Composing Nonlinear Solvers

Matthew Knepley

Computational and Applied Mathematics
Rice University

MIT Aeronautics and Astronautics
Boston, MA May 10, 2016

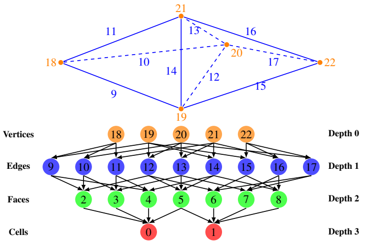
PETSc is one of the most popular software libraries in scientific computing.

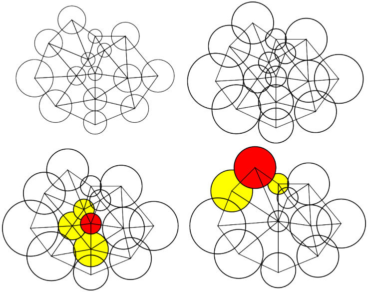
As a principal architect since 2001, I developed

- unstructured meshes (model, algorithms, implementation),
- nonlinear preconditioning (model, algorithms),
- FEM discretizations (data structures, solvers optimization),
- optimizations for multicore and GPU architectures.

What is PETSc?

Knepley, Karpeev, Sci. Prog., 2009. Brune, Knepley, Scott, SISC, 2013.





As a principal architect since 2001, I developed

- unstructured meshes (model, algorithms, implementation),
- nonlinear preconditioning (model, algorithms),
- FEM discretizations (data structures, solvers optimization),
- optimizations for multicore and GPU architectures.

What is PETSc?

Brune, Knepley, Smith, and Tu, SIAM Review, 2015.

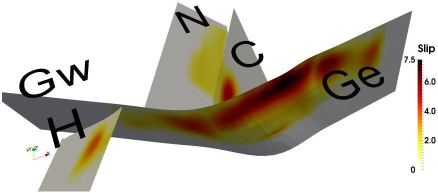
Type	Sym	Statement	Abbreviation
Additive	$+$	$\vec{x} + \alpha(\mathcal{M}(\mathcal{F}, \vec{x}, \vec{b}) - \vec{x})$ $+ \beta(\mathcal{N}(\mathcal{F}, \vec{x}, \vec{b}) - \vec{x})$	$\mathcal{M} + \mathcal{N}$
Multiplicative	$*$	$\mathcal{M}(\mathcal{F}, \mathcal{N}(\mathcal{F}, \vec{x}, \vec{b}), \vec{b})$	$\mathcal{M} * \mathcal{N}$
Left Prec.	$-_L$	$\mathcal{M}(\vec{x} - \mathcal{N}(\mathcal{F}, \vec{x}, \vec{b}), \vec{x}, \vec{b})$	$\mathcal{M} -_L \mathcal{N}$
Right Prec.	$-_R$	$\mathcal{M}(\mathcal{F}(\mathcal{N}(\mathcal{F}, \vec{x}, \vec{b})), \vec{x}, \vec{b})$	$\mathcal{M} -_R \mathcal{N}$
Inner Lin. Inv.	$\backslash$	$\vec{y} = \vec{J}(\vec{x})^{-1}\vec{r}(\vec{x}) = \mathrm{K}(\vec{J}(\vec{x}), \vec{y}_0, \vec{b})$	$\mathcal{N} \backslash \mathrm{K}$

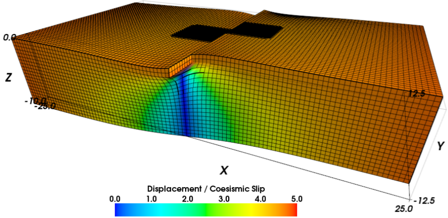
As a principal architect since 2001, I developed

- unstructured meshes (model, algorithms, implementation),
- nonlinear preconditioning (model, algorithms),
- FEM discretizations (data structures, solvers optimization),
- optimizations for multicore and GPU architectures.

What is PETSc?

Aagaard, Knepley, and Williams, J. of Geophysical Research, 2013.



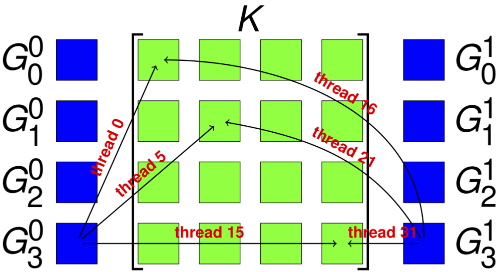


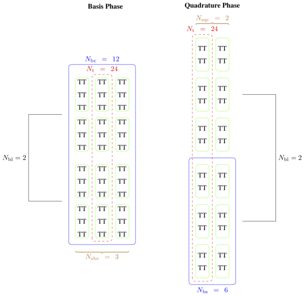
As a principal architect since 2001, I developed

- unstructured meshes (model, algorithms, implementation),
- nonlinear preconditioning (model, algorithms),
- FEM discretizations (data structures, solvers optimization),
- optimizations for multicore and GPU architectures.

What is PETSc?

Knepley and Terrel, Transactions on Mathematical Software, 2012.





As a principal architect since 2001, I developed

- unstructured meshes (model, algorithms, implementation),
- nonlinear preconditioning (model, algorithms),
- FEM discretizations (data structures, solvers optimization),
- optimizations for multicore and GPU architectures.

Main Idea

When you see an
approximate solver,

I see a preconditioner.

Main Idea

When you see an
approximate solver,

I see a preconditioner.

There are *many* approaches for saddle-point problems:

- Block preconditioners
- Schur complement methods
- Multigrid with special smoothers

$$\begin{pmatrix} F & B & M \\ B^T & 0 & 0 \\ N & 0 & K \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \\ T \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \\ q \end{pmatrix}$$

However, today it is hard to compare & combine them and combine in a **hierarchical** manner. For instance we might want,

There are *many* approaches for saddle-point problems:

- Block preconditioners
- Schur complement methods
- Multigrid with special smoothers

$$\begin{pmatrix} F & B & M \\ B^T & 0 & 0 \\ N & 0 & K \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \\ T \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \\ q \end{pmatrix}$$

However, today it is hard to compare & combine them and combine in a **hierarchical** manner. For instance we might want,

There are *many* approaches for saddle-point problems:

- Block preconditioners
- Schur complement methods
- Multigrid with special smoothers

$$\begin{pmatrix} F & B & M \\ B^T & 0 & 0 \\ N & 0 & K \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \\ T \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \\ q \end{pmatrix}$$

However, today it is hard to compare & combine them and combine in a **hierarchical** manner. For instance we might want,

> a Gauss-Siedel iteration between blocks of $(\mathbf{u}, p)$ and T,
> and a full Schur complement factorization for $\mathbf{u}$ and p.

There are *many* approaches for saddle-point problems:

- Block preconditioners
- Schur complement methods
- Multigrid with special smoothers

$$\begin{pmatrix} F & B & M \\ B^T & 0 & 0 \\ N & 0 & K \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \\ T \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \\ q \end{pmatrix}$$

However, today it is hard to compare & combine them and combine in a **hierarchical** manner. For instance we might want,

> an upper triangular Schur complement factorization for $\mathbf{u}$ and p, and geometric multigrid for the $\mathbf{u}$ block.

There are *many* approaches for saddle-point problems:

- Block preconditioners
- Schur complement methods
- Multigrid with special smoothers

$$\begin{pmatrix} F & B & M \\ B^T & 0 & 0 \\ N & 0 & K \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \\ T \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \\ q \end{pmatrix}$$

However, today it is hard to compare & combine them and combine in a **hierarchical** manner. For instance we might want,

> algebraic multigrid for the full $(\mathbf{u}, p)$ system,
> using a block triangular Gauss-Siedel smoother on each level,
> and use identity for the (p, p) block.

Approach for efficient, robust, scalable linear solvers

Need solvers to be:

- **Composable**: separately developed solvers may be easily combined, by non-experts, to form a more powerful solver
- **Nested**: outer solvers call inner solvers
- **Hierarchical**: outer solvers may iterate over all variables for a global problem, while nested inner solvers handle smaller subsets of physics, smaller physical subdomains, or coarser meshes
- **Extensible**: users can easily customize/extend

Composable Linear Solvers for Multiphysics, Brown, Knepley, May, McInnes, Smith, IPDPS, 2012.

FieldSplit Customization

- Analysis
  - `-pc_fieldsplit_<`**split num**`>_fields 2,1,5`
  - `-pc_fieldsplit_detect_saddle_point`
- Synthesis
  - `-pc_fieldsplit_type`
  - `-pc_fieldsplit_real_diagonal`
    Use diagonal blocks of operator to build PC
- Schur complements
  - `-pc_fieldsplit_schur_precondition <self,user,diag>`
    How to build preconditioner for S
  - `-pc_fieldsplit_schur_factorization_type <diag,lower,upper,full>`
    Which off-diagonal parts of the block factorization to use

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_ksp_max_its 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
```

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

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
```

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

The Great Solver Schism: Monolithic or Split?

