Stable high order methods for time-domain wave propagation in complex geometries and heterogeneous media

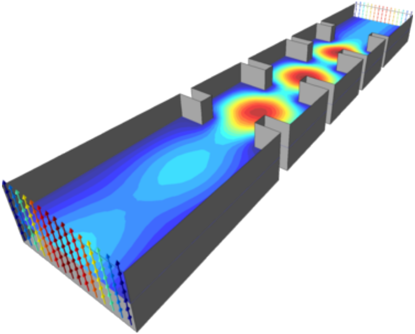
Kaihang Guo

Department of Computational and Applied Mathematics
Rice University

Numerical simulation of wave propagation

Many procedures requires **accurately** and **efficiently** solving time-dependent wave equations in realistic settings.

- Imaging (seismic, medical)
- Engineering design (scattering, design)
- Computational physics (aeroacoustics, astrophysics)



https://www.comsol.com/model/image/12737/big.png

Discontinuous Galerkin (DG) methods for waves

- Unstructured (tetrahedral) meshes for geometric flexibility.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Explicit time stepping: high performance on many-core.

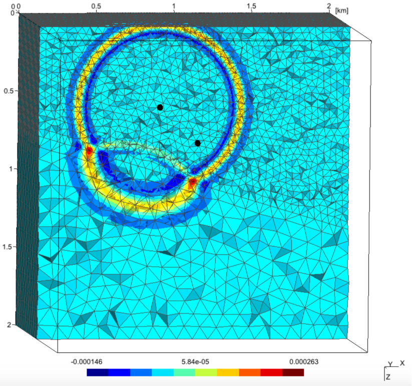
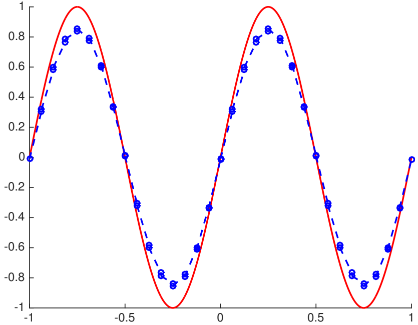


Figure courtesy of Axel Modave.

Discontinuous Galerkin (DG) methods for waves

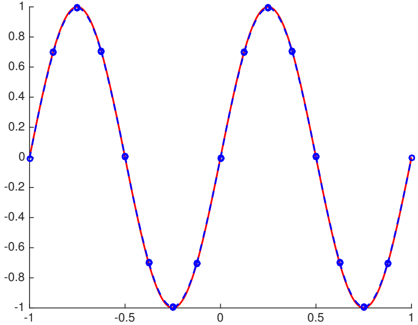
- Unstructured (tetrahedral) meshes for geometric flexibility.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Explicit time stepping: high performance on many-core.



Fine linear approximation.

Discontinuous Galerkin (DG) methods for waves

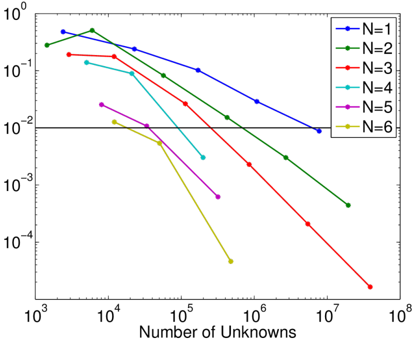
- Unstructured (tetrahedral) meshes for geometric flexibility.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Explicit time stepping: high performance on many-core.



Coarse quadratic approximation.

Discontinuous Galerkin (DG) methods for waves

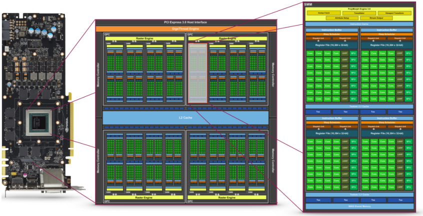
- Unstructured (tetrahedral) meshes for geometric flexibility.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Explicit time stepping: high performance on many-core.



Max errors vs. dofs.

Discontinuous Galerkin (DG) methods for waves

- Unstructured (tetrahedral) meshes for geometric flexibility.
- High order: low numerical dissipation and dispersion.
- High order approximations: more accurate per unknown.
- Explicit time stepping: high performance on many-core.

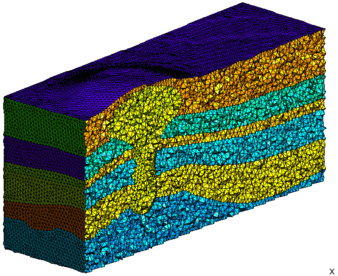


Graphics processing units (GPU).

Time-domain nodal DG methods

Assume $u(\boldsymbol{x}, t) = \sum \boldsymbol{u}_j \phi_j(\boldsymbol{x})$ on D^k

- Compute numerical flux at face nodes (non-local).
- Compute RHS of (local) ODE.
- Evolve (local) solution using explicit time integration (RK, AB, etc).



Mesh courtesy of J.F. Remacle

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x}$$

Example: advection equation.

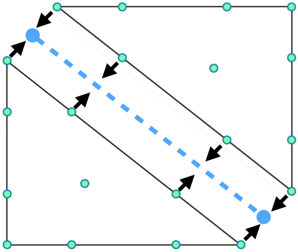
$$\boldsymbol{M}_{ij} = \int_{D^k} \phi_j(\boldsymbol{x})\phi_i(\boldsymbol{x})$$

$$\mathbf{L}_f = \mathbf{M}^{-1}\mathbf{M}_f.$$

Time-domain nodal DG methods

Assume $u(\boldsymbol{x}, t) = \sum \boldsymbol{u}_j \phi_j(\boldsymbol{x})$ on D^k

- Compute numerical flux at face nodes (non-local).
- Compute RHS of (local) ODE.
- Evolve (local) solution using explicit time integration (RK, AB, etc).



$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{D}_x \mathbf{u} + \sum_{\text{faces}} \mathbf{L}_f \left(\text{flux}\right).$$

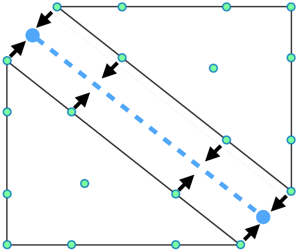
$$\boldsymbol{M}_{ij} = \int_{D^k} \phi_j(\boldsymbol{x}) \phi_i(\boldsymbol{x})$$

$$\mathbf{L}_f = \mathbf{M}^{-1} \mathbf{M}_f.$$

Time-domain nodal DG methods

Assume $u(\boldsymbol{x}, t) = \sum \boldsymbol{u}_j \phi_j(\boldsymbol{x})$ on D^k

- Compute numerical flux at face nodes (non-local).
- Compute RHS of (local) ODE.
- Evolve (local) solution using explicit time integration (RK, AB, etc).



