What are Composable Solvers?

Matt Knepley

Computer Science and Engineering
University at Buffalo

From Modeling to Learning with HPC
ICERM, Providence, RI
Sep 14, 2025

University at Buffalo
Center for Hybrid Rocket
Exascale Simulation Technology

Never believe anything

until you run it.

What are Composable Solvers?

Optimal solvers often

have many subsolves,

pieced together.

What are Composable Solvers?

PETSc is designed to easily:

- Replicate solvers from papers

What are Composable Solvers?

PETSc is designed to easily:

- Replicate solvers from papers
- Combine disparate solvers (precondition)

What are Composable Solvers?

PETSc is designed to easily:

- Replicate solvers from papers
- Combine disparate solvers (precondition)
  - Additively

What are Composable Solvers?

PETSc is designed to easily:

- Replicate solvers from papers
- Combine disparate solvers (precondition)
  - Additively
  - Multiplicatively

What are Composable Solvers?

PETSc is designed to easily:

- Replicate solvers from papers
- Combine disparate solvers (precondition)
  - Additively
  - Multiplicatively
  - Hierarchically

What are Composable Solvers?

PETSc is designed to easily:

- Replicate solvers from papers
- Combine disparate solvers (precondition)
    - Additively
    - Multiplicatively
    - Hierarchically
    - In subdomains

What are Composable Solvers?

PETSc is designed to easily:

- Replicate solvers from papers
- Combine disparate solvers (precondition)
- Solve in substeps

What are Composable Solvers?

PETSc is designed to easily:

- Replicate solvers from papers
- Combine disparate solvers (precondition)
- Solve in substeps
- Solve diagnostically

What are Composable Solvers?

PETSc is designed to easily:

- Replicate solvers from papers
- Combine disparate solvers (precondition)
- Solve in substeps
- Solve diagnostically
- Solve in post-process

What are Composable Solvers?

Linear

(Brown, M. Knepley, et al. 2012)

Nonlinear

(Brune, Matthew G. Knepley, B. F. Smith, and Tu 2015)

Solver People

Bill Gropp

Barry Smith

Jed Brown

Outline

Allen-Cahn Solver

Stokes Solvers

Patch Solvers

Preconditioners for Faults

Lessons

Programming with Options

ex55: Allen-Cahn problem in 2D

- constant mobility
- triangular elements

Geometric multigrid method for saddle point variational inequalities:

```
./ex55 -ksp_type fgmres -pc_type mg -mg_levels_ksp_type fgmres
-mg_levels_pc_type fieldsplit -mg_levels_pc_fieldsplit_detect_saddle_point
-mg_levels_pc_fieldsplit_type schur -da_grid_x 65 -da_grid_y 65
-mg_levels_pc_fieldsplit_schur_fact_type full
-mg_levels_pc_fieldsplit_schur_precondition selfp
-mg_levels_fieldsplit_1_ksp_type gmres -mg_coarse_ksp_type preonly
-mg_levels_fieldsplit_1_pc_type none   -mg_coarse_pc_type svd
-mg_levels_fieldsplit_0_ksp_type preonly
-mg_levels_fieldsplit_0_pc_type sor    -pc_mg_levels 5
-mg_levels_fieldsplit_0_pc_sor_forward -pc_mg_galerkin
-snes_vi_monitor -ksp_monitor_true_residual -snes_atol 1.e-11
-mg_levels_ksp_monitor -mg_levels_fieldsplit_ksp_monitor
-mg_levels_ksp_max_it 2 -mg_levels_fieldsplit_ksp_max_it 5
```

Programming with Options

ex55: Allen-Cahn problem in 2D

Programming with Options

ex55: Allen-Cahn problem in 2D

Run flexible GMRES with 5 levels of multigrid as the preconditioner

```
./ex55 -ksp_type fgmres -pc_type mg -pc_mg_levels 5
  -da_grid_x 65 -da_grid_y 65
```

Programming with Options

ex55: Allen-Cahn problem in 2D

Run flexible GMRES with 5 levels of multigrid as the preconditioner

```
./ex55 -ksp_type fgmres -pc_type mg -pc_mg_levels 5
  -da_grid_x 65 -da_grid_y 65
```

Use the Galerkin process to compute the coarse grid operators

```
-pc_mg_galerkin
```

Programming with Options

ex55: Allen-Cahn problem in 2D

Run flexible GMRES with 5 levels of multigrid as the preconditioner

```
./ex55 -ksp_type fgmres -pc_type mg -pc_mg_levels 5
  -da_grid_x 65 -da_grid_y 65
```

Use the Galerkin process to compute the coarse grid operators

```
-pc_mg_galerkin
```

Use SVD as the coarse grid saddle point solver

```
-mg_coarse_ksp_type preonly -mg_coarse_pc_type svd
```

Programming with Options

ex55: Allen-Cahn problem in 2D

Programming with Options

ex55: Allen-Cahn problem in 2D

Smoother: Flexible GMRES (2 iterates) with a Schur complement PC

```
-mg_levels_ksp_type fgmres -mg_levels_pc_fieldsplit_detect_saddle_point
-mg_levels_ksp_max_it 2 -mg_levels_pc_type fieldsplit
-mg_levels_pc_fieldsplit_type schur
-mg_levels_pc_fieldsplit_schur_fact_type full
-mg_levels_pc_fieldsplit_schur_precondition selfp
```

Programming with Options

ex55: Allen-Cahn problem in 2D

Smoother: Flexible GMRES (2 iterates) with a Schur complement PC

```
-mg_levels_ksp_type fgmres -mg_levels_pc_fieldsplit_detect_saddle_point
-mg_levels_ksp_max_it 2 -mg_levels_pc_type fieldsplit
-mg_levels_pc_fieldsplit_type schur
-mg_levels_pc_fieldsplit_schur_fact_type full
-mg_levels_pc_fieldsplit_schur_precondition selfp
```

Schur complement solver: GMRES (5 iterates) with no preconditioner

```
-mg_levels_fieldsplit_1_ksp_type gmres
-mg_levels_fieldsplit_1_pc_type none -mg_levels_fieldsplit_ksp_max_it 5
```

Programming with Options

ex55: Allen-Cahn problem in 2D

Smoother: Flexible GMRES (2 iterates) with a Schur complement PC

```
-mg_levels_ksp_type fgmres -mg_levels_pc_fieldsplit_detect_saddle_point
-mg_levels_ksp_max_it 2 -mg_levels_pc_type fieldsplit
-mg_levels_pc_fieldsplit_type schur
-mg_levels_pc_fieldsplit_schur_fact_type full
-mg_levels_pc_fieldsplit_schur_precondition selfp
```

Schur complement solver: GMRES (5 iterates) with no preconditioner

```
-mg_levels_fieldsplit_1_ksp_type gmres
-mg_levels_fieldsplit_1_pc_type none -mg_levels_fieldsplit_ksp_max_it 5
```

A_{00} inverse action: Use only the lower diagonal part of A_{00}

```
-mg_levels_fieldsplit_0_ksp_type preonly
-mg_levels_fieldsplit_0_pc_type sor
-mg_levels_fieldsplit_0_pc_sor_forward
```

Outline

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Block-Jacobi (Exact), Cohouet & Chabard, IJNMF, 1988.

```
-ksp_type gmres -pc_type fieldsplit -pc_fieldsplit_type additive
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type lu
 -fieldsplit_pressure_ksp_type preonly -fieldsplit_pressure_pc_type jacobi
```

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Block-Jacobi (Inexact), Cohouet & Chabard, IJNMF, 1988.