Monolithic

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

Split

- 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

Outline

Abstract System

Out prototypical nonlinear equation is:

$$\mathcal{F}(\vec{x}) = \vec{b} \tag{1}$$

and we define the residual as

$$\vec{r}(\vec{x}) = \mathcal{F}(\vec{x}) - \vec{b} \tag{2}$$

Abstract System

Out prototypical nonlinear equation is:

$$\mathcal{F}(\vec{x}) = \vec{b} \tag{1}$$

and we define the (linear) residual as

$$\vec{r}(\vec{x}) = A\vec{x} - \vec{b} \tag{3}$$

Linear Left Preconditioning

The modified equation becomes

$$P^{-1}\left(A\vec{x} - \vec{b}\right) = 0 \tag{4}$$

Linear Left Preconditioning

The modified defect correction equation becomes

$$P^{-1}\left(A\vec{x}_i - \vec{b}\right) = \vec{x}_{i+1} - \vec{x}_i \tag{5}$$

Additive Combination

The linear iteration

$$\vec{x}_{i+1} = \vec{x}_i - (\alpha P^{-1} + \beta Q^{-1})(A\vec{x}_i - \vec{b}) \tag{6}$$

becomes the nonlinear iteration

Additive Combination

The linear iteration

$$\vec{x}_{i+1} = \vec{x}_i - (\alpha P^{-1} + \beta Q^{-1})\vec{r}_i \tag{7}$$

becomes the nonlinear iteration

Additive Combination

The linear iteration

$$\vec{x}_{i+1} = \vec{x}_i - (\alpha P^{-1} + \beta Q^{-1})\vec{r}_i \tag{7}$$

becomes the nonlinear iteration

$$\vec{x}_{i+1} = \vec{x}_i + \alpha(\mathcal{N}(\mathcal{F}, \vec{x}_i, \vec{b}) - \vec{x}_i) + \beta(\mathcal{M}(\mathcal{F}, \vec{x}_i, \vec{b}) - \vec{x}_i) \tag{8}$$

Nonlinear Left Preconditioning

From the additive combination, we have

$$P^{-1}\vec{r} \Longrightarrow \vec{x}_i - \mathcal{N}(\mathcal{F}, \vec{x}_i, \vec{b}) \tag{9}$$

so we define the preconditioning operation as

$$\vec{r}_L \equiv \vec{x} - \mathcal{N}(\mathcal{F}, \vec{x}, \vec{b}) \tag{10}$$

Multiplicative Combination

The linear iteration

$$\vec{x}_{i+1} = \vec{x}_i - (P^{-1} + Q^{-1} - Q^{-1}AP^{-1})\vec{r}_i \qquad (11)$$

becomes the nonlinear iteration

Multiplicative Combination

The linear iteration

$$\vec{x}_{i+1/2} = \vec{x}_i - P^{-1}\vec{r}_i \tag{12}$$

$$\vec{x}_i = \vec{x}_{i+1/2} - Q^{-1}\vec{r}_{i+1/2} \tag{13}$$

becomes the nonlinear iteration

Multiplicative Combination

The linear iteration

$$\vec{x}_{i+1/2} = \vec{x}_i - P^{-1}\vec{r}_i \tag{12}$$

$$\vec{x}_i = \vec{x}_{i+1/2} - Q^{-1}\vec{r}_{i+1/2} \tag{13}$$

becomes the nonlinear iteration

$$\vec{x}_{i+1} = \mathcal{M}(\mathcal{F}, \mathcal{N}(\mathcal{F}, \vec{x}_i, \vec{b}), \vec{b}) \tag{14}$$

Nonlinear Right Preconditioning

For the linear case, we have

$$AP^{-1}\vec{y} = \vec{b} \tag{15}$$

$$\vec{x} = P^{-1}\vec{y} \tag{16}$$

so we define the preconditioning operation as

$$\vec{y} = \mathcal{M}(\mathcal{F}(\mathcal{N}(\mathcal{F}, \cdot, \vec{b})), \vec{x}_i, \vec{b}) \tag{17}$$

$$\vec{x} = \mathcal{N}(\mathcal{F}, \vec{y}, \vec{b}) \tag{18}$$

Nonlinear Preconditioning

Type	Sym	Statement	Abbreviation
Additive	$+$	$\vec{x} + \alpha(\mathcal{M}(\mathcal{F}, \vec{x}, \vec{b}) - \vec{x})$ $+ \beta(\mathcal{N}(\mathcal{F}, \vec{x}, \vec{b}) - \vec{x})$	$\mathcal{M} + \mathcal{N}$
Multiplicative	$*$	$\mathcal{M}(\mathcal{F}, \mathcal{N}(\mathcal{F}, \vec{x}, \vec{b}), \vec{b})$	$\mathcal{M} * \mathcal{N}$
Left Prec.	$-_L$	$\mathcal{M}(\vec{x} - \mathcal{N}(\mathcal{F}, \vec{x}, \vec{b}), \vec{x}, \vec{b})$	$\mathcal{M} -_L \mathcal{N}$
Right Prec.	$-_R$	$\mathcal{M}(\mathcal{F}(\mathcal{N}(\mathcal{F}, \vec{x}, \vec{b})), \vec{x}, \vec{b})$	$\mathcal{M} -_R \mathcal{N}$
Inner Lin. Inv.	$\backslash$	$\vec{y} = \vec{J}(\vec{x})^{-1}\vec{r}(\vec{x}) = \mathrm{K}(\vec{J}(\vec{x}), \vec{y}_0, \vec{b})$	$\mathcal{N} \backslash \mathrm{K}$

Composing Scalable Nonlinear Algebraic Solvers,
Brune, Knepley, Smith, and Tu, SIAM Review, 2015.

Outline

Outline

2 Theory
- Convergence Rates
- Induction Theorem

Rate of Convergence

What should be a Rate of Convergence? [Ptak, 1977]:

1. It should relate quantities which may be measured or estimated during the actual process
2. It should describe accurately in particular the initial stage of the process, not only its asymptotic behavior …

$$\|x_{n+1} - x^*\| \leq c\|x_n - x^*\|^q$$

Rate of Convergence

What should be a Rate of Convergence? [Ptak, 1977]:

1. It should relate quantities which may be measured or estimated during the actual process
2. It should describe accurately in particular the initial stage of the process, not only its asymptotic behavior . . .

$$\|x_{n+1} - x_n\| \leq c\|x_n - x_{n-1}\|^q$$

Rate of Convergence

What should be a Rate of Convergence? [Ptak, 1977]:

1. It should relate quantities which may be measured or estimated during the actual process
2. It should describe accurately in particular the initial stage of the process, not only its asymptotic behavior …

$$\|x_{n+1} - x_n\| \leq \omega(\|x_n - x_{n-1}\|)$$

where we have for all $r \in (0, R]$

$$\sigma(r) = \sum_{n=0}^{\infty} \omega^{(n)}(r) < \infty$$

Nondiscrete Induction

Define an approximate set $Z(r)$, where $x^* \in Z(0)$ implies $f(x^*) = 0$.

Nondiscrete Induction

Define an approximate set $Z(r)$, where $x^* \in Z(0)$ implies $f(x^*) = 0$.

For Newton's method, we use

$$Z(r) = \left\{ x \middle| \|f'(x)^{-1} f(x)\| \leq r, d(f'(x)) \geq h(r), \|x - x_0\| \leq g(r) \right\},$$

where

$$d(A) = \inf_{\|x\| \geq 1} \|Ax\|,$$

and $h(r)$ and $g(r)$ are positive functions.

Nondiscrete Induction

Define an approximate set $Z(r)$, where $x^* \in Z(0)$ implies $f(x^*) = 0$.

For $r \in (0, R]$,

$$Z(r) \subset U(Z(\omega(r)), r)$$

implies

$$Z(r) \subset U(Z(0), \sigma(r)).$$

Nondiscrete Induction

For the fixed point iteration

$$x_{n+1} = Gx_n,$$

if I have

$$x_0 \in Z(r_0)$$

and for $x \in Z(r)$,

$$\begin{aligned} \|Gx - x\| &\leq r \\ Gx &\in Z(\omega(r)) \end{aligned}$$

then

Nondiscrete Induction

For the fixed point iteration

$$x_{n+1} = Gx_n,$$

if I have

$$x_0 \in Z(r_0)$$

and for $x \in Z(r)$,

$$\begin{aligned} \|Gx - x\| &\leq r \\ Gx &\in Z(\omega(r)) \end{aligned}$$

then

$$\begin{aligned} x^* &\in Z(0) \\ x_n &\in Z(\omega^{(n)}(r_0)) \end{aligned}$$

Nondiscrete Induction

For the fixed point iteration

$$x_{n+1} = Gx_n,$$

if I have

$$x_0 \in Z(r_0)$$

and for $x \in Z(r)$,

$$\begin{aligned} \|Gx - x\| &\leq r \\ Gx &\in Z(\omega(r)) \end{aligned}$$

then

$$\begin{aligned} \|x_{n+1} - x_n\| &\leq \omega^{(n)}(r_0) \\ \|x_n - x^*\| &\leq \sigma(\omega^{(n)}(r_0)) \end{aligned}$$

Nondiscrete Induction

For the fixed point iteration

$$x_{n+1} = G x_n,$$

if I have

$$x_0 \in Z(r_0)$$

and for $x \in Z(r)$,

$$\begin{aligned} \|Gx - x\| &\leq r \\ Gx &\in Z(\omega(r)) \end{aligned}$$

then

$$\begin{aligned} \|x_n - x^*\| &\leq \sigma(\omega(\|x_n - x_{n-1}\|)) \\ &= \sigma(\|x_n - x_{n-1}\|) - \|x_n - x_{n-1}\| \end{aligned}$$

Newton's Method

$$\omega_{\mathcal{N}}(r) = cr^2$$

Newton's Method

$$\omega_{\mathcal{N}}(r) = \frac{r^2}{2\sqrt{r^2+a^2}}$$
$$\sigma_{\mathcal{N}}(r) = r + \sqrt{r^2+a^2} - a$$

where

$$a = \frac{1}{k_0}\sqrt{1-2k_0 r_0},$$

k_0 is the (scaled) Lipschitz constant for f', and
r_0 is the (scaled) initial residual.

