior one might hope for. Looking at figure 3, it is interesting to see that the BDS limiter is by far the most destabilizing (as a measure of L^∞-error) in the very short time frame of a couple hundred timesteps. In fact, on closer perusal, it seems that this behavior is being caused by the local maximum principle behavior over the stencil, where the BDS limiter almost completely eradicates all the sub-stencil extrema in the solution. That is, on the stencil $\mathfrak{X}_i$ the minmod function $\varrho_{\boldsymbol{x}}$ tends to smear out (and linearize) almost all subcellular features within the domain of influence of $\mathfrak{X}_i$. This perturbation away from the initial state of the system causes a spike in the L^∞-error, that calms after several thousand applications of the RHS operator (at RKSPP(6,4)).

Another interesting observation and comparison between the BDS limiter and the nonregularized solution can be seen in figures 4 and 5, where the BDS limiter dramatically smooths along sharp gradients. Again, this is the long time behavior of

p	**Regularization**	$\frac{L^2\text{error}}{L^\infty\text{error}}$	Parameters
1	None	$\left(\frac{0.257596956362332}{0.473192221849165}\right)$	No tuning
1	BDS limiter	$\left(\frac{0.255916107446837}{0.514338985954558}\right)$	No tuning
1	Modal filter	$\left(\frac{0.258484813737977}{0.474687497032311}\right)$	$\alpha = 10, \tilde{s} = 4$
1	Artificial diffusion	$\left(\frac{0.257078135845103}{0.469509688235644}\right)$	$\epsilon = 2.5 \times 10^{-6}, \kappa = -1$
2	None	$\left(\frac{0.170738764376758}{0.354097814066434}\right)$	No tuning
2	BDS limiter	$\left(\frac{0.223121738467467}{0.455107484317801}\right)$	No tuning
2	Modal filter	$\left(\frac{0.175711150561175}{0.348555573453897}\right)$	$\alpha = 10, \tilde{s} = 4$
2	Artificial diffusion	$\left(\frac{0.178933438994748}{0.339032435941110}\right)$	$\epsilon = 2.5 \times 10^{-6}, \kappa = -1$
3	None	$\left(\frac{0.126613945270710}{0.257554046914193}\right)$	No tuning
3	BDS limiter	$\left(\frac{0.218639108980124}{0.482120911655806}\right)$	No tuning
3	Modal filter	$\left(\frac{1.84964279569021}{0.733852764586520}\right)$	$\alpha = 10, \tilde{s} = 2$
3	Modal filter	$\left(\frac{0.722437687707741}{0.596532542993910}\right)$	$\alpha = 10, \tilde{s} = s$
3	Artificial diffusion	$\left(\frac{0.128502876699574}{0.267339777576446}\right)$	$\epsilon = 2.5 \times 10^{-7}, \kappa = -1$
3	Artificial diffusion	$\left(\frac{0.152133348851204}{0.318468351656306}\right)$	$\epsilon = 2.5 \times 10^{-6}, \kappa = -1$
3	Artificial diffusion	$\left(\frac{0.193842852653945}{0.390162789772915}\right)$	$\epsilon = 2.5 \times 10^{-6}, \kappa = -5$

Table 1: The L^2 and L^∞-errors of the approximate solutions after one full rotation with respect to (4.1), setting a fixed $h = 1/256$, $dt = 8 \times 10^{-4}$ and using Runge–Kutta SSPRK$(6, 4)$. For the artificial diffusion runs, all have used $c = 0, \mathscr{B} = 1$.

the limiter, as in the short time, the edge gradients are actually amplified, though after multiple applications the solutions stabilizes to a very regularized form. We have also observed that though the BDS limiter is very regularizing in the long term, the short term behavior can be potentially problematic for the auxiliary formulation, where the gradient field $\boldsymbol{\sigma}$ can become highly unstable, such that scalar transport coupled to physical diffusion becomes quite delicate. The reason for this is related again to the above mentioned behavior, where the minmod function resets the cellular values of the functions without reconstructing the gradient values $\boldsymbol{\sigma}$. Hence, on the subsequent timestep evaluation, the base field ι is smooth, but the gradient field $\boldsymbol{\sigma}$ is constructed from the previous timestep prior to limiting, and is thus inconsistent with the new field. The simple solution to this is to run the limiter multiple times per Runge-Kutta stage, once before the auxiliary equation is computed, and once after the RHS operator is evaluated. While this can be very effective at the level of regularization, it can also become prohibitively expensive computationally. It should also be noted that this behavior relative to the auxiliary forms of $\boldsymbol{\sigma}$ is a common

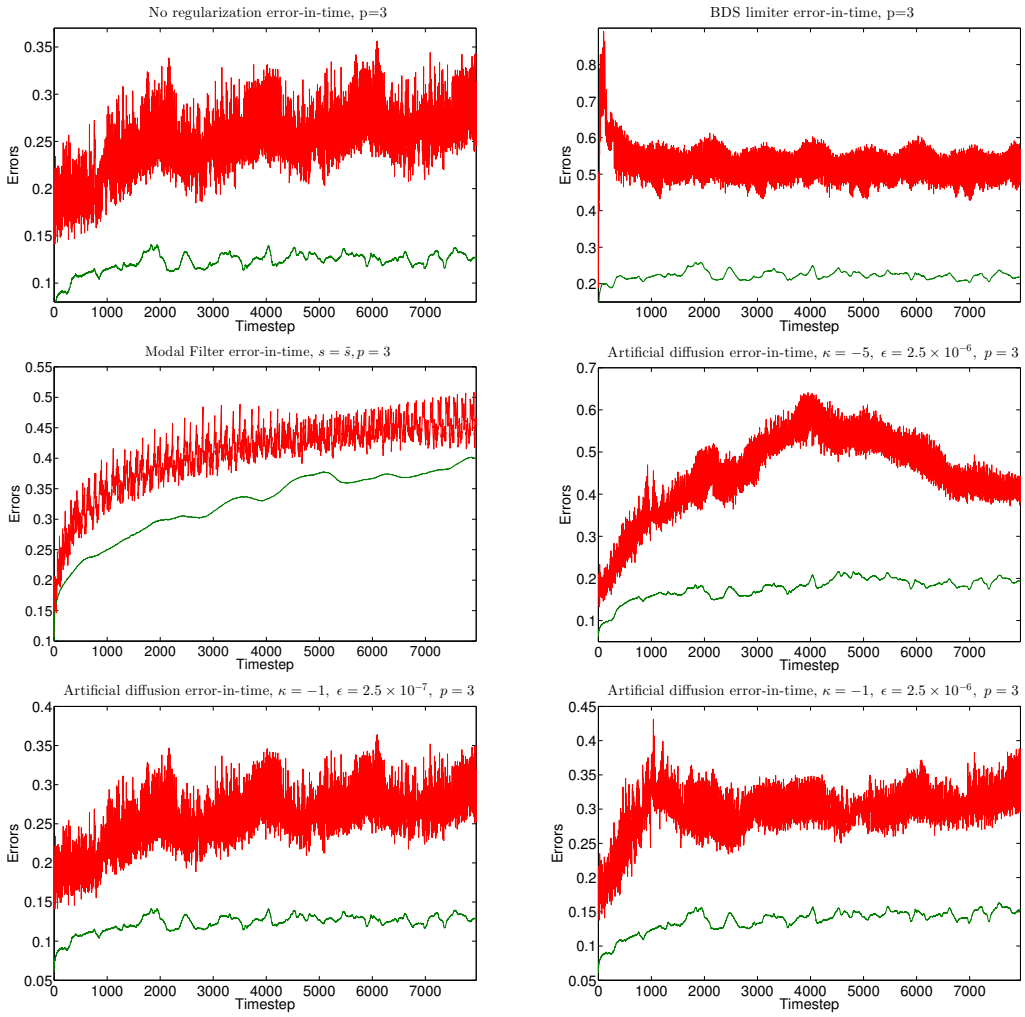

Figure 3: The error-in-time behavior over 7956 timesteps. The bottom graph in each is the L^2-error, and the top is the L^∞-error.

feature of nearly all slopelimiters. Finally, observe in table 2 and figure 5 that a final curious feature of the BDS limiter, is an induced bias relative to the fixed geometry of the unstructured mesh. This behavior is easy to understand, once one understands that the algorithm from section 3.1 is ordinal in the sense that the order in which vertices are evaluated determines the relative weighting of the redistribution factor.

