Scalable Implicit Methods for Free Surface Flows in Glaciology

Scalable Ice-sheet Solvers and Infrastructure for Petascale, High-resolution, Unstructured Simulations (SISIPHUS)
DOE ASCR ISICLES

Jed Brown[1,2], Iulian Grindeanu[1], Dmitry Karpeev[1], Barry F. Smith[1], and Timothy J. Tautges[1]

[1]Argonne National Laboratory
[2]ETH Zürich

2011-07-20

Outline

Objectives

Some opinionated commentary

Conservative steady-state energy transport

Code verification

Coupling software

High order with unassembled representations

Antarctic Ocean-Ice Interaction

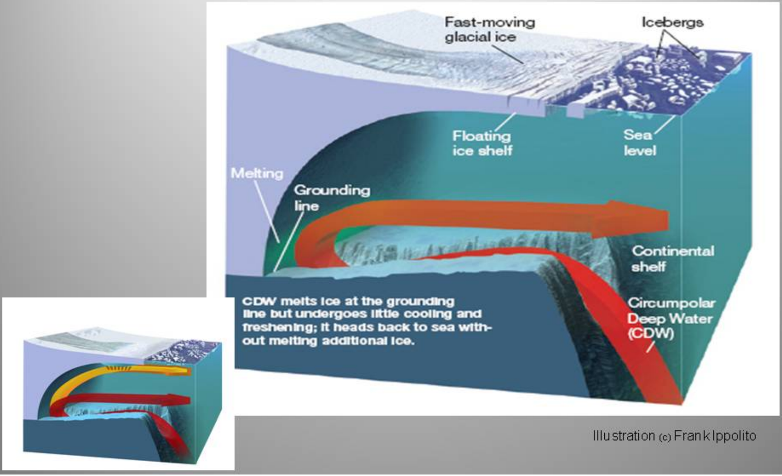


Illustration (c) Frank Ippolito

Bindschadler 2008

Glaciology and society

- Quantitative analysis of ice dynamics in changing climate
- Inversion to determine current state
- Stability and sensitivity, mostly at grounding line
- Predict sea level rise as a function of sea surface temperature [energy policy]

Our efforts

- Unstructured meshing, geometric models of boundaries
- Fully implicit formulations to enable analysis
- Implicit solver components
  - saddle points, anisotropy, heterogeneity, transport
  - composability and multi-physics coupling
- High order accuracy and high throughput
- Adjoints using restricted AD

Outline

Bathymetry and stickyness distribution

- Bathymetry:
  - Aspect ratio $\varepsilon = [H]/[x] \ll 1$
  - Need surface *and* bed slopes to be small
- Stickyness distribution:
  - Limiting cases of plug flow versus vertical shear
  - Stress ratio: $\lambda = [\tau_{xz}]/[\tau_{\text{membrane}}]$
  - Discontinuous: frozen to slippery transition at ice stream margins
- Standard approach in glaciology:
  Taylor expand in ε and sometimes λ, drop higher order terms.

$\lambda \gg 1$ Shallow Ice Approximation (SIA), no horizontal coupling

$\lambda \ll 1$ Shallow Shelf Approximation (SSA), 2D elliptic solve in map-plane

  - Hydrostatic and various hybrids, 2D or 3D elliptic solves

- Bed slope is discontinuous and of order 1.
  - Taylor expansions no longer valid
  - Numerics require high resolution, subgrid parametrization, short time steps
  - Still get low quality results in the regions of most interest.