Newton's Method

$$\omega_{\mathcal{N}}(r) = \frac{r^2}{2\sqrt{r^2 + a^2}}$$
$$\sigma_{\mathcal{N}}(r) = r + \sqrt{r^2 + a^2} - a$$

This estimate is *tight* in that the bounds hold with equality for some function f,

$$f(x) = x^2 - a^2$$

using initial guess

$$x_0 = \frac{1}{k_0}.$$

Also, if equality is attained for some n_0, this holds for all $n \geq n_0$.

Newton's Method

$$\omega_{\mathcal{N}}(r) = \frac{r^2}{2\sqrt{r^2 + a^2}}$$
$$\sigma_{\mathcal{N}}(r) = r + \sqrt{r^2 + a^2} - a$$

If $r \gg a$, meaning we have an inaccurate guess,

$$\omega_{\mathcal{N}}(r) \approx \frac{1}{2} r,$$

whereas if $r \ll a$, meaning we are close to the solution,

$$\omega_{\mathcal{N}}(r) \approx \frac{1}{2a} r^2.$$

Left vs. Right

Left:

$$\mathcal{F}(x) \Longrightarrow x - \mathcal{N}(\mathcal{F}, x, b)$$

Right:

$$x \Longrightarrow y = \mathcal{N}(\mathcal{F}, x, b)$$

Heisenberg vs. Schrödinger Picture

Left vs. Right

Left:

$$\mathcal{F}(x) \Longrightarrow x - \mathcal{N}(\mathcal{F}, x, b)$$

Right:

$$x \Longrightarrow y = \mathcal{N}(\mathcal{F}, x, b)$$

Heisenberg vs. Schrödinger Picture

$\mathcal{M} -_R \mathcal{N}$

We start with $x \in Z(r)$, apply $\mathcal{N}$ so that

$$y \in Z(\omega_{\mathcal{N}}(r)),$$

and then apply $\mathcal{M}$ so that

$$x' \in Z(\omega_{\mathcal{M}}(\omega_{\mathcal{N}}(r))).$$

Thus we have

$$\omega_{\mathcal{M} -_R \mathcal{N}} = \omega_{\mathcal{M}} \circ \omega_{\mathcal{N}}$$

Outline

Non-Abelian

$\mathcal{N} -_R$ NRICH

$$\begin{aligned}
\omega_{\mathcal{N}} \circ \omega_{\text{NRICH}} &= \frac{1}{2}\frac{r^2}{\sqrt{r^2 + a^2}} \circ cr, \\
&= \frac{1}{2}\frac{c^2 r^2}{\sqrt{c^2 r^2 + a^2}}, \\
&= \frac{1}{2}\frac{cr^2}{\sqrt{r^2 + (a/c)^2}}, \\
&= \frac{1}{2}c\frac{r^2}{\sqrt{r^2 + \tilde{a}^2}},
\end{aligned}$$

Non-Abelian

$$\mathcal{N} -_R \text{NRICH: } \tfrac{1}{2}c\frac{r^2}{\sqrt{r^2+\tilde{a}^2}}$$

NRICH $-_R \mathcal{N}$

$$\begin{aligned}
\omega_{\text{NRICH}} \circ \omega_{\mathcal{N}} &= cr \circ \frac{1}{2}\frac{r^2}{\sqrt{r^2+a^2}}, \\
&= \frac{1}{2}c\frac{r^2}{\sqrt{r^2+a^2}}, \\
&= \frac{1}{2}c\frac{r^2}{\sqrt{r^2+a^2}}.
\end{aligned}$$

Non-Abelian

$$\mathcal{N} -_R \text{NRICH: } \frac{1}{2} c \frac{r^2}{\sqrt{r^2 + \tilde{a}^2}}$$

$$\text{NRICH} -_R \mathcal{N}\text{: } \frac{1}{2} c \frac{r^2}{\sqrt{r^2 + a^2}}$$

The first method also changes the onset of second order convergence.

Composed Rates of Convergence

Theorem

If ω_1 and ω_2 are convex rates of convergence, then $\omega = \omega_1 \circ \omega_2$ is a rate of convergence.

Composed Rates of Convergence

Theorem

If ω_1 and ω_2 are convex rates of convergence, then $\omega = \omega_1 \circ \omega_2$ is a rate of convergence.

First we show that

$$\omega(s) \leq \frac{s}{r}\omega(r),$$

which means that convex rates of convergence are non-decreasing.

This implies that compositions of convex rates of convergence are also convex and non-decreasing.

Composed Rates of Convergence

Theorem

If ω_1 and ω_2 are convex rates of convergence, then $\omega = \omega_1 \circ \omega_2$ is a rate of convergence.

Then we show that

$$\omega(r) < r \qquad \forall r \in (0, R)$$

by contradiction.

Composed Rates of Convergence

Theorem

If ω_1 and ω_2 are convex rates of convergence, then $\omega = \omega_1 \circ \omega_2$ is a rate of convergence.

This is enough to show that

$$\omega_1(\omega_2(r)) < \omega_1(r),$$

and in fact

$$(\omega_1 \circ \omega_2)^{(n)}(r) < \omega_1^{(n)}(r).$$

Multidimensional Induction Theorem

Preconditions

Theorem

Let

- p *(1 for our case) and* m *(2 for our case) be two positive integers,*
- X *be a complete metric space and* $D \subset X^p$*,*
- $G : D \to X^p$ *and* $F : D \to X^{p+1}$ *be defined by* $Fu = (u, Gu)$*,*
- $F_k = P_k F$*,* $-p + 1 \leq k \leq m$*, the components of* F*,*
- $P = P_m$*,*
- $Z(r) \subset D$ *for each* $r \in T^p$*,*
- ω *be a rate of convergence of type* (p, m) *on* T*,*
- $u_0 \in D$ *and* $r_0 \in T^p$*.*

Multidimensional Induction Theorem

Theorem

If the following conditions hold

$$\begin{aligned} u_0 &\in Z(r_0), \\ PFZ(r) &\subset Z(\tilde{\omega}(r)), \\ \|F_k u - F_{k+1} u\| &\le \omega_k(r), \end{aligned}$$

for all $r \in T^p$, $u \in Z(r)$, *and* $k = 0, \ldots, m-1$, *then*

1. u_0 *is admissible, and* $\exists x^* \in X$ *such that* $(P_k u_n)_{n \ge 0} \to x^*$,
2. *and the following relations hold for* $n > 1$,

$$\begin{aligned} Pu_n &\in Z(\tilde{\omega}(r_0)), \\ \|P_k u_n - P_{k+1} u_n\| &\le \omega_k^{(n)}(r_0), \qquad 0 \le k \le m-1, \\ \|P_k u_n - x^*\| &\le \sigma_k(\tilde{\omega}(r_0)), \qquad 0 \le k \le m; \end{aligned}$$

Multidimensional Induction Theorem

Theorem

If the following conditions hold

$$\begin{aligned} u_0 &\in Z(r_0), \\ PFZ(r) &\subset Z(\tilde{\omega}(r)), \\ \|F_k u - F_{k+1} u\| &\leq \omega_k(r), \end{aligned}$$

for all $r \in T^p$, $u \in Z(r)$, and $k = 0, \ldots, m-1$, then

1. *u_0 is admissible, and $\exists x^* \in X$ such that $(P_k u_n)_{n \geq 0} \to x^*$,*
2. *and the following relations hold for $n > 1$,*

$$\|P_k u_n - x^*\| \leq \sigma_k(r_n), \qquad 0 \leq k \leq m.$$

where $r_n \in T^p$ and $Pu_{n-1} \in Z(r_n)$.