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type additive
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type gamg
 -fieldsplit_pressure_ksp_type preonly -fieldsplit_pressure_pc_type jacobi
```

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Gauss-Seidel (Inexact), Elman, DTIC, 1994.

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type multiplicative
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type gamg
 -fieldsplit_pressure_ksp_type preonly -fieldsplit_pressure_pc_type jacobi
```

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Gauss-Seidel (Inexact), Elman, DTIC, 1994.

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type multiplicative
 -pc_fieldsplit_0_fields 1 -pc_fieldsplit_1_fields 0
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type gamg
 -fieldsplit_pressure_ksp_type preonly -fieldsplit_pressure_pc_type jacobi
```

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Diagonal Schur Complement, Olshanskii, et.al., Numer. Math., 2006.

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type schur
-pc_fieldsplit_schur_factorization_type diag
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type gamg
 -fieldsplit_pressure_ksp_type minres  -fieldsplit_pressure_pc_type none
```

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Lower Schur Complement, May and Moresi, PEPI, 2008.

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type schur
-pc_fieldsplit_schur_factorization_type lower
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type gamg
 -fieldsplit_pressure_ksp_type minres  -fieldsplit_pressure_pc_type none
```

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Upper Schur Complement, May and Moresi, PEPI, 2008.

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type schur
-pc_fieldsplit_schur_factorization_type upper
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type gamg
 -fieldsplit_pressure_ksp_type minres  -fieldsplit_pressure_pc_type none
```

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Uzawa Iteration, Uzawa, 1958

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type schur
-pc_fieldsplit_schur_factorization_type upper
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type lu
 -fieldsplit_pressure_ksp_type richardson -fieldsplit_pressure_pc_type jacobi
 -fieldsplit_pressure_ksp_max_it 1
```

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Full Schur Complement, Schur, 1905.

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type schur
-pc_fieldsplit_schur_factorization_type full
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type lu
 -fieldsplit_pressure_ksp_rtol 1e-10 -fieldsplit_pressure_pc_type jacobi
```

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

SIMPLE, Patankar and Spalding, IJHMT, 1972.

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type schur
-pc_fieldsplit_schur_factorization_type full
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type lu
 -fieldsplit_pressure_ksp_rtol 1e-10 -fieldsplit_pressure_pc_type jacobi
 -fieldsplit_pressure_inner_ksp_type preonly
  -fieldsplit_pressure_inner_pc_type jacobi
 -fieldsplit_pressure_upper_ksp_type preonly
  -fieldsplit_pressure_upper_pc_type jacobi
```

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Least-Squares Commutator, Kay, Loghin and Wathen, SISC, 2002.

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type schur
-pc_fieldsplit_schur_factorization_type full
-pc_fieldsplit_schur_precondition self
 -fieldsplit_velocity_ksp_type gmres -fieldsplit_velocity_pc_type lu
 -fieldsplit_pressure_ksp_rtol 1e-5 -fieldsplit_pressure_pc_type lsc
```

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

Solver Configuration: No New Code

ex31: P_2/P_1 Stokes Problem with Temperature on Unstructured Mesh

Additive Schwarz + Full Schur Complement, Elman, Howle, Shadid, Shuttleworth, and Tuminaro, SISC, 2006.

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type additive
-pc_fieldsplit_0_fields 0,1 -pc_fieldsplit_1_fields 2
 -fieldsplit_0_ksp_type fgmres -fieldsplit_0_pc_type fieldsplit
 -fieldsplit_0_pc_fieldsplit_type schur
 -fieldsplit_0_pc_fieldsplit_schur_factorization_type full
   -fieldsplit_0_fieldsplit_velocity_ksp_type preonly
   -fieldsplit_0_fieldsplit_velocity_pc_type lu
   -fieldsplit_0_fieldsplit_pressure_ksp_rtol 1e-10
   -fieldsplit_0_fieldsplit_pressure_pc_type jacobi
 -fieldsplit_temperature_ksp_type preonly
 -fieldsplit_temperature_pc_type lu
```

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

Solver Configuration: No New Code

ex31: P_2/P_1 Stokes Problem with Temperature on Unstructured Mesh

Upper Schur Comp. + Full Schur Comp. + Least-Squares Comm.

