

Overview of PhD Work & Future Research Plans

Brendan Keith

Hooke Research Fellowship Presentation
January 8, 2018

Outline

Prelude: Background

Outline

Prelude: Background

Act 1: Numerical Methods

- Discrete least-squares methods
- DPG methods
- DPG* methods
- Goal-oriented methods

Outline

Prelude: Background

Act 1: Numerical Methods

- Discrete least-squares methods
- DPG methods
- DPG* methods
- Goal-oriented methods

Act 2: Applications

- Solids
- Fluids
- Waves
- ~~Mixing patterns~~

Outline

Prelude: Background

Act 1: Numerical Methods

- Discrete least-squares methods
- DPG methods
- DPG* methods
- Goal-oriented methods

Act 2: Applications

- Solids
- Fluids
- Waves
- ~~Mixing patterns~~

~~Act 3: Software~~

- ~~ESEAS~~
- ~~hp 2D & hp 3D~~
- ~~Camellia~~

Outline

Prelude: Background

Act 1: Numerical Methods

- Discrete least-squares methods
- DPG methods
- DPG* methods
- Goal-oriented methods

Act 2: Applications

- Solids
- Fluids
- Waves
- ~~Mixing patterns~~

~~Act 3: Software~~

- ~~ESEAS~~
- ~~hp2D & hp3D~~
- ~~Camellia~~

Afterword: Future Acts

- NA Group
- OCIAM
- OxPDE

More information on my website:

<https://brendankeith.github.io>

Prelude

About My Work

PhD candidate at ICES

Supervisor: Dr Leszek Demkowicz

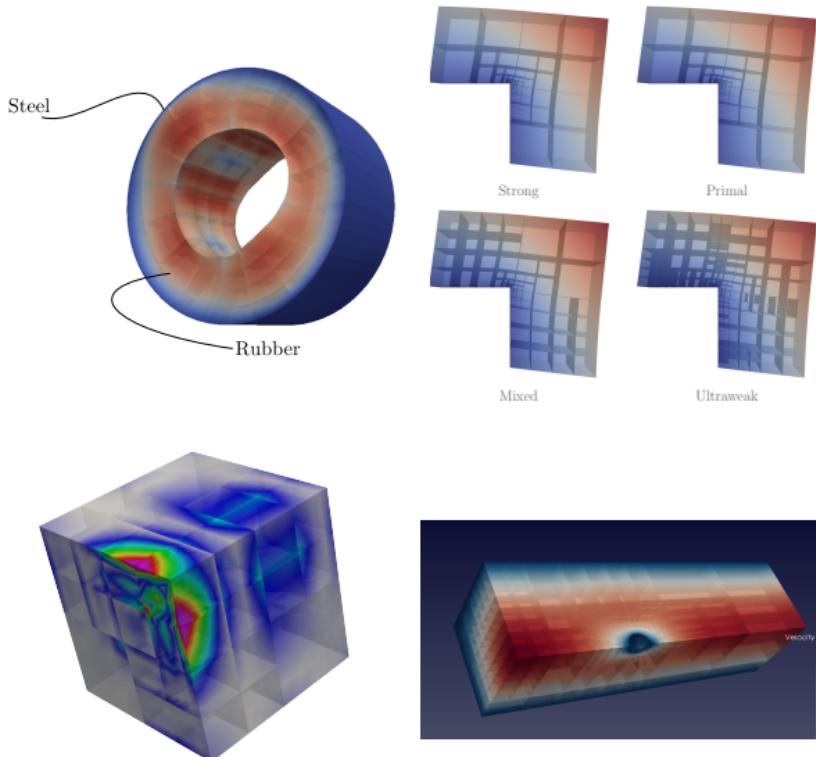
- Study discontinuous Petrov–Galerkin (DPG) methods