Multidimensional Induction Theorem

Theorem

If the following conditions hold

$$\begin{aligned} u_0 &\in Z(r_0), \\ PFZ(r) &\subset Z(\tilde{\omega}(r)), \\ \|F_k u - F_{k+1} u\| &\leq \omega_k(r), \end{aligned}$$

for all $r \in T^p$, $u \in Z(r)$, *and* $k = 0, \ldots, m-1$, *then*

1. u_0 *is admissible, and* $\exists x^* \in X$ *such that* $(P_k u_n)_{n \geq 0} \to x^*$,
2. *and the following relations hold for* $n > 1$,

$$\begin{aligned} Pu_n &\in Z(\tilde{\omega}(r_0)), \\ \|P_k u_n - P_{k+1} u_n\| &\leq \omega_k^{(n)}(r_0), \qquad 0 \leq k \leq m-1, \\ \|P_k u_n - x^*\| &\leq \sigma_k(\tilde{\omega}(r_0)), \qquad 0 \leq k \leq m; \end{aligned}$$

Multidimensional Induction Theorem

Theorem

If the following conditions hold

$$\begin{aligned} u_0 &\in Z(r_0), \\ PFZ(r) &\subset Z(\omega \circ \psi(r)), \\ \|F_0 u - F_1 u\| &\leq r, \\ \|F_1 u - F_2 u\| &\leq \psi(r), \end{aligned}$$

for all $r \in T^p$, $u \in Z(r)$, *and* $k = 0, \ldots, m-1$, *then*

1. u_0 *is admissible, and* $\exists x^* \in X$ *such that* $(P_k u_n)_{n \geq 0} \to x^*$,
2. *and the following relations hold for* $n > 1$,

$$\begin{aligned} Pu_n &\in Z(\tilde{\omega}(r_0)), \\ \|P_k u_n - P_{k+1} u_n\| &\leq \omega_k^{(n)}(r_0), \qquad 0 \leq k \leq m-1, \\ \|P_k u_n - x^*\| &\leq \sigma_k(\tilde{\omega}(r_0)), \qquad 0 \leq k \leq m; \end{aligned}$$

Composed Newton Methods

Theorem

Suppose that we have two nonlinear solvers

- $\mathcal{M}$, Z_1, ω,
- $\mathcal{N}$, Z_0, ψ,

and consider $\mathcal{M} -_R \mathcal{N}$, *meaning a single step of* $\mathcal{N}$ *for each step of* $\mathcal{M}$.

Concretely, take $\mathcal{M}$ *to be the Newton iteration, and* $\mathcal{N}$ *the Chord method. Then the assumptions of the theorem above are satisfied using* $Z = Z_1$ *and*

$$\omega(r) = \{\psi(r), \omega \circ \psi(r)\},$$

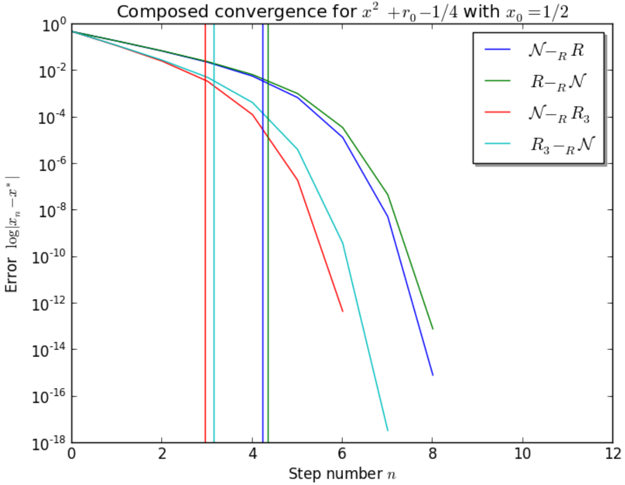
giving us the existence of a solution, and both a priori and a posteriori bounds on the error.

Example

$$f(x) = x^2 + (0.0894427)^2$$

n	$\|x_{n+1} - x_n\|$	$\|x_{n+1} - x_n\| - w^{(n)}(r_0)$	$\|x_n - x^*\| - s(w^{(n)}(r_0))$
0	1.9990e+00	$< 10^{-16}$	$< 10^{-16}$
1	9.9850e-01	$< 10^{-16}$	$< 10^{-16}$
2	4.9726e-01	$< 10^{-16}$	$< 10^{-16}$
3	2.4470e-01	$< 10^{-16}$	$< 10^{-16}$
4	1.1492e-01	$< 10^{-16}$	$< 10^{-16}$
5	4.5342e-02	$< 10^{-16}$	$< 10^{-16}$
6	1.0251e-02	$< 10^{-16}$	$< 10^{-16}$
7	5.8360e-04	$< 10^{-16}$	$< 10^{-16}$
8	1.9039e-06	$< 10^{-16}$	$< 10^{-16}$
9	2.0264e-11	$< 10^{-16}$	$< 10^{-16}$
10	0.0000e+00	$< 10^{-16}$	$< 10^{-16}$

Example



Matrix iterations also 1D scalar once you diagonalize

Pták's nondiscrete induction and its application to matrix iterations, Liesen, IMA J. Num. Anal.,

Outline

Outline

SNES ex16

3D Large Deformation Elasticity

$$\int_\Omega F \cdot S : \nabla v \, d\Omega + \int_\Omega \texttt{loading} \, \mathbf{e}_y \cdot v \, d\Omega = 0 \tag{19}$$

F Deformation gradient

S Second Piola-Kirchhoff tensor

Saint Venant-Kirchhoff model of hyperelasticity

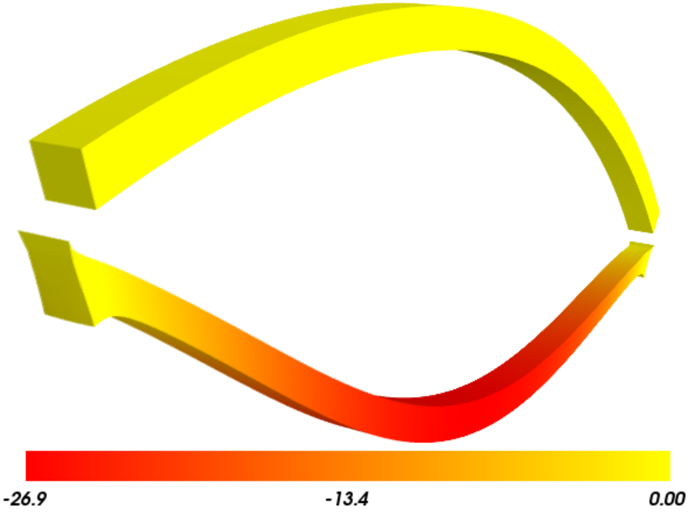
Ω `-arc` *angle* subsection of a cylindrical shell

`-height` *thickness*

`-rad` *inner radius*

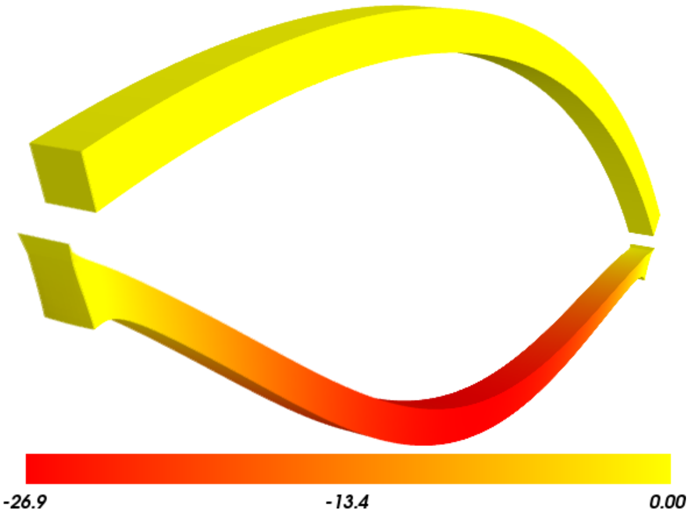
`-width` *width*

Large Deformation Elasticity



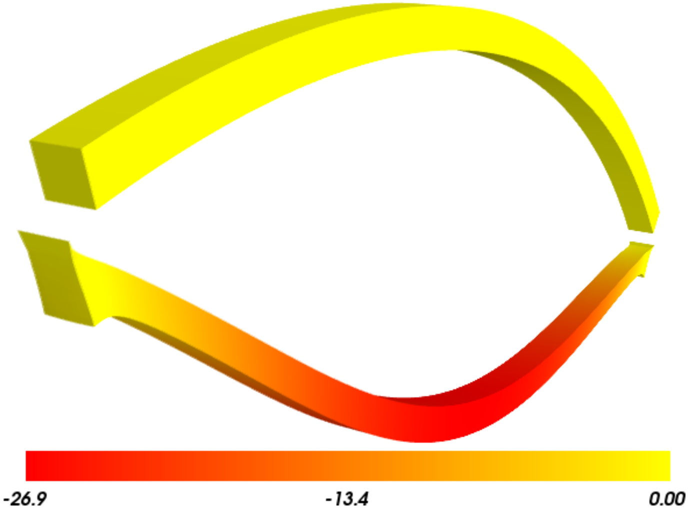
Unstressed and stressed configurations for the elasticity test problem.

Large Deformation Elasticity



Coloration indicates vertical displacement in meters.