$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \underbrace{\mathbf{D}_x \mathbf{u}}_{\text{Volume kernel}} + \underbrace{\sum_{\text{faces}} \mathbf{L}_f \left(\text{flux}\right)}_{\text{Surface kernel}}.$$

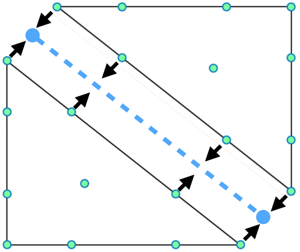
$$\boldsymbol{M}_{ij} = \int_{D^k} \phi_j(\boldsymbol{x}) \phi_i(\boldsymbol{x})$$

$$\mathbf{L}_f = \mathbf{M}^{-1} \mathbf{M}_f.$$

Time-domain nodal DG methods

Assume $u(\boldsymbol{x}, t) = \sum \boldsymbol{u}_j \phi_j(\boldsymbol{x})$ on D^k

- Compute numerical flux at face nodes (non-local).
- Compute RHS of (local) ODE.
- Evolve (local) solution using explicit time integration (RK, AB, etc).



$$\underbrace{\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}}_{\text{Update kernel}} = \underbrace{\mathbf{D}_x \mathbf{u}}_{\text{Volume kernel}} + \underbrace{\sum_{\text{faces}} \mathbf{L}_f \left(\text{flux}\right)}_{\text{Surface kernel}}.$$

$$\boldsymbol{M}_{ij} = \int_{D^k} \phi_j(\boldsymbol{x}) \phi_i(\boldsymbol{x})$$

$$\mathbf{L}_f = \mathbf{M}^{-1} \mathbf{M}_f.$$

Outline

1 Weight-adjusted DG (WADG): high order heterogeneous media

2 Elastic-acoustic coupled media

3 Bernstein-Bezier WADG: high order efficiency

Outline

Energy stable discontinuous Galerkin formulations

- Model problem: acoustic wave equation (pressure-velocity system)

$$\frac{1}{c^2}\frac{\partial p}{\partial t} = \nabla \cdot \boldsymbol{u}, \qquad \frac{\partial \boldsymbol{u}}{\partial t} = \nabla p$$

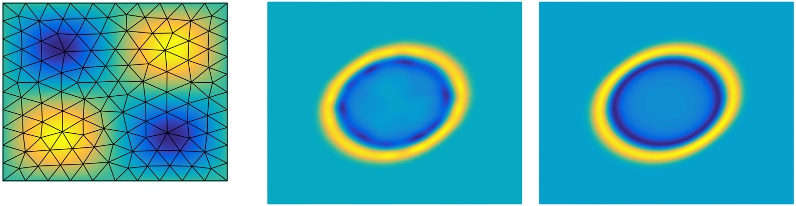
- Local formulation

$$\int_{D^k} \frac{1}{c^2}\frac{\partial p}{\partial t} q = \int_{D^k} \nabla \cdot \boldsymbol{u} q + \frac{1}{2}\int_{\partial D^k} \left([\![\boldsymbol{u}]\!] \cdot \boldsymbol{n} + \tau_p [\![p]\!] \right) q$$
$$\int_{D^k} \frac{\partial \boldsymbol{u}}{\partial t} \boldsymbol{v} = \int_{D^k} \nabla p \cdot \boldsymbol{v} + \frac{1}{2}\int_{\partial D^k} \left([\![p]\!] + \tau_u [\![\boldsymbol{u}]\!] \cdot \boldsymbol{n} \right) \boldsymbol{v}$$

- High order accuracy, semi-discrete energy stability

$$\frac{\partial}{\partial t}\left(\sum_k \int_{D^k} \frac{p^2}{c^2} + |\boldsymbol{u}|^2 \right) = -\sum_k \int_{\partial D^k} \tau_p [\![p]\!]^2 + \tau_u [\![\boldsymbol{u} \cdot \boldsymbol{n}]\!]^2 \leq 0.$$

High order approximation of smoothly varying media



(a) Mesh and exact c^2 (b) Piecewise const. c^2 (c) High order c^2

- Piecewise const. c^2: energy stable and efficient, but inaccurate.

$$\frac{1}{c^2(\boldsymbol{x})}\frac{\partial p}{\partial t} + \nabla\cdot\boldsymbol{u} = 0, \qquad \frac{\partial \boldsymbol{u}}{\partial t} + \nabla p = 0.$$

- High order wavespeeds: weighted mass matrices. Stable, but expensive (pre-computation + storage of matrix inverses)!

$$\boldsymbol{M}_{1/c^2}\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = \boldsymbol{A}_h\boldsymbol{U}, \qquad \left(\boldsymbol{M}_{1/c^2}\right)_{ij} = \int_{D^k}\frac{1}{c^2(\boldsymbol{x})}\phi_j(\boldsymbol{x})\phi_i(\boldsymbol{x}).$$

Weight-adjusted DG (WADG)

- Weight-adjusted DG: provably energy stable approx. of $\boldsymbol{M}_{1/c^2}$

$$\boldsymbol{M}_{1/c^2}\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} \approx \boldsymbol{M}\left(\boldsymbol{M}_{c^2}\right)^{-1}\boldsymbol{M}\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = \boldsymbol{A}_h\boldsymbol{U}.$$

- New evaluation reuses implementation for constant wavespeed

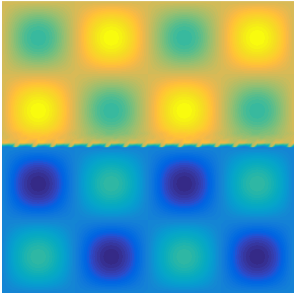
$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = \underbrace{\boldsymbol{M}^{-1}\left(\boldsymbol{M}_{c^2}\right)}_{\text{modified update}} \qquad \underbrace{\boldsymbol{M}^{-1}\boldsymbol{A}_h\boldsymbol{U}}_{\text{constant wavespeed RHS}}$$

- Low storage matrix-free application of $\boldsymbol{M}^{-1}\boldsymbol{M}_{c^2}$ using **quadrature**-based interpolation and L^2 projection matrices $\boldsymbol{V}_q, \boldsymbol{P}_q$.