```
-ksp_type fgmres -pc_type fieldsplit -pc_fieldsplit_type schur
-pc_fieldsplit_0_fields 0,1 -pc_fieldsplit_1_fields 2
-pc_fieldsplit_schur_factorization_type upper
 -fieldsplit_0_ksp_type fgmres -fieldsplit_0_pc_type fieldsplit
 -fieldsplit_0_pc_fieldsplit_type schur
 -fieldsplit_0_pc_fieldsplit_schur_factorization_type full
   -fieldsplit_0_fieldsplit_velocity_ksp_type preonly
   -fieldsplit_0_fieldsplit_velocity_pc_type lu
   -fieldsplit_0_fieldsplit_pressure_ksp_rtol 1e-10
   -fieldsplit_0_fieldsplit_pressure_pc_type jacobi
 -fieldsplit_temperature_ksp_type gmres
 -fieldsplit_temperature_pc_type lsc
```

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

The Great Solver Schism: Monolithic or Split?

- Direct solvers
- Coupled Schwarz
- Coupled Neumann-Neumann (use unassembled matrices)
- Coupled Multigrid

- Physics-split Schwarz (based on relaxation)
- Physics-split Schur (based on factorization)
  - SIMPLE, PCD, LSC
  - segregated smoothers
  - Augmented Lagrangian

Need to understand

- Local spectral properties
- Compatibility properties

- Global coupling strengths

Preferred data structures depend on which method is used.

Outline

Patch Solver People

Lawrence Mitchell

Patrick Farrell

Smoothers for

$$L + \alpha K$$

can suffer as $\alpha \to \infty$ if

$$\mathcal{N}(K) \neq \emptyset.$$

Smoothers for

$$-\nabla \cdot 2\nu\epsilon(\mathbf{u}) + (\mathbf{u} \cdot \nabla)\mathbf{u} - \alpha\nabla(\nabla \cdot \mathbf{u})$$

can suffer as $\alpha \to \infty$ if

$$\mathcal{N}(\nabla(\nabla \cdot \mathbf{u})) \neq \emptyset.$$

The Schur complement is almost

$$S^{-1} \approx -(\nu + \alpha) M_p^{-1}$$

but the velocity smoother is hard.

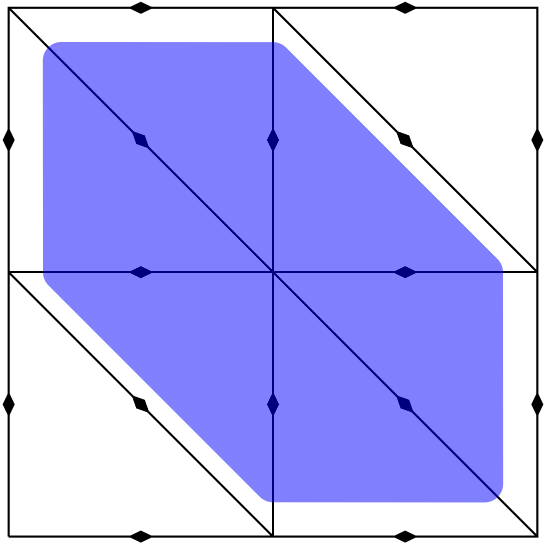
Patch smoothers satisfying

$$\mathcal{N}(K) = \sum_i V_i \bigcap \mathcal{N}(K)$$

are robust.

(Schöberl 1999)

Parameter-Robust Smoothers



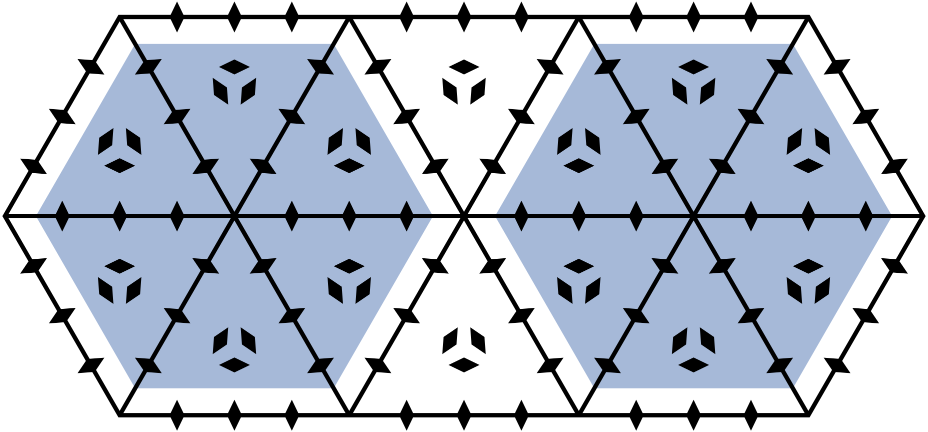
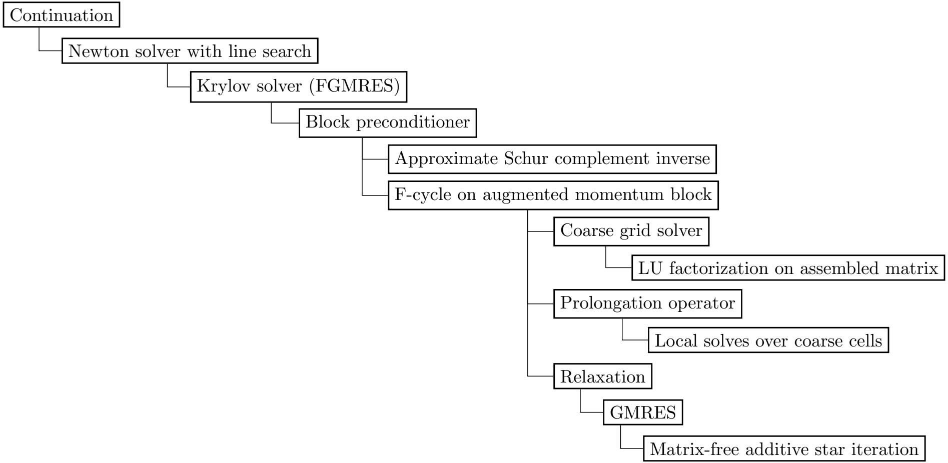


FIG. 3.1. *Star patch for* $\mathbb{BDM}_2$*-elements.*

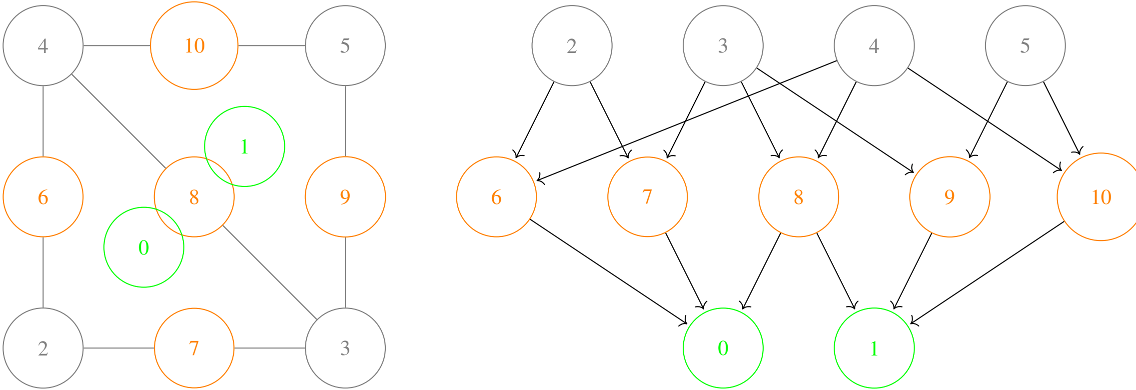
Parameter-Robust Smoothers

Incompressible Navier-Stokes

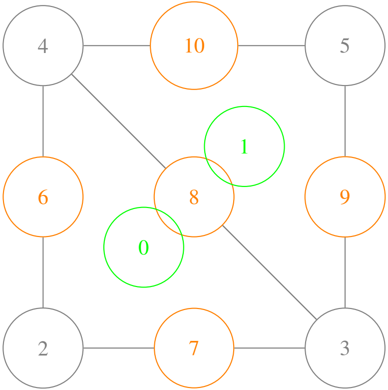


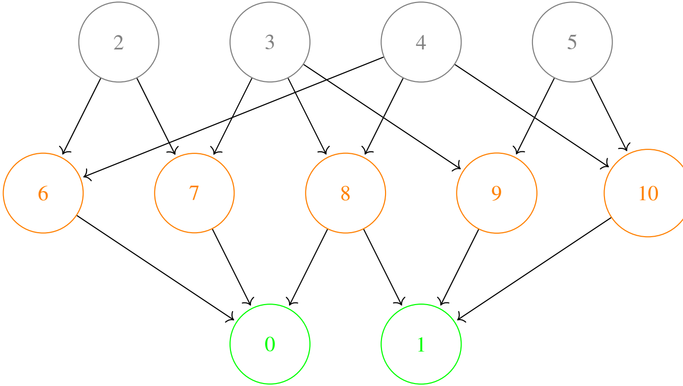
(Farrell, Mitchell, and Wechsung 2019)

Hasse Diagram (Wikipedia 2015)



DMPlex (Lange, Mitchell, Matthew G. Knepley, and Gorman 2016)





Solver for the $\mathcal{H}(\mathrm{div})$ Riesz map