About My Work

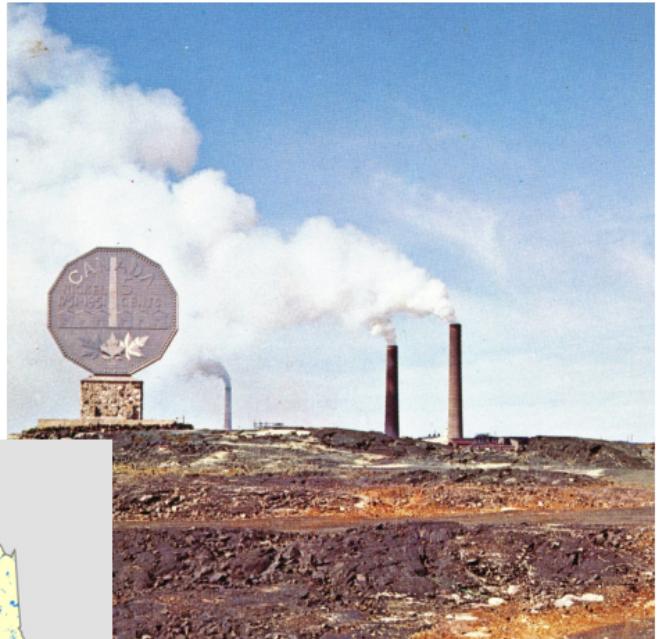
PhD candidate at ICES

Supervisor: Dr Leszek Demkowicz

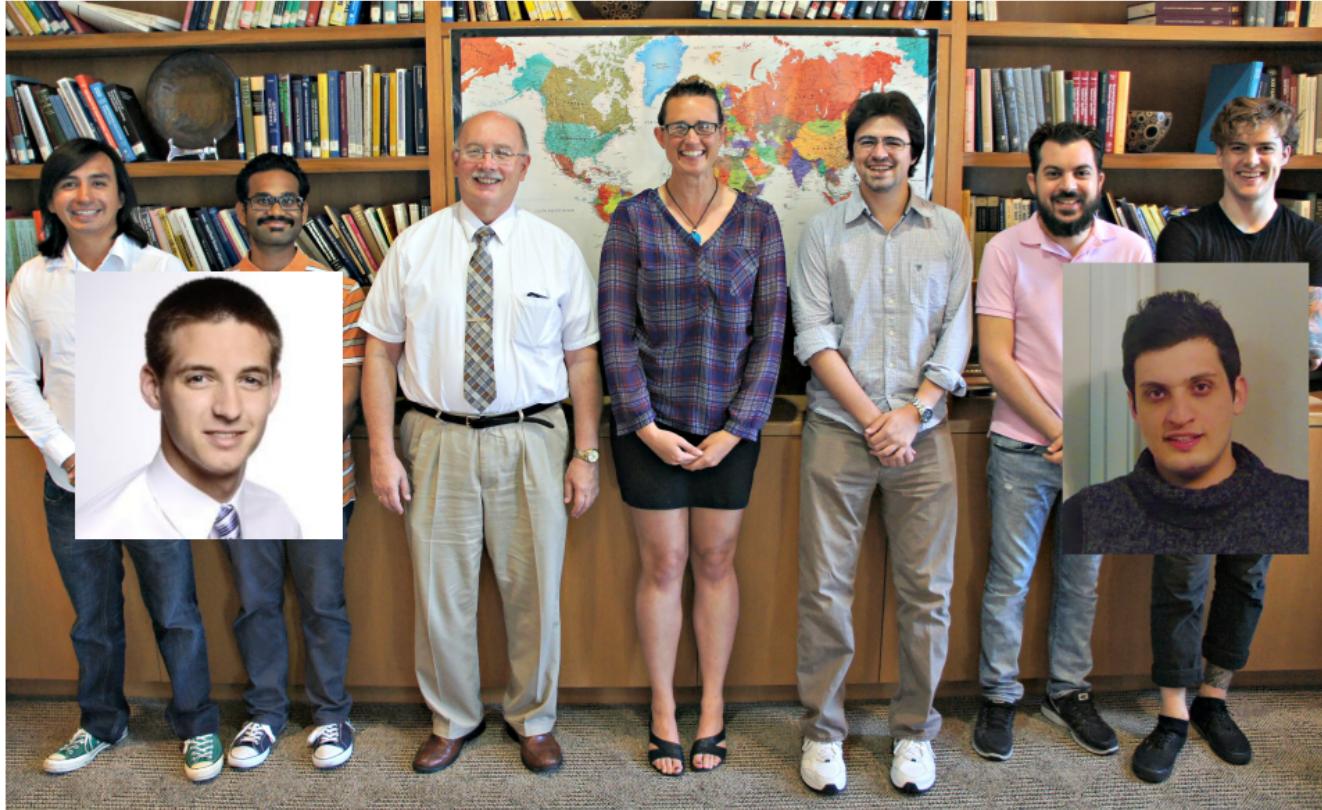
- Study discontinuous Petrov–Galerkin (DPG) methods
- Discovered DPG* methods
- Defined discrete least-squares (DLS) methods
- 3D simulations
- Adaptive mesh refinement
- Goal-oriented methods
- Applications!



About Me



About the DPG Group



About Life at UT

The Texas Applied Mathematics and Engineering Symposium (<http://tames.io>)



Act 1: Numerical Methods

Stability

Petrov–Galerkin methods

Let $b : U \times V \rightarrow \mathbb{R}$ be a continuous bilinear form.

Stability

Petrov–Galerkin methods

Let $b : U \times V \rightarrow \mathbb{R}$ be a continuous bilinear form.

Continuous stability

$$\inf_{u \in U} \sup_{v \in V} \frac{b(u, v)}{\|u\|_U \|v\|_V} = \gamma > 0 \quad \not\Rightarrow$$

Discrete stability

$$\inf_{u_h \in U_h} \sup_{v_h \in V_h} \frac{b(u_h, v_h)}{\|u_h\|_U \|v_h\|_V} = \gamma_h > 0$$

Stability

Petrov–Galerkin methods

Let $b : U \times V \rightarrow \mathbb{R}$ be a continuous bilinear form.

Continuous stability

$$\inf_{u \in U} \sup_{v \in V} \frac{b(u, v)}{\|u\|_U \|v\|_V} = \gamma > 0 \quad \not\Rightarrow$$

Discrete stability

$$\inf_{u_h \in U_h} \sup_{v_h \in V_h} \frac{b(u_h, v_h)}{\|u_h\|_U \|v_h\|_V} = \gamma_h > 0$$

Q: How to satisfy the discrete inf-sup condition

$$\sup_{v_h \in V_h} \frac{b(u_h, v_h)}{\|v_h\|_V} \geq \gamma_h \|u_h\|_U, \quad \forall u_h \in U_h \quad ?$$

Stability

Petrov–Galerkin methods

Let $b : U \times V \rightarrow \mathbb{R}$ be a continuous bilinear form.

Continuous stability

$$\inf_{u \in U} \sup_{v \in V} \frac{b(u, v)}{\|u\|_U \|v\|_V} = \gamma > 0 \quad \nRightarrow$$

Discrete stability

$$\inf_{u_h \in U_h} \sup_{v_h \in V_h} \frac{b(u_h, v_h)}{\|u_h\|_U \|v_h\|_V} = \gamma_h > 0$$

Q: How to satisfy the discrete inf-sup condition

$$\sup_{v_h \in V_h} \frac{b(u_h, v_h)}{\|v_h\|_V} \geq \gamma_h \|u_h\|_U, \quad \forall u_h \in U_h \quad ?$$

A: Fix U_h and increase the dimension of V_h until satisfied.

Discrete Least-Squares Methods

Q: How to define a discrete solution when $\dim(V_h) > \dim(U_h)$?

B. Keith, S. Petrides, F. Fuentes, and L. Demkowicz.

Discrete least-squares finite element methods.

Comput. Methods Appl. Mech. Engrg., 327:226–255, 2017.

Discrete Least-Squares Methods

Q: How to define a discrete solution when $\dim(V_h) > \dim(U_h)$?

A: By seeking a (discrete) least-squares best fit u_h^{opt} .

Discrete Least-Squares Methods

Q: How to define a discrete solution when $\dim(V_h) > \dim(U_h)$?

A: By seeking a (discrete) least-squares best fit u_h^{opt} .