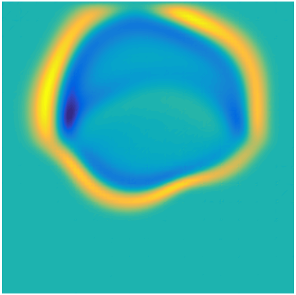
$$\left(\boldsymbol{M}\right)^{-1}\boldsymbol{M}_{c^2} = \underbrace{\boldsymbol{M}^{-1}\boldsymbol{V}_q^T\boldsymbol{W}}_{\boldsymbol{P}_q}\operatorname{diag}\left(c^2\right)\boldsymbol{V}_q.$$

Chan, Hewett, Warburton (2017). Weight-adjusted DG methods: wave propagation in heterogeneous media.

WADG: nearly identical to DG w/weighted mass matrices



(a) $c^2(x,y)$

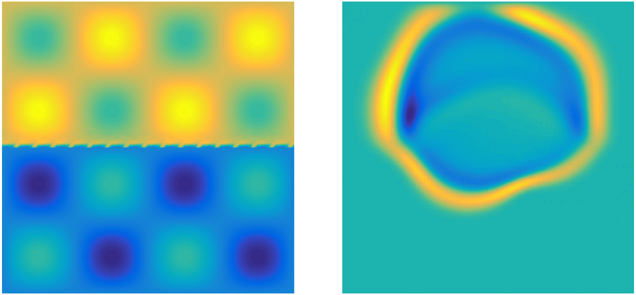


(b) Standard DG

Figure: Standard vs. weight-adjusted DG with spatially varying c^2.

- The L^2 error is $O(h^{N+1})$, but the difference between the DG and WADG solutions is $O(h^{N+2})$!

WADG: nearly identical to DG w/weighted mass matrices

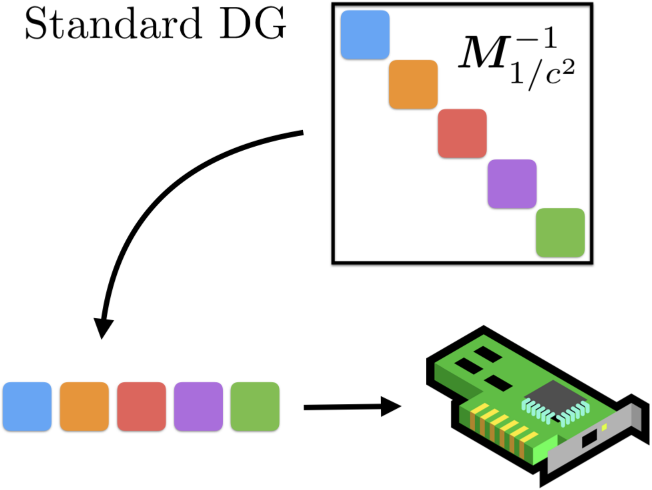


(a) $c^2(x, y)$ (b) Weighted-adjusted DG

Figure: Standard vs. weight-adjusted DG with spatially varying c^2.

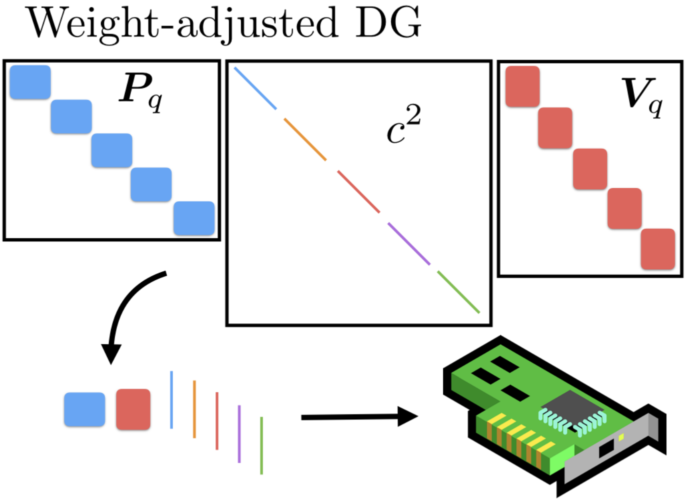
- The L^2 error is $O(h^{N+1})$, but the difference between the DG and WADG solutions is $O(h^{N+2})$!

WADG: more efficient than storing M_{1/c^2}^{-1} on GPUs



Efficiency on GPUs: reduce memory accesses and data movement!

WADG: more efficient than storing M_{1/c^2}^{-1} on GPUs



Efficiency on GPUs: reduce memory accesses and data movement!

Outline

Matrix-valued weights and elastic wave propagation

- Symmetric velocity-stress formulation (entries of $\boldsymbol{A}_i$ are ± 1 or 0)

$$\rho \frac{\partial \boldsymbol{v}}{\partial t} = \sum_{i=1}^{d} \boldsymbol{A}_i^T \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{x}_i}, \qquad \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{\sigma}}{\partial t} = \sum_{i=1}^{d} \boldsymbol{A}_i \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}_i}.$$

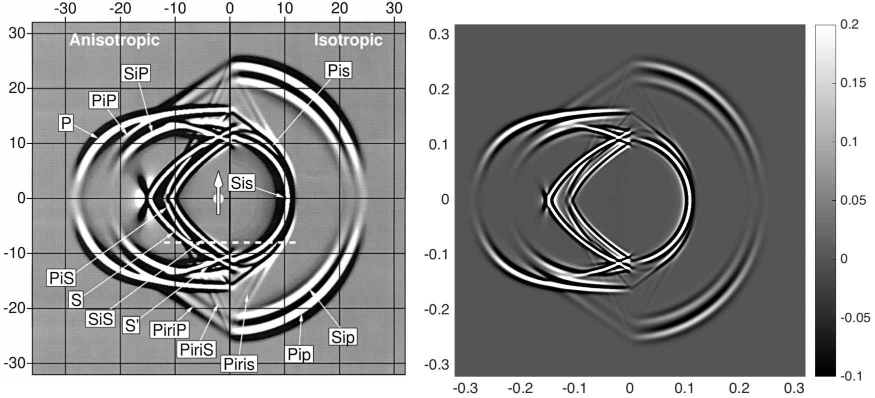
- DG formulation: *simple* penalty fluxes, matrix-weighted mass matrix

$$\boldsymbol{A}_1^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \qquad \boldsymbol{M}_{\boldsymbol{C}^{-1}} = \begin{pmatrix} \boldsymbol{M}_{C_{11}^{-1}} & \dots & \boldsymbol{M}_{C_{1d}^{-1}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{M}_{C_{d1}^{-1}} & \dots & \boldsymbol{M}_{C_{dd}^{-1}} \end{pmatrix}$$