```
-ksp_type cg
-pc_type mg
-mg_levels_ksp_type richardson
-mg_levels_ksp_richardson_scale 0.333333
-mg_levels_pc_type patch
-mg_levels_patch_pc_patch_local_type additive
-mg_levels_patch_pc_patch_construct_type star
-mg_levels_patch_pc_patch_construct_dim 0
```

(Farrell, Matthew G Knepley, Mitchell, and Wechsung 2021)

Many papers followed

(Adler, Benson, et al. 2021)

(Adler, He, et al. 2022)

(Laakmann, Farrell, and Mitchell 2022)

(Abu-Labdeh, MacLachlan, and Farrell 2023)

(Laakmann, Hu, and Farrell 2023)

on different problems.

Composable Solvers

Incompressible Viscoresistive MHD

- Continuation in $\mathrm{Re_m}$ and Re (in stationary case)
  - Newton/Picard iteration with line search
    - Krylov solver (FGMRES)
      - Block preconditioner with $(\mathbf{u}, p)$ and $(\mathbf{E}, \mathbf{B})$-block
        - Solver for $(\mathbf{u}, p)$-block
        - Schur complement approximation with $\tilde{\mathcal{S}}^{(\mathbf{u},p)}$
          - Solver for $\tilde{\mathcal{S}}^{(\mathbf{u},p)}$

(Laakmann, Farrell, and Mitchell 2022)

Composable Solvers

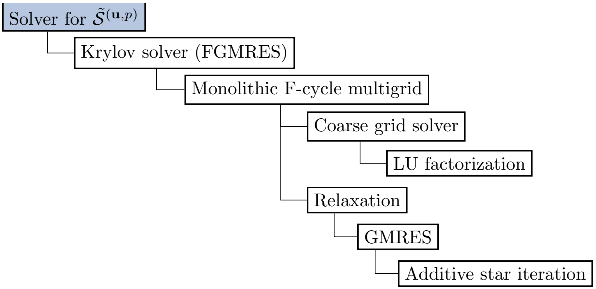
Incompressible Viscoresistive MHD

- Solver for $(\mathbf{u}, p)$-block
  - Krylov solver (FGMRES)
    - Block preconditioner
      - Approximate Schur complement inverse
        - Exact pressure mass matrix inverse
      - F-cycle on augmented momentum block
        - Coarse grid solver
          - LU factorization
        - Relaxation
          - GMRES
            - Additive star iteration

(Laakmann, Farrell, and Mitchell 2022)

Composable Solvers

Incompressible Viscoresistive MHD



(Laakmann, Farrell, and Mitchell 2022)

Outline

- Allen-Cahn Solver
- Stokes Solvers