Bathymetry and stickyness distribution

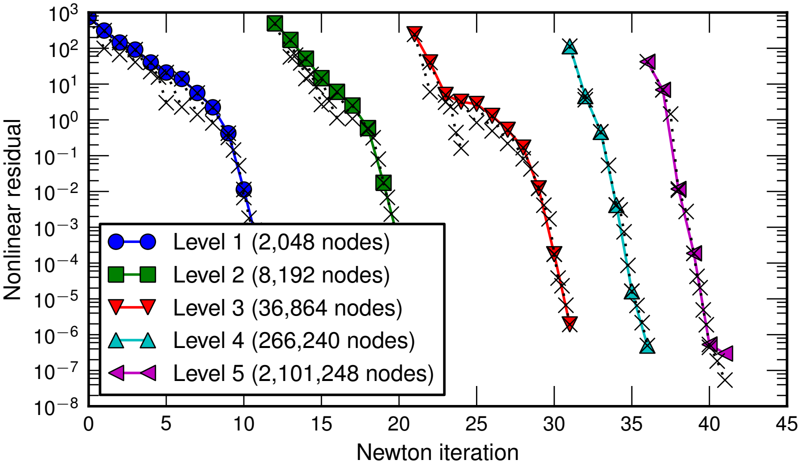
- Bathymetry:
  - Aspect ratio $\varepsilon = [H]/[x] \ll 1$
  - Need surface *and* bed slopes to be small
- Stickyness distribution:
  - Limiting cases of plug flow versus vertical shear
  - Stress ratio: $\lambda = [\tau_{xz}]/[\tau_{\text{membrane}}]$
  - Discontinuous: frozen to slippery transition at ice stream margins
- Standard approach in glaciology:
  Taylor expand in ε and sometimes λ, drop higher order terms.

$\lambda \gg 1$ Shallow Ice Approximation (SIA), no horizontal coupling

$\lambda \ll 1$ Shallow Shelf Approximation (SSA), 2D elliptic solve in map-plane

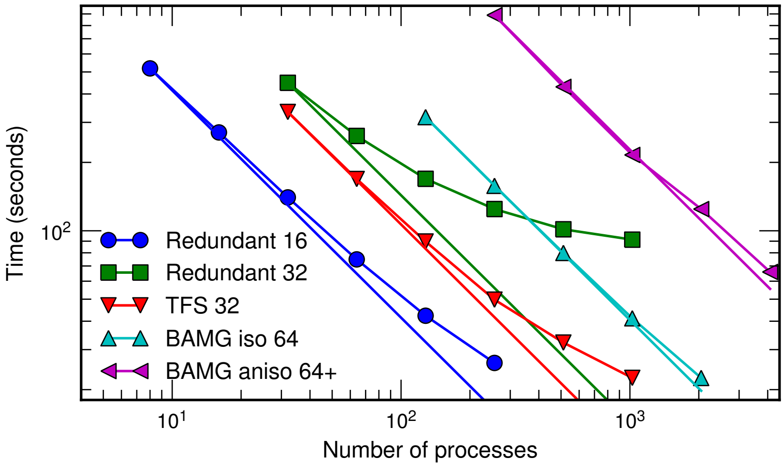
  - Hydrostatic and various hybrids, 2D or 3D elliptic solves
- Bed slope is discontinuous and of order 1.
  - Taylor expansions no longer valid
  - Numerics require high resolution, subgrid parametrization, short time steps
  - Still get low quality results in the regions of most interest.

Textbook multigrid efficiency



Grid-sequenced Newton-Krylov solution of test X. The solid lines denote nonlinear iterations, and the dotted lines with $\times$ denote linear residuals.

Strong scaling on Shaheen



Strong scaling on Shaheen for different size coarse levels problems and different coarse level solvers. The straight lines on the strong scaling plot have slope -1 which is optimal.

Outline

Objectives

Some opinionated commentary

Conservative steady-state energy transport

Code verification

Coupling software

High order with unassembled representations

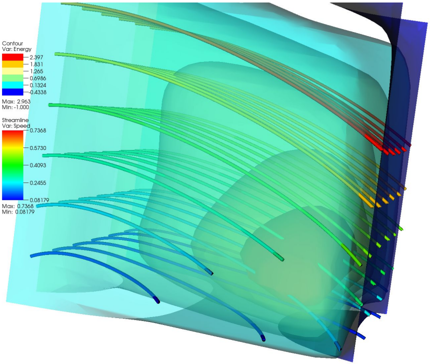
Conservative two-phase formulation

Find momentum density ρu, pressure p, and total energy density E:

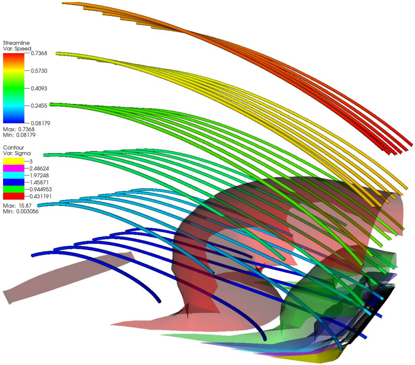
$$(\rho u)_t + \nabla\cdot(\rho u \otimes u - \eta D u_i + p1) - \rho g = 0$$
$$\rho_t + \nabla\cdot \rho u = 0$$
$$E_t + \nabla\cdot\big((E+p)u - k_T \nabla T - k_\omega \nabla \omega\big) - \eta D u_i : D u_i - \rho u \cdot g = 0$$

- Solve for density ρ, ice velocity u_i, temperature T, and melt fraction ω using constitutive relations.
  - Simplified constitutive relations can be solved explicitly.
  - Temperature, moisture, and strain-rate dependent rheology η.
  - High order FEM, typically Q_3 momentum & energy, SUPG (yuck).
- DAEs solved implicitly after semidiscretizing in space.
- Newton solver converges quadratically.
- Thermocoupled steady state in one nonlinear solve
  - no time stepping needed, total cost similar to 3 semi-implicit steps
  - useful for inverse problems and stability analysis
- (Somewhat) robust preconditioning using nested field-split

Block on inclined plate, nominal $\mathrm{Re} = 0.24$, $\mathrm{Pe} = 120$



Contours of Energy, melt fraction up to 15%, density ratio 2.



Contours of viscous heat production, $1/r$ singularity at corners.

Relative effect of the blocks

$$J = \begin{pmatrix} J_{uu} & J_{up} & J_{uE} \\ J_{pu} & 0 & 0 \\ J_{Eu} & J_{Ep} & J_{EE} \end{pmatrix}.$$

J_{uu} Viscous/momentum terms, nearly symmetric, variable coefficionts, anisotropy from Newton.

J_{up} Weak pressure gradient, viscosity dependence on pressure (small), gravitational contribution (pressure-induced density variation). Large, nearly balanced by gravitational forcing.

J_{uE} Viscous dependence on energy, very nonlinear, not very large.

J_{pu} Divergence (mass conservation), nearly equal to J_{up}^T.

J_{Eu} Sensitivity of energy on momentum, mostly advective transport. Large in boundary layers with large thermal/moisture gradients.

J_{Ep} Thermal/moisture diffusion due to pressure-melting, $u \cdot \nabla$.

J_{EE} Advection-diffusion for energy, very nonlinear at small regularization. Advection-dominated except in boundary layers and stagnant ice, often balanced in vertical.

How much nesting?

$$P_1 = \begin{pmatrix} J_{uu} & J_{up} & J_{uE} \\ 0 & B_{pp} & 0 \\ 0 & 0 & J_{EE} \end{pmatrix}$$

- B_{pp} is a mass matrix in the pressure space weighted by inverse of kinematic viscosity.
- Elman, Mihajlović, Wathen, JCP 2011 for non-dimensional isoviscouse Boussinesq.
- Works well for non-dimensional problems on the cube, not for realistic parameters.

$$P = \begin{bmatrix} \begin{pmatrix} J_{uu} & J_{up} \\ J_{pu} & 0 \end{pmatrix} & \\ \begin{pmatrix} J_{Eu} & J_{Ep} \end{pmatrix} & J_{EE} \end{bmatrix}$$

- Inexact inner solve using upper-triangular with B_{pp} for Schur.
- Another level of nesting.
- GCR tolerant of inexact inner solves.
- Outer converges in 1 or 2 iterations.

- Low-order preconditioning full-accuracy unassembled high order operator.
- Build these on command line with PETSc `PCFieldSplit`.