- Weight-adjusted approx. to $(\boldsymbol{M}_{\boldsymbol{C}^{-1}})^{-1}$ decouples each component

$$\boldsymbol{M}_{\boldsymbol{C}^{-1}}^{-1} \approx \left(\boldsymbol{I} \otimes \boldsymbol{M}^{-1}\right) \boldsymbol{M}_{\boldsymbol{C}} \left(\boldsymbol{I} \otimes \boldsymbol{M}^{-1}\right).$$

Simple to incorporate anisotropic media

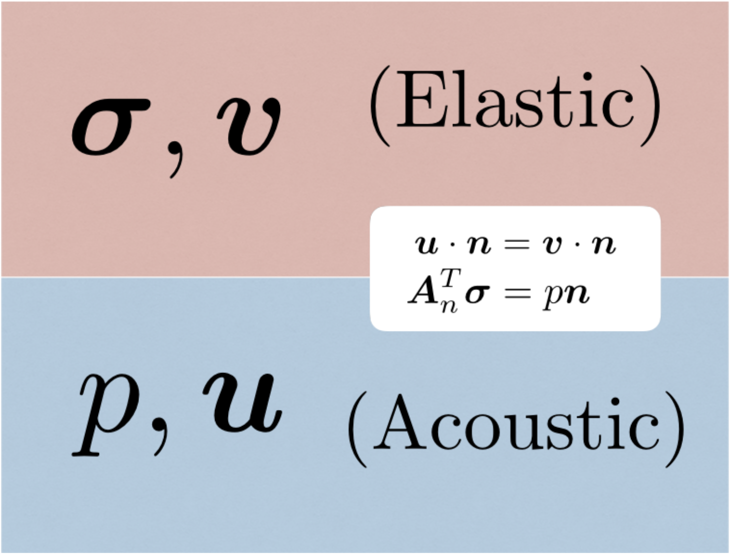


(a) Reference solution (b) WADG solution

Figure: Anisotropic media simply involves modifying the definition of $\boldsymbol{C}$.

Komatitsch, Barnes, Tromp (2000). *Simulation of anisotropic wave propagation based upon a spectral element method.*
Chan (2018). Weight-adjusted DG methods: matrix-valued weights and elastic wave prop. in heterogeneous media.

Energy stable acoustic-elastic coupling



Energy stable acoustic-elastic coupling

$$(\mathfrak{F}\boldsymbol{q})^* = \mathfrak{F}^-\boldsymbol{q}^- + \frac{\boldsymbol{n}\cdot[\![\mathbf{S}]\!] + \rho^+ c_p^+[\![\boldsymbol{v}]\!]}{\rho^+ c_p^+ + \rho^- c_p^-}\begin{pmatrix}\boldsymbol{n}\otimes\boldsymbol{n}\\ \rho^- c_p^- \boldsymbol{n}\end{pmatrix}.$$

$$\begin{aligned}(\mathfrak{F}\boldsymbol{q})^* = \mathfrak{F}^-\boldsymbol{q}^- &+ \frac{c_p^- c_p^+ \boldsymbol{n}\cdot[\![\mathbf{S}]\!] + c_p^-(\lambda^+ + 2\mu^+)[\![\boldsymbol{v}]\!]}{c_p^+(\lambda^- + 2\mu^-) + c_p^-(\lambda^+ + 2\mu^+)}\begin{pmatrix}\boldsymbol{n}\otimes\boldsymbol{n}\\ \rho^- c_p^- \boldsymbol{n}\end{pmatrix} + \left(\frac{c_s^- c_s^+}{\mu^+ c_s^- + \mu^- c_s^+}\boldsymbol{s}\cdot[\![\mathbf{S}]\!] + \frac{c_s^-\mu^+}{\mu^+ c_s^- + \mu^- c_s^+}\boldsymbol{s}\cdot[\![\boldsymbol{v}]\!]\right)\begin{pmatrix}\mathrm{sym}(\boldsymbol{s}\otimes\boldsymbol{n})\\ \rho^- c_s^- \boldsymbol{s}\end{pmatrix}\\
&+ \left(\frac{c_s^- c_s^+}{\mu^+ c_s^- + \mu^- c_s^+}\boldsymbol{t}\cdot[\![\mathbf{S}]\!] + \frac{c_s^-\mu^+}{\mu^+ c_s^- + \mu^- c_s^+}\boldsymbol{t}\cdot[\![\boldsymbol{v}]\!]\right)\begin{pmatrix}\mathrm{sym}(\boldsymbol{t}\otimes\boldsymbol{n})\\ \rho^- c_s^- \boldsymbol{t}\end{pmatrix} = \mathfrak{F}^-\boldsymbol{q}^- + \frac{c_p^- c_p^+ \boldsymbol{n}\cdot[\![\mathbf{S}]\!] + c_p^-(\lambda^+ + 2\mu^+)[\![\boldsymbol{v}]\!]}{c_p^+(\lambda^- + 2\mu^-) + c_p^-(\lambda^+ + 2\mu^+)}\begin{pmatrix}\boldsymbol{n}\otimes\boldsymbol{n}\\ \rho^- c_p^- \boldsymbol{n}\end{pmatrix}\\
&- \frac{c_s^- c_s^+}{\mu^+ c_s^- + \mu^- c_s^+}\begin{pmatrix}\mathrm{sym}(\boldsymbol{n}\otimes(\boldsymbol{n}\times(\boldsymbol{n}\times[\![\mathbf{S}]\!])))\\ \rho^- c_s^- \boldsymbol{n}\times(\boldsymbol{n}\times[\![\mathbf{S}]\!])\end{pmatrix} - \frac{c_s^-\mu^+}{\mu^+ c_s^- + \mu^- c_s^+}\begin{pmatrix}\mathrm{sym}(\boldsymbol{n}\otimes(\boldsymbol{n}\times(\boldsymbol{n}\times[\![\boldsymbol{v}]\!])))\\ \rho^- c_s^- \boldsymbol{n}\times(\boldsymbol{n}\times[\![\boldsymbol{v}]\!])\end{pmatrix},\end{aligned}$$

$$(\mathfrak{F}\boldsymbol{q})^* = \mathfrak{F}^-\boldsymbol{q}^- + \frac{\boldsymbol{n}\cdot[\![\mathbf{S}]\!] + \rho^+ c_p^+[\![\boldsymbol{v}]\!]}{\rho^+ c_p^+ + \rho^- c_p^-}\begin{pmatrix}\boldsymbol{n}\otimes\boldsymbol{n}\\ \rho^- c_p^- \boldsymbol{n}\end{pmatrix} - \frac{1}{\rho^- c_s^-}\begin{pmatrix}\mathrm{sym}(\boldsymbol{n}\otimes(\boldsymbol{n}\times(\boldsymbol{n}\times[\![\mathbf{S}]\!])))\\ \rho^- c_s^- \boldsymbol{n}\times(\boldsymbol{n}\times[\![\mathbf{S}]\!])\end{pmatrix}.$$