- Patch Solvers
- Preconditioners for Faults
- Lessons

A modern, community simulator for crustal deformation

Brad Aagaard

Charles Williams

Open-source, community code

- Modular design
- Testing
- Documentation
- Distribution

PyLith 1.0 released in 2007

PyLith

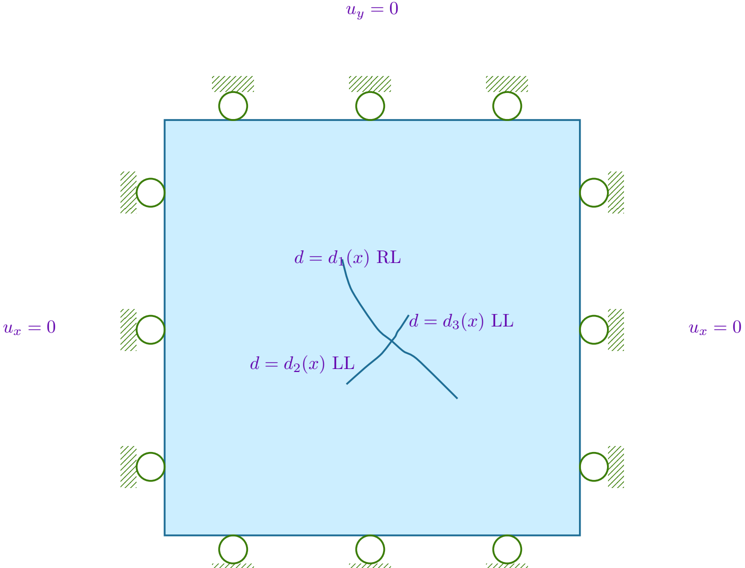
A modern, community simulator for crustal deformation

Elasticity
(Aagaard, Matthew G. Knepley, and Williams 2013)

Poroelasticity
(Walker, Matthew G. Knepley, Aagaard, and Williams 2023)

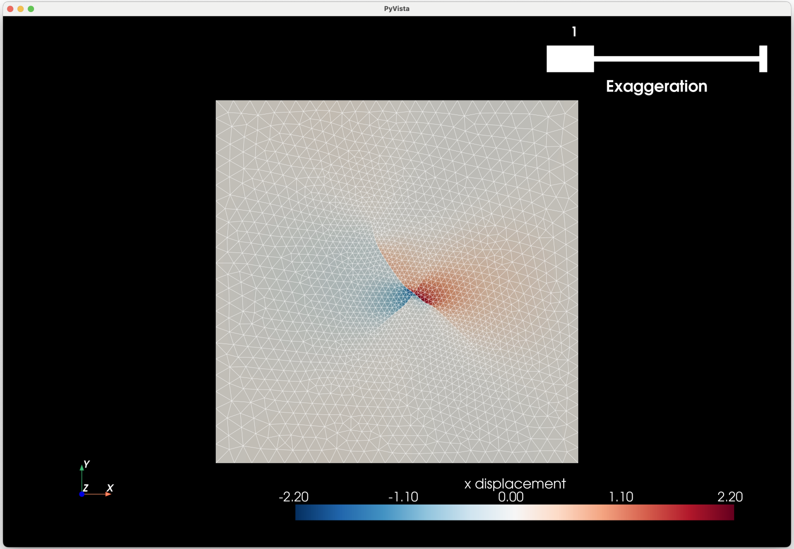
2D Crustal Strike-Slip

2019 Ridgecrest



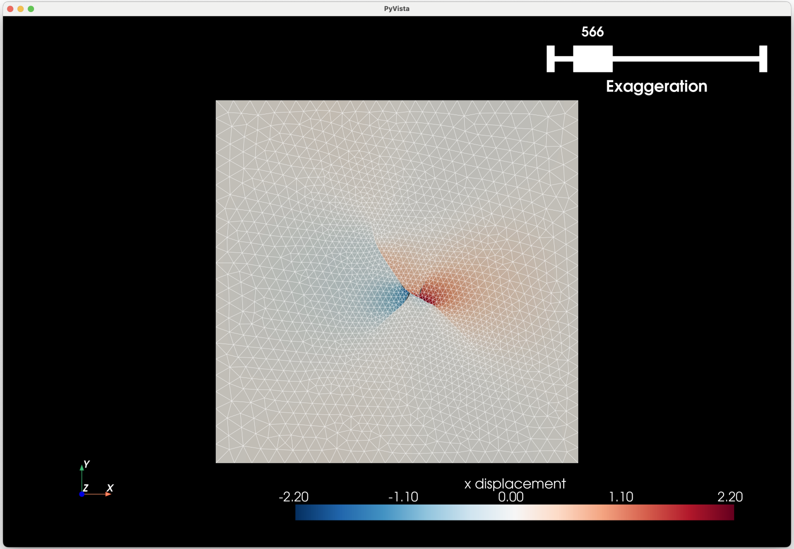
2D Crustal Strike-Slip

2019 Ridgecrest



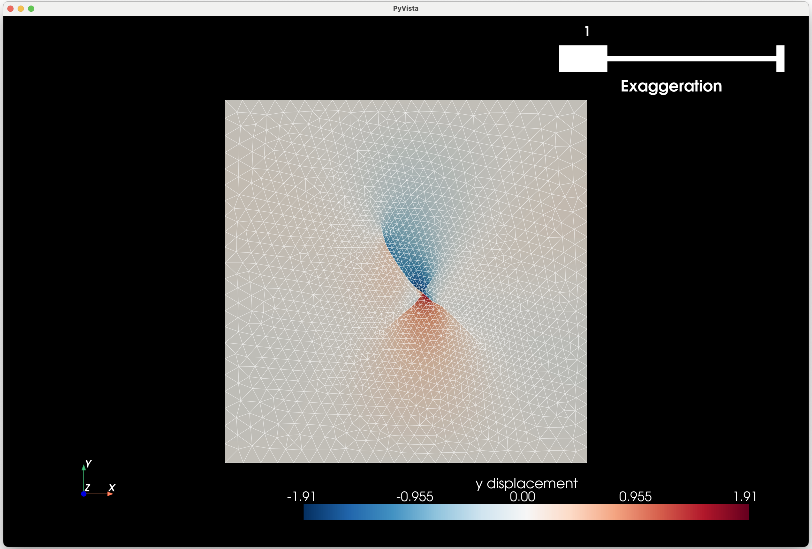
2D Crustal Strike-Slip

2019 Ridgecrest



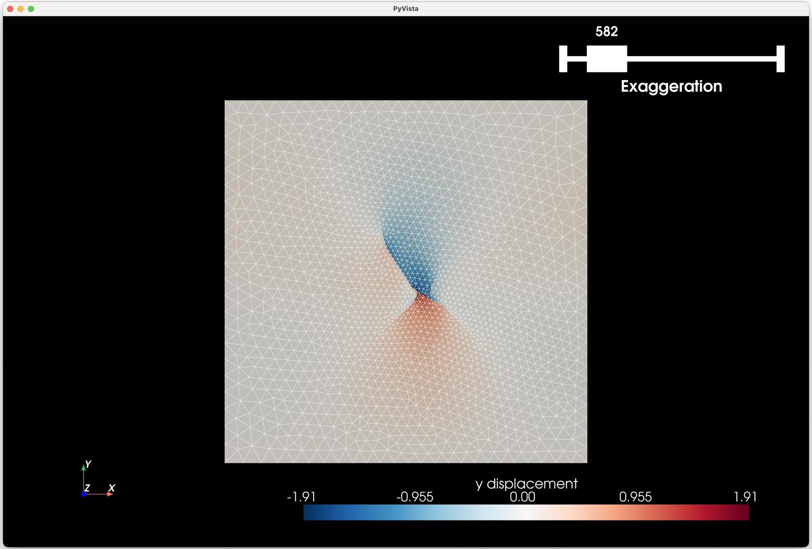
2D Crustal Strike-Slip

2019 Ridgecrest

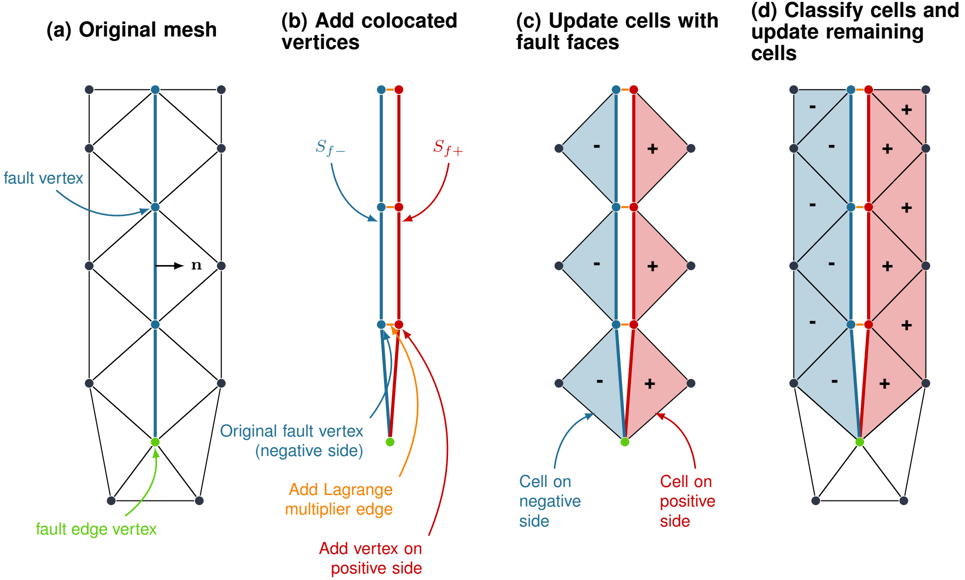


2D Crustal Strike-Slip

2019 Ridgecrest



Cohesive Formulation

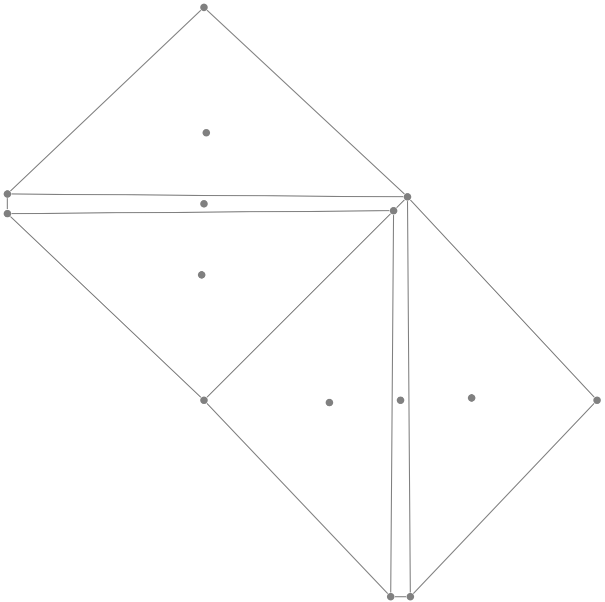


Governing Equations

$$\int_{\Omega_f} \rho(\mathbf{x}) \frac{\partial \mathbf{v}}{\partial t} \, d\Omega = \int_{\Gamma_{f+}} \boldsymbol{\sigma} \cdot \mathbf{n} + \boldsymbol{\lambda} \, d\Gamma + \int_{\Gamma_{f-}} \boldsymbol{\sigma} \cdot \mathbf{n} - \boldsymbol{\lambda} \, d\Gamma$$

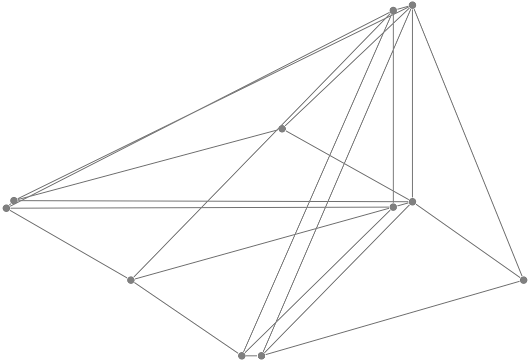
$$\mathbf{u}^{+} - \mathbf{u}^{-} = \mathbf{d}(\mathbf{x}, t)$$

Test Mesh



Test Mesh

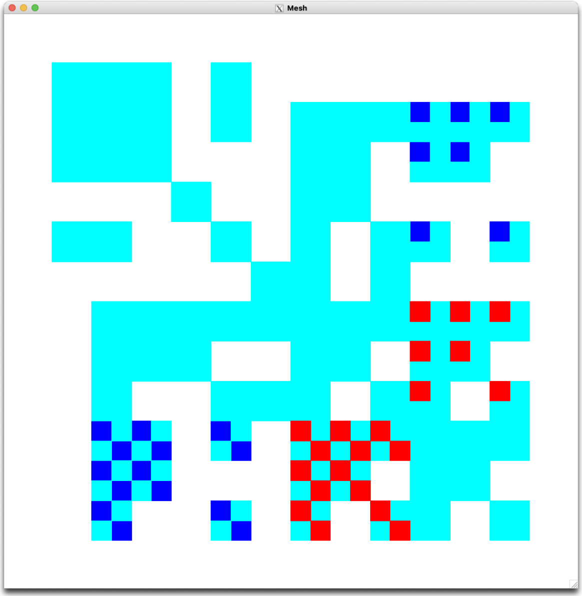
3D



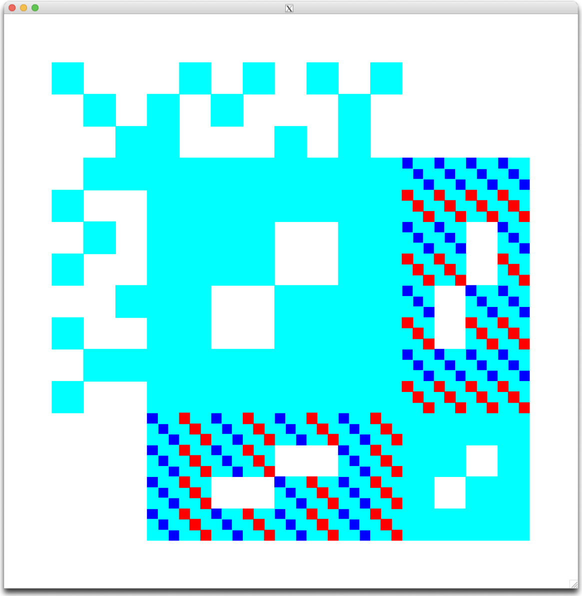
We have a saddle-point system,

$$\begin{pmatrix} E & C \\ C^T & 0 \end{pmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ d \end{pmatrix}$$

Jacobian



Jacobian



Schur complement Solver

