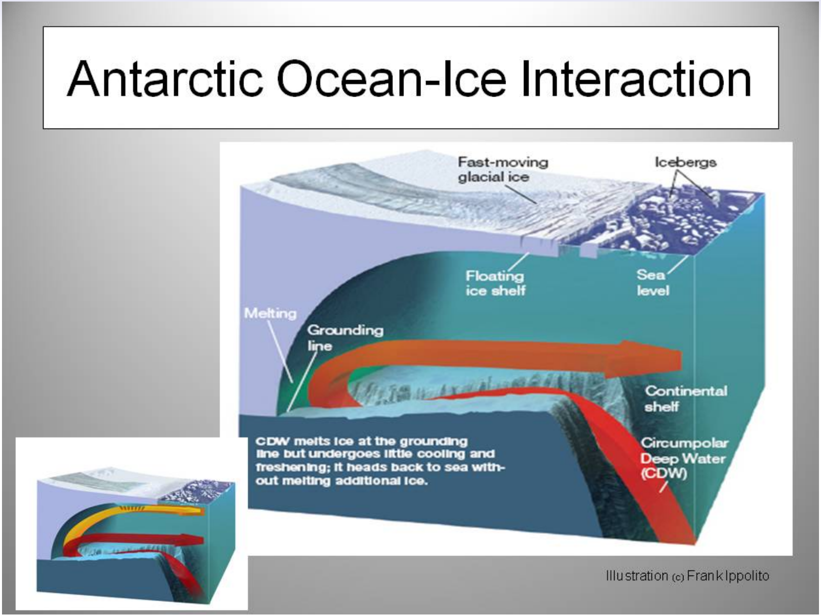
Implicit integration of 3D ice sheet flow using hybrid factorization/relaxation block preconditioning

