Scalable and Composable Implicit Solvers for Polythermal Ice Flow with Steep Topography

Jed Brown[1],
Matt Knepley[2], Dave May[3], Barry Smith[1]

[1]Mathematics and Computer Science Division, Argonne National Laboratory
[2]Computation Institute, University of Chicago
[3]ETH Zürich

SCA 2012-04-03

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  - Taylor expansions no longer valid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.
- Basal sliding parameters are discontinuous.

Hydrostatic equations for ice sheet flow

- Valid when $w_x \ll u_z$, independent of basal friction (Schoof&Hindmarsh 2010)
- Eliminate p and w from Stokes by incompressibility:
  3D elliptic system for $u = (u, v)$

$$-\nabla \cdot \left[\eta \begin{pmatrix} 4u_x + 2v_y & u_y + v_x & u_z \\ u_y + v_x & 2u_x + 4v_y & v_z \end{pmatrix} \right] + \rho g \overline{\nabla} h = 0$$

$$\eta(\theta, \gamma) = \frac{B(\theta)}{2} (\gamma_0 + \gamma)^{\frac{1-\mathfrak{n}}{2\mathfrak{n}}}, \qquad \mathfrak{n} \approx 3$$

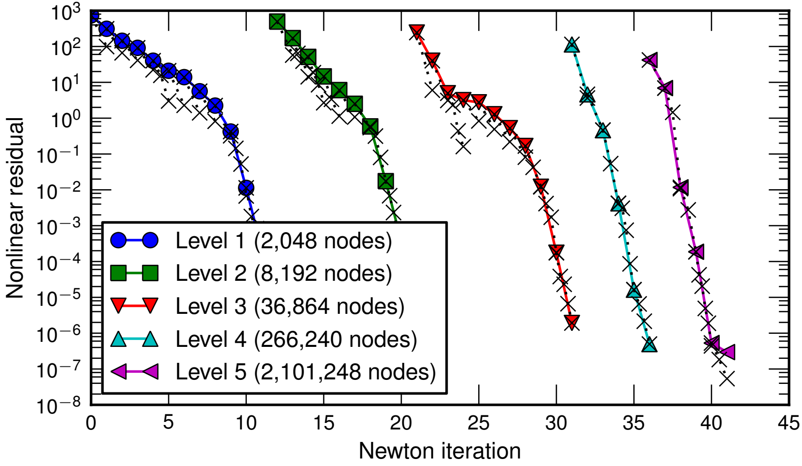
$$\gamma = u_x^2 + v_y^2 + u_x v_y + \frac{1}{4}(u_y + v_x)^2 + \frac{1}{4}u_z^2 + \frac{1}{4}v_z^2$$

and slip boundary $\sigma \cdot n = \beta^2 u$ where

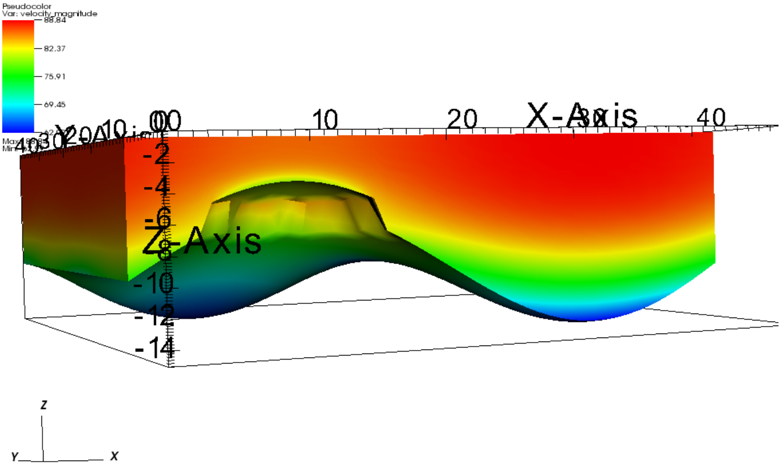
$$\beta^2(\gamma_b) = \beta_0^2 (\epsilon_b^2 + \gamma_b)^{\frac{\mathfrak{m}-1}{2}}, \qquad 0 < \mathfrak{m} \leq 1$$

$$\gamma_b = \frac{1}{2}(u^2 + v^2)$$

- Q_1 FEM with Newton-Krylov-Multigrid solver in PETSc: `src/snes/examples/tutorials/ex48.c`



