Filtering for Discontinuous Galerkin Methods

Jennifer K. Ryan
jryan@kth.se

A bit about me (www.jenniferkayryan.com)

- From the US
- Have held (permanent) academic positions in US, UK, Netherlands, and now Sweden
- Held visiting positions at: Los Alamos National Lab, Uppsala, Heinrich-Heine, Cambridge
- Research:
  - *General:* Develop high-order numerical methods for nonlinear hyperbolic conservation laws.
  - *More Specifically:* Multi-wavelet multi-resolution analysis, Accuracy extraction through SIAC filtering.
  - *Funding:* European Commission, US National Science Foundation, US Air Force Office of Scientific Research

A bit about you

How many of you...

- are PhD students?
- work in hyperbolic equations?
- work with filters?

Plan for today

- Morning:
  - Review of conservation laws and some properties of DG
  - Motivation for filtering
  - Types of filters
- Afternoon:
  - Superconvergence and SIAC filtering.

Useful resources

To add to/reiterate Matteo's suggestions:

- Books and Lecture Notes:
  - Jan S. Hesthaven and Tim Warburton, "Nodal discontinuous Galerkin Methods: Algorithms, Analysis, and Applications", Springer, 2008.
  - Bernardo Cockburn, Chi-Wang Shu, Claes Johnson, Eitan Tadmore, and Alfio Quarteroni, "Advanced Numerical Approximation of Nonlinear Hyperbolic Equations", Springer, 1997.
  - Chi-Wang Shu, "Discontinuous Galerkin Methods: General Approach and Stability".
- Codes:
  - Tim Warburton (Nodal DG Matlab code)
  - Jan Hesthaven (various)
  - Jesse Chan (Trixi – a Julia package)

Questions?

Part I:

A review of DG for conservation laws

https://github.com/numwisk/DG-summer-school.git

DG for conservation laws

Assume that the model equation is of the form

$$\mathbf{u}_t + \nabla \cdot f(\mathbf{u}) = \mathbf{0}$$

with

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u_0}(\mathbf{x})$$

and boundary conditions to make the problem well posed.

Let's form the DG scheme ...

DG for conservation laws

Examples of $f(\mathbf{u})$:

- Linear advection: $f(\mathbf{u}) = A\mathbf{u}$, where A is a real, constant matrix.
- Variable coefficient: $f(\mathbf{u}) = A(\mathbf{x})\mathbf{u}$
- Burgers equation: $f(u) = \frac{u^2}{2}$

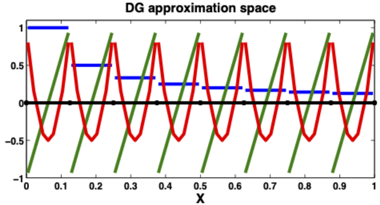
Note: If $f(\mathbf{u})$ is non-linear we need to ensure that we choose a physically-relevant week solution.

Step 1: The Approximation Space

Approximation space consists of piecewise polynomials of degree $\leq p$.

$$V_h^k = \{v \in L^2(\Omega) : v \in \mathbb{P}^p(\tau_e), \forall \tau_e \in \mathcal{T}_h\}$$

where $\mathcal{T}_h$ is some tessellation of our domain.



For simplicity, we take the Legendre polynomials

Step 2: The Variational Formulation

The DG scheme is given by:

Find $u_h(\mathbf{x}, t) \in V_h^p$ such that

$$\int_{\tau_e} (u_h)_t v_h(x)\, dx - \int_{\tau_e} f(u_h) \nabla (v_h)\, dx + \int_{\partial \tau_e} \hat{n} \cdot \widehat{f} v_h\, ds = 0$$

for all $v_h \in V_h^k$. Here, $\tau_e \in \mathcal{T}$ and $\hat{f}$ is the numerical flux.

Step 3: The Flux

The fluxes are generally chosen to be monotone. More specifically,

- locally Lipschitz and consistent with the flux $f(u)$, i.e., $\hat{f}(u,u) = f(u)$,
- a nondecreasing function of its first argument, and
- a nonincreasing function of its second argument.

We implement the local Lax-Friedrichs flux

$$\hat{f}(a,b) = \frac{1}{2}(f(a) + f(b) - \lambda(b-a))$$

with $\lambda = \max_{\min(a,b) \leq s \leq \max(a,b)} |f'(s)|$.

This helps to enforce weak continuity at element interfaces.