- Traditional upwind acoustic-elastic fluxes are complex to derive.
- Cannot prove energy stability in the case of heterogeneous media.

Wilcox, Stadler, Burstedde, Ghattas (2010). *A high-order discontinuous Galerkin method for wave propagation through coupled elastic–acoustic media.*

Energy stable acoustic-elastic coupling

$$\boldsymbol{A}_n = \boldsymbol{A}_1 n_x + \boldsymbol{A}_2 n_y + \boldsymbol{A}_3 n_z \qquad \text{(Elastic)}$$

$$\frac{1}{2}\left(\boldsymbol{A}_n^T(\boldsymbol{\sigma}^+ - \boldsymbol{\sigma}) + \tau_{\boldsymbol{v}} \boldsymbol{A}_n^T \boldsymbol{A}_n (\boldsymbol{v}^+ - \boldsymbol{v})\right)$$

$$\frac{1}{2}\left(\boldsymbol{A}_n(\boldsymbol{v}^+ - \boldsymbol{v}) + \tau_{\boldsymbol{\sigma}} \boldsymbol{A}_n \boldsymbol{A}_n^T(\boldsymbol{\sigma}^+ - \boldsymbol{\sigma}) \cdot \boldsymbol{n}\right) \boldsymbol{n}$$

$$\frac{1}{2}\left((\boldsymbol{u}^+ - \boldsymbol{u}) \cdot \boldsymbol{n} + \tau_p (p^+ - p)\right)$$

$$\frac{1}{2}\left((p^+ - p) + \tau_{\boldsymbol{u}} (\boldsymbol{u}^+ - \boldsymbol{u}) \cdot \boldsymbol{n}\right) \boldsymbol{n}$$

$$\text{(Acoustic)}$$

Energy stable acoustic-elastic coupling

(Elastic)

$$\frac{1}{2}\boldsymbol{A}_n\left(\boldsymbol{n}\boldsymbol{n}^T(\boldsymbol{u}-\boldsymbol{v})+\tau_{\boldsymbol{\sigma}}\left(p\boldsymbol{n}-\boldsymbol{A}_n^T\boldsymbol{\sigma}\right)\right)$$

$$\frac{1}{2}\boldsymbol{n}^T\left(p\boldsymbol{n}-\boldsymbol{A}_n^T\boldsymbol{\sigma}+\left(\boldsymbol{I}-\boldsymbol{n}\boldsymbol{n}^T\right)\boldsymbol{A}_n^T\boldsymbol{\sigma}+\tau_{\boldsymbol{v}}(\boldsymbol{u}-\boldsymbol{v})\right)$$

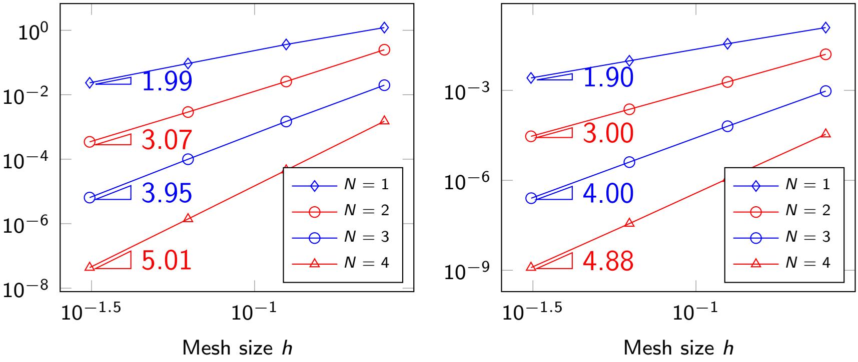
$$\boldsymbol{u}\cdot\boldsymbol{n}=\boldsymbol{v}\cdot\boldsymbol{n}$$
$$\boldsymbol{A}_n^T\boldsymbol{\sigma}=p\boldsymbol{n}$$

$$\frac{1}{2}\boldsymbol{n}^T\left(\boldsymbol{v}-\boldsymbol{u}+\tau_p\left(\boldsymbol{A}_n^T\boldsymbol{\sigma}-p\boldsymbol{n}\right)\right)$$

$$\frac{1}{2}\boldsymbol{n}\boldsymbol{n}^T\left(\boldsymbol{A}_n^T\boldsymbol{\sigma}-p\boldsymbol{n}+\tau_{\boldsymbol{u}}(\boldsymbol{v}-\boldsymbol{u})\right)$$

(Acoustic)

Numerical results: coupled acoustic-elastic media

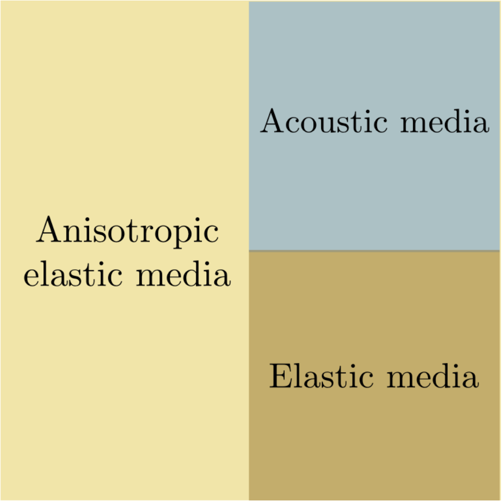


(a) Snell's law solution

(b) Scholte wave solution

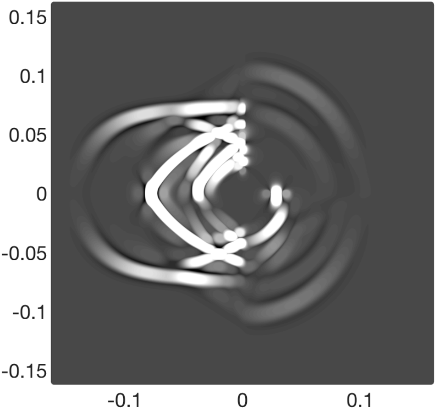
High order convergence of L^2 error for acoustic-elastic media.