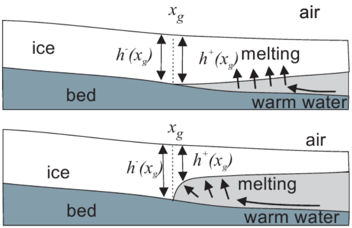
Jed Brown

Copper Mountain 2010-04-08

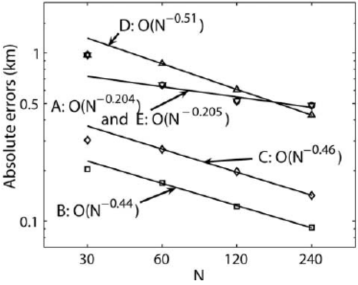


Bindschadler 2008

Grounding lines

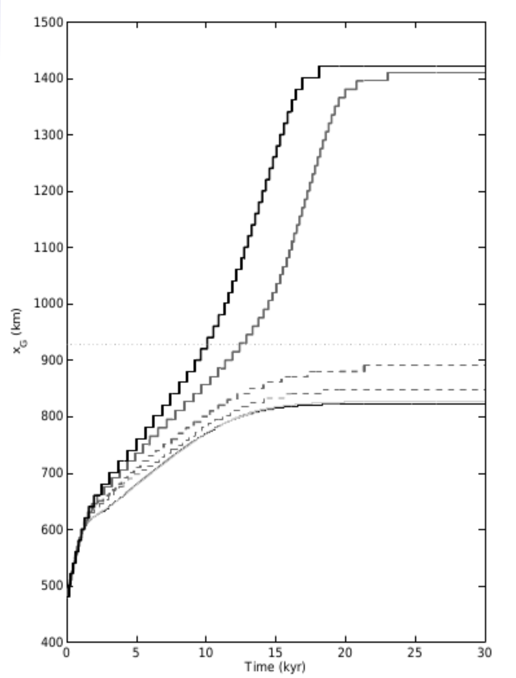


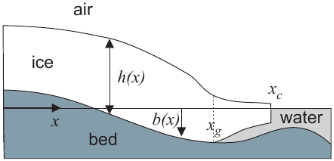
Schoof 2007



Bueler et. al. 2005

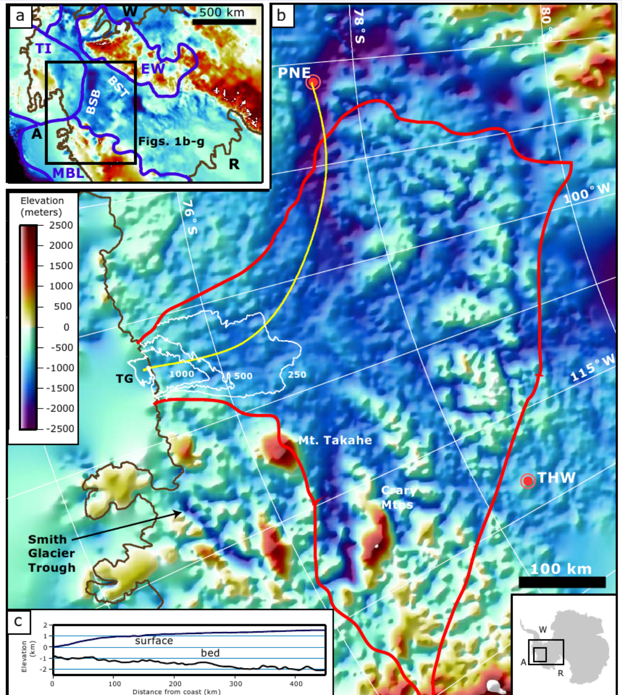
- ocean circulation is very sensitive to grounding line geometry, feedback
- current models are less than first-order accurate at margins
- extremely high resolution needed for qualitatively correct results on Eulerian meshes
- non-shallow physics applies in vicinity of grounding line





(Schoof 2007)

Evolution of grounding line location on 20, 15, 10, 7.5 and 2.5 kilometer meshes in one horizontal dimension. (*Durand et al. 2009*)



Holt et al.

2006

Non-Newtonian Stokes system: velocity $\boldsymbol{u}$, pressure p

$$-\nabla\cdot(\eta D\boldsymbol{u}) + \nabla p - \boldsymbol{f} = 0$$
$$\nabla\cdot\boldsymbol{u} = 0$$

$$D\boldsymbol{u} = \tfrac{1}{2}\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right)$$
$$\gamma(D\boldsymbol{u}) = \tfrac{1}{2}D\boldsymbol{u}:D\boldsymbol{u}$$
$$\eta(\gamma) = B(\Theta,\dots)\big(\epsilon+\gamma\big)^{\frac{\mathfrak{p}-2}{2}}$$
$$\mathfrak{p} = 1 + \tfrac{1}{\mathfrak{n}} \approx \tfrac{4}{3}$$
$$T = \mathbf{1} - \boldsymbol{n}\otimes\boldsymbol{n}$$

with boundary conditions

$$(\eta D\boldsymbol{u} - p\mathbf{1})\cdot\boldsymbol{n} = \begin{cases}\mathbf{0} & \text{free surface}\\ -\rho_w z\boldsymbol{n} & \text{ice-ocean interface}\end{cases}$$

$$\boldsymbol{u} = \mathbf{0} \qquad \text{frozen bed}, \Theta < \Theta_0$$

$$\left.\begin{aligned}\boldsymbol{u}\cdot\boldsymbol{n} &= \boldsymbol{g}_{\text{melt}}(T\boldsymbol{u},\dots)\\ T(\eta D\boldsymbol{u} - p\mathbf{1})\cdot\boldsymbol{n} &= \boldsymbol{g}_{\text{slip}}(T\boldsymbol{u},\dots)\end{aligned}\right\}\text{nonlinear slip}, \Theta \ge \Theta_0$$

$$\boldsymbol{g}_{\text{slip}}(T\boldsymbol{u}) = \beta_{\mathfrak{m}}(\dots)|T\boldsymbol{u}|^{\mathfrak{m}-1}T\boldsymbol{u}$$

Navier $\mathfrak{m} = 1$, Weertman $\mathfrak{m} \approx \frac{1}{3}$, Coulomb $\mathfrak{m} = 0$.