Large Deformation Elasticity



P. Wriggers, Nonlinear Finite Element Methods, Springer, 2008.

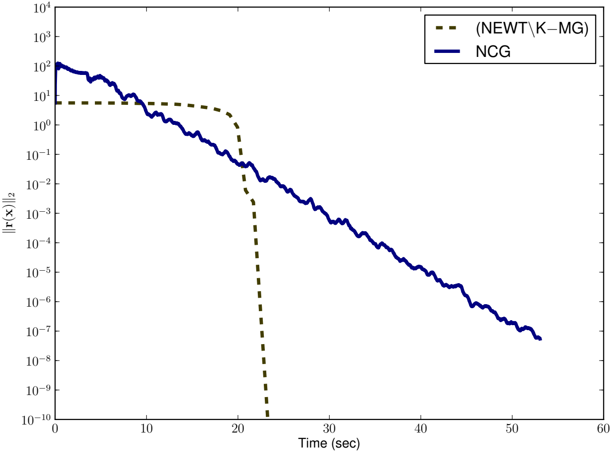
Large Deformation Elasticity
Running

SNES example 16:

```
cd src/snes/examples/tutorials
make ex16
./ex16 -da_grid_x 401 -da_grid_y 9 -da_grid_z 9
  -height 3 -width 3
  -rad 100 -young 100 -poisson 0.2
  -loading -1 -ploading 0
```

Plain SNES Convergence

$(\mathcal{N}\backslash K - MG)$ and NCG

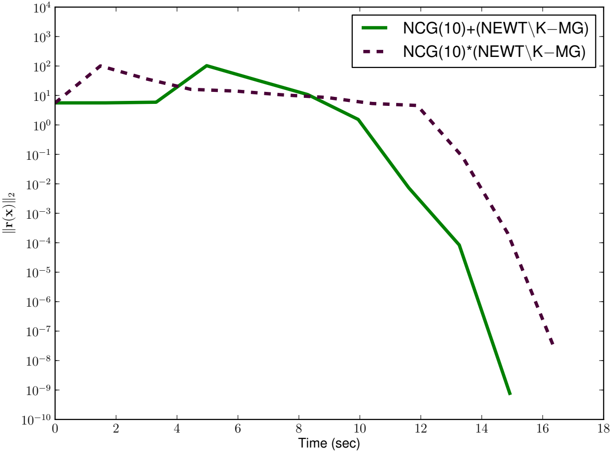


Plain SNES Convergence

Solver	T	N. It	L. It	Func	Jac	PC	NPC
NCG	53.05	4495	0	8991	–	–	–
$(\mathcal{N}\backslash K - MG)$	23.43	27	1556	91	27	1618	–

Composed SNES Convergence

$\mathrm{NCG}(10) + (\mathcal{N}\backslash\mathrm{K} - \mathrm{MG})$ and $\mathrm{NCG}(10) * (\mathcal{N}\backslash\mathrm{K} - \mathrm{MG})$

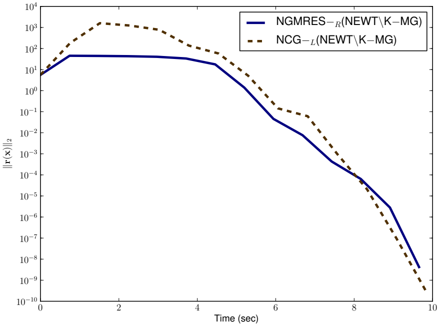


Composed SNES Convergence

Solver	T	N. It	L. It	Func	Jac	PC	NPC
NCG	53.05	4495	0	8991	–	–	–
$(\mathcal{N}\backslash \mathrm{K} - \mathrm{MG})$	23.43	27	1556	91	27	1618	–
NCG(10) $+(\mathcal{N}\backslash \mathrm{K} - \mathrm{MG})$	14.92	9	459	218	9	479	–
NCG(10) $*(\mathcal{N}\backslash \mathrm{K} - \mathrm{MG})$	16.34	11	458	251	11	477	–

Peconditioned SNES Convergence

NGMRES $-_R$ $(\mathcal{N}\backslash\mathrm{K}-\mathrm{MG})$ and NCG $-_L$ $(\mathcal{N}\backslash\mathrm{K}-\mathrm{MG})$



Peconditioned SNES Convergence

Solver	T	N. It	L. It	Func	Jac	PC	NPC
NCG	53.05	4495	0	8991	–	–	–
$(\mathcal{N}\backslash K-MG)$	23.43	27	1556	91	27	1618	–
NCG(10) $+(\mathcal{N}\backslash K-MG)$	14.92	9	459	218	9	479	–
NCG(10) $*(\mathcal{N}\backslash K-MG)$	16.34	11	458	251	11	477	–
NGMRES $-_R(\mathcal{N}\backslash K-MG)$	9.65	13	523	53	13	548	13
NCG $-_L(\mathcal{N}\backslash K-MG)$	9.84	13	529	53	13	554	13

Outline

SNES ex19

Driven Cavity Flow

$$-\Delta \vec{u} + \nabla \times \Omega = 0$$

$$-\Delta \Omega + \nabla \cdot (\vec{u}\Omega) - GR\nabla_x T = 0$$

$$-\Delta T + PR\nabla \cdot (\vec{u} T) = 0$$

SNES ex19

Driven Cavity Flow

$$-\Delta \vec{u} + \nabla \times \Omega = 0$$

$$\nabla \cdot (\vec{u}\Omega) - GR\nabla_x T = 0$$

$$-\Delta T + PR\nabla \cdot (\vec{u}T) = 0$$

Driven Cavity Problem

SNES ex19.c

```
./ex19 -lidvelocity 100 -grashof 1e2
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_monitor_short -snes_converged_reason -snes_view
```

Driven Cavity Problem

SNES ex19.c

```
./ex19 -lidvelocity 100 -grashof 1e2
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_monitor_short -snes_converged_reason -snes_view

lid velocity = 100, prandtl # = 1, grashof # = 100
  0 SNES Function norm 768.116
  1 SNES Function norm 658.288
  2 SNES Function norm 529.404
  3 SNES Function norm 377.51
  4 SNES Function norm 304.723
  5 SNES Function norm 2.59998
  6 SNES Function norm 0.00942733
  7 SNES Function norm 5.20667e-08
Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE iterations 7
```

Driven Cavity Problem

SNES ex19.c

```
./ex19 -lidvelocity 100 -grashof 1e4
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_monitor_short -snes_converged_reason -snes_view
```

Driven Cavity Problem

SNES ex19.c

```
./ex19 -lidvelocity 100 -grashof 1e4
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_monitor_short -snes_converged_reason -snes_view

lid velocity = 100, prandtl # = 1, grashof # = 10000
  0 SNES Function norm 785.404
  1 SNES Function norm 663.055
  2 SNES Function norm 519.583
  3 SNES Function norm 360.87
  4 SNES Function norm 245.893
  5 SNES Function norm 1.8117
  6 SNES Function norm 0.00468828
  7 SNES Function norm 4.417e-08
Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE iterations 7
```

Driven Cavity Problem

SNES ex19.c

```
./ex19 -lidvelocity 100 -grashof 1e5
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_monitor_short -snes_converged_reason -snes_view
```

Driven Cavity Problem

SNES ex19.c

```
./ex19 -lidvelocity 100 -grashof 1e5
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_monitor_short -snes_converged_reason -snes_view

lid velocity = 100, prandtl # = 1, grashof # = 100000
  0 SNES Function norm 1809.96
Nonlinear solve did not converge due to DIVERGED_LINEAR_SOLVE iterations 0
```

Driven Cavity Problem

SNES ex19.c

```
./ex19 -lidvelocity 100 -grashof 1e5
  -da_grid_x 16 -da_grid_y 16 -da_refine 2 -pc_type lu
  -snes_monitor_short -snes_converged_reason -snes_view

lid velocity = 100, prandtl # = 1, grashof # = 100000
  0 SNES Function norm 1809.96
  1 SNES Function norm 1678.37
  2 SNES Function norm 1643.76
  3 SNES Function norm 1559.34
  4 SNES Function norm 1557.6
  5 SNES Function norm 1510.71
  6 SNES Function norm 1500.47
  7 SNES Function norm 1498.93
  8 SNES Function norm 1498.44
  9 SNES Function norm 1498.27
 10 SNES Function norm 1498.18
 11 SNES Function norm 1498.12
 12 SNES Function norm 1498.11
 13 SNES Function norm 1498.11
 14 SNES Function norm 1498.11
 ...
```

Nonlinear Preconditioning

```
./ex19 -lidvelocity 100 -grashof 5e4 -da_refine 4 -snes_monitor_short
 -snes_type newtonls -snes_converged_reason
 -pc_type lu
```

```
lid velocity = 100, prandtl # = 1, grashof # = 50000
  0 SNES Function norm 1228.95
  1 SNES Function norm 1132.29
  2 SNES Function norm 1026.17
  3 SNES Function norm 925.717
  4 SNES Function norm 924.778
  5 SNES Function norm 836.867
  ⋮
 21 SNES Function norm 585.143
 22 SNES Function norm 585.142
 23 SNES Function norm 585.142
 24 SNES Function norm 585.142
  ⋮
```

Nonlinear Preconditioning

```
./ex19 -lidvelocity 100 -grashof 5e4 -da_refine 4 -snes_monitor_short
 -snes_type fas -snes_converged_reason
 -fas_levels_snes_type gs -fas_levels_snes_max_it 6
```

```
lid velocity = 100, prandtl # = 1, grashof # = 50000
  0 SNES Function norm 1228.95
  1 SNES Function norm 574.793
  2 SNES Function norm 513.02
  3 SNES Function norm 216.721
  4 SNES Function norm 85.949
Nonlinear solve did not converge due to DIVERGED_INNER iterations 4
```