Example with isotropic-anisotropic acoustic-elastic media

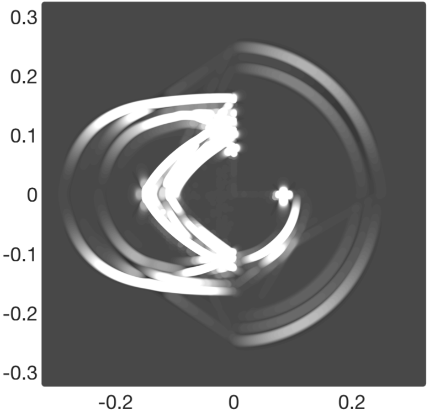


Komatitsch, Barnes, Tromp (2000). *Simulation of anisotropic wave propagation based upon a spectral element method.*
Guo, Acosta, Chan (2019). A weight-adjusted DG method for wave propagation in coupled elastic-acoustic media.

Example with isotropic-anisotropic acoustic-elastic media



(a) $T = 30\mu s$



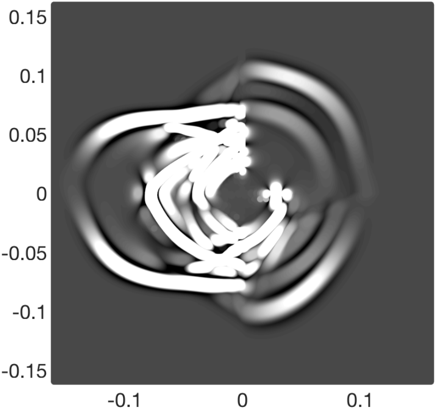
(b) $T = 60\mu s$

Piecewise constant anisotropic-isotropic acoustic-elastic media.

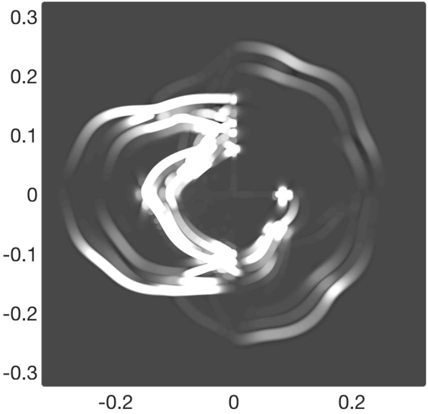
Komatitsch, Barnes, Tromp (2000). *Simulation of anisotropic wave propagation based upon a spectral element method.*

Guo, Acosta, Chan (2019). A weight-adjusted DG method for wave propagation in coupled elastic-acoustic media.

Example with isotropic-anisotropic acoustic-elastic media



(a) $T = 30\mu s$



(b) $T = 60\mu s$

Piecewise smoothly varying anisotropic-isotropic acoustic-elastic media.

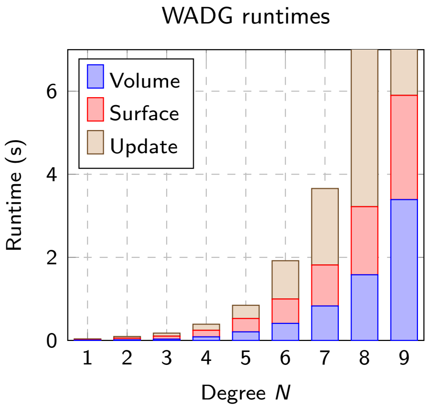
Komatitsch, Barnes, Tromp (2000). *Simulation of anisotropic wave propagation based upon a spectral element method.*

Guo, Acosta, Chan (2019). A weight-adjusted DG method for wave propagation in coupled elastic-acoustic media.

Outline

Computational costs at high orders of approximation

Problem: WADG at high orders becomes **expensive**!

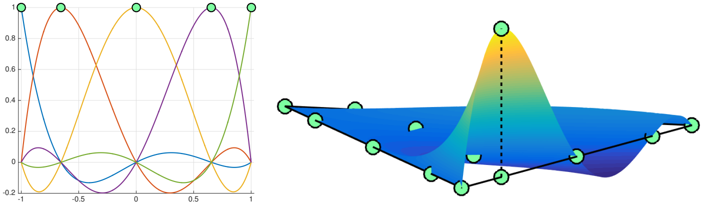


- Large **dense** matrices: $O(N^6)$ work per element.
- Idea: choose basis such that matrices are **sparse**.

WADG runtimes for 50 timesteps, 98304 elements.

BBDG: Bernstein-Bezier DG methods

- Nodal DG: $O(N^6)$ cost in 3D vs $O(N^3)$ degrees of freedom.
- Switch to Bernstein basis: sparse and structured matrices.
- Optimal $O(N^3)$ application of differentiation and lifting matrices.

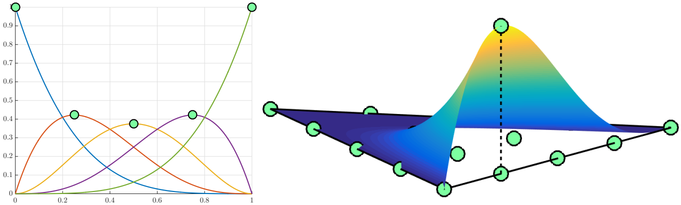


Nodal bases in one, two, and three dimensions.

Chan, Warburton 2017. GPU-accelerated Bernstein-Bezier discontinuous Galerkin methods for wave propagation (SISC).

BBDG: Bernstein-Bezier DG methods

- Nodal DG: $O(N^6)$ cost in 3D vs $O(N^3)$ degrees of freedom.
- Switch to Bernstein basis: sparse and structured matrices.
- Optimal $O(N^3)$ application of differentiation and lifting matrices.

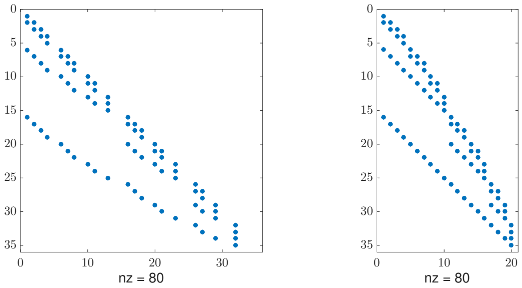


Bernstein bases in one, two, and three dimensions.

