Strongly Coupled Solvers with Loosely Coupled Software

Jed Brown[1,2], Dave May[2], and Barry Smith[1]

[1]Argonne National Laboratory
[2]ETH Zürich

2011-07-21

Example problems

- ALE free-surface flows
- Saddle-point problems (incompressibility, contact)
- Stiff waves
- Radiation hydrodynamics
- Multi-domain problems (e.g. fluid-structure interaction)
- Full space PDE-constrained optimization

Stiff wave prototype (low-Mach shallow water)

$$J = \begin{pmatrix} A & \nabla\cdot \\ h\nabla & D \end{pmatrix} \qquad S = A - (\nabla\cdot)D^{-1}h\nabla$$

- A and D are "nice", but proportional to $1/\Delta t$
- MG needs low-order upwinded smoothers for h-ellipticity
- S is well-behaved, nearly-parabolic, classical multigrid works
- Representative: "Schur complements make things better"

The Great Solver Schism: Monolithic or Split?

Monolithic

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

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

Split

- Physics-split Schwarz (based on relaxation)
- Physics-split Schur (based on factorization)
  - "block" preconditioners/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.

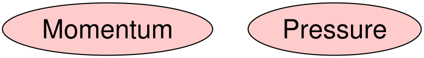
Outline

Coupling software

Applications

- Ice Flow
- Geodynamics

Multi-physics coupling in PETSc

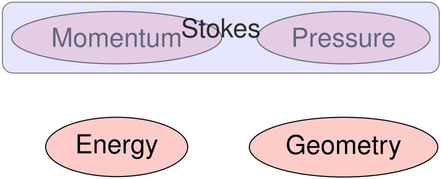


- package each “physics” independently
- solve single-physics and coupled problems
- semi-implicit and fully implicit
- reuse residual and Jacobian evaluation unmodified
- direct solvers, fieldsplit inside multigrid, multigrid inside fieldsplit without recompilation
- use the best possible matrix format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

Multi-physics coupling in PETSc

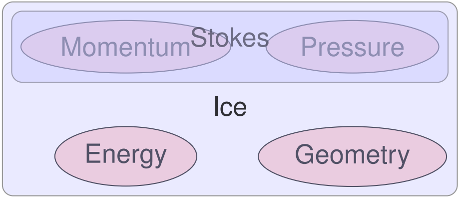
- package each "physics" independently
- solve single-physics and coupled problems
- semi-implicit and fully implicit
- reuse residual and Jacobian evaluation unmodified
- direct solvers, fieldsplit inside multigrid, multigrid inside fieldsplit without recompilation
- use the best possible matrix format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

Multi-physics coupling in PETSc



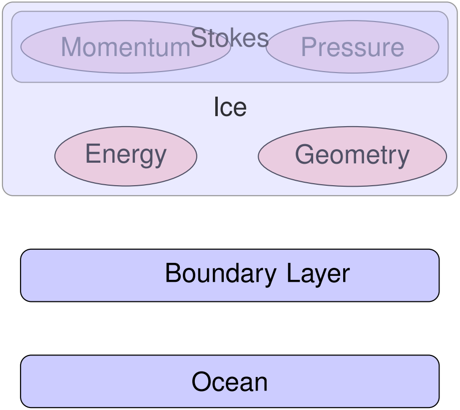
- package each “physics” independently
- solve single-physics and coupled problems
- semi-implicit and fully implicit
- reuse residual and Jacobian evaluation unmodified
- direct solvers, fieldsplit inside multigrid, multigrid inside fieldsplit without recompilation
- use the best possible matrix format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

MatGetLocalSubMatrix() spaces

- Newton method for $F(x) = 0$ solves

$$J(x)\delta x = -F(x)$$

$$J = \begin{pmatrix} J_{aa} & J_{ab} & J_{ac} \\ J_{ba} & J_{bb} & J_{bc} \\ J_{ca} & J_{cb} & J_{cc} \end{pmatrix}.$$

- Conceptually, there are three spaces in parallel
  - V Globally assembled space
  - V_i Global space for a single physics i
  - $\overline{V}_i$ Local space (with ghosts) for a single physcs i
  - $\overline{V}$ $\prod_i \overline{V}_i$ Concatenation of all single-physics local spaces
- Different components need different relationships

$V_i \to V$ field-split

$\overline{V} \to V$ coupled Neumann domain decomposition methods

$\overline{V}_i$ natural language for modular residual evaluation and assembly

MatGetLocalSubMatrix() spaces

Spaces

V Globally assembled space

V_i Global space for a single physics i

$\overline{V}_i$ Local space (with ghosts) for a single physcs i

$\overline{V}$ $\prod_i \overline{V}_i$ Concatenation of all single-physics local spaces

- Multiple physics $x = [x_a, x_b, x_c]$

I_i Map indices from V_i to V.