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



- Bathymetry is essentially discontinuous on any grid
- Shallow ice approximation produces oscillatory solutions
- Nonlinear and linear solvers have major problems or fail
- Grid sequenced Newton-Krylov multigrid works
  as well as in the smooth case

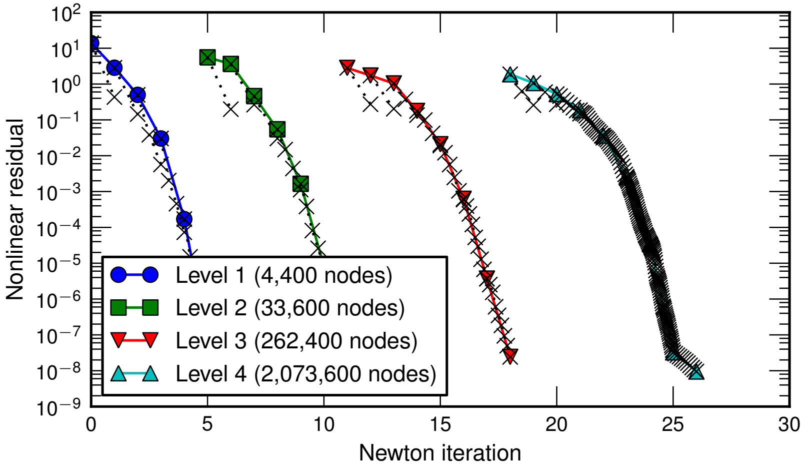
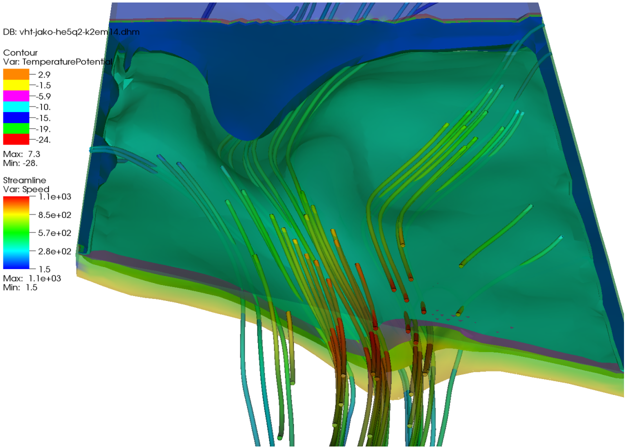


Figure: Grid sequenced Newton-Krylov convergence for test Y. The "cliff" has 58° angle in the red line (12×125 meter elements), 73° for the cyan line (6×62 meter elements).


DB: vht-jako-he5q2-k2em14.dhm
Contour
Var: TemperaturePotential
2.9
-1.5
-5.9
-10.
-15.
-19.
-24.
Max: 7.3
Min: -28.
Streamline
Var: Speed
1.1e+03
8.5e+02
5.7e+02
2.8e+02
1.5
Max: 1.1e+03
Min: 1.5

Polythermal ice

- Interface tracking methods (Greve's SICOPOLIS)
  - Different fields for temperate and cold ice.
  - Lagrangian or Eulerian, problems with changing topology
  - No discrete conservation
- Interface capturing
  - Enthalpy: Aschwanden, Bueler, Khroulev, Blatter (J. Glac. 2012)
    - Not in conservation form
    - Only conservative for infinitesimal melt fraction
  - Energy
    - Conserves mass, momentum, and energy for arbitrary melt fraction
    - Implicit equation of state

Conservative (non-Boussinesq) two-phase ice flow

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
- DAEs solved implicitly after semidiscretizing in space.
- Preconditioning using nested fieldsplit

The Great Solver Schism: Monolithic or Split?