Chan, Warburton 2017. GPU-accelerated Bernstein-Bezier discontinuous Galerkin methods for wave propagation (SISC).

BBDG: Bernstein-Bezier DG methods

- Nodal DG: $O(N^6)$ cost in 3D vs $O(N^3)$ degrees of freedom.
- Switch to Bernstein basis: sparse and structured matrices.
- Optimal $O(N^3)$ application of differentiation and lifting matrices.

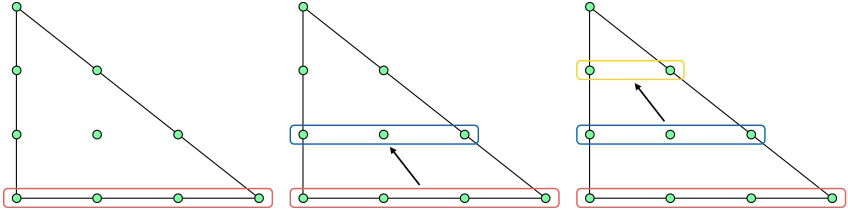


Tetrahedral Bernstein differentiation and degree elevation matrices.

Chan, Warburton 2017. GPU-accelerated Bernstein-Bezier discontinuous Galerkin methods for wave propagation (SISC).

BBDG: Bernstein-Bezier DG methods

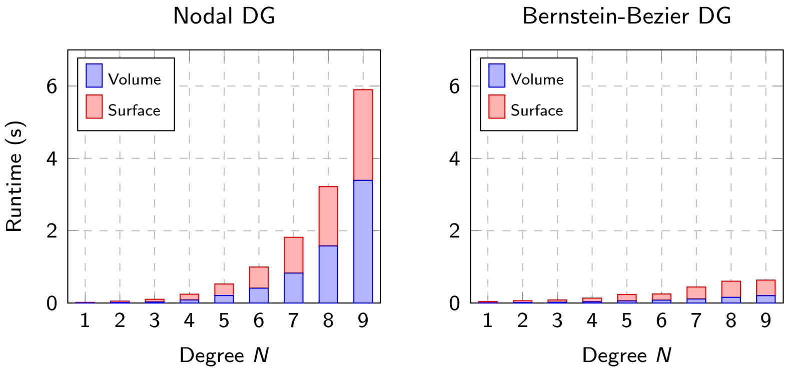
- Nodal DG: $O(N^6)$ cost in 3D vs $O(N^3)$ degrees of freedom.
- Switch to Bernstein basis: sparse and structured matrices.
- Optimal $O(N^3)$ application of differentiation and lifting matrices.



Optimal $O(N^3)$ complexity "slice-by-slice" application of Bernstein lift.

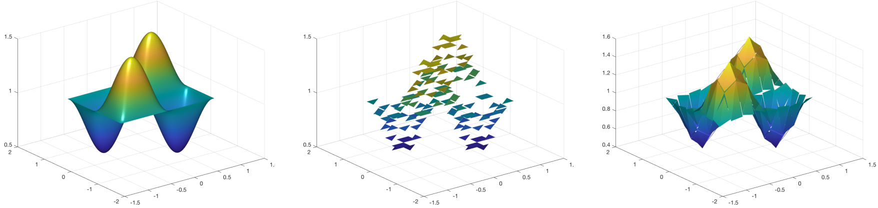
Chan, Warburton 2017. GPU-accelerated Bernstein-Bezier discontinuous Galerkin methods for wave propagation (SISC).

BBDG: efficient volume, surface kernels



$$\underbrace{\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}}_{\text{Update kernel}} = \underbrace{\mathbf{D}_x\mathbf{u}}_{\text{Volume kernel}} + \underbrace{\sum_{\text{faces}} \mathbf{L}_f\,(\text{flux})}_{\text{Surface kernel}}, \qquad \mathbf{L}_f = \mathbf{M}^{-1}\mathbf{M}_f.$$

A faster BBWADG update kernel



(a) Exact c^2 (b) $M = 0$ approximation (c) $M = 1$ approximation

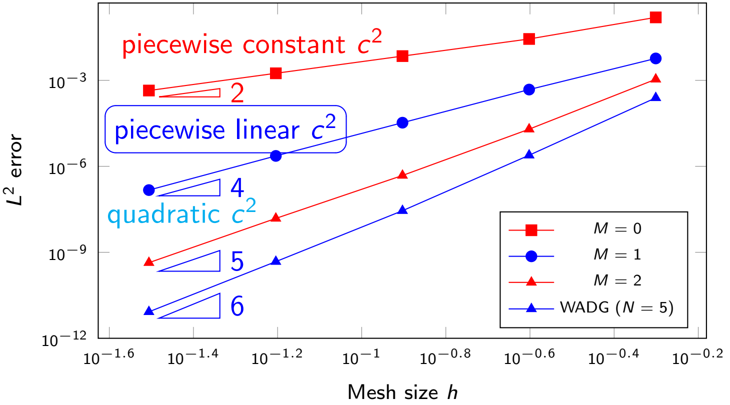
- Exploit continuous WADG steps: given $u(\boldsymbol{x})$, compute

$$P_N\left(u(\boldsymbol{x})c^2(\boldsymbol{x})\right), \qquad P_N = L^2 \text{ projection operator.}$$

- Our approach: approx. $c^2(\boldsymbol{x})$ with degree M polynomial, use fast Bernstein algorithms for polynomial multiplication and projection.

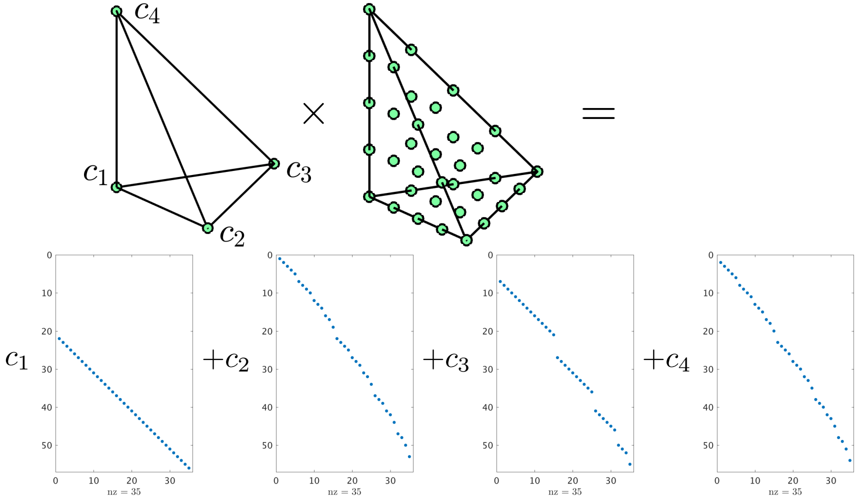
- Can reuse fast $O(N^3)$ Bernstein-based volume and surface kernels.