NB: For fixed U_h , the optimally stable FEM is always a minimum residual method!

Stiffness matrix:

$$B_{ij} = b(u_j, v_i)$$

Load vector:

$$l_i = \ell(v_i)$$

Discrete Least-Squares Methods

Q: How to define a discrete solution when $\dim(V_h) > \dim(U_h)$?

A: By seeking a (discrete) least-squares best fit u_h^{opt} .

NB: For fixed U_h , the optimally stable FEM is always a minimum residual method!

Problem:

Stiffness matrix:

$$B^T B u = B^T l$$

$$B_{ij} = b(u_j, v_i)$$

Load vector:

$$l_i = \ell(v_i)$$

Discrete Least-Squares Methods

Q: How to define a discrete solution when $\dim(V_h) > \dim(U_h)$?

A: By seeking a (discrete) least-squares best fit u_h^{opt} .

NB: For fixed U_h , the optimally stable FEM is always a minimum residual method!

Problem:

Stiffness matrix:

$$B^T B u = B^T l$$

$$B_{ij} = b(u_j, v_i)$$

But!

Load vector:

$$\text{cond}(B^T B) = \text{cond}(B)^2$$

$$l_i = \ell(v_i)$$

Discrete Least-Squares Methods

Q: How to define a discrete solution when $\dim(V_h) > \dim(U_h)$?

A: By seeking a (discrete) least-squares best fit u_h^{opt} .

NB: For fixed U_h , the optimally stable FEM is always a minimum residual method!

Problem:

Stiffness matrix:

$$B_{ij} = b(u_j, v_i)$$

Load vector:

$$l_i = \ell(v_i)$$

~~$$B^T B u = B^T l$$~~

$$B^T G^{-1} B u = B^T G^{-1} l$$

where

$$G \approx BB^T$$

Discrete Least-Squares Methods

Q: How to define a discrete solution when $\dim(V_h) > \dim(U_h)$?

A: By seeking a (discrete) least-squares best fit u_h^{opt} .

NB: For fixed U_h , the optimally stable FEM is always a minimum residual method!

Problem:

Stiffness matrix:

$$B_{ij} = b(u_j, v_i)$$

Load vector:

$$l_i = \ell(v_i)$$

~~$$B^T B u = B^T l$$~~

$$B^T G^{-1} B u = B^T G^{-1} l$$

where

$$G \approx BB^T$$

DPG: G is block-diagonal

DPG* Methods

Q: How to define a discrete solution when $\dim(V_h) < \dim(U_h)$?

L. Demkowicz, J. Gopalakrishnan, and B. Keith.
Analysis of a DPG* method for the Poisson equation.
In preparation, 2018.

B. Keith, L. Demkowicz, and J. Gopalakrishnan.
DPG* method.
ICES Report 17-25, The University of Texas at Austin, 2017.

DPG* Methods

Q: How to define a discrete solution when $\dim(V_h) < \dim(U_h)$?

A: By seeking a minimum norm solution. i.e.

$$\mathbf{u} = \arg \min_{\mathbf{u}} \mathbf{u}^T \mathbf{G} \mathbf{u} \quad \text{subject to} \quad \mathbf{B} \mathbf{u} = \mathbf{l}.$$

DPG* Methods

Q: How to define a discrete solution when $\dim(V_h) < \dim(U_h)$?

A: By seeking a minimum norm solution. i.e.

$$\mathbf{u} = \arg \min_{\mathbf{u}} \mathbf{u}^T \mathbf{G} \mathbf{u} \quad \text{subject to} \quad \mathbf{B} \mathbf{u} = \mathbf{l}.$$

With DPG, we solve

$$\mathbf{B}^T \mathbf{G}^{-1} \mathbf{B} \mathbf{u} = \mathbf{B}^T \mathbf{G}^{-1} \mathbf{l}$$

DPG* Methods

Q: How to define a discrete solution when $\dim(V_h) < \dim(U_h)$?

A: By seeking a minimum norm solution. i.e.

$$\mathbf{u} = \arg \min_{\mathbf{u}} \mathbf{u}^T \mathbf{G} \mathbf{u} \quad \text{subject to} \quad \mathbf{B} \mathbf{u} = \mathbf{l}.$$

With DPG, we solve

$$\mathbf{B}^T \mathbf{G}^{-1} \mathbf{B} \mathbf{u} = \mathbf{B}^T \mathbf{G}^{-1} \mathbf{l}$$

With DPG*, we solve

$$\mathbf{B} \mathbf{G}^{-1} \mathbf{B}^T \mathbf{w} = \mathbf{l}$$

DPG* Methods

Q: How to define a discrete solution when $\dim(V_h) < \dim(U_h)$?

A: By seeking a minimum norm solution. i.e.

$$\mathbf{u} = \arg \min_{\mathbf{u}} \mathbf{u}^T \mathbf{G} \mathbf{u} \quad \text{subject to} \quad \mathbf{B} \mathbf{u} = \mathbf{l}.$$

With DPG, we solve

$$\mathbf{B}^T \mathbf{G}^{-1} \mathbf{B} \mathbf{u} = \mathbf{B}^T \mathbf{G}^{-1} \mathbf{l}$$

With DPG*, we solve

$$\mathbf{B} \mathbf{G}^{-1} \mathbf{B}^T \mathbf{w} = \mathbf{l}$$

and then post-process \mathbf{w} :

$$\mathbf{u} = \mathbf{G}^{-1} \mathbf{B}^T \mathbf{w}$$

Highlights

- ★ DPG/DPG* always deliver SPD (HPD) stiffness matrices

Highlights

- ★ DPG/DPG* always deliver SPD (HPD) stiffness matrices
- ★ DPG/DPG* have built-in stability

Highlights

- ★ DPG/DPG* always deliver SPD (HPD) stiffness matrices
- ★ DPG/DPG* have built-in stability
- ★ DPG has a built-in error estimator

Highlights

- ★ DPG/DPG* always deliver SPD (HPD) stiffness matrices
- ★ DPG/DPG* have built-in stability
- ★ DPG has a built-in error estimator
- ★ DPG* works for some problems without uniqueness

Goal-Oriented Methods

Consider the following abstract model:

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathcal{F}(\mathbf{x}, \mathbf{u}; \boldsymbol{\mu}), \\ \mathbf{y} = \mathcal{G}(\mathbf{x}). \end{cases}$$

Solution variable: \mathbf{x}

Input: \mathbf{u}

Model parameter(s): $\boldsymbol{\mu}$

QoI (or output): \mathbf{y} .