Other critical equations

- Mesh motion: $\boldsymbol{x}$

$$-\nabla\cdot\boldsymbol{\sigma} = 0 \qquad \boldsymbol{\sigma} = \mu\left[2D\boldsymbol{w} + (\nabla\boldsymbol{w})^T\nabla\boldsymbol{w}\right] + \lambda\,\mathrm{tr}(\nabla\boldsymbol{w})\mathbf{1}$$

surface: $(\dot{\boldsymbol{x}} - \boldsymbol{u})\cdot\boldsymbol{n} = q_{BL},\; T\boldsymbol{\sigma}\cdot\boldsymbol{n} = 0 \qquad \boldsymbol{w} = \boldsymbol{x} - \boldsymbol{x}_0$

- Heat transport: Θ (enthalpy)

$$\frac{\partial}{\partial t}\Theta + (\boldsymbol{u} - \dot{\boldsymbol{x}})\cdot\nabla\Theta$$

$$-\nabla\cdot\left[\kappa_T(\Theta)\nabla T(\Theta) + \kappa_\omega\nabla\omega(\Theta) + \boldsymbol{q}_D(\Theta)\right] - \eta D\boldsymbol{u} : D\boldsymbol{u} = 0$$

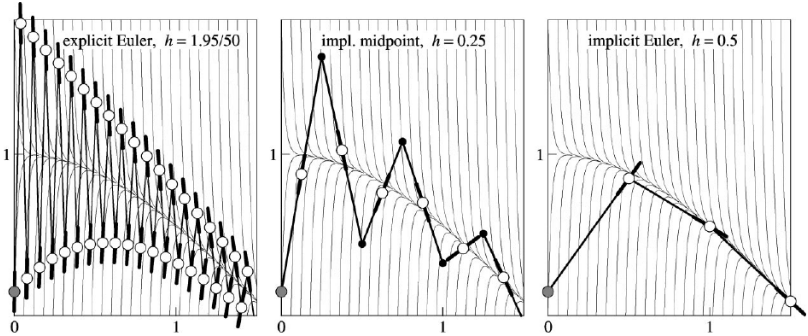
- ALE advection
- Thermal diffusion
- Moisture diffusion/Darcy flow
- Strain heating

Note: $\kappa(\Theta)$ and $\boldsymbol{q}_D(\Theta)$ are very sensitive near $\Theta = \Theta_0$

Summary of primal variables in DAE

u	velocity	algebraic
p	pressure	algebraic
x	mesh location	algebraic in domain, differential at surface
Θ	enthalpy	differential

Stiff integrators



$$\dot{x} + 50(x - \cos t) = 0$$

- $\dot{x} = \lambda x$
- $\mathcal{R}(h\lambda) = x^{n+1}/x^n$
- A-stable: $|\mathcal{R}(\{\Re[z] \leq 0\})| \leq 1$
- L-stable: $\lim_{z\to\infty} \mathcal{R}(z) = 0$

---

[1] Hairer and Wanner, 1999

Barriers

Dahlquist's second barrier

An A-stable linear multistep method has order $p \leq 2$.

Diagonally implicit Runge-Kutta

A DIRK evaluates the first stage to order $q = 1$.

Circumvent with general linear methods

$$\begin{bmatrix} Y \\ X^{n+1} \end{bmatrix} = \begin{bmatrix} A & U \\ B & V \end{bmatrix} \begin{bmatrix} h\dot{Y} \\ X^n \end{bmatrix}$$

- stage values $Y = \{y_1, \ldots, y_s\}$
- Nordsieck vector passed between steps

$$X = \{x_1, \ldots, x_r\} = \{x, h\dot{x}, h^2\ddot{x}, \ldots\}$$

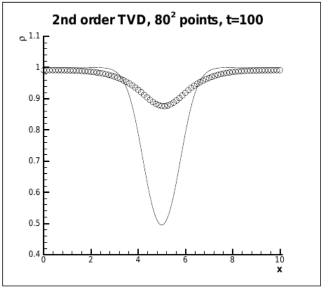
Special class: IRKS (inherent Runge-Kutta stability)[2]