The Linear Advection Equation

Consider the model problem

$$\begin{aligned}\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} &= 0, \qquad \Omega = [-1, 1]\\ u(x, 0) &= f(x)\\ u(-1, t) &= u(1, t)\end{aligned}$$

DG Scheme: Variational Formulation

The DG scheme is: Find $u_h \in V_h^p$ such that

$$\int_{x_k}^{x_{k+1}} (u_h)_t v_h(x)\, dx - \int_{x_k}^{x_{k+1}} (au_h)(v_h)_x\, dx + \widehat{au_h}_{x_{k+1}} v_{x_{k+1}}^- - \widehat{au_h}_{x_k} v_{x_k}^+ = 0, \quad k = 1, \dots, N.$$

for all $v_h \in V_h^p$. Assume

$$u_h(x,t) = \sum_{\ell=1}^{p+1} u_k^{(\ell)}(t)\varphi^{(\ell)}(r), \quad v_h(x) = \varphi^{(m)}(r),\ m = 1, \dots, p+1,$$

where $r = \frac{2}{\Delta x_k}(x - x_k) - 1 \in [-1, 1]$ is a local coordinate mapping.

DG Scheme: Implementation

Denote $\mathbf{u}_k(t) = [u_k^{(1)}(t),\, u_k^{(2)}(t), \ldots,\, u_k^{(p+1)}(t)]^T$. Then

$$\frac{\Delta x_k}{2}\mathcal{M}\frac{d}{dt}\mathbf{u}_k(t) = a\mathcal{S}\mathbf{u}_k(t) - \left[\widehat{au}_{h x_{k+1}} - \widehat{au}_{h x_k}(-1)^m\right]$$

where

$$\mathcal{M}(m,\ell) = \int_{-1}^{1} \varphi^{(\ell)}(r)\varphi^{(m)}(r)\,dr, \quad \mathcal{S}(m,\ell) = \int_{-1}^{1} \varphi^{(\ell)}(r)\frac{d}{dr}\varphi^{(m)}(r)\,dr$$

DG Scheme: Initial Condition

- Initial modes are obtained using an L^2-projection:

$$M\mathbf{u}_k = \int_{-1}^{1} u_0\left(x_k + \frac{\Delta x_k}{2}(r+1)\right)\varphi\, dr$$

- Other types of projection are possible. For example, Gauss-Radau:

$$\begin{aligned} M_{p\times p}\mathbf{u}_k &= \int_{-1}^{1} u_0\left(x_k + \frac{\Delta x_k}{2}(r+1)\right)\varphi\, dr \\ u_h(x_{k+1}^{\pm}, 0) &= u_0(x_{k+1}^{\pm}) \end{aligned}$$

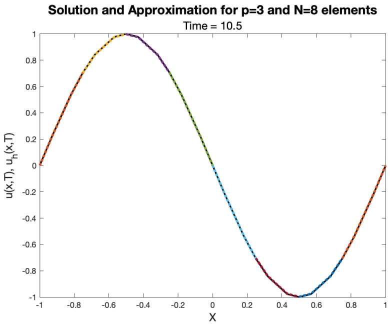
- Is it possible **not** to do a projection?

DG Scheme: Initial Conditions

For our purposes, we will concentrate on the initial conditions:

- $u(x, 0) = \sin(4\pi x)$
- $u(x, 0) =$
$$\begin{cases} \frac{1}{6}(G(x, \beta, z-\delta) + G(x, \beta, z+\delta) + 4G(x, \beta, z)), & [-0.8, -0.6], \\ 1, & x \in [-0.4, -0.2], \\ 1 - |10(x - 0.1)|, & x \in [0, 0.2], \\ \frac{1}{6}(F(x, \alpha, a-\delta) + F(x, \alpha, a+\delta) + 4F(x, \alpha, z)), & x \in [0.4, 0.4], \\ 0, & \text{for all other } x \end{cases},$$
where $G(x, \beta, z) = e^{-\beta(x-z)^2}$ and $F(x, \alpha, a) = \sqrt{\max\left(1 - \alpha^2(x-\alpha)^2, 0\right)}$ and $a = 0.5,\ z = -0.7,\ \delta = 0.005,\ \alpha = 10$, and $\beta = \log(2)/36\delta^2$.

Linear Advection



Code can be downloaded from
https://github.com/numwisk/DG-summer-school.git

from Hesthaven & Warburton

...WHEW!