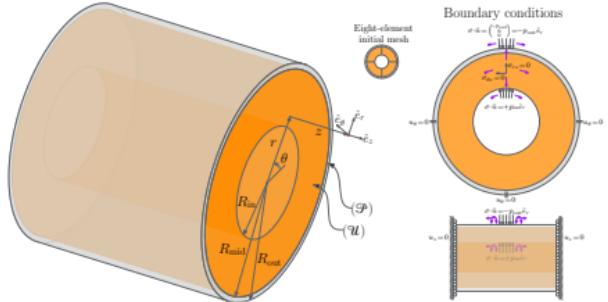
Research interest:

- By designing numerical methods with only the given output \mathbf{y} in mind, efficiency can sometimes be greatly improved.
- Example: goal-oriented adaptive mesh refinement.

Act 2: Applications

Structural Mechanics

Sheathed hose

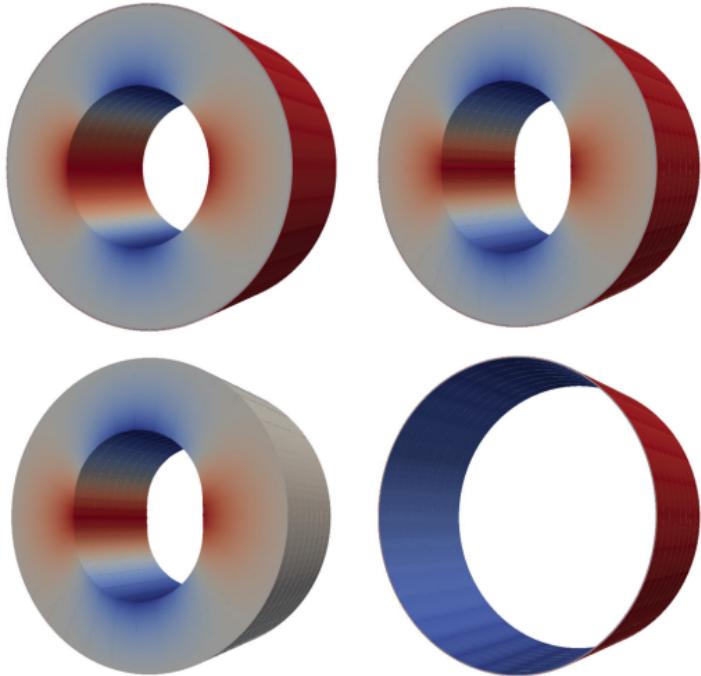


$$(E_{\text{steel}} = 200\text{GPA}, E_{\text{Rubber}} = 0.01\text{GPA})$$

Coupled formulations

- All coupled formulations are **mutually compatible** throughout the domain.

Circumferential stress: $\sigma_{\theta\theta}$



F. Fuentes, B. Keith, L. Demkowicz, and P. L. Tallec.

Coupled variational formulations of linear elasticity and the DPG methodology.
J. Comput. Phys., 348:715–731, 2017.

B. Keith, F. Fuentes, and L. Demkowicz.

The DPG methodology applied to different variational formulations of linear elasticity.
Comput. Methods Appl. Mech. Engrg., 309:579–609, 2016.

Fluid Mechanics

Oldroyd-B fluid

Conservation of mass and momentum:

$$\begin{aligned}\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \boldsymbol{\sigma} &= \rho \mathbf{f} \quad \text{on } \Omega \times (0, T) . \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{on } \Omega \times (0, T) .\end{aligned}$$

Constitutive law:

$$\boldsymbol{\sigma} = -p \mathbf{I} + 2\mu_S \boldsymbol{\varepsilon}(\mathbf{u}) + \mathbf{T},$$

where

$$\mathbf{T} + \lambda \mathcal{L}_u \mathbf{T} = 2\mu_P \boldsymbol{\varepsilon}(\mathbf{u}).$$

Lie derivative:

$$\mathcal{L}_u \mathbf{T} = \frac{\partial \mathbf{T}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{T} - (\nabla \mathbf{u} \mathbf{T} - \mathbf{T} \nabla^T \mathbf{u}).$$

Fluid Mechanics

Oldroyd-B fluid

Conservation of mass and momentum:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \boldsymbol{\sigma} = \rho \mathbf{f} \quad \text{on } \Omega \times (0, T). \\ \nabla \cdot \mathbf{u} = 0 \quad \text{on } \Omega \times (0, T).$$

Constitutive law:

$$\boldsymbol{\sigma} = -p \mathbf{I} + 2\mu_S \boldsymbol{\varepsilon}(\mathbf{u}) + \mathbf{T},$$

where

$$\mathbf{T} + \lambda \mathcal{L}_u \mathbf{T} = 2\mu_P \boldsymbol{\varepsilon}(\mathbf{u}).$$

Lie derivative:

$$\mathcal{L}_u \mathbf{T} = \frac{\partial \mathbf{T}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{T} - (\nabla \mathbf{u} \mathbf{T} - \mathbf{T} \nabla^T \mathbf{u}).$$

Quantity of interest

Drag coefficient

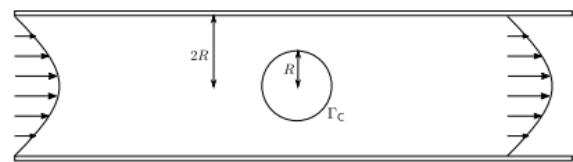
$$K(t) = \frac{1}{\mu \bar{u}} \int_{\Gamma_C} \hat{t} \cdot \hat{\mathbf{e}}_x \, ds.$$

Γ_C : boundary of cylinder

\hat{t} : traction

$\mu = \mu_S + \mu_P$: viscosity

\bar{u} : average inflow velocity.