```
-ksp_gmres_restart 100
-pc_type fieldsplit
-pc_use_amat
-pc_fieldsplit_type schur

-pc_fieldsplit_schur_factorization_type lower
-pc_fieldsplit_schur_precondition selfp
-pc_fieldsplit_schur_scale 1.0

-fieldsplit_displacement_ksp_type preonly
-fieldsplit_displacement_pc_type ml

-fieldsplit_lagrange_multiplier_fault_ksp_type preonly
-fieldsplit_lagrange_multiplier_fault_pc_type ml
```

Schur complement Solver

```
-snes_ksp_ew
-snes_ksp_ew_rtol0 1e-4

-ksp_gmres_restart 100

-fieldsplit_displacement_ksp_type gmres
-fieldsplit_displacement_ksp_max_it 10
```

Schur complement Solver

- $S = -C^T EC$ connects both sides
- Difficult to precondition S
- Not scalable with problem size

Using topology and layout information,

in DMPlex and PetscSection,

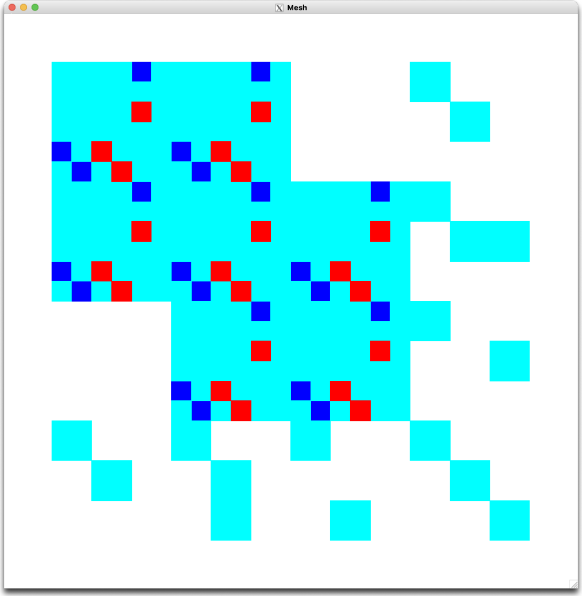
we compute a reordering

and blocking automatically.

Reordering