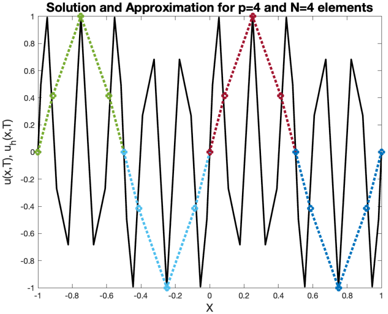
Part II: Filtering

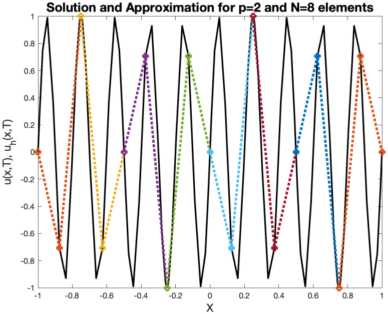
Reasons to filter

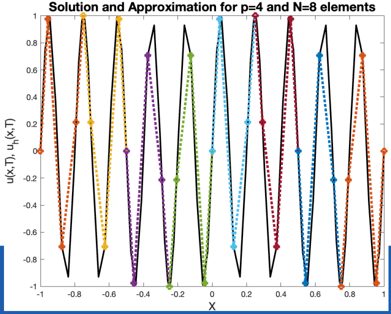
Why do we filter?

- Aliasing
- Gibbs oscillations
- Accuracy extraction (afternoon)

Aliasing


Solution and Approximation for p=4 and N=4 elements
u(x,T), u_h(x,T)
x


Solution and Approximation for p=2 and N=8 elements
u(x,T), u_h(x,T)
x

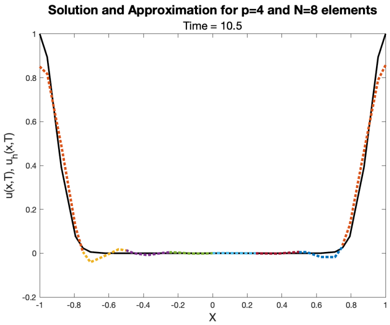

Solution and Approximation for p=4 and N=8 elements
u(x,T), u_h(x,T)
x

Aliasing: Exercise 1

https://github.com/numwisk/DG-summer-school.git

- Construct a smooth initial condition such that aliasing occurs.
- Why does it occur?
- How do you know the instability is caused by aliasing?
- Is there a way to ensure that aliasing doesn't occur?

Aliasing



Solution and Approximation for p=4 and N=8 elements
Time = 10.5
$u(x,T)$, $u_h(x,T)$
X
1
0.8
0.6
0.4
0.2
0
-0.2
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

Aliasing

- Caused by underresolution (not enough points per wavelength).
- Frequently encountered when we utilize and interpolation perspective

Error Estimate: Projection

Theorem

Assume that $v \in \mathcal{H}^m(I)$ *and that* v_h *represents a polynomial of projection order* $p+1$. *Then*

$$\|v - v_h\|_{I,q} \leq (p+1)^{\rho - m} |v|_{I,m},$$

where

$$\rho = \begin{cases} \frac{3}{2}q, & 0 \leq q \leq 1 \\ 2q - \frac{1}{2}, & q \geq 1 \end{cases}$$

and $0 \leq q \leq m$.

Error Estimate: Projection

Theorem
If $v \in \mathcal{H}^m(I)$, $m \geq 1$ *then*

$$\|v^{(q)} - v^{(q)}\|_{I,0} \leq \left[\frac{(p+2-\sigma)!}{(p+2+\sigma-4q)!}\right]^{1/2} |v|_{I,\sigma},$$

where $\sigma = \min(p+2, m)$ *and* $q \leq m$.

For $p >> m$, $\quad \|v^{(q)} - v^{(q)}\|_{I,0} \leq (p+1)^{2q-m}|v|_{I,m}$.

Error Estimate: Interpolation

However, for interpolation we have

Theorem

Assume that $u \in \mathcal{H}^m(\tau_k)$, $m > \frac{1}{2}$, and that u_h represents a piecewise polynomial interpolation of order $p+1$. Then

$$\|u - u_h\|_{\Omega,q,h} \leq C \frac{h^{\sigma-q}}{(p+1)^{m-2q-1/2}} |u|_{\Omega,\sigma,h}$$

for $0 \leq q \leq \sigma$, $\sigma = \min(p+2, m)$.