Confined cylinder domain.

Fluid Mechanics

Oldroyd-B fluid

Conservation of mass and momentum:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \boldsymbol{\sigma} = \rho \mathbf{f} \quad \text{on } \Omega \times (0, T). \\ \nabla \cdot \mathbf{u} = 0 \quad \text{on } \Omega \times (0, T).$$

Constitutive law:

$$\boldsymbol{\sigma} = -p \mathbf{I} + 2\mu_S \boldsymbol{\varepsilon}(\mathbf{u}) + \mathbf{T},$$

where

$$\mathbf{T} + \lambda \mathcal{L}_u \mathbf{T} = 2\mu_P \boldsymbol{\varepsilon}(\mathbf{u}).$$

Lie derivative:

$$\mathcal{L}_u \mathbf{T} = \frac{\partial \mathbf{T}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{T} - (\nabla \mathbf{u} \mathbf{T} - \mathbf{T} \nabla^T \mathbf{u}).$$

Quantity of interest

Drag coefficient

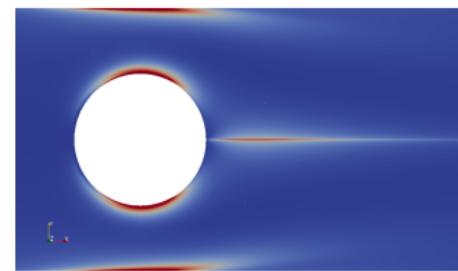
$$K(t) = \frac{1}{\mu \bar{u}} \int_{\Gamma_C} \hat{t} \cdot \hat{\mathbf{e}}_x \, ds.$$

Γ_C : boundary of cylinder

\hat{t} : traction

$\mu = \mu_S + \mu_P$: viscosity

\bar{u} : average inflow velocity.



Close-up of rescaled T_{11} -component from $\lambda = 0.9$.

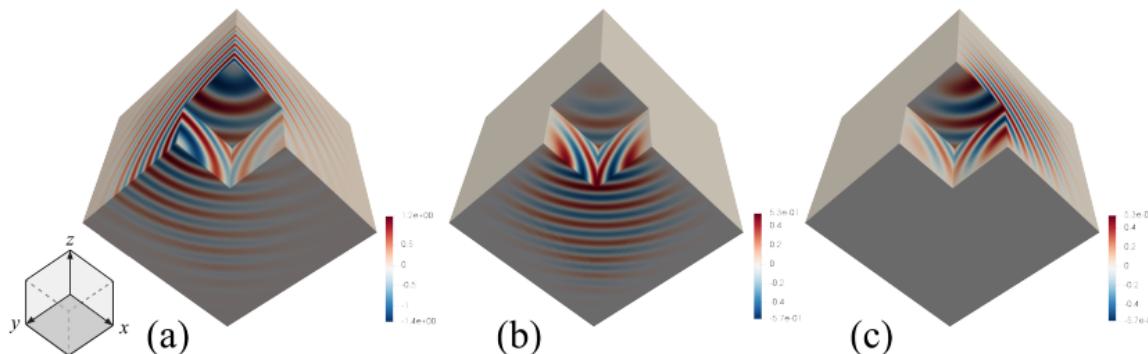
Wave Mechanics

- Acoustics
- Electromagnetics
- Elastodynamics

$$-\Delta p - \omega^2 p = f$$

$$\frac{1}{\mu} \nabla \times \nabla \times \mathbf{E} - \omega^2 \epsilon \mathbf{E} = i\omega \mathbf{J}^{\text{imp}}$$

$$-\nabla \cdot (\mathbf{C} : \nabla \mathbf{u}) - \rho \omega^2 \mathbf{u} = \mathbf{f}$$



Electromagnetic wave scattering of the discrete electric field \mathbf{E} .

(a) The x -component; (b) the y -component; (c) the z -component. Only the real part of the solution is visualized.

A. Vaziri Astaneh, B. Keith, and L. Demkowicz.

On perfectly matched layers and non-symmetric variational formulations.

Submitted, 2017.

Afterword: Future Acts

Viscoelastic Rate-Type Fluids With Stress Diffusion

Potential collaboration with Prof. Endre Süli

Oldroyd-B fluid **with stress diffusion**

Conservation of mass and momentum:

$$\begin{aligned} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \boldsymbol{\sigma} &= \rho \mathbf{f} \quad \text{on } \Omega \times (0, T). \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{on } \Omega \times (0, T). \end{aligned}$$

Constitutive law:

$$\boldsymbol{\sigma} = -p \mathbf{I} + 2\mu_S \boldsymbol{\varepsilon}(\mathbf{u}) + \mathbf{T},$$

where

$$\mathbf{T} + \lambda \mathcal{L}_u \mathbf{T} = 2\mu_P \boldsymbol{\varepsilon}(\mathbf{u}) + \epsilon \Delta \mathbf{T}.$$

Lie derivative:

$$\mathcal{L}_u \mathbf{T} = \frac{\partial \mathbf{T}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{T} - (\nabla \mathbf{u} \mathbf{T} - \mathbf{T} \nabla^T \mathbf{u}).$$

M. Bulíček, J. Málek, V. Průša, and E. Süli.

PDE analysis of a class of thermodynamically compatible viscoelastic rate-type fluids with stress-diffusion.

arXiv:1707.02350 (math.AP), 2017.

J. Málek, V. Průša, T. Skrivan, and E. Süli.

Thermodynamics of viscoelastic rate-type fluids with stress diffusion.

arXiv:1706.06277 (physics.flu-dyn), 2017.