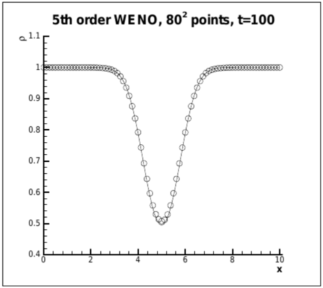
- A-stable
- L-stable
- order p, stage order q, $p = q = r - 1 = s - 1$
- diagonally implicit
- Asymptotically correct error estimates for present method *and* method of order $p + 1$.
- Implemented in PETSc's TSGL:
  - implicit DAE form: $f(t, x, \dot{x}) = 0$
  - orders $p = 1, \ldots, 5$
  - adaptive-order, adaptive-step controller
  - plugin architecture for controllers
  - make new methods available to the controller by giving their tableau, error estimates computed automatically
  - solve $f(t, x, x_0 + \alpha x) = 0$ with SNES

---

[2] Butcher, Jackiewicz, Wright 2007, *On error propagation in general linear methods for ordinary differential equations*

Why do we want high order methods?

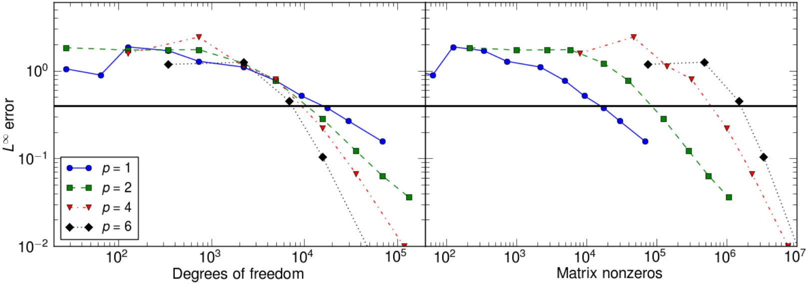




- high accuracy with reasonable computational effort
- qualitatively correct answers over long time scales
- stronger conservation statements
- better performance at high aspect ratio
  - Quintic velocities permit a space that strongly enforces $\nabla \cdot \boldsymbol{u} = 0$ with inf-sup constant independent of aspect ratio.

[3] Shu 2001

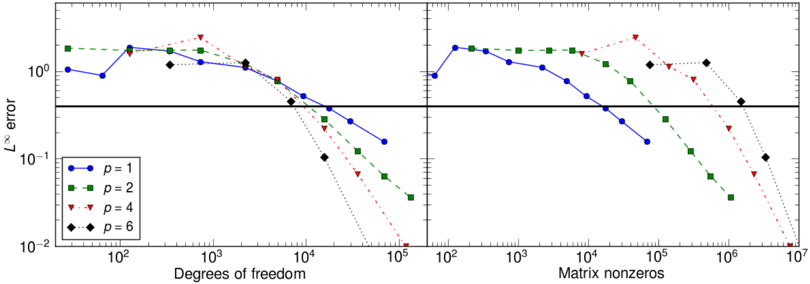
High order methods are expensive



Order p:

- element matrices have p^6 nonzeros
- cost $\mathcal{O}(p^7) - \mathcal{O}(p^9)$ to assemble
- **Very** expensive to precondition and solve with

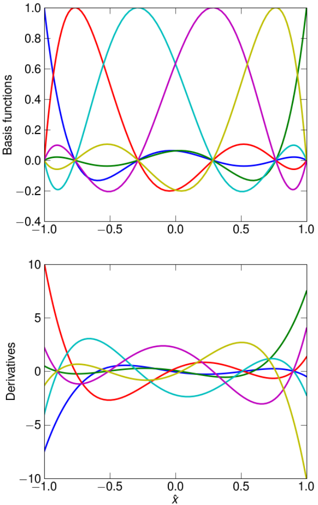
High order methods are expensive



Order p:

- element matrices have p^6 nonzeros
- cost $\mathcal{O}(p^7) - \mathcal{O}(p^9)$ to assemble
- **Very** expensive to precondition and solve with
- High order methods must be competitive **per degree of freedom** in order to be practical

Nodal hp-version finite element methods



1D reference element

- Lagrange interpolants on Legendre-Gauss-Lobatto points
- Quadrature $\hat{R}$, weights $\hat{W}$
- Evaluation: $\hat{B}, \hat{D}$

3D reference element

$$\hat{\boldsymbol{W}} = \hat{W} \otimes \hat{W} \otimes \hat{W}$$

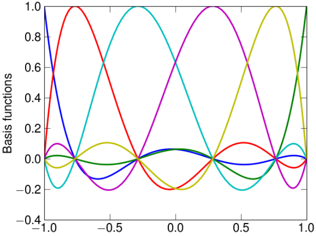
$$\hat{\boldsymbol{B}} = \hat{B} \otimes \hat{B} \otimes \hat{B}$$

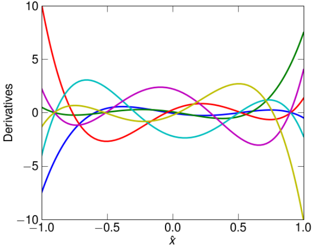
$$\hat{\boldsymbol{D}}_0 = \hat{D} \otimes \hat{B} \otimes \hat{B}$$

$$\hat{\boldsymbol{D}}_1 = \hat{B} \otimes \hat{D} \otimes \hat{B}$$

$$\hat{\boldsymbol{D}}_2 = \hat{B} \otimes \hat{B} \otimes \hat{D}$$

Nodal hp-version finite element methods





1D reference element

- Lagrange interpolants on Legendre-Gauss-Lobatto points
- Quadrature $\hat{R}$, weights $\hat{W}$
- Evaluation: $\hat{B}, \hat{D}$

3D reference element

$$\hat{\boldsymbol{W}} = \hat{W} \otimes \hat{W} \otimes \hat{W}$$

$$\hat{\boldsymbol{B}} = \hat{B} \otimes \hat{B} \otimes \hat{B}$$

$$\hat{\boldsymbol{D}}_0 = \hat{D} \otimes \hat{B} \otimes \hat{B}$$

$$\hat{\boldsymbol{D}}_1 = \hat{B} \otimes \hat{D} \otimes \hat{B}$$

$$\hat{\boldsymbol{D}}_2 = \hat{B} \otimes \hat{B} \otimes \hat{D}$$

These tensor product operations are very efficient, 10–20+ times faster than sparse mat-vec

Operations on physical elements

Mapping to physical space

$$x^e : \hat{K} \to K^e, \quad J^e_{ij} = \partial x^e_i / \partial \hat{x}_j, \quad (J^e)^{-1} = \partial \hat{x} / \partial x^e$$

Element operations in physical space

$$\boldsymbol{B}^e = \hat{\boldsymbol{B}} \qquad \boldsymbol{W}^e = \hat{\boldsymbol{W}} \Lambda(|J^e(\boldsymbol{r})|)$$

$$\boldsymbol{D}^e_i = \Lambda\left(\frac{\partial \hat{x}_0}{\partial x_i}\right) \hat{\boldsymbol{D}}_0 + \Lambda\left(\frac{\partial \hat{x}_1}{\partial x_i}\right) \hat{\boldsymbol{D}}_1 + \Lambda\left(\frac{\partial \hat{x}_2}{\partial x_i}\right) \hat{\boldsymbol{D}}_2$$

$$(\boldsymbol{D}^e_i)^T = \hat{\boldsymbol{D}}_0^T \Lambda\left(\frac{\partial \hat{x}_0}{\partial x_i}\right) + \hat{\boldsymbol{D}}_1^T \Lambda\left(\frac{\partial \hat{x}_1}{\partial x_i}\right) + \hat{\boldsymbol{D}}_2^T \Lambda\left(\frac{\partial \hat{x}_2}{\partial x_i}\right)$$