The modal filter also demonstrates interesting behavior. Notice in table 1 that especially at $p = 1$ the modal filter demonstrates very little change in accuracy from the nonregularized solution, though it is slightly more diffuse. As p increases, the modal filter moves further away from the nonregularized solution, though the choice of $\tilde{s}$ has a large effect on the error. The reason for this is again fairly simple. In the

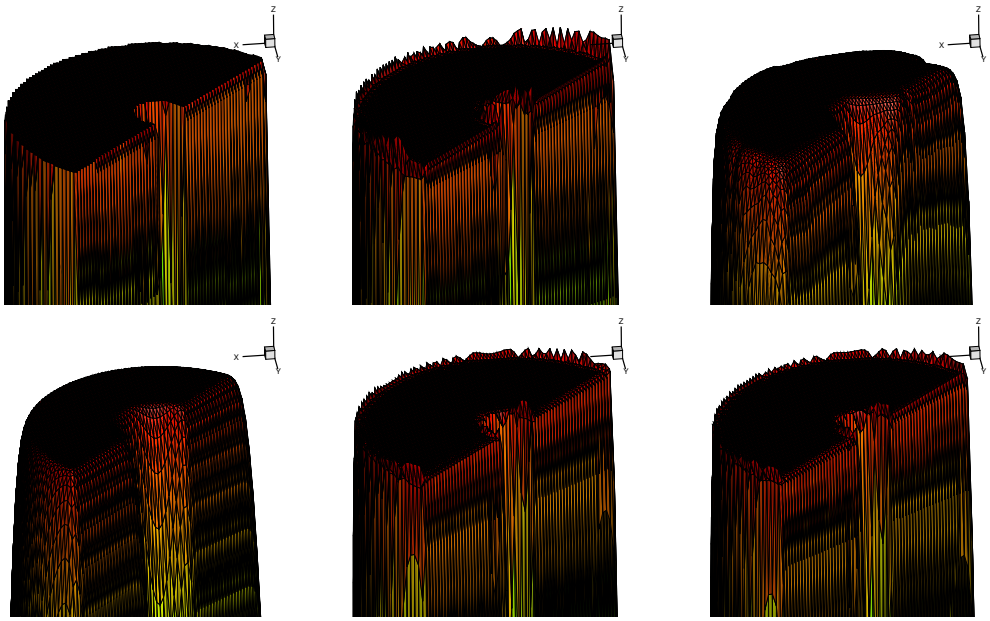

Figure 4: Here we show the sharpest discontinuity island, in clockwise order from top left, with the initial state, then end states after one half of a clockwise rotation at $p = 3$, $h = 1/256$ using: L^2-projection, no regularization, BDS limiter, artificial diffusion with $\epsilon = 2.5 \times 10^{-6}$ and $\kappa = -5$, artificial diffusion with $\epsilon = 2.5 \times 10^{-6}$ and $\kappa = -1$, and a modal filter with $\tilde{s} = 2$.

modal filter the average value (i.e. the constant mode in the polynomial expansion) is not (in general) ever adjusted, it is only the higher order modes that are diffused (one can of course diffuse these modes, but this leads to a very globally diffuse method). Thus the impact on the solution is similar to slopelimiter, where when the value of $\tilde{s}$ is set too high/low relative to the weight of the modal expansion, the effect is either minimal or the amount of diffusion can be catastrophic as seen in table 1, when $p = 3$ and $\tilde{s} = 2$. Nevertheless when tuned correctly, the modal filter can smooth the solution very nicely, and as seens in figures 3 and 5, causes less eratic response in the error profile. Nevertheless, the modal filter does not preserve local maxima or minima, but rather globally diffuse the solution in a way very similar to the BDS slopelimiter.

Finally, the artificial diffusion solutions offer a dramatically different type of regularization behavior than the modal filter or the slopelimited solutions. As shown in figure (3), in the case of the nonregularized solution and the modal filter, the solution error slowly but steadily grows in time, as would be expected from solutions with no discontinuities. In the BDS limiter one witnesses the opposite behavior. Large initial error is induced, and is subsequently dampened over the timesteps until the system seems to reach a relatively steady state in the error behavior. In the case

p	**Regularization**	$\frac{L^2\text{error}}{L^\infty\text{error}}$	Parameters
3	None	$\left(\frac{0.121672034988564}{0.261596521446734}\right)$	No tuning
3	BDS limiter	$\left(\frac{0.221578306991342}{0.542707704298091}\right)$	No tuning
3	Modal filter	$\left(\frac{0.653552134431015}{0.629007003412417}\right)$	$\alpha = 10, \tilde{s} = 2$
3	Modal filter	$\left(\frac{0.278944826656417}{0.451577854625951}\right)$	$\alpha = 10, \tilde{s} = s$
3	Artificial diffusion	$\left(\frac{0.190501786334666}{0.298360381752994}\right)$	$\epsilon = 5 \times 10^{-4}, \kappa = -1$
3	Artificial diffusion	$\left(\frac{0.188748772436751}{0.292264396683535}\right)$	$\epsilon = 2.5 \times 10^{-5}, \kappa = -1$

Table 2: The L^2 and L^∞-errors of the approximate solutions after a quarter rotation with respect to (4.1), setting a fixed $h = 1/512$, $dt = 1 \times 10^{-4}$ and using Runge–Kutta SSPRK$(6, 4)$, again using $c = 0, \mathscr{B} = 1$.

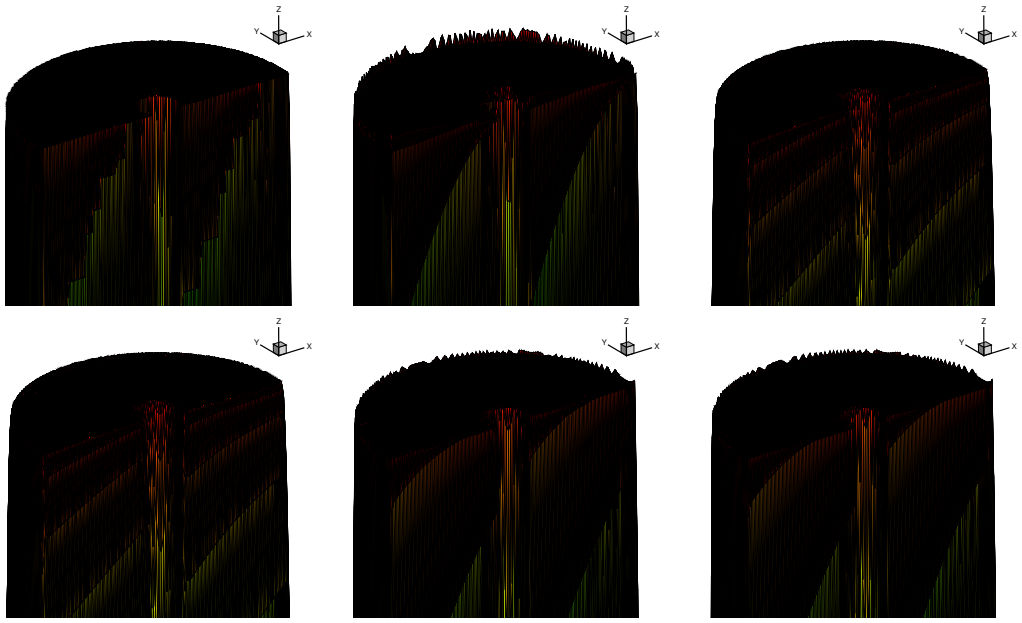

Figure 5: The same simulation as in figure 4, including a single refinement to $h = 1/512$. In clockwise order from top left, with the initial state, then end states after $1/4$ clockwise rotation at $p = 3$: L^2-projection, no regularization, BDS limiter, artificial diffusion with $\epsilon = 2.5 \times 10^{-5}$ and $\kappa = -1$, artificial diffusion with $\epsilon = 5 \times 10^{-4}$ and $\kappa = -1$, and a modal filter with $\tilde{s} = s$.

of artificial diffusion, the tuning parameters provide enough flexibility so that the error behavior over time can be tuned to essentially whatever response type is most physically/mathematically justifiable.