Monolithic

- Direct solvers
- Coupled Schwarz
- Coupled Neumann-Neumann (need unassembled matrices)
- Coupled multigrid

X Need to understand local spectral and compatibility properties of the coupled system

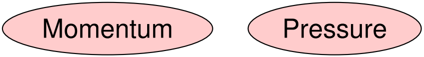
Split

- Physics-split Schwarz (based on relaxation)
- Physics-split Schur (based on factorization)
  - approximate commutators SIMPLE, PCD, LSC
  - segregated smoothers
  - Augmented Lagrangian
  - “parabolization” for stiff waves

X Need to understand global coupling strengths

- Preferred data structures depend on which method is used.
- Interplay with geometric multigrid.

Multi-physics coupling in PETSc

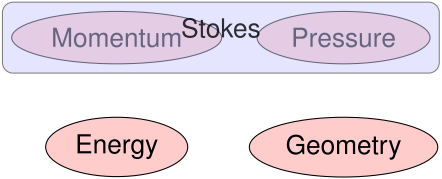


- package each "physics" independently
- solve single-physics and coupled problems
- semi-implicit and fully implicit
- reuse residual and Jacobian evaluation unmodified
- direct solvers, fieldsplit inside multigrid, multigrid inside fieldsplit without recompilation
- use the best possible matrix format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

Multi-physics coupling in PETSc

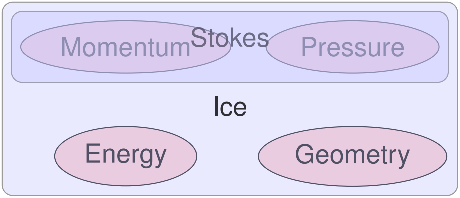
- package each "physics" independently
- solve single-physics and coupled problems
- semi-implicit and fully implicit
- reuse residual and Jacobian evaluation unmodified
- direct solvers, fieldsplit inside multigrid, multigrid inside fieldsplit without recompilation
- use the best possible matrix format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

Multi-physics coupling in PETSc



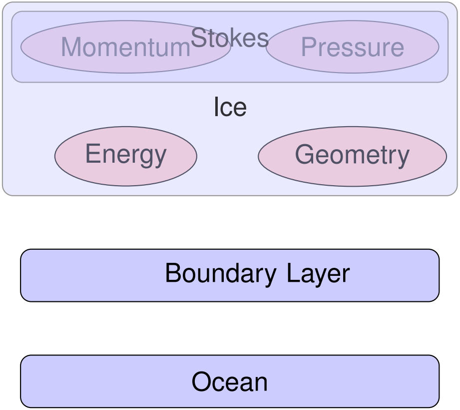
- package each "physics" independently
- solve single-physics and coupled problems
- semi-implicit and fully implicit
- reuse residual and Jacobian evaluation unmodified
- direct solvers, fieldsplit inside multigrid, multigrid inside fieldsplit without recompilation
- use the best possible matrix format 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

Splitting for Multiphysics

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

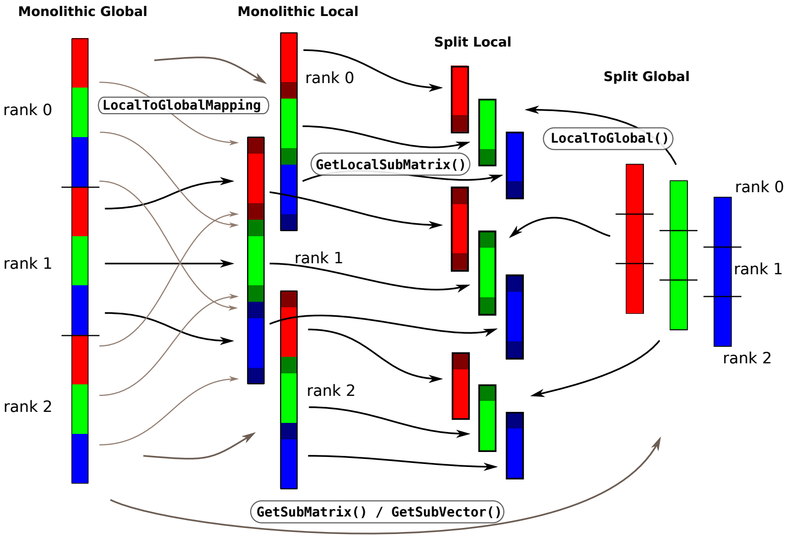
- Relaxation: `-pc_fieldsplit_type [additive,multiplicative,symmetric_multiplicative]`