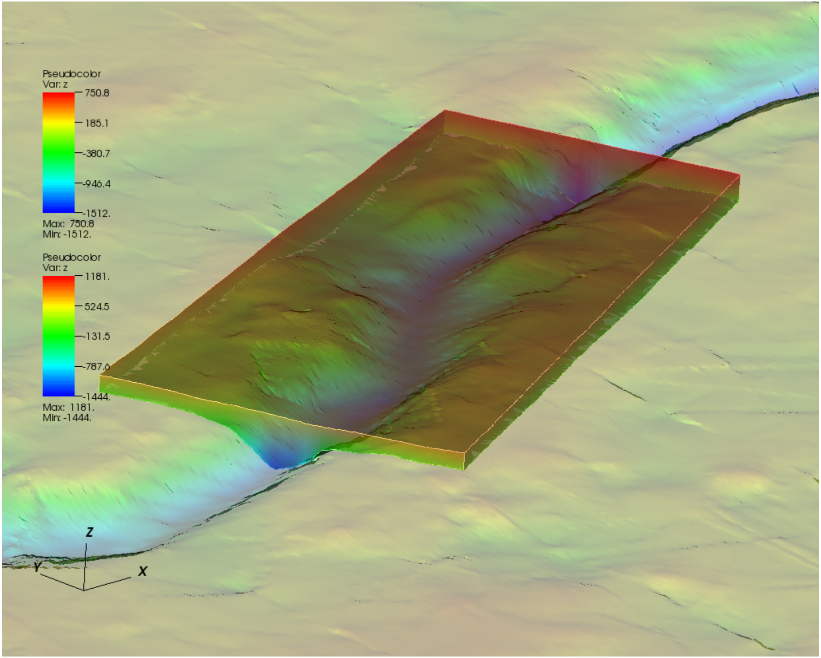
Everything is better as a smoother

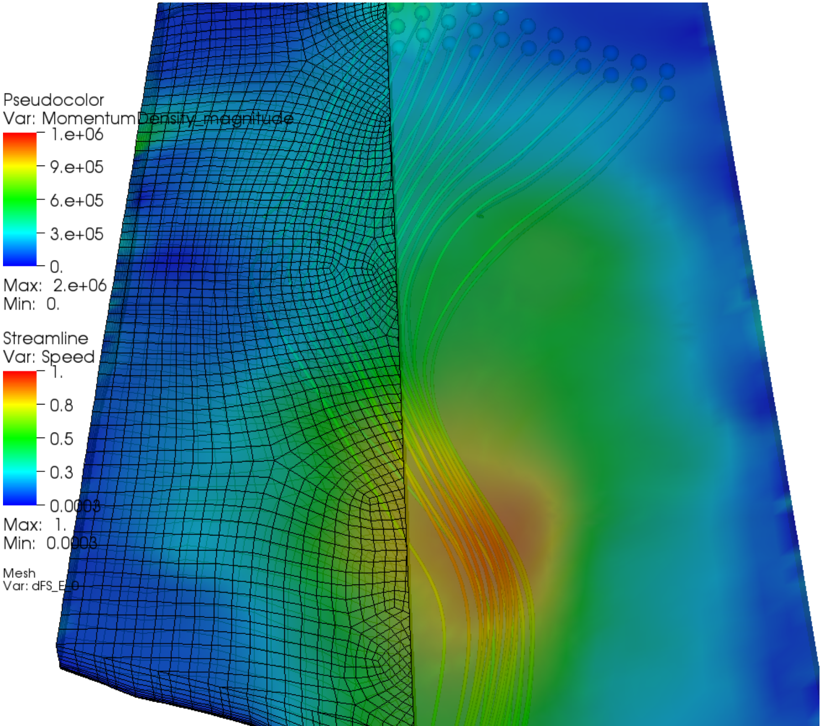
Block preconditioners work alright, but…