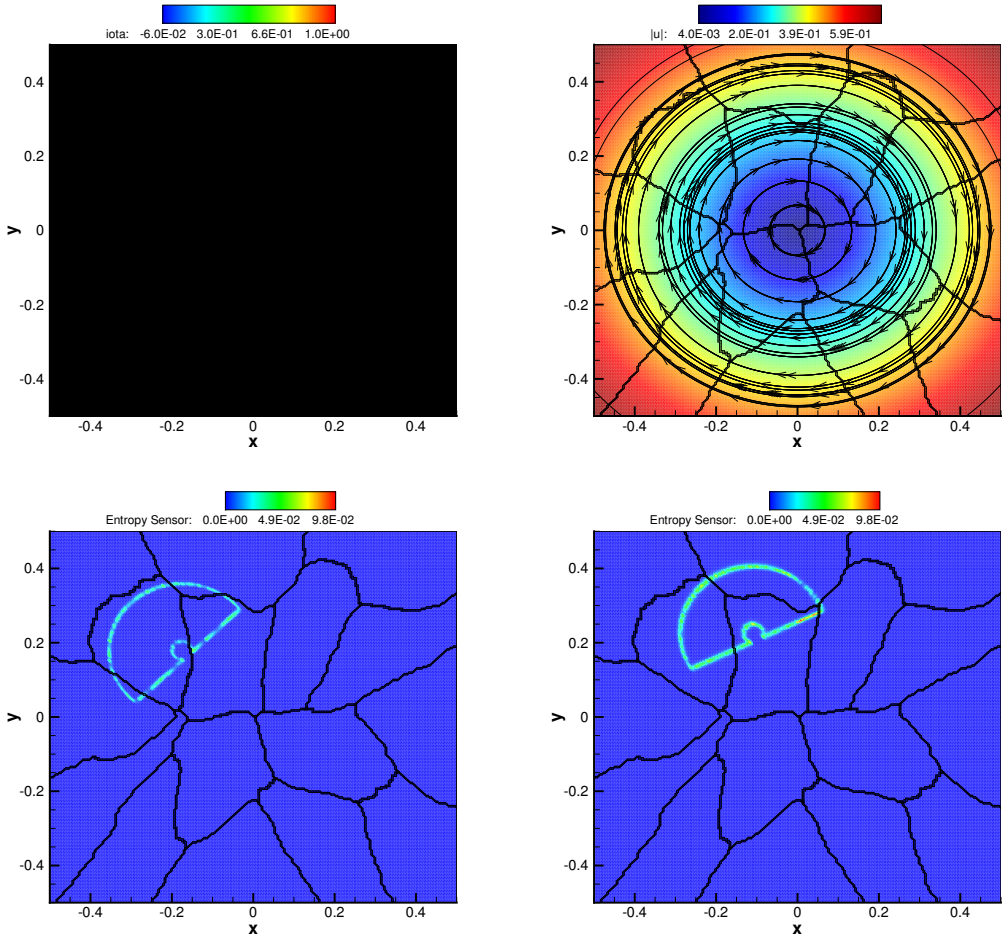

Figure 6: The top left shows an overhead view of the L^2-projected initial condition with mesh outline, while the top right is norm of the constant background velocity field with streamlines included. The larger subdomains here outine the 16 processor subdomains of this parallel simulation. The bottom left is the entropy sensor S after 15 timesteps, and the bottom right S after 405 timesteps.

To explain this behavior let us first explain the response of the entropy sensor S_{ij} to a solution profile. The entropy sensor can be used to isolate areas where the nonlinearities in the solution (not necessarily in the PDE itself) start to dominate its modal expansion. In this example case, these areas correspond spatially to coordinates where the gradient of the solution becomes unbounded. This is observable in figure 6, where the entropy sensor gets closest to zero, from below, near the sharpest interchanges, and the resulting domain of disorder tends to grow in time. As a result, the larger the κ the sharper, or tighter, becomes the domain of influence wherein the

artificial diffusion ϵ_i is acting.

For example, in figure 3 the solution of the artificial diffusion at relatively tight settings $\kappa = -1$ with low artificial diffusion $\epsilon = 2.5 \times 10^{-7}$ has an error behavior that is reminiscent of the modal filter and the nonregularized solution. This is because the diffusion is of too low in magnitude to diffuse the Runge oscillations along the steep gradients. However, when the diffusion is slightly raised to $\epsilon = 2.5 \times 10^{-6}$, the behavior becomes quite different. Now, as seen in Table 1, at low p the optimal error can actually be reduced by diffusing out the spurious modes along the edges (without allowing this diffusion to spread to neighboring regions). This behavior is interesting to observe in figure 3, where the error does indeed grow in time still, as with the nonregularized solution, though does so without inducing a strong perturbative spike like the BDS limiter. In fact, even at $p = 3$ the error after one full revolution is still substantially smaller than that of the BDS limiter, for example. Moreso, one can see in figures 4 and 5 that this solution has almost completely dampened the undershoots and overshoots in the solution, and moreover, it has achieved this without excising any subcellular and sub-stencil extrema from the solution.

These settings can nevertheless be quite delicate. For example, in the looser setting on S_{ij} where $\kappa = -5$ and $\epsilon = 2.5 \times 10^{-6}$, the behavior of the artificial diffusion can tend to dominate, and in so doing begin to pollute the solution with spurious oscillatory modes. That is, even though the error behavior of this solution in Table 1 seems to suggest a better solution than, for example, the BDS limiter, and even though figure 3 suggests that after a stabilization in the diffusive modes the diffusion starts to dominate the solution and the error, the solution itself demonstrates a very noticeable ripple forming across the top of the plateau (graphs not shown). This plateau should, of course, remain at a constant value along the scalar transport, and so the fact that it is picking up spurious modes far into the domain that are bleeding over from the Runge oscillations along the edges is reason for concern. This of course only serves to underscore the delicacy of requiring tuning parameters in complicated largescale algorithms.