Nonlinear Preconditioning

```
./ex19 -lidvelocity 100 -grashof 5e4 -da_refine 4 -snes_monitor_short
 -snes_type fas -snes_converged_reason
 -fas_levels_snes_type gs -fas_levels_snes_max_it 6
  -fas_coarse_snes_converged_reason
```

```
lid velocity = 100, prandtl # = 1, grashof # = 50000
  0 SNES Function norm 1228.95
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 12
  1 SNES Function norm 574.793
    Nonlinear solve did not converge due to DIVERGED_MAX_IT its 50
  2 SNES Function norm 513.02
    Nonlinear solve did not converge due to DIVERGED_MAX_IT its 50
  3 SNES Function norm 216.721
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 22
  4 SNES Function norm 85.949
    Nonlinear solve did not converge due to DIVERGED_LINE_SEARCH its 42
Nonlinear solve did not converge due to DIVERGED_INNER iterations 4
```

Nonlinear Preconditioning

```
./ex19 -lidvelocity 100 -grashof 5e4 -da_refine 4 -snes_monitor_short
 -snes_type fas -snes_converged_reason
 -fas_levels_snes_type gs -fas_levels_snes_max_it 6
  -fas_coarse_snes_linesearch_type basic
  -fas_coarse_snes_converged_reason
```

```
lid velocity = 100, prandtl # = 1, grashof # = 50000
  0 SNES Function norm 1228.95
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 6
  ⋮
 47 SNES Function norm 78.8401
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 5
 48 SNES Function norm 73.1185
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 6
 49 SNES Function norm 78.834
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 5
 50 SNES Function norm 73.1176
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 6
  ⋮
```

Nonlinear Preconditioning

```
./ex19 -lidvelocity 100 -grashof 5e4 -da_refine 4 -snes_monitor_short
 -snes_type nrichardson -npc_snes_max_it 1 -snes_converged_reason
 -npc_snes_type fas -npc_fas_coarse_snes_converged_reason
  -npc_fas_levels_snes_type gs -npc_fas_levels_snes_max_it 6
  -npc_fas_coarse_snes_linesearch_type basic
```

```
lid velocity = 100, prandtl # = 1, grashof # = 50000
  0 SNES Function norm 1228.95
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 6
  1 SNES Function norm 552.271
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 27
  2 SNES Function norm 173.45
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 45
  ⋮
 43 SNES Function norm 3.45407e-05
    Nonlinear solve converged due to CONVERGED_SNORM_RELATIVE its 2
 44 SNES Function norm 1.6141e-05
    Nonlinear solve converged due to CONVERGED_SNORM_RELATIVE its 2
 45 SNES Function norm 9.13386e-06
Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE iterations 45
```

Nonlinear Preconditioning

```
./ex19 -lidvelocity 100 -grashof 5e4 -da_refine 4 -snes_monitor_short
 -snes_type ngmres -npc_snes_max_it 1 -snes_converged_reason
 -npc_snes_type fas -npc_fas_coarse_snes_converged_reason
  -npc_fas_levels_snes_type gs -npc_fas_levels_snes_max_it 6
  -npc_fas_coarse_snes_linesearch_type basic
```

```
lid velocity = 100, prandtl # = 1, grashof # = 50000
  0 SNES Function norm 1228.95
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 6
  1 SNES Function norm 538.605
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 13
  2 SNES Function norm 178.005
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 24
  ⋮
 27 SNES Function norm 0.000102487
    Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 2
 28 SNES Function norm 4.2744e-05
    Nonlinear solve converged due to CONVERGED_SNORM_RELATIVE its 2
 29 SNES Function norm 1.01621e-05
Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE iterations 29
```

Nonlinear Preconditioning

```
./ex19 -lidvelocity 100 -grashof 5e4 -da_refine 4 -snes_monitor_short
 -snes_type ngmres -npc_snes_max_it 1 -snes_converged_reason
 -npc_snes_type fas -npc_fas_coarse_snes_converged_reason
 -npc_fas_levels_snes_type newtonls -npc_fas_levels_snes_max_it 6
  -npc_fas_levels_snes_linesearch_type basic
  -npc_fas_levels_snes_max_linear_solve_fail 30
  -npc_fas_levels_ksp_max_it 20 -npc_fas_levels_snes_converged_reason
  -npc_fas_coarse_snes_linesearch_type basic
lid velocity = 100, prandtl # = 1, grashof # = 50000
  0 SNES Function norm 1228.95
    Nonlinear solve did not converge due to DIVERGED_MAX_IT its 6
    ⋮
        Nonlinear solve converged due to CONVERGED_SNORM_RELATIVE its 1
    ⋮
  1 SNES Function norm 0.1935
  2 SNES Function norm 0.0179938
  3 SNES Function norm 0.00223698
  4 SNES Function norm 0.000190461
  5 SNES Function norm 1.6946e-06
Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE iterations 5
```

Nonlinear Preconditioning

```
./ex19 -lidvelocity 100 -grashof 5e4 -da_refine 4 -snes_monitor_short
 -snes_type composite -snes_composite_type additiveoptimal
 -snes_composite_sneses fas,newtonls -snes_converged_reason
 -sub_0_fas_levels_snes_type gs -sub_0_fas_levels_snes_max_it 6
   -sub_0_fas_coarse_snes_linesearch_type basic
 -sub_1_snes_linesearch_type basic -sub_1_pc_type mg
```

```
lid velocity = 100, prandtl # = 1, grashof # = 50000
  0 SNES Function norm 1228.95
  1 SNES Function norm 541.462
  2 SNES Function norm 162.92
  3 SNES Function norm 48.8138
  4 SNES Function norm 11.1822
  5 SNES Function norm 0.181469
  6 SNES Function norm 0.00170909
  7 SNES Function norm 3.24991e-08
Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE iterations 7
```

Nonlinear Preconditioning

```
./ex19 -lidvelocity 100 -grashof 5e4 -da_refine 4 -snes_monitor_short
 -snes_type composite -snes_composite_type multiplicative
 -snes_composite_sneses fas,newtonls -snes_converged_reason
 -sub_0_fas_levels_snes_type gs -sub_0_fas_levels_snes_max_it 6
   -sub_0_fas_coarse_snes_linesearch_type basic
 -sub_1_snes_linesearch_type basic -sub_1_pc_type mg
```

```
lid velocity = 100, prandtl # = 1, grashof # = 50000
  0 SNES Function norm 1228.95
  1 SNES Function norm 544.404
  2 SNES Function norm 18.2513
  3 SNES Function norm 0.488689
  4 SNES Function norm 0.000108712
  5 SNES Function norm 5.68497e-08
Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE iterations 5
```

Nonlinear Preconditioning

Solver	T	N. It	L. It	Func	Jac	PC	NPC
$(\mathcal{N}\backslash K - MG)$	9.83	17	352	34	85	370	–
NGMRES $-_R$ $(\mathcal{N}\backslash K - MG)$	7.48	10	220	21	50	231	10
FAS	6.23	162	0	2382	377	754	–
FAS $+ (\mathcal{N}\backslash K - MG)$	8.07	10	197	232	90	288	–
FAS $* (\mathcal{N}\backslash K - MG)$	4.01	5	80	103	45	125	–
NRICH $-_L$ FAS	3.20	50	0	1180	192	384	50
NGMRES $-_R$ FAS	1.91	24	0	447	83	166	24

Nonlinear Preconditioning

See discussion in:

Composing Scalable Nonlinear Algebraic Solvers,
Peter Brune, Matthew Knepley, Barry Smith, and Xuemin Tu,

SIAM Review, **57**(4), 535–565, 2015.

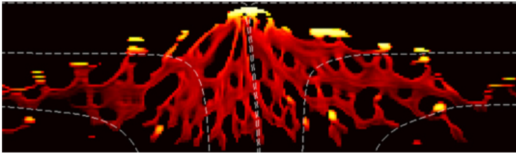
http://www.mcs.anl.gov/uploads/cels/papers/P2010-0112.pdf

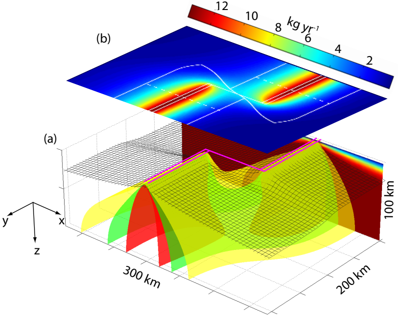
Outline

3 Experiments
- Composition
- Multilevel
- Magma Dynamics

Magma Dynamics

- Couples scales
  - Subduction
  - Magma Migration
- Physics
  - Incompressible fluid
  - Porous solid
  - Variable porosity
- Deforming matrix
  - Compaction pressure
- Code generation
  - FEniCS
- Multiphysics Preconditioning
  - PETSc FieldSplit