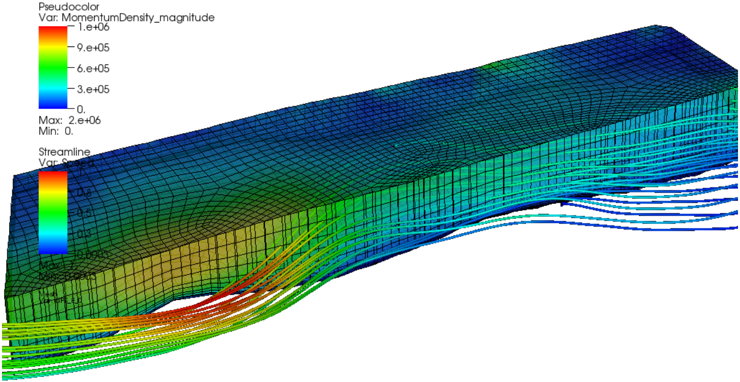
- nested iteration requires more dot products
- more iterations: coarse levels don't "see" each other
- finer grained kernels: lower arithmetic intensity, even more limited by memory bandwidth

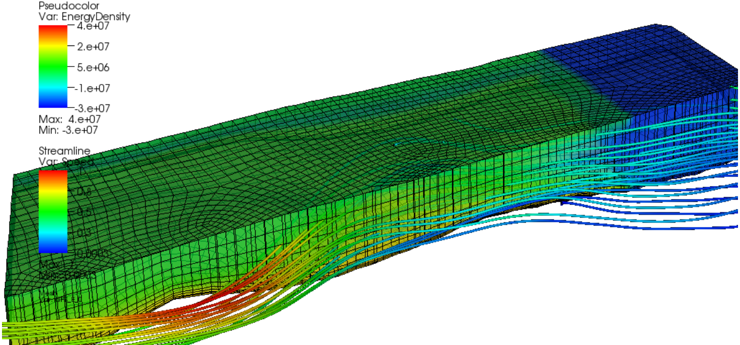
Coupled multigrid

- need compatible coarsening
  - can do algebraically (Adams 2004) but would need to assemble
- stability issues for lowest order $Q_1 - P_0^{\text{disc}}$
  - Rannacher-Turek looks great, but no discrete Korn's inequality
- coupled "Vanka" smoothers difficult to implement with high performance, especially for FEM
- block preconditioners as smoothers reuse software better
- one level by reducing order for the coarse space, more levels need non-nested geometric MG or go all-algebraic and pay for matrix assembly and setup


Pseudocolor
Var: z
750.8
185.1
-380.7
-946.4
-1512.
Max: 750.8
Min: -1512.
Pseudocolor
Var: z
1181.
524.5
-131.5
-787.6
-1444.
Max: 1181.
Min: -1444.
Z
Y
X


Pseudocolor
Var: MomentumDensity_magnitude
1.e+06
9.e+05
6.e+05
3.e+05
0.
Max: 2.e+06
Min: 0.
Streamline
Var: Speed
1.
0.8
0.5
0.3
0.0003
Max: 1.
Min: 0.0003
Mesh
Var: dFS_E_0


Pseudocolor
Var: MomentumDensity_magnitude
1.e+06
9.e+05
6.e+05
3.e+05
0.
Max: 2.e+06
Min: 0.
Streamline


Pseudocolor
Var: EnergyDensity
4.e+07
2.e+07
5.e+06
-1.e+07
-3.e+07
Max: 4.e+07
Min: -3.e+07
Streamline

Outline

Symbolic form of large-deformation elasticity

Find displacement vector u such that:

$$\int_\Omega \nabla v : \Pi = 0, \quad \forall v$$

where

$F = I - \nabla u$	Deformation gradient
$E = (F^T F - I)/2$	Green-Lagrange tensor
$S = \lambda(\text{tr} E) I + 2\mu E$	Second Piola-Kirchoff tensor
$\Pi = F \cdot S$	First Piola-Kirchoff tensor

```
def weak_form(u, du, v, dv):
  I = eye(3)                        # Identity tensor
  F = I - du                        # Deformation gradient
  E = (F.T*F - I)/2                 # Green-Lagrange tensor
  S = lmbda*E.trace()*I + 2*mu*E    # Second Piola-Kirchoff tensor
  Pi = F * S                        # First Piola-Kirchoff tensor
  return dv.dot(Pi)
```