4.2 **Validation**

The test from §4.1 offers clean insight into the response of the algorithms, but it remains to see how the regularization methods perfom on large application models. To wit, we take the shallow water equations fully coupled to Exner's sediment conservation law:

$$\begin{aligned} \partial_t H + \nabla_x \cdot \boldsymbol{q} - \nabla_x \cdot (\mathfrak{W} \nabla_x H) &= 0, \\ \partial_t \boldsymbol{q} + \nabla_x \cdot \left(\boldsymbol{q} \otimes \boldsymbol{u} + \tfrac{1}{2} g H^2 \right) &= g H \nabla_x b + \nabla_x \cdot (\mathfrak{N} \nabla_x \boldsymbol{q}) + S, \\ \partial_t b_i + \nabla_x \cdot \tilde{\boldsymbol{q}}_i - \nabla_x \cdot (\mathfrak{D} \nabla_x b_i) &= 0, \end{aligned} \tag{4.4}$$

where $\tilde{\boldsymbol{q}}$ is the flux formulation for the *sediment discharge* $\tilde{\boldsymbol{q}} = \tilde{\boldsymbol{q}}(H, \boldsymbol{q}, b)$, with the *momentum flux* $\boldsymbol{q} = H\boldsymbol{u}$. The total height of the *water column* H is a linear com-

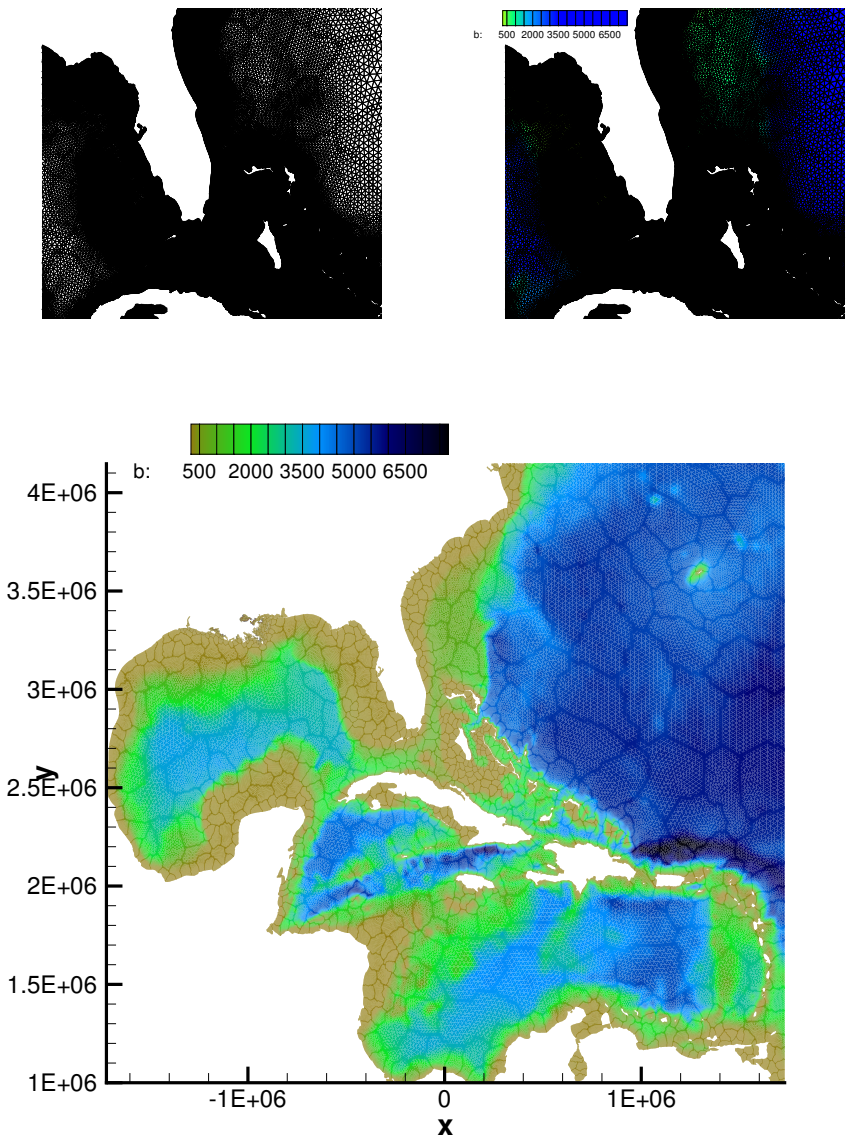

Figure 7: The bathymetry b and mesh for the north Atlantic unstructured grid. The top left shows a zoom in of the mesh around the Florida coast, where the top right gives the corresponding bathymetry. The bottom shows the entire domain. Light colored subdomains indicate the location of each of the 1024 processors used in parallel to run these simulations.

bination of the *bathymetric bed load* b with layered strata $b = \sum_{i=1}^{\ell} b_i$ and the *free*

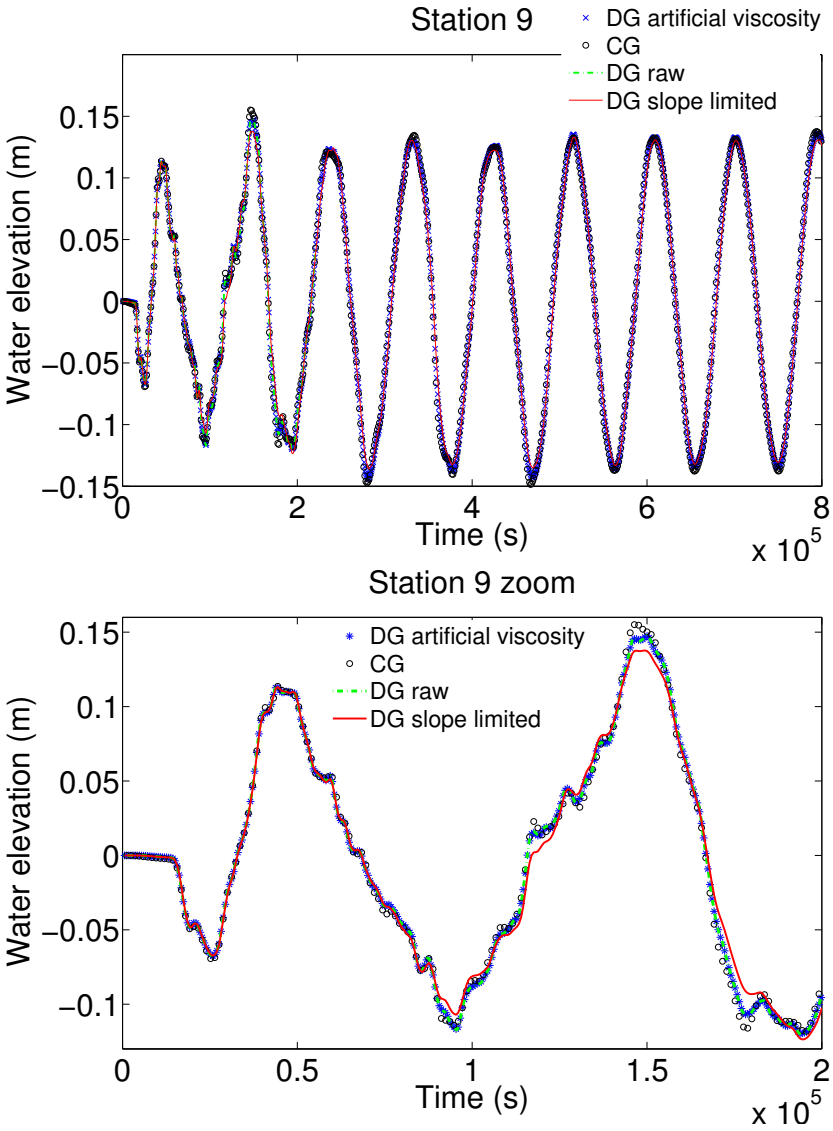

Figure 8: Here we show the tidal response at station 9, Boston, Boston Harbor, MA: $+42^\circ\ 21'\ 17.9994"$, $-71^\circ\ 3'\ 6.012"$.

surface height ζ, such that $H = \zeta + b$ where g is the gravitational constant.