Viscoelastic Rate-Type Fluids With Stress Diffusion

Potential collaboration with Prof. Endre Süli

Oldroyd-B fluid with stress diffusion

Conservation of mass and momentum:

$$\begin{aligned} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \boldsymbol{\sigma} &= \rho \mathbf{f} \quad \text{on } \Omega \times (0, T). \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{on } \Omega \times (0, T). \end{aligned}$$

Constitutive law:

$$\boldsymbol{\sigma} = -p \mathbf{I} + 2\mu_S \boldsymbol{\varepsilon}(\mathbf{u}) + \mathbf{T},$$

where

$$\mathbf{T} + \lambda \mathcal{L}_u \mathbf{T} = 2\mu_P \boldsymbol{\varepsilon}(\mathbf{u}) + \epsilon \Delta \mathbf{T}.$$

Lie derivative:

$$\mathcal{L}_u \mathbf{T} = \frac{\partial \mathbf{T}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{T} - (\nabla \mathbf{u} \mathbf{T} - \mathbf{T} \nabla^T \mathbf{u}).$$

M. Bulíček, J. Málek, V. Průša, and E. Süli.

PDE analysis of a class of thermodynamically compatible viscoelastic rate-type fluids with stress-diffusion.

arXiv:1707.02350 (math.AP), 2017.

- No systematic derivation of such fluid models with a stress diffusion term until now.
- Analyze such models with adaptive DG-type methods.
- Possible applications include polymer melts and die extrusion in 3D.

J. Málek, V. Průša, T. Skrivan, and E. Süli.

Thermodynamics of viscoelastic rate-type fluids with stress diffusion.

arXiv:1706.06277 (physics.flu-dyn), 2017.

Other Possibilities

The Numerical Analysis Group

Prof. Patrick Farrell Various things.

Prof. Andy Wathen Preconditioners.

OCIAM/The Wolfson Centre

Prof. Andreas Münch Liquid polymers. Phase-field models.

Prof. Sarah Waters Fluid flows.

Prof. Dominic Vella Swelling-induced pattern formation.

OxPDE

Prof. John Ball Nonlinear minimum residual theory; applications to elasticity.

Other Possibilities

The Numerical Analysis Group

Prof. Patrick Farrell Various things.

Prof. Andy Wathen Preconditioners.

OCIAM/The Wolfson Centre

Prof. Andreas Münch Liquid polymers. Phase-field models.

Prof. Sarah Waters Fluid flows.

Prof. Dominic Vella Swelling-induced pattern formation.

OxPDE

Prof. John Ball Nonlinear minimum residual theory; applications to elasticity.

And anyone else with an interest in my work!

References

- L. Demkowicz, J. Gopalakrishnan, and B. Keith.
Analysis of a DPG* method for the Poisson equation.
In preparation, 2018.
- F. Fuentes, B. Keith, L. Demkowicz, and S. Nagaraj.
Orientation embedded high order shape functions for
the exact sequence elements of all shapes.
Comput. Math. Appl., 70(4):353–458, 2015.
- F. Fuentes, B. Keith, L. Demkowicz, and P. L. Tallec.
Coupled variational formulations of linear elasticity and
the DPG methodology.
J. Comput. Phys., 348:715–731, 2017.
- B. Keith, L. Demkowicz, and J. Gopalakrishnan.
DPG* method.
ICES Report 17-25, The University of Texas at Austin, 2017.
- B. Keith, F. Fuentes, and L. Demkowicz.
The DPG methodology applied to different variational
formulations of linear elasticity.
Comput. Methods Appl. Mech. Engrg., 309:579–609,
2016.
- B. Keith, P. Knechtges, N. V. Roberts, S. Elgeti, M. Behr,
and L. Demkowicz.
An ultraweak DPG method for viscoelastic fluids.
J. Non-Newton. Fluid Mech., 247:107–122, 2017.
- B. Keith, S. Petrides, F. Fuentes, and L. Demkowicz.
Discrete least-squares finite element methods.
Comput. Methods Appl. Mech. Engrg., 327:226–255,
2017.
- B. Keith and N. V. Roberts.
Goal-oriented adaptive mesh refinement with a DPG
method for viscoelastic fluids.
In preparation, 2018.
- B. Keith, A. Vaziri Astaneh, and L. Demkowicz.
Goal-oriented adaptive mesh refinement for
non-symmetric functional settings.
arXiv:1711.01996 (math.NA), 2017.
- A. Vaziri Astaneh, B. Keith, and L. Demkowicz.
On perfectly matched layers and non-symmetric
variational formulations.
Submitted, 2017.

Thank you!

Extra slides

Goal-oriented adaptivity

Problem:

$$\begin{cases} \text{Find } u \in U : \\ b(u, v) = \ell(v), \quad \forall v \in V \end{cases}$$

Dual Problem:

$$\begin{cases} \text{Find } v \in V : \\ b(u, v) = g(u), \quad \forall u \in U \end{cases}$$

Goal-oriented adaptivity

Problem:

$$\begin{cases} \text{Find } u \in U : \\ b(u, v) = \ell(v), \quad \forall v \in V \end{cases}$$

Dual Problem:

$$\begin{cases} \text{Find } v \in V : \\ b(u, v) = g(u), \quad \forall u \in U \end{cases}$$

DPG–DPG* orthogonality:

$$g(u - u_h) = b(u - u_h, v)$$

B. Keith, L. Demkowicz, and J. Gopalakrishnan.

DPG* method.

ICES Report 17-25, The University of Texas at Austin, 2017.

B. Keith, A. Vaziri Astaneh, and L. Demkowicz.

Goal-oriented adaptive mesh refinement for non-symmetric functional settings.

arXiv:1711.01996 (math.NA), 2017.

Goal-oriented adaptivity

Problem:

$$\begin{cases} \text{Find } u \in U : \\ b(u, v) = \ell(v), \quad \forall v \in V \end{cases}$$

Dual Problem:

$$\begin{cases} \text{Find } v \in V : \\ b(u, v) = g(u), \quad \forall u \in U \end{cases}$$

DPG–DPG* orthogonality:

$$g(u - u_h) = b(u - u_h, v) = b(u - u_h, v - v_h)$$

B. Keith, L. Demkowicz, and J. Gopalakrishnan.

DPG* method.

ICES Report 17-25, The University of Texas at Austin, 2017.

B. Keith, A. Vaziri Astaneh, and L. Demkowicz.

Goal-oriented adaptive mesh refinement for non-symmetric functional settings.

arXiv:1711.01996 (math.NA), 2017.