$$\begin{bmatrix} A & \\ & D \end{bmatrix}^{-1} \qquad \begin{bmatrix} A & \\ C & D \end{bmatrix}^{-1} \qquad \begin{bmatrix} A & \\ & 1 \end{bmatrix}^{-1} \left(1 - \begin{bmatrix} A & B \\ & 1 \end{bmatrix} \begin{bmatrix} A & \\ C & D \end{bmatrix}^{-1} \right)$$

  - Gauss-Seidel inspired, works when fields are loosely coupled
- Factorization: `-pc_fieldsplit_type schur`

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

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



Work in Split Local space, matrix data structures reside in any space.

Multiphysics Assembly Code: Jacobians

```
FormJacobian_Coupled(SNES snes,Vec X,Mat J,Mat B,...) {
  // Access components as for residuals
  MatGetLocalSubMatrix(B,is[0],is[0],&Buu);
  MatGetLocalSubMatrix(B,is[0],is[1],&Buk);
  MatGetLocalSubMatrix(B,is[1],is[0],&Bku);
  MatGetLocalSubMatrix(B,is[1],is[1],&Bkk);
  FormJacobianLocal_U(user,&infou,u,k,Buu);          // single physics
  FormJacobianLocal_UK(user,&infou,&infok,u,k,Buk); // coupling
  FormJacobianLocal_KU(user,&infou,&infok,u,k,Bku); // coupling
  FormJacobianLocal_K(user,&infok,u,k,Bkk);          // single physics
  MatRestoreLocalSubMatrix(B,is[0],is[0],&Buu);
  // More restores
```

- Assembly code is independent of matrix format
- Single-physics code is used unmodified for coupled problem
- No-copy fieldsplit:

  `-pack_dm_mat_type nest -pc_type fieldsplit`
- Coupled direct solve:

  `-pack_dm_mat_type aij -pc_type lu -pc_factor_mat_solver_package mumps`

Stokes Example

The common block preconditioners for Stokes require only options:

```
-pc_type fieldsplit
-pc_field_split_type

-fieldsplit_0_pc_type ml
-fieldsplit_0_ksp_type preonly
```

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix}$$

Stokes Example

The common block preconditioners for Stokes require only options:

```
-pc_type fieldsplit
-pc_field_split_type additive
-fieldsplit_0_pc_type ml
-fieldsplit_0_ksp_type preonly
-fieldsplit_1_pc_type jacobi
-fieldsplit_1_ksp_type preonly
```

$$\begin{pmatrix} \hat{A} & 0 \\ 0 & I \end{pmatrix}$$

Cohouet and Chabard, *Some fast 3D finite element solvers for the generalized Stokes problem*, 1988.

Stokes Example

The common block preconditioners for Stokes require only options:

```
-pc_type fieldsplit
-pc_field_split_type
multiplicative

-fieldsplit_0_pc_type ml
-fieldsplit_0_ksp_type preonly

-fieldsplit_1_pc_type jacobi
-fieldsplit_1_ksp_type preonly
```

$$\begin{pmatrix} \hat{A} & B \\ 0 & I \end{pmatrix}$$

Elman, *Multigrid and Krylov subspace methods for the discrete Stokes equations*, 1994.

Stokes Example

The common block preconditioners for Stokes require only options:

```
-pc_type fieldsplit
-pc_field_split_type schur

-fieldsplit_0_pc_type ml
-fieldsplit_0_ksp_type preonly

-fieldsplit_1_pc_type none
-fieldsplit_1_ksp_type minres
 -pc_fieldsplit_schur_factorization_type diag
```

$$\begin{pmatrix} \hat{A} & 0 \\ 0 & -\hat{S} \end{pmatrix}$$

May and Moresi, *Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics*, 2007.