The eddy viscosity is taken to satisfy $\mathfrak{N} = \mathfrak{N}_\star + \epsilon_{\mathfrak{N}}$, which is the sum of native $\mathfrak{N}_\star$ plus artificial $\epsilon_{\mathfrak{N}}$ viscosity. Here $\mathfrak{W}$ is the wave diffusion comprised of native $\mathfrak{W}_\star$ and artificial $\epsilon_{\mathfrak{W}}$ wave diffusion, and $\mathfrak{D}$ is the sedimentary eddy viscosity made up of $\mathfrak{D}_\star$ and $\epsilon_{\mathfrak{D}}$. The remaining source term $S = S(\boldsymbol{x}, t)$ connotes the remaining first order forcings in the system, such as winds in hurricane storm surge models [8, 17], and tidal constituents, etc. Also, note that when we write b and its corresponding

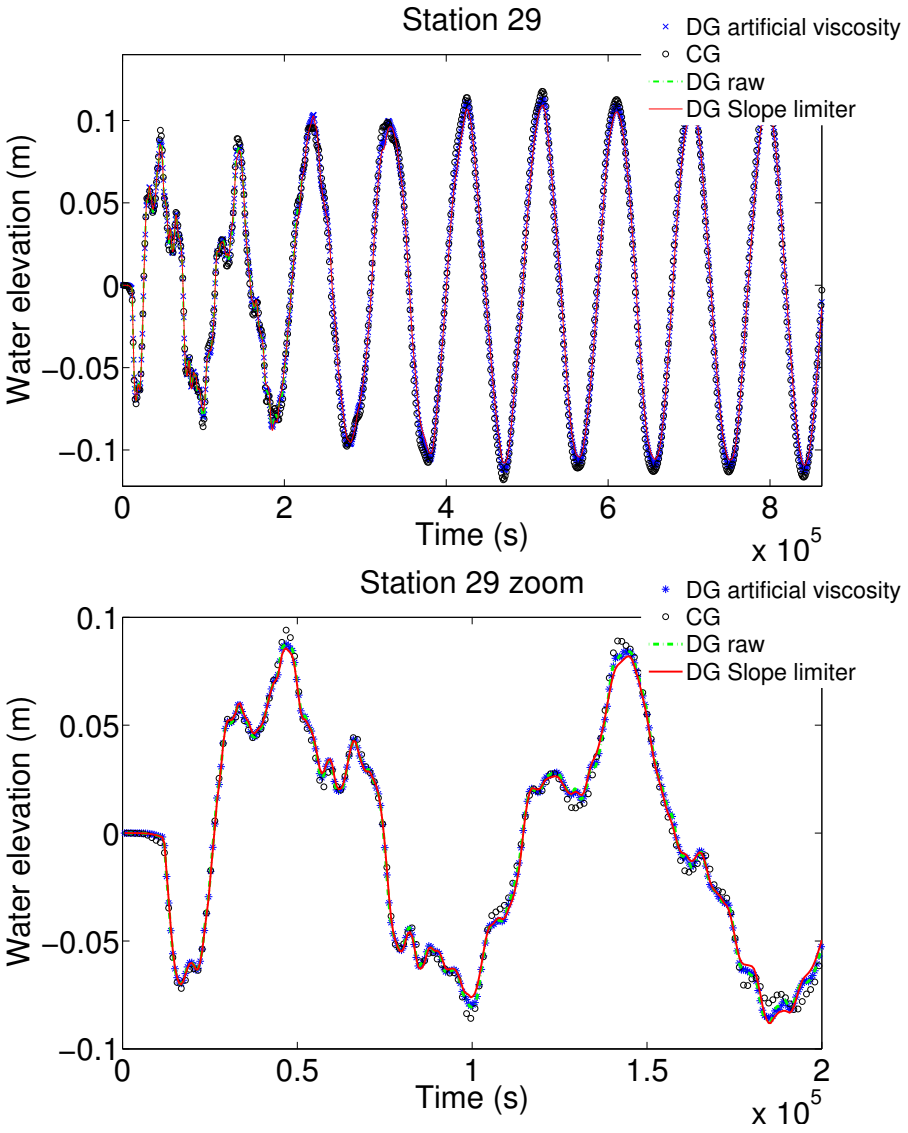

Figure 9: The tidal response at station 29, Ocean City, Fishing Pier, MD: 29, $+38°\ 19'\ 36.012"$, $-75°\ 4'\ 59.988"$.

flux in the Exner equation with no index, we mean to restrict to the case of the single stratum, $\ell = 1$.

Our test case will be an initial boundary problem on a large north Atlantic unstructured grid comprised of 492179 elements (see figure 7). We use this as a benchmark problem to test coastal tidal forcing estimates, where the primary forcing fields are the tidal constituents. In this example we restrict to the $O1$ lunar diurnal constituent. The native eddy viscosity is set to $\mathfrak{N}_\star = 5$, with a quarter day ramp up of the tides

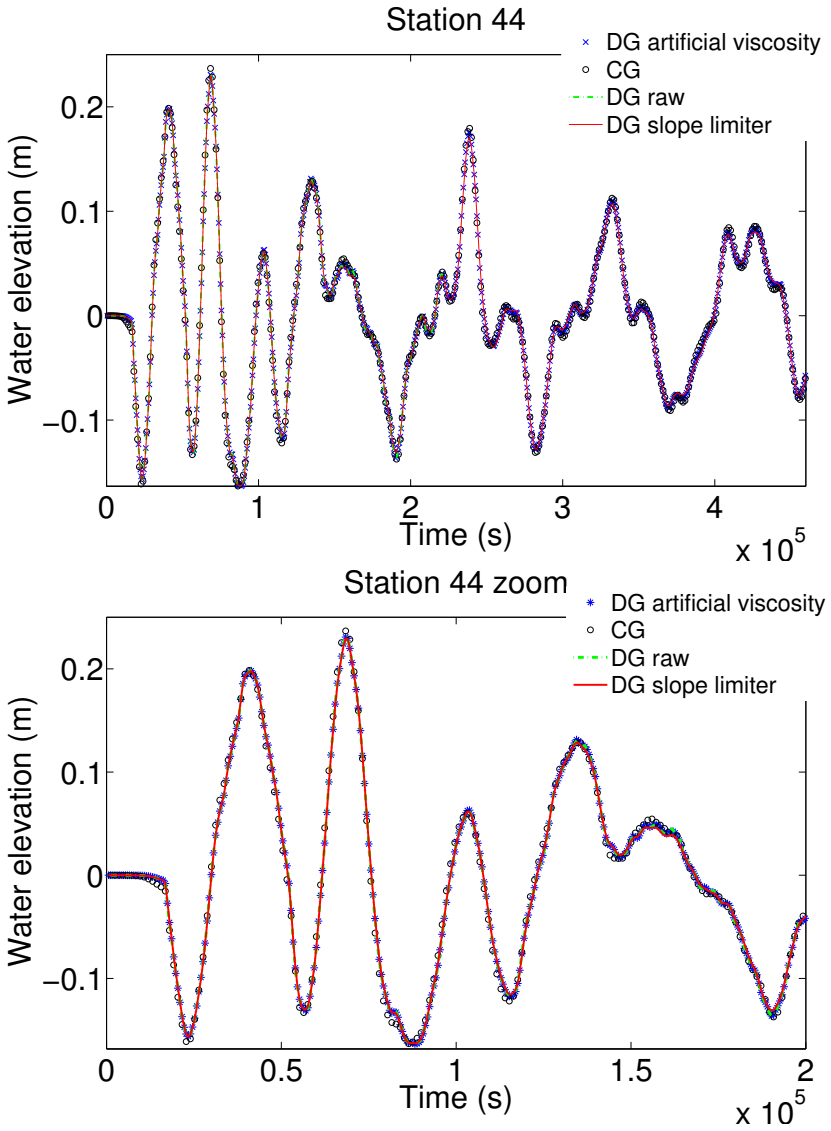

Figure 10: Here we show the tidal response at station 44, Charleston, Cooper River Entrance, SC: $+32°\ 46'\ 54.0114"$, $-79°\ 55'\ 29.9994"$.

to avoid shocking the system. In our example the bathymetry is taken to be time-independent and thus the continuity equation simply tracks the evolution of the free surface ζ.