Aliasing: (Optional) Exercise

Let us now consider the case where $f(u) = a(x)u(x,t)$ and $a(x) = (1 - x^2)^5 + 1$. The DG scheme is:

$$\mathcal{M}^k \frac{d}{dt}\mathbf{u}_h^k + \mathcal{S}\mathbf{f}_h^k = \frac{1}{2}\oint_{x_k^+}^{x_{k+1}^-} \hat{n}\cdot[[f_h^k]]\ell^k(x),\, dx$$

- Projection of flux

$$f_h^k(x) = \mathcal{P}_h(a(x)u_h^k(x)) = \sum_{i=1}^{p+1} f_h^k(x_i^k)\ell_i^k(x)$$

Aliasing: (Optional) Exercise

$$\mathcal{M}^k \frac{d}{dt}\mathbf{u}_h^k + \mathcal{S}\mathbf{f}_h^k = \frac{1}{2}\oint_{x_k^+}^{x_{k+1}^-} \hat{n} \cdot [[f_h^k]]\ell^k(x),\, dx$$

- Interpolation of flux

$$f_h^k(x) = \mathcal{I}_h(a(x)u_h^k(x)) = \sum_{i=1}^{p+1} a(x_i^k)u_h^k(x_i, t)\ell_i^k(x)$$

1. Alter the code at https://github.com/numwisk/DG-summer-school.git to utilize the interpolation perspective for the flux representation.
2. What is the difference in the two schemes?
3. Is it possible to use interpolation for the flux?

Aliasing: The Remedy

Note that for the stability estimate we have

$$\frac{1}{2}\frac{d}{dt}\|u_h\|_\Omega \leq C_1\|u_h\|_\Omega + C_2(h, a(x))(p+1)^{1-m}|u|_{\Omega,m}$$

The main culprit:

$$\|\mathcal{I}_h\frac{dv}{dx} - \frac{dv}{dx}(\mathcal{I}_h v)\|$$

How do we fix this?

Aliasing: The Remedy

Add a filter function:

$$u_h^*(x,t) = \sum_{m=1}^{p+1} \sigma\left(\frac{m-1}{p}\right) u_k^{(m)} \varphi^{(m)}(x)$$

where

$$\sigma(\eta) \begin{cases} = 1, & \eta = 0 \\ \leq q, & 0 \leq \eta < 1 \\ = 0, & \eta \geq 1 \end{cases}.$$

- $\sigma(\eta) = 1$ ensures consistency.
- $\sigma(\eta) \leq 1$ dissipates high modes.
- $\sigma(\eta) = 0$???

Filters

Definition (Filter)

A filter, σ, of order $2m$ is a smooth, even function whose support is $[-1, 1]$ and such that

$$\sigma(0) = 1$$

and

$$\begin{aligned} \sigma^{(\alpha)}(0) =&0, \quad 1 \leq \alpha \leq 2m-1 \\ \sigma^{(\alpha)}(1) =&0, \quad 1 \leq \alpha \leq 2m-1 \end{aligned}$$

H. Vandeven, Family of Spectral filters for discontinuous Problems, Journal of Scientific Computing, 6 (1991), pp. 159-192.

Filters

Some filtering options:

- $\sigma(\eta) = 1 - \alpha\eta^{2m}$
- $\sigma(\eta) = exp(-\alpha\eta^{2m})$

Note: We filter at every time step(?)

Filters

Filters are equivalent to adding dissipation to our model equation:

$$\begin{aligned}\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} =& \epsilon(-1)^{2m+1}\left[\frac{\partial}{\partial x}(1-x^2)\frac{\partial}{\partial x}\right]^{2m} u, \qquad \Omega = [-1, 1] \\ u(x, 0) =& f(x) \\ u(-1, t) =& u(1, t)\end{aligned}$$

Gibbs

The same ideas can be utilized for Gibbs oscillations:

Filters: Exercise 2

1. Use the wavetest.m initial condition in the linear advection code and run the code for final time of 10.5.
2. Is it possible to choose the points-per-wavelength so that there are no Gibbs oscillations? Why or why not?
3. Use the Filter1D.m code to filter the approximation. How often must you filter in order to control the oscillations?
4. How would you implement a multi-dimensional filter?

Part III: Filtering for
Accuracy Extraction
SIAC Filtering