Olshanskii, Peters, and Reusken *Uniform preconditioners for a parameter dependent saddle point problem with application to generalized Stokes interface equations*, 2006.

Stokes Example

The common block preconditioners for Stokes require only options:

```
-pc_type fieldsplit
-pc_field_split_type schur
```

```
-fieldsplit_0_pc_type ml
-fieldsplit_0_ksp_type preonly
```

```
-fieldsplit_1_pc_type none
-fieldsplit_1_ksp_type minres
```

$$\begin{pmatrix} \hat{A} & 0 \\ B^T & \hat{S} \end{pmatrix}$$

```
-pc_fieldsplit_schur_factorization_type lower
```

May and Moresi, *Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics*, 2007.

Stokes Example

The common block preconditioners for Stokes require only options:

```
-pc_type fieldsplit
-pc_field_split_type schur

-fieldsplit_0_pc_type ml
-fieldsplit_0_ksp_type preonly

-fieldsplit_1_pc_type none
-fieldsplit_1_ksp_type minres
-pc_fieldsplit_schur_factorization_type upper
```

$$\begin{pmatrix} \hat{A} & B \\ 0 & \hat{S} \end{pmatrix}$$

May and Moresi, *Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics*, 2007.

Stokes Example

The common block preconditioners for Stokes require only options:

```
-pc_type fieldsplit
-pc_field_split_type schur

-fieldsplit_0_pc_type ml
-fieldsplit_0_ksp_type preonly

-fieldsplit_1_pc_type lsc
-fieldsplit_1_ksp_type minres
 -pc_fieldsplit_schur_factorization_type full
```

$$\begin{pmatrix} \hat{A} & B \\ 0 & \hat{S}_{\text{LSC}} \end{pmatrix}$$

May and Moresi, *Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics*, 2007.

Elman, Howle, Shadid, Shuttleworth, and Tuminaro, *Block preconditioners based on approximate commutators*, 2006.

Stokes Example

The common block preconditioners for Stokes require only options:

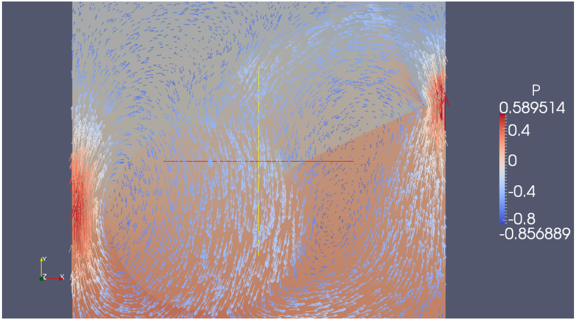
```
-pc_type fieldsplit
-pc_field_split_type schur
 -pc_fieldsplit_schur_factorization_type
```

$$\begin{pmatrix} I & 0 \\ B^T A^{-1} & I \end{pmatrix} \begin{pmatrix} \hat{A} & 0 \\ 0 & \hat{S} \end{pmatrix} \begin{pmatrix} I & A^{-1}B \\ 0 & I \end{pmatrix}$$

Coupled MG for Stokes, split smoothers

$$J = \begin{pmatrix} A & B^T \\ B & C \end{pmatrix}$$

$$P_{\text{smooth}} = \begin{pmatrix} A_{\text{SOR}} & 0 \\ B & M \end{pmatrix}$$




```
-pc_type mg -pc_mg_levels 5 -pc_mg_galerkin
-mg_levels_pc_type fieldsplit
-mg_levels_pc_fieldsplit_block_size 3
-mg_levels_pc_fieldsplit_0_fields 0,1
-mg_levels_pc_fieldsplit_1_fields 2
-mg_levels_fieldsplit_0_pc_type sor
```