Goal-oriented adaptivity

Problem:

$$\begin{cases} \text{Find } u \in U : \\ b(u, v) = \ell(v), \quad \forall v \in V \end{cases}$$

Dual Problem:

$$\begin{cases} \text{Find } v \in V : \\ b(u, v) = g(u), \quad \forall u \in U \end{cases}$$

DPG–DPG* orthogonality:

$$g(u - u_h) = b(u - u_h, v) = b(u - u_h, v - v_h) = \ell(v - v_h)$$

B. Keith, L. Demkowicz, and J. Gopalakrishnan.

DPG* method.

ICES Report 17-25, The University of Texas at Austin, 2017.

B. Keith, A. Vaziri Astaneh, and L. Demkowicz.

Goal-oriented adaptive mesh refinement for non-symmetric functional settings.

arXiv:1711.01996 (math.NA), 2017.

Goal-oriented adaptivity

Problem:

$$\begin{cases} \text{Find } u \in U : \\ b(u, v) = \ell(v), \quad \forall v \in V \end{cases}$$

Dual Problem:

$$\begin{cases} \text{Find } v \in V : \\ b(u, v) = g(u), \quad \forall u \in U \end{cases}$$

DPG–DPG* orthogonality:

$$\begin{aligned} g(u - u_h) &= b(u - u_h, v) = b(u - u_h, v - v_h) = \ell(v - v_h) \\ &\leq \underbrace{\|\mathcal{B}u_h - \ell\|_{V'}}_{\lesssim \eta(u_h)} \underbrace{\|v - v_h\|_V}_{\lesssim \eta'(v_r)} \end{aligned}$$

B. Keith, L. Demkowicz, and J. Gopalakrishnan.

DPG* method.

ICES Report 17-25, The University of Texas at Austin, 2017.

B. Keith, A. Vaziri Astaneh, and L. Demkowicz.

Goal-oriented adaptive mesh refinement for non-symmetric functional settings.

arXiv:1711.01996 (math.NA), 2017.

Goal-oriented adaptivity

Problem:

$$\begin{cases} \text{Find } u \in U : \\ b(u, v) = \ell(v), \quad \forall v \in V \end{cases}$$

Dual Problem:

$$\begin{cases} \text{Find } v \in V : \\ b(u, v) = g(u), \quad \forall u \in U \end{cases}$$

DPG–DPG* orthogonality:

$$\begin{aligned} g(u - u_h) &= b(u - u_h, v) = b(u - u_h, v - v_h) = \ell(v - v_h) \\ &\leq \underbrace{\|\mathcal{B}u_h - \ell\|_{V'}}_{\lesssim \eta(u_h)} \underbrace{\|v - v_h\|_V}_{\lesssim \eta'(v_r)} \end{aligned}$$

$$g(u - u_h) \lesssim \eta(u_h) \eta'(v_r)$$

B. Keith, L. Demkowicz, and J. Gopalakrishnan.

DPG* method.

ICES Report 17-25, The University of Texas at Austin, 2017.

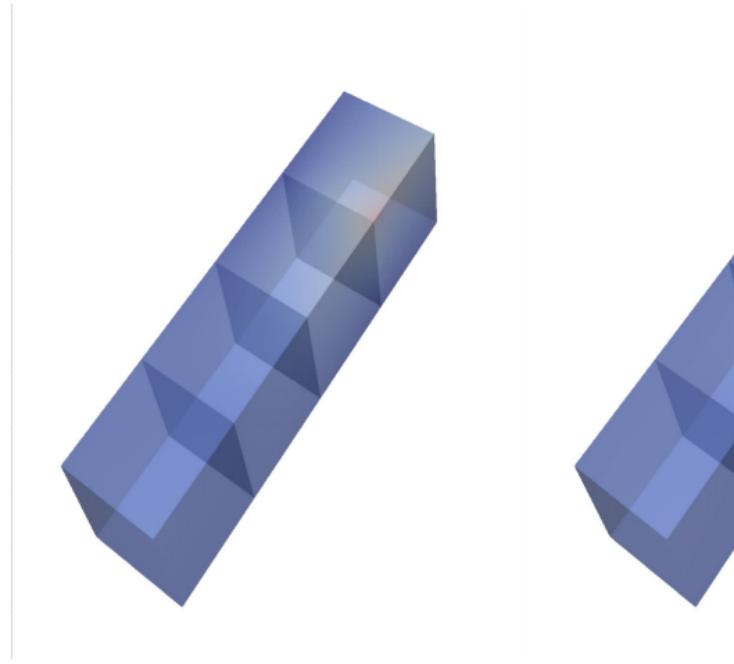
B. Keith, A. Vaziri Astaneh, and L. Demkowicz.

Goal-oriented adaptive mesh refinement for non-symmetric functional settings.

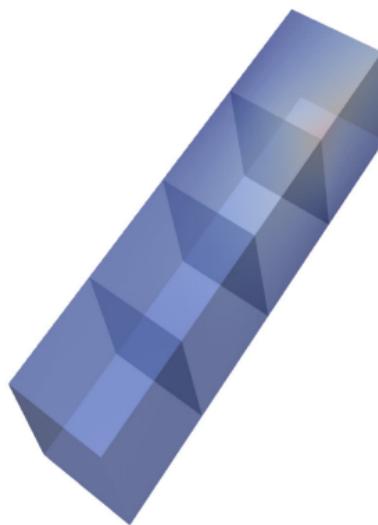
arXiv:1711.01996 (math.NA), 2017.

Goal-Oriented Adaptivity

Solution-oriented strategy



Goal-oriented strategy (1 of 3)



B. Keith and N. V. Roberts.

Goal-oriented adaptive mesh refinement with a DPG method for viscoelastic fluids.
In preparation, 2018.

B. Keith, A. Vaziri Astaneh, and L. Demkowicz.

Goal-oriented adaptive mesh refinement for non-symmetric functional settings.
arXiv:1711.01996 (math.NA), 2017.