Symbolic form of large-deformation elasticity

Find displacement vector u such that:

$$\int_\Omega \nabla v : \Pi = 0, \quad \forall v$$

where

$F = I - \nabla u$	Deformation gradient
$E = (F^T F - I)/2$	Green-Lagrange tensor
$S = \lambda(\mathrm{tr} E)I + 2\mu E$	Second Piola-Kirchoff tensor
$S = \lambda(J^2 - I)C^{-1} + \mu(I - C^{-1})$	Neo-Hookean material, $(C = F^T F)$
$\Pi = F \cdot S$	First Piola-Kirchoff tensor

```
def weak_form(u, du, v, dv):
  I = eye(3)                      # Identity tensor
  F = I - du                      # Deformation gradient
  E = (F.T*F - I)/2               # Green-Lagrange tensor
  S = lmbda*E.trace()*I + 2*mu*E  # Second Piola-Kirchoff tensor
  Pi = F * S                      # First Piola-Kirchoff tensor
  return dv.dot(Pi)
```

Manufactured solution

- Choose a solution u_{exact} with rich derivatives

```
def solution(x,y,z, a,b,c):
  return Matrix([cos(x) * exp(y) * z + sin(z),
                 sin(x) * tanh(y) + x * cosh(z),
                 exp(x) * sinh(y) + y * log(1+z**2)])
```

- Apply strong-form nonlinear differential operator symbolically to define

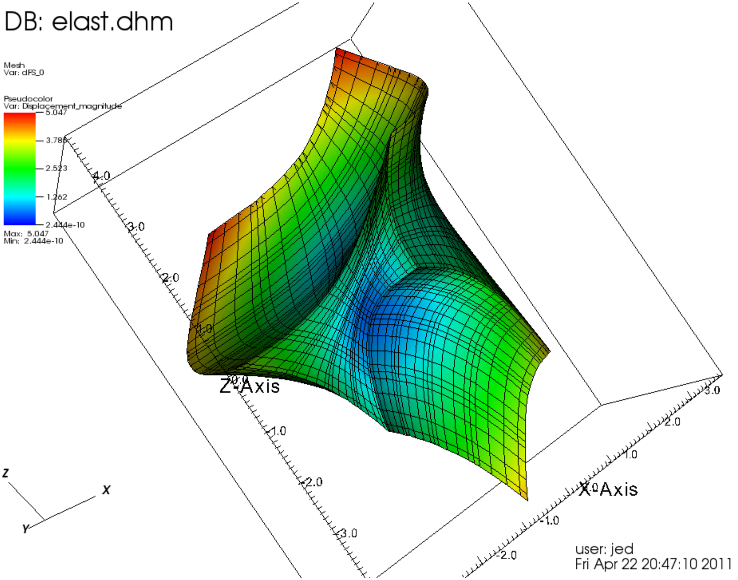
$$f(x,y,z) = \nabla \cdot \Pi(\nabla u_{\text{exact}})$$

- Solve finite element problem for u_h

$$\int_\Omega \nabla v : \Pi(\nabla u_h) = \int v \cdot f(x,y,z), \quad \forall v$$

- Compute norms of $u_h - u_{\text{exact}}$.