The results are shown in figures 8–11. We take here the continuous Galerkin ADCIRC model as our benchmark solution [8], and run at $p = 1$ with a timestep of $dt = 0.1$. The results are somewhat subtle, though striking. First, the solution does not run to a stable completion without the addition of a limiter of some sort, either

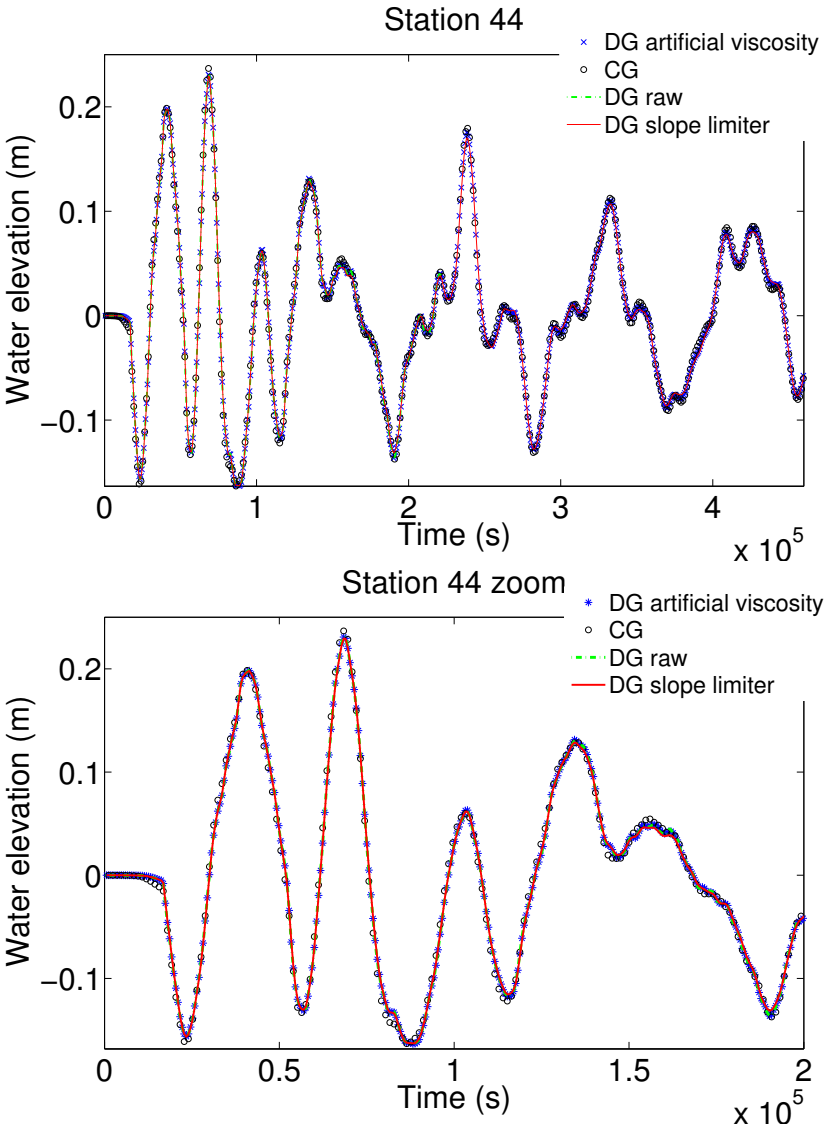

Figure 11: Here we show the tidal response at station 100, Corpus Christi, Gulf of Mexico, TX: $+27°\ 34'\ 47.9994"$, $-97°\ 13'\ 0.0114"$.

slopelimiting or artificial diffusion. It should also be noted here, that no wetting and drying algorithm is being used for this test example, which would add an additional limiter into the mix. Looking at figure 8, it is clear that many of the differences occur in the first two to three days of the simulation. In the zoomed in plot it is easy to observe that the BDS limiter is seemingly reducing the energy of the wave, lowering crest amplitudes and raising trough amplitudes. In contrast, the artificial viscosity limiter, run using $\kappa = -1$ and $\epsilon = 20\ \mathrm{m}^2\ \mathrm{s}^{-1}$, provides a solution that is virtually

indistinguishable from the native DG solution (DG raw, which, as seen in figure 8 is only stable until about 2×10^5 seconds), with the one exception being that it runs to completion. This same behavior is seen to occur again in figures 9–11. These subtleties notwithstanding, all of the DG solutions perform relatively well against the CG benchmark solution.

§5 Conclusion

In nonlinear PDEs complicated regularity can be expected in the mathematical solution, inducing important questions about the nature of the physical regularity as well as numerical regularity of the solution. To study, analyze, and understand the prominent features that arise, and can even dominate nonlinear system solution behavior, one must utilize numerical regularization techniques. In the area of modal discontinuous Galerkin methods, three salient techniques are: slopelimiting, modal filtering, and artificial diffusion.

We have discussed the implementation and derivation of a very effective slope limiter, known as the BDS limiter. We further reviewed modal spectral filters, and developed a new artificial diffusion technique based on a novel entropy sensor formulation. Numerical studies have helped to develop a deeper understanding of when and how these techniques can be used to regularize solutions to nonlinear advection-diffusion-reaction equations.

We have found that when one is solving systems at relatively low order, e.g. $p = 1$, when diffusion and/or viscosity can be neglected such that the auxiliary variables $\boldsymbol{\sigma}$ do not play into the solution behavior, and when reaction terms g have nonoscillatory subcellular behavior, then the BDS limiter is extremely appealing for producing stable, computationally efficient solutions to broad classes of nonlinear applications. It should be further noted that the BDS limiter is not difficult to extend to preserve derived quantities, such as positivity preservation, etc.

In cases where $p > 1$, or when solution behavior demonstrates strong dependencies on the gradients $\boldsymbol{\sigma}$ and have areas where diffusion/viscosity or nonlinear reactions dominate, then modal filtering or artificial diffusion can be much more effective as tools for solution regularization. Modal filtering is appealing in that it is fairly easy to implement, has no effect on the CFL condition of the system, is easy to tune, and is computationally efficient. However, modal filtering suffers from the fact that it dampens local extrema. One could of course extend the filter to having dependencies on a local entropy sensor, in order to make a local spectral viscosity method that more feectively addresses this issue.

Along these lines, artificial diffusion (or artificial viscosity) can be used and very finely tuned to dampen spurious solution artifacts by virtue of its coupled entropy sensor. It is highly tunable, and thus can display broad types of regularization across the solution. However, the artificial diffusion can stiffen the timestepping when diffusive modes are not already present in the system, and it can be very challenging to

tune for large nonlinear application models.

Areas for future study include the relationship of nonlinear operator eigenmodes to temporal stabilization of the solution using an extended form of linear stability theory. We are also looking to test these regularizing procedures on large scale applications models, such as storm surge and barrier island erosion models over many millions of degrees of freedom.

§6 Acknowledgements

The first author would like to thank Kyle Mandli and Stewart Stafford for helpful comments, insights, and conversation.

References

[1] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *J. Comput. Phys.*, 114(1):45–58, 1994. ISSN 0021-9991. doi: 10.1006/jcph.1994.1148. URL http://dx.doi.org/10.1006/jcph.1994.1148.

[2] D. N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2001/02. ISSN 0036-1429. doi: 10.1137/S0036142901384162. URL http://dx.doi.org.ezproxy.lib.utexas.edu/10.1137/S0036142901384162.

[3] D.N. Arnold, F. Brezzi, B. Cockburn, and D. Marini. Discontinuous Galerkin methods for elliptic problems. In *Discontinuous Galerkin methods (Newport, RI, 1999)*, volume 11 of *Lect. Notes Comput. Sci. Eng.*, pages 89–101. Springer, Berlin, 2000.

[4] G. E. Barter and D.L. Darmofal. Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation. *Journal of Computational Physics*, 229(5):1810–1827, Mar 1 2010. ISSN 0021-9991. doi: {10.1016/j.jcp.2009.11.010}.

[5] T. Barth and D.C. Jesperson. The design and application of upwind schems and unstructured meshes. *AIAA*, pages 89–0366, 1989.