Goal-Oriented Adaptivity

Goal-oriented adaptive strategy

1. Solve the primal and dual DPG problems, for u_h and v_h .
2. Estimate the global error in the quantity of interest and cease further computations if the estimate is sufficiently small.
3. Estimate local errors in the quantity of interest by computing η_K and η'_K for each element K in the mesh and mark those elements for refinement as dictated by a user-determined marking strategy.
4. Refine all marked elements and construct a new mesh with a user-determined refinement strategy.

Discrete Least-Squares Methods

Factorization, etc.

We are solving the **discrete** least squares problem

$$\mathbf{u}^{\text{opt}} = \arg \min_{\mathbf{u} \in \mathbb{R}^N} \|\mathbf{L}^{-1}(\mathbf{B}\mathbf{u} - \mathbf{l})\|_2.$$

or

$$(\mathbf{L}^{-1}\mathbf{B})^\top(\mathbf{L}^{-1}\mathbf{B}) \mathbf{u} = (\mathbf{L}^{-1}\mathbf{B})^\top(\mathbf{L}^{-1}\mathbf{l}). \quad (\star\star)$$

Discrete Least-Squares Methods

Factorization, etc.

We are solving the **discrete** least squares problem

$$\mathbf{u}^{\text{opt}} = \arg \min_{\mathbf{u} \in \mathbb{R}^N} \|\mathbf{L}^{-1}(\mathbf{B}\mathbf{u} - \mathbf{l})\|_2.$$

or

$$(\mathbf{L}^{-1}\mathbf{B})^T(\mathbf{L}^{-1}\mathbf{B}) \mathbf{u} = (\mathbf{L}^{-1}\mathbf{B})^T(\mathbf{L}^{-1}\mathbf{l}). \quad (**)$$

- Note that $(**)$ suggests an efficient way to locally construct $\mathbf{B}^T \mathbf{G}^{-1} \mathbf{B}$.
- It hints at solution strategies based on least squares solvers.
Trefethen and Bau (SIAM, 1997) belabour least squares problems and there is a untapped literature on sparse least squares solvers.
- There are many stable solution techniques for SPD linear systems, however the matrix $\mathbf{B}^T \mathbf{G}^{-1} \mathbf{B}$ has a **squared** condition number.
- This opportunity does not exist with FOSLS because it does not discretize the Riesz map, $\mathcal{R}_V : V \rightarrow V'$.

Discrete Least-Squares Methods

Factorization, etc.

We are solving the **discrete** least squares problem

$$\mathbf{u}^{\text{opt}} = \arg \min_{\mathbf{u} \in \mathbb{R}^N} \|\mathbf{L}^{-1}(\mathbf{B}\mathbf{u} - \mathbf{l})\|_2.$$

or

$$(\mathbf{L}^{-1}\mathbf{B})^T(\mathbf{L}^{-1}\mathbf{B}) \mathbf{u} = (\mathbf{L}^{-1}\mathbf{B})^T(\mathbf{L}^{-1}\mathbf{l}). \quad (**)$$

- Note that $(**)$ suggests an efficient way to locally construct $\mathbf{B}^T \mathbf{G}^{-1} \mathbf{B}$.
- It hints at solution strategies based on least squares solvers.
Trefethen and Bau (SIAM, 1997) belabour least squares problems and there is a untapped literature on sparse least squares solvers.
- There are many stable solution techniques for SPD linear systems, however the matrix $\mathbf{B}^T \mathbf{G}^{-1} \mathbf{B}$ has a **squared** condition number.
- This opportunity does **not** exist with FOSLS because it does not discretize the Riesz map, $\mathcal{R}_V : V \rightarrow V'$.

Discrete Least-Squares Methods

Factorization, etc.

We are solving the **discrete** least squares problem

$$\mathbf{u}^{\text{opt}} = \arg \min_{\mathbf{u} \in \mathbb{R}^N} \|\mathbf{L}^{-1}(\mathbf{B}\mathbf{u} - \mathbf{l})\|_2.$$

or

$$(\mathbf{L}^{-1}\mathbf{B})^T(\mathbf{L}^{-1}\mathbf{B}) \mathbf{u} = (\mathbf{L}^{-1}\mathbf{B})^T(\mathbf{L}^{-1}\mathbf{l}). \quad (**)$$

- Note that $(**)$ suggests an efficient way to locally construct $\mathbf{B}^T \mathbf{G}^{-1} \mathbf{B}$.
- It hints at solution strategies based on least squares solvers.
Trefethen and Bau (SIAM, 1997) belabour least squares problems and there is a untapped literature on sparse least squares solvers.
- There are many stable solution techniques for SPD linear systems, however the matrix $\mathbf{B}^T \mathbf{G}^{-1} \mathbf{B}$ has a **squared** condition number.
- This opportunity does **not** exist with FOSLS because it does not discretize the Riesz map, $\mathcal{R}_V : V \rightarrow V'$.

Discrete Least-Squares Methods

Factorization, etc.

We are solving the **discrete** least squares problem

$$\mathbf{u}^{\text{opt}} = \arg \min_{\mathbf{u} \in \mathbb{R}^N} \|\mathbf{L}^{-1}(\mathbf{B}\mathbf{u} - \mathbf{l})\|_2.$$

or

$$(\mathbf{L}^{-1}\mathbf{B})^T(\mathbf{L}^{-1}\mathbf{B}) \mathbf{u} = (\mathbf{L}^{-1}\mathbf{B})^T(\mathbf{L}^{-1}\mathbf{l}). \quad (**)$$

- Note that $(**)$ suggests an efficient way to locally construct $\mathbf{B}^T \mathbf{G}^{-1} \mathbf{B}$.
- It hints at solution strategies based on least squares solvers.
Trefethen and Bau (SIAM, 1997) belabour least squares problems and there is a untapped literature on sparse least squares solvers.
- There are many stable solution techniques for SPD linear systems, however the matrix $\mathbf{B}^T \mathbf{G}^{-1} \mathbf{B}$ has a **squared** condition number.
- This opportunity does **not** exist with FOSLS because it does not discretize the Riesz map, $\mathcal{R}_V : V \rightarrow V'$.