Manufactured solution



Convergence rates

Mesh		# Nodes	$\|u_h - u\|_2$ Error	$\mathcal{O}$	$\|u_h - u\|_\infty$ Error	$\mathcal{O}$	$\|\nabla u_h - \nabla u\|_2$ Error	$\mathcal{O}$	$\|\nabla u_h - \nabla u\|_\infty$ Error	$\mathcal{O}$
Q_1	1^3	8	1.79e+00	—	6.50e-01	—	3.70e+00	—	1.08e+00	—
Q_1	2^3	27	5.49e-01	1.71	3.40e-01	0.93	1.61e+00	1.20	6.92e-01	0.64
Q_1	4^3	125	1.53e-01	1.84	1.26e-01	1.43	8.01e-01	1.01	4.51e-01	0.62
Q_1	8^3	729	3.94e-02	1.96	3.73e-02	1.76	3.98e-01	1.01	2.81e-01	0.68
Q_1	16^3	4913	9.95e-03	1.99	1.01e-02	1.88	1.98e-01	1.01	1.57e-01	0.84
Q_1	32^3	35937	2.49e-03	2.00	2.61e-03	1.95	9.92e-02	1.00	8.32e-02	0.92
Q_3	1^3	64	4.14e-02	—	2.71e-02	—	2.90e-01	—	1.63e-01	—
Q_3	2^3	343	2.06e-03	4.33	2.06e-03	3.72	2.39e-02	3.60	1.14e-02	3.84
Q_3	4^3	2197	1.81e-04	3.51	2.06e-04	3.32	4.23e-03	2.50	2.88e-03	1.98
Q_3	8^3	15625	1.22e-05	3.89	1.87e-05	3.46	5.79e-04	2.87	5.84e-04	2.30
Q_5	1^3	216	3.76e-03	—	2.90e-03	—	4.69e-02	—	3.16e-02	—
Q_5	2^3	1331	7.58e-05	5.63	5.92e-05	5.61	1.62e-03	4.86	1.05e-03	4.91
Q_5	4^3	9261	7.33e-07	6.69	6.61e-07	6.48	2.59e-05	5.97	1.76e-05	5.90
Q_9	1^3	1000	5.81e-05	—	5.04e-05	—	1.42e-03	—	1.05e-03	—
Q_9	2^3	6859	6.27e-08	9.86	7.59e-08	9.38	1.63e-06	9.77	1.60e-06	9.36

Outline

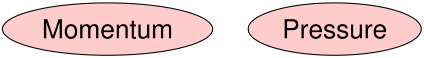
ALE form

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

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

- mesh motion equations (Laplace-Beltrami or pseudo-elasticity)
- $(\dot{x} - u) \cdot n =$ accumulution
- “just” geometry
- Stokes problem
- temperature dependence of rheology
- convective terms and strain heating in heat transport
- thermal advection-diffusion

Multi-physics coupling in PETSc

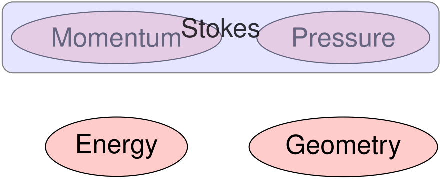


- package each "physics" independently
- solve single-physics and coupled problems
- semi-implicit and fully implicit
- reuse residual and Jacobian evaluation unmodified
- direct solvers, fieldsplit inside multigrid, multigrid inside fieldsplit without recompilation
- use the best possible matrix format for each physics (e.g. symmetric block size 3)
- matrix-free anywhere
- multiple levels of nesting

Multi-physics coupling in PETSc

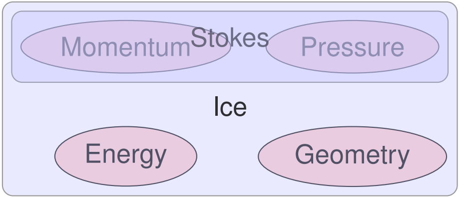
- package each "physics" independently
- solve single-physics and coupled problems
- semi-implicit and fully implicit
- reuse residual and Jacobian evaluation unmodified
- direct solvers, fieldsplit inside multigrid, multigrid inside fieldsplit without recompilation
- use the best possible matrix format for each physics (e.g. symmetric block size 3)
- matrix-free anywhere
- multiple levels of nesting