Nonlinear solvers in PETSc SNES

LS, TR	Newton-type with line search and trust region
NRichardson	Nonlinear Richardson, usually preconditioned
VIRS, VIRSAUG, and VISS	reduced space and semi-smooth methods for variational inequalities
QN	Quasi-Newton methods like BFGS
NGMRES	Nonlinear GMRES
NCG	Nonlinear Conjugate Gradients
SORQN	Multiplicative Schwarz quasi-Newton
GS	Nonlinear Gauss-Seidel/multiplicative Schwarz sweeps
FAS	Full approximation scheme (nonlinear multigrid)
MS	Multi-stage smoothers, often used with FAS for hyperbolic problems
Shell	Your method, often used as a (nonlinear) preconditioner

Quasi-Newton revisited: ameliorating setup costs

- Newton-Krylov with analytic Jacobian

Lag	FunctionEval	JacobianEval	PCSetUp	PCApply
1 bt	12	8	8	31
1 cp	31	6	6	24
2 bt		— diverged —		
2 cp	41	4	4	35
3 cp	50	4	4	44

- Jacobian-free Newton-Krylov with lagged preconditioner

Lag	FunctionEval	JacobianEval	PCSetUp	PCApply
1 bt	23	11	11	31
2 bt	48	4	4	36
3 bt	64	3	3	52
4 bt	87	3	3	75

- Limited-memory Quasi-Newton/BFGS with lagged solve for H_0

Restart	H_0	FunctionEval	JacobianEval	PCSetUp	PCApply
1 cp	10^{-5}	17	4	4	35
1 cp	preonly	21	5	5	10
3 cp	10^{-5}	21	3	3	43
3 cp	preonly	23	3	3	11
6 cp	10^{-5}	29	2	2	60
6 cp	preonly	29	2	2	14

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?

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

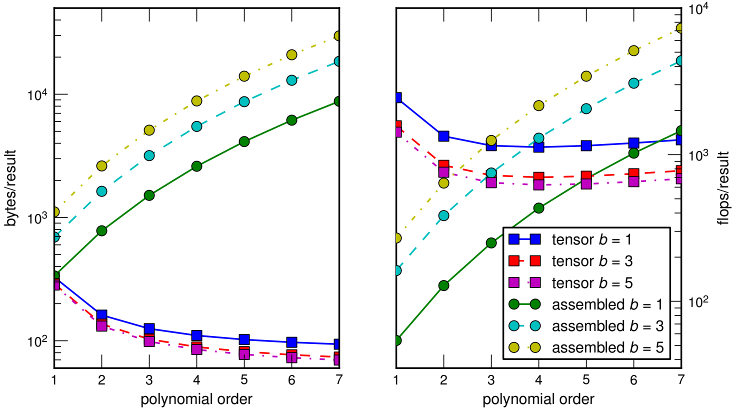
- B_{pp} is a mass matrix in the pressure space weighted by inverse of kinematic viscosity.
- Elman, Mihajlović, Wathen, JCP 2011 for non-dimensional isoviscous Boussinesq.
- Works well for non-dimensional problems on the cube, not for realistic parameters.

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

- 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`.

Performance of assembled versus unassembled



- High order Jacobian stored unassembled using coefficients at quadrature points, can use local AD
- Choose approximation order at run-time, independent for each field
- Precondition high order using assembled lowest order method
- Implementation $> 70\%$ of FPU peak, SpMV bandwidth wall $< 4\%$

Hardware Arithmetic Intensity

Operation	Arithmetic Intensity (flops/B)
Sparse matrix-vector product	1/6
Dense matrix-vector product	1/4
Unassembled matrix-vector product	≈ 8
High-order residual evaluation	> 5

Processor	BW (GB/s)	Peak (GF/s)	Balanced AI (F/B)
Sandy Bridge 6-core	21*	150	7.2
Magny Cours 16-core	42*	281	6.7
Blue Gene/Q node	43	205	4.8
Tesla M2050	144	515	3.6