Global problem is defined by assembly

$$\mathcal{E} = [\mathcal{E}^e]$$

where $\mathcal{E}^e$ maps global dofs to element dofs

Residuals

- Continuous weak form: find $u \in \boldsymbol{\mathcal{V}}_D$ such that

$$\int_\Omega v \cdot f_0(u, \nabla u) + \nabla v : f_1(u, \nabla u) = 0 \qquad \forall v \in \boldsymbol{\mathcal{V}}_0$$

- Fully discrete form

$$\sum_e \mathcal{E}_e^T \Big[(\boldsymbol{B}^e)^T \boldsymbol{W}^e \Lambda(f_0(u^e, \nabla u^e)) + \sum_{i=0}^{2} (\boldsymbol{D}_i^e)^T \boldsymbol{W}^e \Lambda(f_1(u^e, \nabla u^e)) \Big] = \boldsymbol{0}$$

with $u^e = \boldsymbol{B}^e \mathcal{E}^e u$ and $\nabla u^e = \{\boldsymbol{D}_i^e \mathcal{E}^e u\}_{i=0}^2$.

1. Get element dofs with $\mathcal{E}^e$, evaluate $u, \nabla u$ at quadrature points
2. Apply the pointwise operations f_0, f_1
3. Weight the residuals with $\boldsymbol{W}^e$
4. Contribute weighted residuals via $\mathcal{E}_e^T (\boldsymbol{B}^e)^T$ and $\mathcal{E}_e^T (\boldsymbol{D}^e)^T$

Jacobians

- Continuous weak form: find $u \in \boldsymbol{\mathcal{V}}_D$ such that

$$v^T F(u) \sim \int_\Omega v \cdot f_0(u, \nabla u) + \nabla v : f_1(u, \nabla u) = 0 \qquad \forall v \in \boldsymbol{\mathcal{V}}_0$$

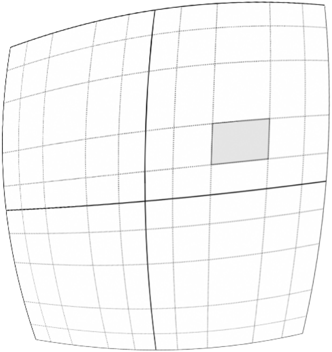
- Weak form of the Jacobian

$$v^T J(w) u \sim \int_\Omega \begin{bmatrix} v^T & \nabla v^T \end{bmatrix} \begin{bmatrix} f_{0,0} & f_{0,1} \\ f_{1,0} & f_{1,1} \end{bmatrix} \begin{bmatrix} u \\ \nabla u \end{bmatrix}$$

$$[f_{i,j}] = \begin{bmatrix} \frac{\partial f_0}{\partial u} & \frac{\partial f_0}{\partial \nabla u} \\ \frac{\partial f_1}{\partial u} & \frac{\partial f_1}{\partial \nabla u} \end{bmatrix} (w, \nabla w)$$

- Frequently much of $[f_{i,j}]$ is computed while evaluating f_i.
  - Inexpensive *taping* for full-accuracy matrix-free Jacobian
  - Code reuse in preconditioner assembly
- The terms in $[f_{i,j}]$ are easy to compute with symbolic math. Possible to automatically generate code.

“Dual order”



- any system of equations
- robust on non-affine elements
- robust to variable coefficients
- leverages existing software
- requires very little coding
- weak (bounded) p-dependence

Changing the inner product

Consider the problem $Ax = b$ discretized with high-order elements. The consistent formulation of this problem with low-order elements is $\hat{A}x = \hat{M}M^{-1}b$.

(Orszag 1980, Deville & Mund 1990, Kim 2007)

What code do you need to write?