```
Mat Object: Jacobian 1 MPI process
  type: seqaij
  rows=24, cols=24, bs=2 variable blocks set
  total: nonzeros=320, allocated nonzeros=320
  total number of mallocs used during MatSetValues calls=0
    IS Object: Block Sizes 1 MPI process
      type: general
    Number of indices in set 5
    0 12
    1 6
    2 2
    3 2
    4 2
```

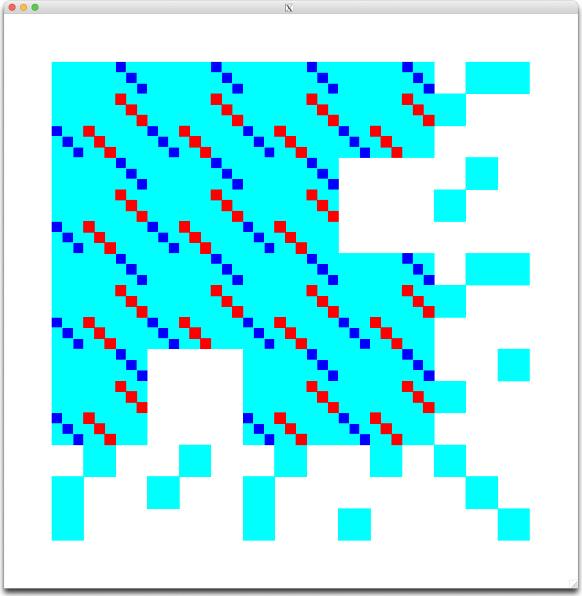
Reordering



Reordering

```
Mat Object: Jacobian 1 MPI process
  type: seqaij
  rows=45, cols=45, bs=3 variable blocks set
  total: nonzeros=1341, allocated nonzeros=1341
  total number of mallocs used during MatSetValues calls=0
    IS Object: Block Sizes 1 MPI process
      type: general
    Number of indices in set 5
    0 27
    1 9
    2 3
    3 3
    4 3
```

Reordering



Multigrid Solver

```
-dm_reorder_section
-dm_reorder_section_type cohesive

-ksp_gmres_restart 100
-pc_type gamg
-mg_fine_pc_type vpbjacobi
```

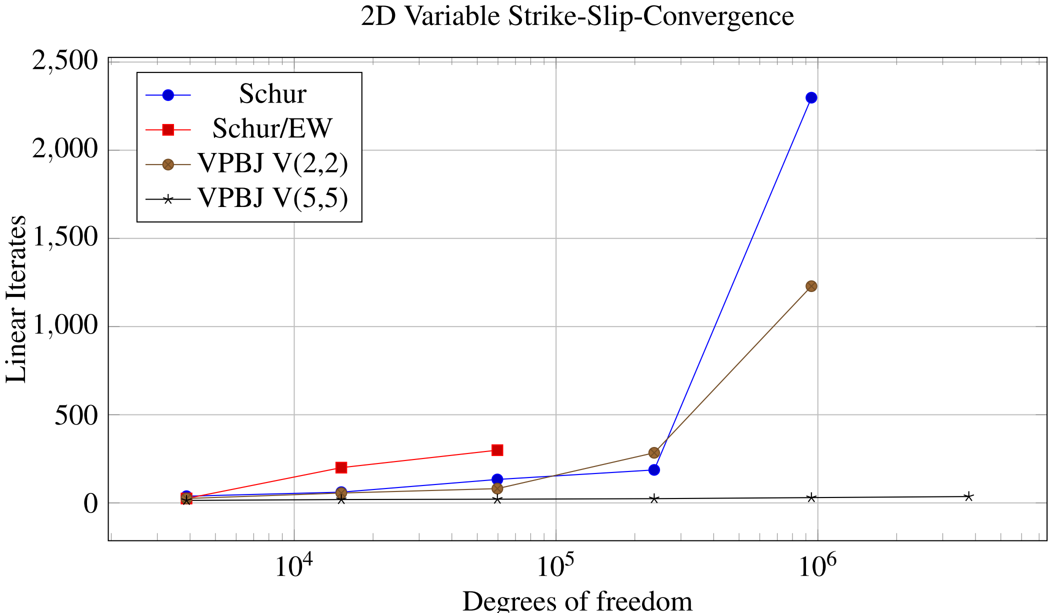
Multigrid Solver

```
-mg_levels_pc_type pbjacobi

-pc_gamg_coarse_eq_limit 200

-mg_fine_ksp_max_it 5
```

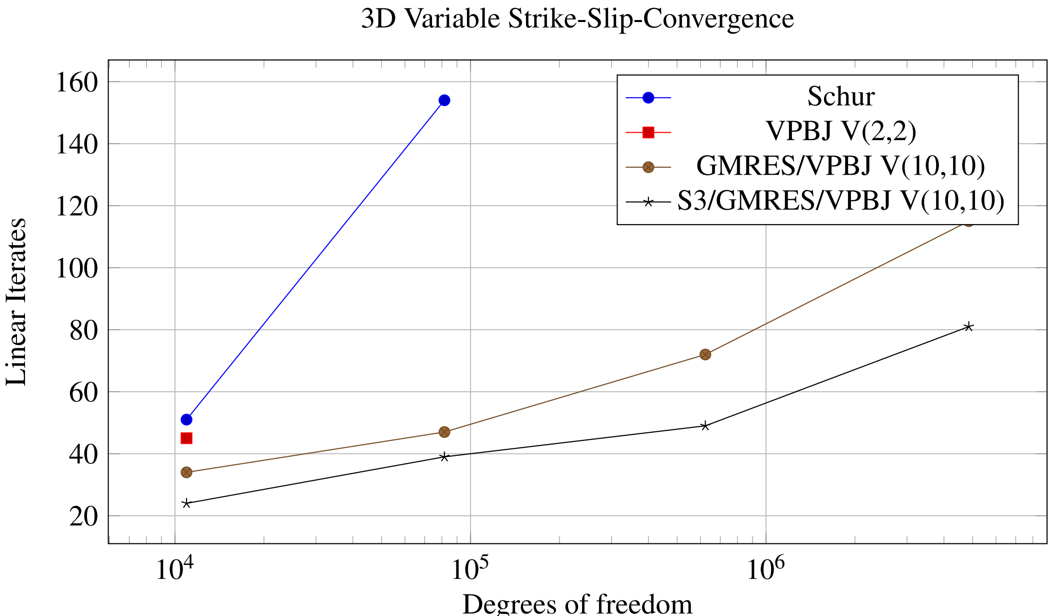
Multigrid Solver