[6] J.B. Bell, C.N. Dawson, and G.R. Shubin. An unsplit, higher order godunov method for scalar conservation laws in multiple dimensions. *Journal of Computational Physics*, 74(1):1 – 24, 1988. ISSN 0021-9991. doi: DOI:10.1016/0021-9991(88)90065-4. URL http://www.sciencedirect.com/science/article/B6WHY-4DD1T8P-N0/2/aba1bf519b0924a0a20968665aa37091.

[7] S. Bunya, E.J. Kubatko, J. J. Westerink, and C. Dawson. A wetting and drying treatment for the Runge-Kutta discontinuous Galerkin solution to the shallow water equations. *Comput. Methods Appl. Mech. Engrg.*, 198(17-20):1548–1562, 2009. ISSN 0045-7825. doi: 10.1016/j.cma.2009.01.008. URL `http://dx.doi.org/10.1016/j.cma.2009.01.008`.

[8] S. Bunya, J. C. Dietrich, J. J. Westerink, B. A. Ebersole, J. M. Smith, J. H. Atkinson, R. Jensen, D. T. Resio, R. A. Luettich, C. Dawson, V. J. Cardone, A. T. Cox, M. D. Powell, H. J. Westerink, and H. J. Roberts. A High-Resolution Coupled Riverine Flow, Tide, Wind, Wind Wave, and Storm Surge Model for Southern Louisiana and Mississippi. Part I: Model Development and Validation. *Monthly Weather Review*, 138(2):345–377, Feb 2010. ISSN 0027-0644. doi: {10.1175/2009MWR2906.1}.

[9] E. Casoni, J. Peraire, and A. Huerta. One-dimensional shock-capturing for high-order discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, 71(6):737–755, FEB 28 2013. ISSN 0271-2091. doi: {10.1002/fld.3682}.

[10] B. Cockburn. An introduction to the discontinuous Galerkin method for convection-dominated problems. In *Advanced numerical approximation of nonlinear hyperbolic equations (Cetraro, 1997)*, volume 1697 of *Lecture Notes in Math.*, pages 151–268. Springer, Berlin, 1998.

[11] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Math. Comp.*, 52(186):411–435, 1989. ISSN 0025-5718.

[12] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.*, 35(6):2440–2463 (electronic), 1998. ISSN 0036-1429.

[13] B. Cockburn and C.-W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16(3):173–261, 2001. ISSN 0885-7474.

[14] B. Cockburn and C.W. Shu. The runge-kutta discontinuous galerkin method for conservation laws v: Multidimensional systems. *Journal of Computational Physics*, 141(2):199 – 224, 1998. ISSN 0021-9991. doi: DOI:10.1006/jcph.1998.5892. URL `http://www.sciencedirect.com/science/article/B6WHY-45J593T-6N/2/34d62c09260f2d7d0af3099d456a3b72`.

[15] B. Cockburn, S.-Y. Lin, and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. III. One-dimensional systems. *J. Comput. Phys.*, 84(1):90–113, 1989. ISSN 0021-9991.

[16] C. Dawson, E.J. Kubatko, J.J. Westerink, C. Trahan, C. Mirabito, C. Michoski, and N. Panda. Discontinuous Galerkin methods for modeling hurricane storm surge. *Advances in Water Resources*, 34(9):1165 – 1176, 2011. ISSN 0309-1708. doi: 10.1016/j.advwatres.2010.11.004. URL `http://www.sciencedirect.com/science/article/pii/S0309170810002137`. New Computational Methods and Software Tools.

[17] C. Dawson, J.J. Westerink, J.C. Feyen, and D. Pothina. Continuous, discontinuous and coupled discontinuous-continuous Galerkin finite element methods for the shallow water equations. *Internatioanl Journal for Numerical Methods in Fluids*, 52(1):63–88, Sep 10 2006. ISSN 0271-2091. doi: {10.1002/fjd.1156}.

[18] M. DuChene, A.M. Spagnuolo, E.J. Kubatko, J.J. Westerink, and C. Dawson. A Framework for Running the ADCIRC Discontinuous Galerkin Storm Surge Model on a GPU. In Sato, M and Matsuoka, S and Sloot, PMA and VanAlbada, GD and Dongarra, J, editor, *Proceedings of the International Conference on Computational Science (ICCS)*, volume 4 of *Procedia Computer Science*, pages 2017–2026, Sara Burgerhartstraat 25, po box 211, 1000 ae Amsterdam, Netherlands, 2011. Elsevier; Univ Tsukuba, Ctr Computat Sci, Elsevier Science BV. doi: {10.1016/j.procs.2011.04.220}. International Conference on Computational Science (ICCS) on the Ascent of Computational Excellence, Campus Nanyang Technolog Univ, Singapore, Singapore, 2011.

[19] L.J. Durlofsky, B. Engquist, and S. Osher. Triangle based adaptive stencils for the solution of hyperbolic conservation laws. *Journal of Computational Physics*, 98(1):64 – 73, 1992. ISSN 0021-9991. doi: DOI:10.1016/0021-9991(92)90173-V. URL `http://www.sciencedirect.com/science/article/B6WHY-4DD1P88-NW/2/14f5775efbf9049e31e12411e2e34238`.

[20] M. Feistauer, J. Felcman, and I. Straškraba. *Mathematical and computational methods for compressible flow*. Numerical mathematics and scientific computation. Oxford University Press, 2003. ISBN 0-19-850588-4.

[21] J.S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods*, volume 54 of *Texts in Applied Mathematics*. Springer, New York, 2008. ISBN 978-0-387-72065-4. doi: 10.1007/978-0-387-72067-8. URL `http://dx.doi.org/10.1007/978-0-387-72067-8`. Algorithms, analysis, and applications.

[22] F. Hindenlang, G.J. Gassner, C. Altmann, A. Beck, M. Staudenmaier, and C.-D. Munz. Explicit discontinuous Galerkin methods for unsteady problems. *Computers & Fluids*, 61(SI):86–93, May 30 2012. ISSN 0045-7930. doi: {10.1016/j.compfluid.2012.03.006}.

[23] D. Ketcheson, M. Parsani, and R. LeVeque. High-order wave propagation algorithms for hyperbolic systems. *SIAM Journal on Scientific Computing*, 35

(1):A351–A377, 2013. doi: 10.1137/110830320. URL http://epubs.siam.org/doi/abs/10.1137/110830320.

[24] A. Kloeckner, T. Warburton, J. Bridge, and J. S. Hesthaven. Nodal discontinuous Galerkin methods on graphics processors. *Journal of Computational Physics*, 228 (21):7863–7882, Nov 2009. ISSN 0021-9991. doi: {10.1016/j.jcp.2009.06.041}.

[25] A. Kloeckner, 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. ISSN 0973-5348. doi: {10.1051/mmnp/20116303}.

[26] E.J. Kubatko, J.J. Westerink, and C. Dawson. hp discontinuous galerkin methods for advection dominated problems in shallow water flow. *Computer Methods in Applied Mechanics and Engineering*, 196(1-3):437 – 451, 2006. ISSN 0045-7825. doi: DOI:10.1016/j.cma.2006.05.002. URL http://www.sciencedirect.com/science/article/B6V29-4M1CYTM-1/2/6c45c85d20d17690046881a795b0b04d.

[27] E.J. Kubatko, J.J. Westerink, and C. Dawson. Semi discrete discontinuous Galerkin methods and stage-exceeding-order, strong-stability-preserving Runge-Kutta time discretizations. *J. Comput. Phys.*, 222(2):832–848, 2007. ISSN 0021-9991. doi: 10.1016/j.jcp.2006.08.005. URL http://dx.doi.org/10.1016/j.jcp.2006.08.005.

[28] E.J. Kubatko, C. Dawson, and J.J. Westerink. Time step restrictions for Runge-Kutta discontinuous Galerkin methods on triangular grids. *J. Comput. Phys.*, 227(23):9697–9710, 2008. ISSN 0021-9991. doi: 10.1016/j.jcp.2008.07.026. URL http://dx.doi.org/10.1016/j.jcp.2008.07.026.