Conventional FEM

- Residuals $v^T F(u)$

```
for each quadrature point:
  sum basis functions:  u, ∇u
  evaluate f_0, f_1
  weight residuals against v, ∇v
```

- Assembly $J(w)$

```
for each quadrature point:
  sum basis functions:  w, ∇w
  evaluate [f_{i,j}]
  for each test function:
    for each trial function:
      sum into K^e[test, trial]
insert K^e into global matrix
```

Dual-order hp-FEM

- Residuals $v^T F(u)$

```
evaluate u, ∇u at quad pts
evaluate f_0, f_1, tape for [f_{i,j}]
weight residuals and transpose
```

- Matrix-free $v^T J(w) u$

```
evaluate u, ∇u at quad points
restore [f_{i,j}](w) from tape
weight residuals
```

$$\begin{bmatrix} v \\ \nabla v \end{bmatrix}^T \begin{bmatrix} f_{0,0} & f_{0,1} \\ f_{1,0} & f_{1,1} \end{bmatrix} \begin{bmatrix} u \\ \nabla u \end{bmatrix}$$

```
transpose
```

- Assemble one or more matrices for preconditioning

Multiphysics

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$$

- Relaxation

$$\begin{bmatrix} A & \\ C & D \end{bmatrix}$$

  - Gauss-Seidel inspired, easy to implement
  - works when fields are loosely coupled
- Factorization

$$\begin{bmatrix} 1 & \\ CA^{-1} & 1 \end{bmatrix} \begin{bmatrix} A & B \\ & S \end{bmatrix}, \qquad S = D - CA^{-1}B$$

  - robust (exact factorization), can often drop lower block
  - how to precondition S which is usually dense?
    - interpret as differential operators, use approximate commutators

Power-law Stokes

- Strong form: Find $(\boldsymbol{u}, p) \in \boldsymbol{\mathcal{V}}_D \times \mathcal{P}$ such that

$$\begin{aligned} -\nabla \cdot (\eta D\boldsymbol{u}) + \nabla p - \boldsymbol{f} &= 0 \\ \nabla \cdot \boldsymbol{u} &= 0 \end{aligned}$$

where

$$\begin{aligned} D\boldsymbol{u} &= \tfrac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right) \\ \gamma(D\boldsymbol{u}) &= \tfrac{1}{2} D\boldsymbol{u} : D\boldsymbol{u} \\ \eta(\gamma) &= B(\Theta, \dots)\left(\epsilon + \gamma\right)^{\frac{\mathfrak{p}-2}{2}}, \quad \mathfrak{p} = 1 + \tfrac{1}{\mathfrak{n}} \approx \tfrac{4}{3} \end{aligned}$$

Power-law Stokes

Weak form of the Newton step

Find $(\boldsymbol{u}, p)$ such that

$$\int_\Omega D\boldsymbol{v} : \left[\eta \mathbf{1} + \eta' D\boldsymbol{w} \otimes D\boldsymbol{w}\right] : D\boldsymbol{u} \\ - p\nabla \cdot \boldsymbol{v} - q\nabla \cdot \boldsymbol{u} = -v \cdot F(\boldsymbol{w}) \qquad \forall (\boldsymbol{v}, q)$$

Matrix form

$$\begin{bmatrix} A(\boldsymbol{w}) & B^T \\ B & \end{bmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = -\begin{pmatrix} F_u(\boldsymbol{w}) \\ 0 \end{pmatrix}$$

Block factorization

$$\begin{bmatrix} A & B^T \\ B & \end{bmatrix} = \begin{bmatrix} 1 & \\ BA^{-1} & 1 \end{bmatrix} \begin{bmatrix} A & B^T \\ & S \end{bmatrix} = \begin{bmatrix} A & \\ B & S \end{bmatrix} \begin{bmatrix} 1 & A^{-1}B^T \\ & 1 \end{bmatrix}$$

where the Schur complement is

$$S = -BA^{-1}B^T.$$

Approximate commutators[4]