One level of p-multigrid

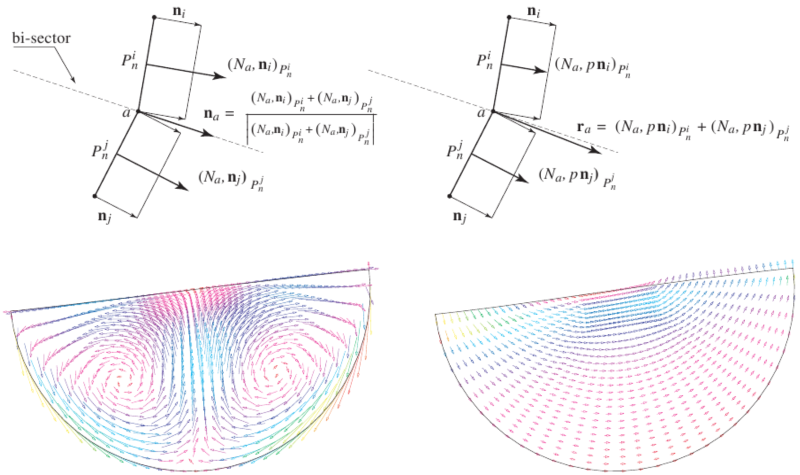
- Want to skip assembly on finest level (for better throughput)
- High order operators lack h-ellipticity
  - Necessary and sufficient condition for existence of pointwise smoother
- Use embedded low-order operator as smoother
  - Rescaled to recover a consistent inner product
  - Does not destroy symmetry for point-block Jacobi
- Polynomial smoothers
  - Target upper part of PBJacobi-preconditioned spectrum
  - Efficient GPU implementation
- Reordered incomplete factorization to couple “columns”
- Operator-dependent interpolation is more delicate
- Strict semi-coarsening requires semi-structured grid

Construction of conservative nodal normals

$$n^i = \int_\Gamma \phi^i n$$

- Exact conservation even with rough surfaces
- Definition is robust in 2D and for first-order elements in 3D
- $\int_\Gamma \phi^i = 0$ for corner basis function of undeformed P_2 triangle
- May be negative for sufficiently deformed quadrilaterals
- Mesh motion should use normals from CAD model
  - Difference between CAD normal and conservative normal introduces correction term to conserve mass within the mesh
  - Anomolous velocities if disagreement is large (fast moving mesh, rough surface)
- Normal field not as smooth/accurate as desirable (and achievable with non-conservative normals)
  - Mostly problematic for surface tension
  - Walkley et al, *On calculation of normals in free-surface flow problems*, 2004

Need for well-balancing



(Behr, *On the application of slip boundary condition on curved surfaces*, 2004)

"No" boundary condition

- Integration by parts produces

$$\int_\Gamma v \cdot T\sigma \cdot n, \qquad \sigma = \eta Du - p1, \qquad T = 1 - n \otimes n$$

- Continuous weak form requires either
  - Dirichlet: $u|_\Gamma = f \implies v|_\Gamma = 0$
  - Neumann/Robin: $\sigma \cdot n|_\Gamma = g(u, p)$
- Discrete problem allows integration of $\sigma \cdot n$ "as is"
  - Extends validity of equations to include Γ
  - Not valid for continuum equations
  - Introduced by Papanastasiou, Malamataris, and Ellwood, 1992 for Navier-Stokes outflow boundaries
  - Griffiths, *The 'no boundary condition' outflow boundary condition*, 1997
    - Proves L^∞ order of accuracy $\mathcal{O}((h + 1/\mathrm{Pe})^{p+1})$ for Galerkin finite elements of order p (linear advection-diffusion)
    - Demonstrates equivalence with collocation at Radau points in outflow element
  - Used in slip boundary conditions by Behr 2004

Outlook

- Unintrusive composition of multigrid and block preconditioning
- We can build many preconditioners from the literature *on the command line*
- User code does not depend on matrix format, preconditioning method, nonlinear solution method, time integration method (implicit or IMEX), or size of coupled system (except for driver).
- Similar infrastructure extends to nonlinear methods
- Preliminary implementations on GPU

In development

- Distributive relaxation and Vanka smoothers
- Improved operator-dependent semi-geometric multigrid
- Automated support for mixing analysis/UQ into levels
- IMEX time stepping for geometry evolution
- Special basis functions for corners