[29] E.J. Kubatko, S. Bunya, C. Dawson, J.J. Westerink, and C. Mirabito. A performance comparison of continuous and discontinuous finite element shallow water models. *J. Sci. Comput.*, 40(1-3):315–339, 2009. ISSN 0885-7474. doi: 10.1007/s10915-009-9268-2. URL http://dx.doi.org/10.1007/s10915-009-9268-2.

[30] Ethan J. Kubatko, Joannes J. Westerink, and Clint Dawson. Semi discrete discontinuous galerkin methods and stage-exceeding-order, strong-stability-preserving runge-kutta time discretizations. *J. Comput. Phys.*, 222(2):832–848, March 2007. ISSN 0021-9991. doi: 10.1016/j.jcp.2006.08.005. URL http://dx.doi.org/10.1016/j.jcp.2006.08.005.

[31] Ethan J. Kubatko, Joannes J. Westerink, and Clint Dawson. hp discontinuous Galerkin methods for advection dominated problems in shallow water flow. *Computer Methods in Applied Mechanics and Engineering*, 196(1-3):437–451, 2006. ISSN 0045-7825. doi: {10.1016/j.cma.2006.05.002}.

[32] D. Kuzmin. A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *J. Comput. Appl. Math.*, 233(12):3077–3085, 2010. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/j.cam.2009.05.028.

[33] D. Kuzmin. Slope limiting for discontinuous Galerkin approximations with a possibly non-orthogonal Taylor basis. *International Journal for Numerical Methods in Fluids*, 71(9):1178–1190, Mar 30 2013. ISSN 0271-2091. doi: {10.1002/fld.3707}.

[34] D. Kuzmin and F. Schieweck. A parameter-free smoothness indicator for high-resolution finite element schemes. *Central European Journal of Mathematics*, 11 (8):1478–1488, Aug 2013. ISSN 1895-1074. doi: {10.2478/s11533-013-0254-4}.

[35] Y. Liu, C.-W. Shu, E. Tadmor, and M. Zhang. Central discontinuous Galerkin methods on overlapping cells with a nonoscillatory hierarchical reconstruction. *SIAM J. Numer. Anal.*, 45(6):2442–2467 (electronic), 2007. ISSN 0036-1429. doi: 10.1137/060666974. URL `http://dx.doi.org/10.1137/060666974`.

[36] Y Maday, S. Kaber, and E. Tadmor. Legendre pseudospectral viscosity method for nonlinear conservation-laws. *SIAM Journal on Numerical Analysis*, 30(2): 321–342, Apr 1993. ISSN 0036-1429. doi: {10.1137/0730016}.

[37] A. Meister, S. Ortleb, and Th. Sonar. Application of spectral filtering to discontinuous Galerkin methods on triangulations. *Numerical Methods for Partial Differential Equations*, 28(6):1840–1868, NOV 2012. ISSN 0749-159X. doi: {10.1002/num.20705}.

[38] C. Michoski, J. A. Evans, P. G. Schmitz, and A. Vasseur. A discontinuous Galerkin method for viscous compressible multifluids. *J. Comput. Phys.*, 229 (6):2249–2266, 2010. ISSN 0021-9991. doi: 10.1016/j.jcp.2009.11.033. URL `http://dx.doi.org/10.1016/j.jcp.2009.11.033`.

[39] C. Michoski, C. Mirabito, C. Dawson, D. Wirasaet, E. J. Kubatko, and J. J. Westerink. Adaptive hierarchic transformations for dynamically p-enriched slope-limiting over discontinuous Galerkin systems of generalized equations. *J. Comput. Phys.*, 230(22):8028–8056, 2011. ISSN 0021-9991. doi: 10.1016/j.jcp.2011.07.009. URL `http://dx.doi.org/10.1016/j.jcp.2011.07.009`.

[40] C. Michoski, C. Mirabito, C. Dawson, D. Wirasaet, E.J. Kubatko, and J.J. Westerink. Dynamic p-enrichment schemes for multicomponent reactive flows. *Advances in Water Resources*, 34(12):1666 – 1680, 2011. ISSN 0309-1708. doi: 10.1016/j.advwatres.2011.09.001. URL `http://www.sciencedirect.com/science/article/pii/S0309170811001679`.

[41] C. Michoski, J.A. Evans, and P.G. Schmitz. Discontinuous Galerkin hp-adaptive methods for multiscale chemical reactors i: Quiescent reactors. *preprint*, 2013.

[42] S. Osher. Convergence of generalized MUSCL schemes. *SIAM J. Numer. Anal.*, 22(5):947–961, 1985. ISSN 0036-1429.

[43] P.-O. Persson and J. Peraire. Sub-cell shock capturing for discontinuous Galerkin methods, 2013. preprint.

[44] S.J. Ruuth. Global optimization of explicit strong-stability-preserving Runge-Kutta methods. *Math. Comp.*, 75(253):183–207 (electronic), 2006. ISSN 0025-5718. doi: 10.1090/S0025-5718-05-01772-2. URL `http://dx.doi.org/10.1090/S0025-5718-05-01772-2`.

[45] C.-W. Shu and S. Osher. Efficient implementation of essentially nonoscillatory shock-capturing schemes. *J. Comput. Phys.*, 77(2):439–471, 1988. ISSN 0021-9991.

[46] Eitan Tadmor and Knut Waagan. ADAPTIVE SPECTRAL VISCOSITY FOR HYPERBOLIC CONSERVATION LAWS. *SIAM JOURNAL ON SCIENTIFIC COMPUTING*, 34(2):A993–A1009, 2012. ISSN 1064-8275. doi: {10.1137/110836456}.

[47] J.W. Thomas. *Numerical partial differential equations: finite difference methods*, volume 22 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1995. ISBN 0-387-97999-9.

[48] D. Wirasaet, S. Tanaka, E. J. Kubatko, J. J. Westerink, and C. Dawson. A performance comparison of nodal discontinuous Galerkin methods on triangles and quadrilaterals. *International Journal for Numerical Methods in Fluids*, 64 (10-12):1336–1362, Dec 30 2010. ISSN 0271-2091. doi: {10.1002/fld.2376}. 15th International Conference on Finite Elements in Flow Problems, Tokyo, Japan, Apr 01-03, 2009.

[49] JG Xin and JE Flaherty. Viscous stabilization of discontinuous Galerkin solutions of hyperbolic conservation laws. *Applied Numerical Mathematics*, 56(3-4): 444–458, Mar-Apr 2006. ISSN 0168-9274. doi: {10.1016/j.apnum.2005.08.001}. 3rd International Conference on Numerical Solutions of Volterra and Delay Equations, Tempe, AZ, MAY, 2004.

[50] Y. Xing and C.W. Shu. High-order finite volume weno schemes for the shallow water equations with dry states. *Advances in Water Resources*, 34(8):1026 – 1038, 2011. ISSN 0309-1708. doi: 10.1016/j.advwatres.2011.05.008. URL `http://www.sciencedirect.com/science/article/pii/S0309170811000996`.

[51] Y. Xing, X. Zhang, and C.-W. Shu. Positivity-preserving high order well-balanced discontinuous Galerkin methods for the shallow water equations. *Advances in Water Resources*, 33(12):1476 – 1493, 2010. ISSN 0309-1708.

doi: 10.1016/j.advwatres.2010.08.005. URL http://www.sciencedirect.com/science/article/pii/S0309170810001491.

[52] V. Zingan, J.-L. Guermond, J. Morel, and B. Popov. Implementation of the entropy viscosity method with the discontinuous Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 253:479–490, Jan 1 2013. ISSN 0045-7825. doi: {10.1016/j.cma.2012.08.018}.

[53] V.N. Zingan. Discontinuous Galerkin finite element method for the nonlinear hyperbolic problems with entropy-based artificial viscosity stabilization. *Dissertation, Texas A & M University*, 2012.