- Scaled mass matrix (Burstedde et al, Grinevich & Olshanskii)

$$-S \sim \operatorname{div}(\operatorname{div}\eta D)^{-1}\nabla \approx \operatorname{div}\nabla(\operatorname{div}\eta\nabla)^{-1}$$
$$\approx \operatorname{div}\nabla(\operatorname{div}\nabla)^{-1}\eta^{-1} \approx \eta^{-1}$$

- Least squares commutator (Elman et al, May & Moresi)

$$-S \sim (\operatorname{div}\nabla)\Big[\operatorname{div}(\operatorname{div}\eta D)\nabla\Big]^{-1}(\operatorname{div}\nabla)$$

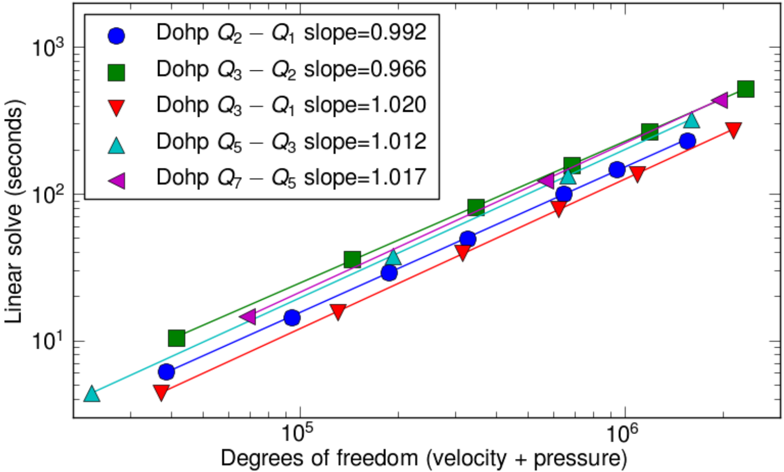
- Shallow water, $\alpha \propto 1/\Delta t$, only retaining the dominant terms

$$S \sim \alpha - \operatorname{div}(\alpha^{-1} g h \nabla)$$

This is a parabolic operator so this case is easier (no need for approximate commutators).

[4] Elman & Tuminaro 2009, *Boundary conditions in approximate commutator preconditioners for the Navier-Stokes equations*.

Power-law Stokes Scaling



Only assembles Q_1 matrices, ML for elliptic pieces

ALE form

After semidiscretization in time ($\alpha \propto 1/\Delta t$) we have a Jacobian

$$\begin{bmatrix} A_{II} & A_{I\Gamma} & & & & \\ & \alpha M_{\Gamma\Gamma} & & -N_{\Gamma\Gamma} & & \\ G_{II} & G_{\Gamma I} & B_{II} & B_{I\Gamma} & C_I^T & D_I \\ G_{I\Gamma} & G_{\Gamma\Gamma} & B_{\Gamma I} & B_{\Gamma\Gamma} & C_\Gamma^T & D_\Gamma \\ G_{Ip} & G_{\Gamma p} & C_I & C_\Gamma & & \\ \alpha E_I & \alpha E_\Gamma & F_I & F_\Gamma & & \alpha M_\Theta + J \end{bmatrix}$$

- pseudo-elasticity for mesh motion
- $(\dot{\boldsymbol{x}} - \boldsymbol{u}) \cdot \boldsymbol{n} =$ accumulution
- "just" geometry
- Stokes problem
- temperature dependence of rheology
- convective terms and strain heating in heat transport
- heat diffusion

Outlook

- High order spatial and temporal accuracy is desirable and feasible for the grounding line dynamics problem.
- High order elements are effective at utilizing modern hardware.
- Coupling between multiple domains is hard.
- Smooth manufactured solutions are not enough to study solver performance.
- Need good software to combine relaxation for loosely coupled processes and factorization for stiff/indefinite coupling.

Tools

- PETSc `http://mcs.anl.gov/petsc`
  - ML, Hypre, MUMPS
- ITAPS `http://itaps.org`
  - MOAB, CGM, Lasso