```
-pc_gamg_agg_nsmooths 3

-mg_fine_ksp_type gmres

-mg_fine_ksp_max_it 10
```

Convergence



Convergence



Outline

Expose better abstractions

Expose better abstractions

at runtime

Expose better abstractions

at runtime

that compose together.

Expose better abstractions

at runtime

that compose together.

(Brown, Matthew G. Knepley, and B. Smith 2015)

Build in Layers

Build in Layers

to allow targeted APIs

Build in Layers

to allow targeted APIs

that preserve understandability.

Build in Layers

to allow targeted APIs

that preserve understandability.

(B. F. Smith and Gropp 1996)

PETSc Integration

PETSc is a set a library interfaces

- We do not seize `main()`
- We do not control output
- We propagate errors from underlying packages
- We present the same interfaces in:
  - C
  - C++
  - Fortran
  - Python
  - Julia

See Gropp in SIAM, OO Methods for Interop SciEng, '99

Advice from Bill Gropp

> *You want to think about how you decompose your data structures, how you think about them globally. [...] If you were building a house, you'd start with a set of blueprints that give you a picture of what the whole house looks like. You wouldn't start with a bunch of tiles and say. "Well I'll put this tile down on the ground, and then I'll find a tile to go next to it." But all too many people try to build their parallel programs by creating the smallest possible tiles and then trying to have the structure of their code emerge from the chaos of all these little pieces. You have to have an organizing principle if you're going to survive making your code parallel.*

(http://www.rce-cast.com/Podcast/rce-28-mpich2.html)

References I

Brown, Jed, Matthew Knepley, David A. May, Lois C. McInnes, and Barry F. Smith (2012). “Composable linear solvers for multiphysics”. In: Proceeedings of the 11th International Symposium on Parallel and Distributed Computing (ISPDC 2012). IEEE Computer Society, pp. 55–62. DOI: 10.1109/ISPDC.2012.16.

Brune, Peter R., Matthew G. Knepley, Barry F. Smith, and Xuemin Tu (2015). “Composing Scalable Nonlinear Algebraic Solvers”. In: SIAM Review 57.4. http://www.mcs.anl.gov/papers/P2010-0112.pdf, pp. 535–565. DOI: 10.1137/130936725. URL: http://www.mcs.anl.gov/papers/P2010-0112.pdf.

Schöberl, Joachim (1999). “Multigrid methods for a parameter dependent problem in primal variables”. In: Numerische Mathematik 84.1, pp. 97–119.

Farrell, Patrick E, Lawrence Mitchell, and Florian Wechsung (2019). “An augmented Lagrangian preconditioner for the 3D stationary incompressible Navier-Stokes equations at high Reynolds number”. In: SIAM Journal on Scientific Computing 41.5, A3073–A3096. eprint: 1810.03315.

Wikipedia (2015). Hasse Diagram. http://en.wikipedia.org/wiki/Hasse_diagram. URL: http://en.wikipedia.org/wiki/Hasse_diagram.

Lange, Michael, Lawrence Mitchell, Matthew G. Knepley, and Gerard J. Gorman (2016). “Efficient mesh management in Firedrake using PETSc-DMPlex”. In: SIAM Journal on Scientific Computing 38.5, S143–S155. DOI: 10.1137/15M1026092. eprint: http://arxiv.org/abs/1506.07749.

Farrell, Patrick E, Matthew G Knepley, Lawrence Mitchell, and Florian Wechsung (2021). “PCPATCH: software for the topological construction of multigrid relaxation methods”. In: ACM Transaction on Mathematical Software 47.3, pp. 1–22. ISSN: 0098-3500. DOI: 10.1145/3445791. eprint: http://arxiv.org/abs/1912.08516.

Adler, James H, Thomas R Benson, Eric C Cyr, Patrick E Farrell, Scott P MacLachlan, and Ray S Tuminaro (2021). “Monolithic Multigrid Methods for Magnetohydrodynamics”. In: SIAM Journal on Scientific Computing 0, S70–S91.

Adler, James H, Yunhui He, Xiaozhe Hu, Scott MacLachlan, and Peter Ohm (2022). “Monolithic multigrid for a reduced-quadrature discretization of poroelasticity”. In: SIAM Journal on Scientific Computing 45.3, S54–S81.

References II

Laakmann, Fabian, Patrick E Farrell, and Lawrence Mitchell (2022). “An augmented Lagrangian preconditioner for the magnetohydrodynamics equations at high Reynolds and coupling numbers”. In: SIAM Journal on Scientific Computing 44.4, B1018–B1044.

Abu-Labdeh, Razan, Scott MacLachlan, and Patrick E Farrell (2023). “Monolithic multigrid for implicit Runge–Kutta discretizations of incompressible fluid flow”. In: Journal of Computational Physics 478, p. 111961.

Laakmann, Fabian, Kaibo Hu, and Patrick E Farrell (2023). “Structure-preserving and helicity-conserving finite element approximations and preconditioning for the Hall MHD equations”. In: Journal of Computational Physics 492, p. 112410.

Aagaard, Brad T., Matthew G. Knepley, and Charles A. Williams (2013). “A Domain Decomposition Approach to Implementing Fault Slip in Finite-Element Models of Quasi-static and Dynamic Crustal Deformation”. In: Journal of Geophysical Research: Solid Earth 118.6, pp. 3059–3079. ISSN: 2169-9356. DOI: 10.1002/jgrb.50217.

Walker, Robert L., Matthew G. Knepley, Brad T. Aagaard, and Charles A. Williams (2023). “Multiphysics Modeling in PyLith: Poroelasticity”. In: Geophysical Journal International 235.3, pp. 2442–2475. DOI: 10.1093/gji/ggad370.

Brown, Jed, Matthew G. Knepley, and Barry Smith (Jan. 2015). “Run-time extensibility and librarization of simulation software”. In: IEEE Computing in Science and Engineering 17.1, pp. 38–45. DOI: 10.1109/MCSE.2014.95.

Smith, Barry F. and William D. Gropp (1996). “The Design of Data-structure-neutral Libraries for the Iterative Solution of Sparse Linear Systems”. In: Scientific Programming 5, pp. 329–336.