R_i Global physics restriction $R_i : V \to V_i$

$$R_i x = x[I_i] = x_i$$

$\bar{I}_i$ Map indices from $\overline{V}_i$ to V_i

$\overline{R}_i$ Extract local single-physics part from global single-physics

$$\overline{R}_i x_i = x_i[\bar{I}_i] = \bar{x}_i$$

$\tilde{I}_i$ Map indices from $\overline{V}_i$ to $\overline{V}$

`MatGetLocalSubMatrix()` spaces

- Globally assembled coupled matrix in terms of assembled single-physics blocks

$$J = \sum_{ij} R_i^T J_{ij} R_j$$

  - Language of Schwarz and fieldsplit
- Assembled single-physics blocks in terms of local single-physics matrices

$$J_{ij} = \overline{R}_i^T \overline{J}_{ij} \overline{R}_j$$

  - Language of assembly and Neumann/FETI domain decomposition
  - `MatSetValuesLocal()`

`MatGetLocalSubMatrix(Mat A,IS rows,IS cols,Mat *B);`

- Primarily for assembly
  - `B` is not guaranteed to implement `MatMult`
  - The communicator for `B` is not specified, only safe to use non-collective ops (unless you check)
- `IS` represents an index set, includes a block size and communicator
- `MatSetValuesBlockedLocal()` is implemented
- MatNest returns nested submatrix, no-copy
- No-copy for Neumann-Neumann formats (unassembled across procs, e.g. BDDC, FETI-DP)
- Most other matrices return a lightweight proxy `Mat`
  - `COMM_SELF`
  - Values not copied, does not implement `MatMult`
  - Translates indices to the language of the parent matrix
  - Multiple levels of nesting are flattened

Outline

Coupling software

Applications

- Ice Flow
- Geodynamics

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

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?

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

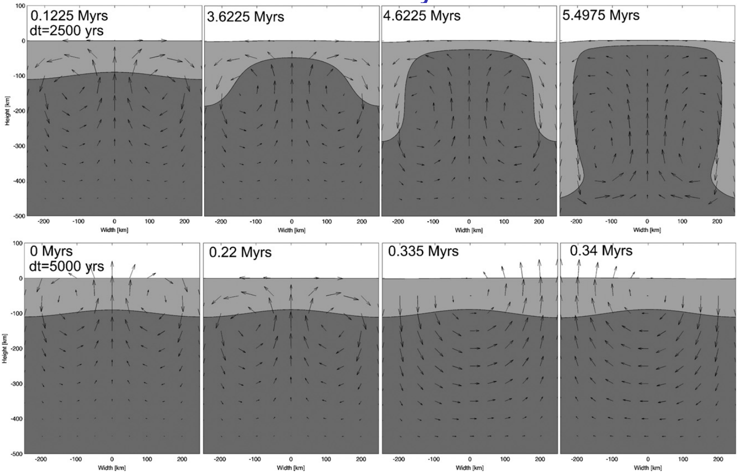
- B_{pp} is a mass matrix in the pressure space weighted by inverse of kinematic viscosity.
- Elman, Mihajlović, Wathen, JCP 2011 for non-dimensional isoviscouse Boussinesq.
- Works well for non-dimensional problems on the cube, not for realistic parameters.

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

- 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`.

The Drunken Seaman instability



- Subduction and mantle convection with a free surface.
- Free surface critical to long-term dynamics (e.g. mountain range formation)
- Advective 0.01 CFL for stability.
- Semi-implicit helps: Kaus, Mühlhaus, and May, 2010

Fully implicit free surface

- Velocity u, pressure p, Lagrangian/ALE coordinates x.

$$J = \begin{pmatrix} J_{uu} & J_{up} & J_{ux} \\ J_{pu} & 0 & J_{ux} \\ J_{xu} & 0 & J_{xx} \end{pmatrix}.$$

- Precondition with

$$P = \begin{bmatrix} \begin{pmatrix} J_{uu} & J_{up} \\ J_{pu} & 0 \end{pmatrix} & \\ \begin{pmatrix} J_{xu} & 0 \end{pmatrix} & J_{xx} \end{bmatrix} \qquad P_{\text{Stokes}} = \begin{pmatrix} J_{uu} & 0 \\ J_{pu} & B_{pp} \end{pmatrix}$$

- ```
  -dm_mat_type nest -pc_type fieldsplit
  -fieldsplit_stokes_pc_type fieldsplit
  -fieldsplit_stokes_pc_fieldsplit_type schur
  -fieldsplit_stokes_pc_fieldsplit_schur_factorization_type lower
  -fieldsplit_stokes_fieldsplit_u_pc_type mg
  ```
- `-dm_mat_type aij -pc_type lu -pc_factor_mat_solver_package mumps`
  ```

Outlook

- Block and symmetric formats, monolithic or nested
- Multigrid inside or outside field splits
- Can use IMEX methods $g(t,x,\dot{x}) = f(t,x)$
- Preallocation for off-diagonal blocks
- Nonlinear solvers for IMEX systems with structure
- General/nonsymmetric pivoting.