Multi-physics coupling in PETSc



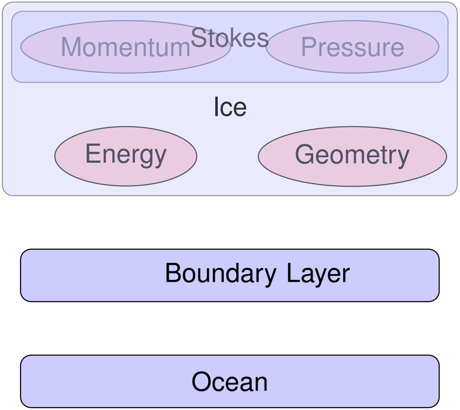
- package each “physics” independently
- solve single-physics and coupled problems
- semi-implicit and fully implicit
- reuse residual and Jacobian evaluation unmodified
- direct solvers, fieldsplit inside multigrid, multigrid inside fieldsplit without recompilation
- use the best possible matrix format for each physics (e.g. symmetric block size 3)
- matrix-free anywhere
- multiple levels of nesting

Multi-physics coupling in PETSc



- package each "physics" independently
- solve single-physics and coupled problems
- semi-implicit and fully implicit
- reuse residual and Jacobian evaluation unmodified
- direct solvers, fieldsplit inside multigrid, multigrid inside fieldsplit without recompilation
- use the best possible matrix format for each physics (e.g. symmetric block size 3)
- matrix-free anywhere
- multiple levels of nesting

Multi-physics coupling in PETSc

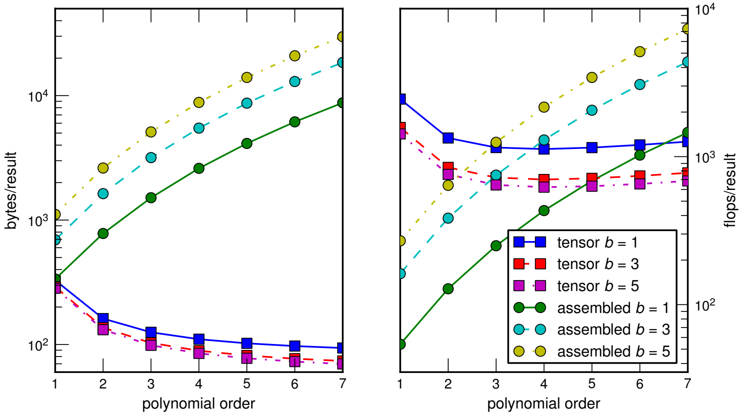


- package each "physics" independently
- solve single-physics and coupled problems
- semi-implicit and fully implicit
- reuse residual and Jacobian evaluation unmodified
- direct solvers, fieldsplit inside multigrid, multigrid inside fieldsplit without recompilation
- use the best possible matrix format for each physics (e.g. symmetric block size 3)
- matrix-free anywhere
- multiple levels of nesting

Outline

- Objectives
- Some opinionated commentary
- Conservative steady-state energy transport
- Code verification
- Coupling software
- High order with unassembled representations

Performance of assembled versus unassembled



- Same linear operator, smaller to not store unassembled
- Use local symbolic math or AD, runtime choice of order, precondition with low-order method
- Dual order h and p FEM: `github.com/jedbrown/dohp`
- PETSc: `mcs.anl.gov/petsc`

Automation

- For unassembled representations, decomposition, and assembly
- Continuous weak form: find u

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

- Weak form of the Jacobian $J(w)$: find u

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

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

- Terms in $[f_{i,j}]$ easy to compute symbolically, AD more scalable.
- Nonlinear terms f_0, f_1 usually have the most expensive nonlinearities in the computation of scalars
  - Equations of state, effective viscosity
  - Compute gradient with reverse-mode, store at quadrature points.
  - Perturb scalars, then use forward-mode to complete the Jacobian.
  - Flip forward/reverse for action of the adjoint.

Outlook

- Stabilized continuous FEM is terrible, need to implement DG for transport.
- Issues defining conservative normals for slip: all-positive (e.g. spline) basis?
- Geometric coupling to surface causes delicate stiffness at large time steps.
- Visualization should have hooks for solving equations of state.
- Need good hierarchical solver diagnostic tools.
- Solution transfer after remeshing.
- Have active set and semi-smooth Newton for contact, but there are many types of contact and much work to be done.
- *Strongly Coupled Solvers with Loosely Coupled Software* CP76 11:00 tomorrow, Room 304.