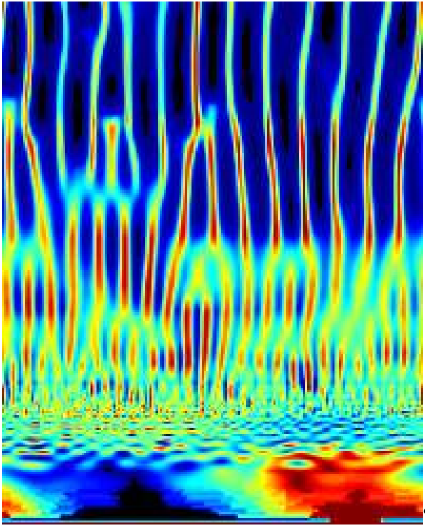




[a] Katz

Magma Dynamics

- Couples scales
  - Subduction
  - Magma Migration
- Physics
  - Incompressible fluid
  - Porous solid
  - Variable porosity
- Deforming matrix
  - Compaction pressure
- Code generation
  - FEniCS
- Multiphysics Preconditioning
  - PETSc FieldSplit

[a]

[a] Katz, Speigelman

Dimensional Formulation

$$\nabla p - \nabla \zeta_\phi (\nabla \cdot \vec{v}^S) - \nabla \cdot \left(2\eta_\phi \dot{\epsilon}^S \right) = 0$$

$$\nabla \cdot \left(-\frac{K_\phi}{\mu} \nabla p + \vec{v}^S \right) = 0$$

$$\frac{\partial \phi}{\partial t} - \nabla \cdot (1 - \phi) \vec{v}^S = 0$$

Closure Conditions

$$K_\phi = K_0 \left(\frac{\phi}{\phi_0} \right)^n$$

$$\eta_\phi = \eta_0 \exp\left(-\lambda(\phi - \phi_0)\right)$$

$$\zeta_\phi = \zeta_0 \left(\frac{\phi}{\phi_0} \right)^{-m}$$

Nondimensional Formulation

$$\nabla p - \nabla \left(\left(\frac{\phi}{\phi_0} \right)^{-m} \nabla \cdot \vec{v}^S \right) - \nabla \cdot \left(2 e^{-\lambda(\phi - \phi_0)} \dot{\epsilon}^S \right) = 0$$

$$\nabla \cdot \left(- \frac{R^2}{r_\zeta + 4/3} \left(\frac{\phi}{\phi_0} \right)^n \nabla p + \vec{v}^S \right) = 0$$

$$\frac{\partial \phi}{\partial t} - \nabla \cdot (1 - \phi) \vec{v}^S = 0$$

Initial and Boundary conditions

Initially

$$\phi = \phi_0 + A\cos(\vec{k} \cdot \vec{x})$$

where

$$A \ll \phi_0$$

and on the top and bottom boundary

$$K_\phi \nabla p \cdot \hat{n} = 0$$

$$\vec{v}^S = \pm\frac{\dot{\gamma}}{2}\hat{x}$$

Newton options

```
-snes_monitor -snes_converged_reason
-snes_type newtonls -snes_linesearch_type bt
-snes_fd_color -snes_fd_color_use_mat -mat_coloring_type greedy
-ksp_rtol 1.0e-10 -ksp_monitor -ksp_gmres_restart 200
-pc_type fieldsplit
 -pc_fieldsplit_0_fields 0,2 -pc_fieldsplit_1_fields 1
 -pc_fieldsplit_type schur -pc_fieldsplit_schur_precondition selfp
  -pc_fieldsplit_schur_factorization_type full
   -fieldsplit_0_pc_type lu
   -fieldsplit_pressure_ksp_rtol 1.0e-9 -fieldsplit_pressure_pc_type gamg
    -fieldsplit_pressure_ksp_monitor
    -fieldsplit_pressure_ksp_gmres_restart 100
    -fieldsplit_pressure_ksp_max_it 200
```

Newton options

Separate porosity

```
-pc_type fieldsplit
 -pc_fieldsplit_0_fields 0,1 -pc_fieldsplit_1_fields 2
 -pc_fieldsplit_type multiplicative
   -fieldsplit_0_pc_type fieldsplit
   -fieldsplit_0_pc_fieldsplit_type schur
   -fieldsplit_0_pc_fieldsplit_schur_precondition selfp
   -fieldsplit_0_pc_fieldsplit_schur_factorization_type full
   -fieldsplit_0_fieldsplit_velocity_pc_type lu
   -fieldsplit_0_fieldsplit_pressure_ksp_rtol 1.0e-9
   -fieldsplit_0_fieldsplit_pressure_pc_type gamg
    -fieldsplit_0_fieldsplit_pressure_ksp_monitor
    -fieldsplit_0_fieldsplit_pressure_ksp_gmres_restart 100
    -fieldsplit_fieldsplit_0_pressure_ksp_max_it 200
```

Early Newton convergence

```
0 TS dt 0.01 time 0
    0 SNES Function norm 5.292194079127e-03
      Linear pressure_ solve converged due to CONVERGED_RTOL its 10
      0 KSP Residual norm 4.618093146920e+00
      Linear pressure_ solve converged due to CONVERGED_RTOL its 10
      1 KSP Residual norm 3.018153330707e-03
      Linear pressure_ solve converged due to CONVERGED_RTOL its 11
      2 KSP Residual norm 4.274869628519e-13
    Linear solve converged due to CONVERGED_RTOL its 2
    1 SNES Function norm 2.766906985362e-06
      Linear pressure_ solve converged due to CONVERGED_RTOL its 8
      0 KSP Residual norm 2.555890235972e-02
      Linear pressure_ solve converged due to CONVERGED_RTOL its 8
      1 KSP Residual norm 1.638293944976e-07
      Linear pressure_ solve converged due to CONVERGED_RTOL its 8
      2 KSP Residual norm 1.771928779400e-14
    Linear solve converged due to CONVERGED_RTOL its 2
    2 SNES Function norm 1.188754322734e-11
  Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 2
1 TS dt 0.01 time 0.01
```

Later Newton convergence

```
0 TS dt 0.01 time 0.63
    0 SNES Function norm 9.366565251786e-03
      Linear pressure_ solve converged due to CONVERGED_RTOL its 16
      Linear pressure_ solve converged due to CONVERGED_RTOL its 16
      Linear pressure_ solve converged due to CONVERGED_RTOL its 16
    Linear solve converged due to CONVERGED_RTOL its 2
    1 SNES Function norm 4.492625910272e-03
    Linear solve converged due to CONVERGED_RTOL its 2
    2 SNES Function norm 3.666181450068e-03
    Linear solve converged due to CONVERGED_RTOL its 2
    3 SNES Function norm 2.523116582272e-03
    Linear solve converged due to CONVERGED_RTOL its 2
    4 SNES Function norm 3.022638159491e-04
    Linear solve converged due to CONVERGED_RTOL its 2
    5 SNES Function norm 9.761317324448e-06
    Linear solve converged due to CONVERGED_RTOL its 2
    6 SNES Function norm 1.147944474432e-08
    Linear solve converged due to CONVERGED_RTOL its 2
    7 SNES Function norm 8.729160299009e-14
  Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE its 7
1 TS dt 0.01 time 0.64
```

Newton failure

```
0 TS dt 0.01 time 0.64
Time 0.64 L_2 Error: 0.494811 [0.0413666, 0.491642, 0.0376071]
    0 SNES Function norm 9.682733054059e-03
    Linear solve converged due to CONVERGED_RTOL iterations 2
    1 SNES Function norm 6.841434267123e-03
    Linear solve converged due to CONVERGED_RTOL iterations 3
    2 SNES Function norm 4.412420553822e-03
    Linear solve converged due to CONVERGED_RTOL iterations 5
    3 SNES Function norm 3.309326919835e-03
    Linear solve converged due to CONVERGED_RTOL iterations 6
    4 SNES Function norm 3.022494350289e-03
    Linear solve converged due to CONVERGED_RTOL iterations 7
    5 SNES Function norm 2.941050948582e-03
    Linear solve converged due to CONVERGED_RTOL iterations 7
   ⋮
    9 SNES Function norm 2.631941422878e-03
    Linear solve converged due to CONVERGED_RTOL iterations 7
   10 SNES Function norm 2.631897334054e-03
    Linear solve converged due to CONVERGED_RTOL iterations 10
   11 SNES Function norm 2.631451174722e-03
    Linear solve converged due to CONVERGED_RTOL iterations 15
   ⋮
```

NCG+Newton options

```
-snes_monitor -snes_converged_reason
-snes_type composite -snes_composite_type multiplicative
 -snes_composite_sneses ncg,newtonls
  -sub_0_snes_monitor -sub_1_snes_monitor
  -sub_0_snes_type ncg -sub_0_snes_linesearch_type cp
   -sub_0_snes_max_it 5
  -sub_1_snes_linesearch_type bt -sub_1_snes_fd_color
   -sub_1_snes_fd_color_use_mat -mat_coloring_type greedy
  -sub_1_ksp_rtol 1.0e-10 -sub_1_ksp_monitor -sub_1_ksp_gmres_restart 200
  -sub_1_pc_type fieldsplit -sub_1_pc_fieldsplit_0_fields 0,2
   -sub_1_pc_fieldsplit_1_fields 1
  -sub_1_pc_fieldsplit_type schur
   -sub_1_pc_fieldsplit_schur_precondition selfp
   -sub_1_pc_fieldsplit_schur_factorization_type full
    -sub_1_fieldsplit_0_pc_type lu
    -sub_1_fieldsplit_pressure_ksp_rtol 1.0e-9
     -sub_1_fieldsplit_pressure_pc_type gamg
     -sub_1_fieldsplit_pressure_ksp_gmres_restart 100
     -sub_1_fieldsplit_pressure_ksp_max_it 200
```