BBWADG: effect of approximating c^2 on accuracy



Approximating smooth $c^2(\boldsymbol{x})$ using L^2 projection:
$O(h^2)$ for $M = 0$, $O(h^4)$ for $M = 1$, $O(h^{M+3})$ for $0 < M \leq N - 2$.

Fast Bernstein polynomial multiplication



Bernstein polynomial multiplication ($M = 1$ shown), $O(N^3)$ cost for fixed M.

Fast Bernstein polynomial projection

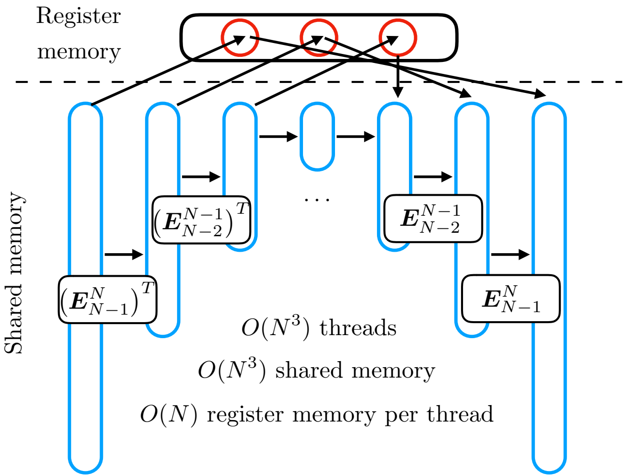
- Given $c^2(\boldsymbol{x})u(\boldsymbol{x})$ as a degree $(N+M)$ polynomial, apply L^2 projection matrix $\boldsymbol{P}_N^{N+M}$ to reduce to degree N.
- Polynomial L^2 projection matrix $\boldsymbol{P}_N^{N+M}$ under Bernstein basis:

$$\boldsymbol{P}_N^{N+M} = \underbrace{\sum_{j=0}^{N} c_j \boldsymbol{E}_{N-j}^{N} \left(\boldsymbol{E}_{N-j}^{N}\right)^T}_{\tilde{\boldsymbol{P}}_N} \left(\boldsymbol{E}_N^{N+M}\right)^T$$

- "Telescoping" form of $\tilde{\boldsymbol{P}}_N$: $O(N^4)$ complexity, more GPU-friendly.

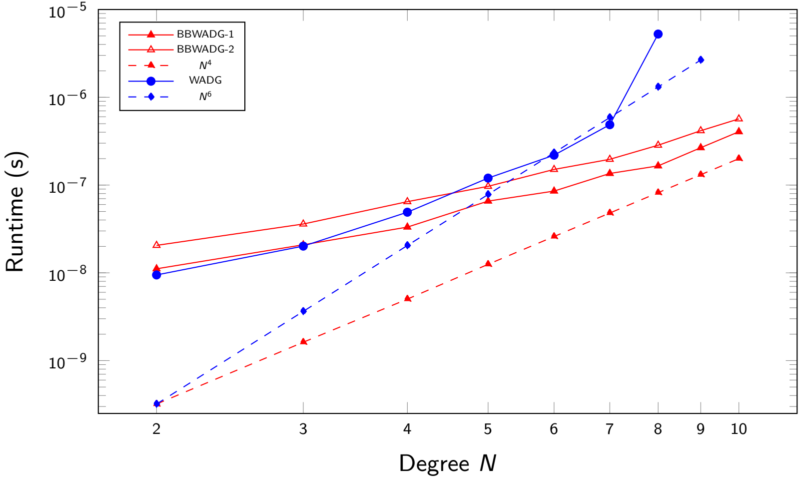
$$\left(c_0 \boldsymbol{I} + \boldsymbol{E}_{N-1}^{N} \left(c_1 \boldsymbol{I} + \boldsymbol{E}_{N-2}^{N-1} \left(c_2 \boldsymbol{I} + \cdots\right) \left(\boldsymbol{E}_{N-2}^{N-1}\right)^T\right) \left(\boldsymbol{E}_{N-1}^{N}\right)^T\right)$$

Sketch of GPU algorithm for $\tilde{P}_N$



$$\left(c_0\boldsymbol{I} + \boldsymbol{E}^N_{N-1}\left(c_1\boldsymbol{I} + \boldsymbol{E}^{N-1}_{N-2}\left(c_2\boldsymbol{I} + \cdots\right)\left(\boldsymbol{E}^{N-1}_{N-2}\right)^T\right)\left(\boldsymbol{E}^N_{N-1}\right)^T\right)$$

BBWADG: computational runtime (3D acoustics)



Per-element runtimes of update kernels for BBWADG vs WADG (acoustic). We observe an asymptotic complexity of $O(N^4)$ per element for $N \gg 1$.

Summary and future work

- Weight-adjusted DG: high order accuracy, provable stability, and efficiency in heterogeneous acoustic-elastic media.
- Current work: stable WADG for moving curved meshes (r-adaptivity) and extension to nonlinear conservation laws.
- This work has been supported by TOTAL E&P Research and Technology USA and the National Science Foundation under DMS-1712639 and DMS-1719818.

Guo, Chan (2020). Bernstein-Bézier weight-adjusted DG methods for wave propagation in heterogeneous media.
Guo, Acosta, Chan (2019). A weight-adjusted DG method for wave propagation in coupled elastic-acoustic media.
Guo, Chan (in preparation). Weight-adjusted high order DG methods on moving curved meshes.
Chan (2018). Weight-adjusted DG methods: matrix-valued weights and elastic wave prop. in heterogeneous media.
Chan, Hewett, Warburton (2017). Weight-adjusted DG methods: wave propagation in heterogeneous media.
Chan, Warburton (2017). GPU-accelerated Bernstein-Bezier discontinuous Galerkin methods for wave propagation.