NCG+Newton convergence

```
0 TS dt 0.01 time 0.64
    0 SNES Function norm 9.682733054059e-03
      0 SNES Function norm 9.682733054059e-03
      1 SNES Function norm 3.705698943518e-02
      2 SNES Function norm 4.981898384331e-02
      3 SNES Function norm 5.710183285964e-02
      4 SNES Function norm 5.476973798534e-02
      5 SNES Function norm 6.464724668855e-02
      0 SNES Function norm 6.464724668855e-02
        0 KSP Residual norm 1.021155502263e+00
        1 KSP Residual norm 9.145207488003e-05
        2 KSP Residual norm 3.899752904206e-09
        3 KSP Residual norm 1.001750831581e-12
      1 SNES Function norm 8.940296814443e-03
    1 SNES Function norm 8.940296814443e-03
    2 SNES Function norm 4.290429277269e-02
    3 SNES Function norm 1.154466745956e-02
    4 SNES Function norm 2.938816182982e-03
    5 SNES Function norm 4.148507767082e-04
    6 SNES Function norm 1.892807106900e-05
    7 SNES Function norm 4.912654244547e-08
    8 SNES Function norm 3.851626525260e-13
1 TS dt 0.01 time 0.65
```

FAS options

Top level

```
-snes_monitor -snes_converged_reason
-snes_type fas -snes_fas_type full -snes_fas_levels 4
 -fas_levels_3_snes_monitor -fas_levels_3_snes_converged_reason
  -fas_levels_3_snes_atol 1.0e-9 -fas_levels_3_snes_max_it 2
  -fas_levels_3_snes_type newtonls -fas_levels_3_snes_linesearch_type bt
  -fas_levels_3_snes_fd_color -fas_levels_3_snes_fd_color_use_mat
  -fas_levels_3_ksp_rtol 1.0e-10 -mat_coloring_type greedy
  -fas_levels_3_ksp_gmres_restart 50 -fas_levels_3_ksp_max_it 200
   -fas_levels_3_pc_type fieldsplit
    -fas_levels_3_pc_fieldsplit_0_fields 0,2
    -fas_levels_3_pc_fieldsplit_1_fields 1
    -fas_levels_3_pc_fieldsplit_type schur
     -fas_levels_3_pc_fieldsplit_schur_precondition selfp
     -fas_levels_3_pc_fieldsplit_schur_factorization_type full
      -fas_levels_3_fieldsplit_0_pc_type lu
      -fas_levels_3_fieldsplit_pressure_ksp_rtol 1.0e-9
       -fas_levels_3_fieldsplit_pressure_pc_type gamg
       -fas_levels_3_fieldsplit_pressure_ksp_gmres_restart 100
       -fas_levels_3_fieldsplit_pressure_ksp_max_it 200
```

FAS options

2nd level

```
-fas_levels_2_snes_monitor -fas_levels_2_snes_converged_reason
 -fas_levels_2_snes_atol 1.0e-9 -fas_levels_2_snes_max_it 2
 -fas_levels_2_snes_type newtonls -fas_levels_2_snes_linesearch_type bt
 -fas_levels_2_snes_fd_color -fas_levels_2_snes_fd_color_use_mat
 -fas_levels_2_ksp_rtol 1.0e-10 -fas_levels_2_ksp_gmres_restart 50
 -fas_levels_2_pc_type fieldsplit
  -fas_levels_2_pc_fieldsplit_0_fields 0,2
  -fas_levels_2_pc_fieldsplit_1_fields 1
  -fas_levels_2_pc_fieldsplit_type schur
   -fas_levels_2_pc_fieldsplit_schur_precondition selfp
   -fas_levels_2_pc_fieldsplit_schur_factorization_type full
    -fas_levels_2_fieldsplit_0_pc_type lu
    -fas_levels_2_fieldsplit_pressure_ksp_rtol 1.0e-9
     -fas_levels_2_fieldsplit_pressure_pc_type gamg
     -fas_levels_2_fieldsplit_pressure_ksp_gmres_restart 100
     -fas_levels_2_fieldsplit_pressure_ksp_max_it 200
```

FAS options

1st level

```
-fas_levels_1_snes_monitor -fas_levels_1_snes_converged_reason
 -fas_levels_1_snes_atol 1.0e-9
 -fas_levels_1_snes_type newtonls -fas_levels_1_snes_linesearch_type bt
 -fas_levels_1_snes_fd_color -fas_levels_1_snes_fd_color_use_mat
 -fas_levels_1_ksp_rtol 1.0e-10 -fas_levels_1_ksp_gmres_restart 50
 -fas_levels_1_pc_type fieldsplit
  -fas_levels_1_pc_fieldsplit_0_fields 0,2
  -fas_levels_1_pc_fieldsplit_1_fields 1
  -fas_levels_1_pc_fieldsplit_type schur
   -fas_levels_1_pc_fieldsplit_schur_precondition selfp
   -fas_levels_1_pc_fieldsplit_schur_factorization_type full
    -fas_levels_1_fieldsplit_0_pc_type lu
    -fas_levels_1_fieldsplit_pressure_ksp_rtol 1.0e-9
    -fas_levels_1_fieldsplit_pressure_pc_type gamg
```

FAS options

Coarse level

```
-fas_coarse_snes_monitor -fas_coarse_snes_converged_reason
 -fas_coarse_snes_atol 1.0e-9
 -fas_coarse_snes_type newtonls -fas_coarse_snes_linesearch_type bt
 -fas_coarse_snes_fd_color -fas_coarse_snes_fd_color_use_mat
 -fas_coarse_ksp_rtol 1.0e-10 -fas_coarse_ksp_gmres_restart 50
 -fas_coarse_pc_type fieldsplit
  -fas_coarse_pc_fieldsplit_0_fields 0,2
  -fas_coarse_pc_fieldsplit_1_fields 1
  -fas_coarse_pc_fieldsplit_type schur
   -fas_coarse_pc_fieldsplit_schur_precondition selfp
   -fas_coarse_pc_fieldsplit_schur_factorization_type full
    -fas_coarse_fieldsplit_0_pc_type lu
    -fas_coarse_fieldsplit_pressure_ksp_rtol 1.0e-9
    -fas_coarse_fieldsplit_pressure_pc_type gamg
```

FAS convergence

```
0 TS dt 0.01 time 0.64
    0 SNES Function norm 9.682733054059e-03
      2 SNES Function norm 4.412420553822e-03
            2 SNES Function norm 8.022096211721e-15
          1 SNES Function norm 2.773743832538e-04
            1 SNES Function norm 5.627093528843e-11
          1 SNES Function norm 4.405884464849e-10
        2 SNES Function norm 8.985059910030e-08
          1 SNES Function norm 4.672651281994e-15
            0 SNES Function norm 3.160322858961e-15
          0 SNES Function norm 4.672651281994e-15
        1 SNES Function norm 1.046571008046e-14
      2 SNES Function norm 1.804845173803e-02
        2 SNES Function norm 2.776600115290e-12
          0 SNES Function norm 1.354009326059e-12
            0 SNES Function norm 5.881604627760e-13
          0 SNES Function norm 1.354011456281e-12
        0 SNES Function norm 2.776600115290e-12
      2 SNES Function norm 9.640723411562e-05
    1 SNES Function norm 9.640723411562e-05
    2 SNES Function norm 1.057876040732e-08
    3 SNES Function norm 5.623618219189e-11
1 TS dt 0.01 time 0.65
```

Outline

User Solve

```
MPI_Comm comm;
SNES snes;
DM dm;
Vec u;

SNESCreate(comm, &snes);
SNESSetDM(snes, dm);
SNESSetFromOptions(snes);
DMCreateGlobalVector(dm, &u);
SNESSolve(snes, NULL, u);
```

Nonlinear Solver Organization

- Hierarchical
  - MG (FAS)
  - DD (Nonlinear FETI-DP)
- Robust
  - Line Search/Trust Region
  - Krylov (NGMRES)
  - Nonlinear ASM
  - Direct (Homotopy, Gröbner Basis)

What is Missing?

Closing Admonition

Never believe *anything*,

unless you can run it.

Closing Admonition

Never believe *anything*,

unless you can run it.

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_factorization_type full
-mg_levels_pc_fieldsplit_schur_precondition user
-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

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

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

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

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

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_factorization_type full
-mg_levels_pc_fieldsplit_schur_precondition diag
```

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
```

Schur complement action: Use only the lower diagonal part of A00

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

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_factorization_type full
-mg_levels_pc_fieldsplit_schur_precondition diag
```

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
```

Schur complement action: Use only the lower diagonal part of A00

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

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_factorization_type full
-mg_levels_pc_fieldsplit_schur_precondition diag
```

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
```

Schur complement action: Use only the lower diagonal part of A00

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

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_factorization_type full
-mg_levels_pc_fieldsplit_schur_precondition diag
```

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
```

Schur complement action: Use only the lower diagonal part of A00

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