

# Nonlinear Preconditioning in PETSc

Matthew Knepley  $\in$  PETSc Team

Computation Institute  
University of Chicago

Challenges in 21st Century Experimental Mathematical  
Computation

ICERM, Providence, RI      July 22, 2014



# The PETSc Team



Bill Gropp



Barry Smith



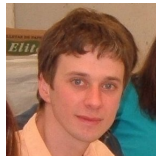
Satish Balay



Jed Brown



Matt Knepley



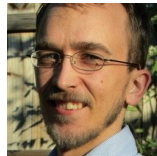
Lisandro Dalcin



Hong Zhang



Mark Adams



Toby Issac

# Why Experiment with Solvers?

- Asymptotic convergence rate
  - Rarely get asymptotics in practice
  - Really want the constant
- Manner of convergence
  - Stationary iterative methods decrease high frequency error quickly
  - Tuminaro, Walker, Shadid, JCP, 180, pp. 549-558 (2002).
- Robustness
  - Line search
  - Solver composition

# Why Experiment with Solvers?

- Asymptotic convergence rate
  - Rarely get asymptotics in practice
  - Really want the constant
- Manner of convergence
  - Stationary iterative methods decrease high frequency error quickly
  - [Tuminaro, Walker, Shadid, JCP, 180, pp. 549-558 \(2002\).](#)
- Robustness
  - Line search
  - Solver composition

# Why Experiment with Solvers?

- Asymptotic convergence rate
  - Rarely get asymptotics in practice
  - Really want the constant
- Manner of convergence
  - Stationary iterative methods decrease high frequency error quickly
  - [Tuminaro, Walker, Shadid, JCP, 180, pp. 549-558 \(2002\).](#)
- Robustness
  - Line search
  - Solver composition

# Outline

- 1 Algorithmics
- 2 Experiments

# Abstract System

Out prototypical nonlinear equation is:

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

and we define the residual as

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

# Abstract System

Out prototypical nonlinear equation is:

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

and we define the (linear) residual as

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



# Linear Left Preconditioning

The modified equation becomes

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

# Linear Left Preconditioning

The modified defect correction equation becomes

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

# Additive Combination

The linear iteration

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

becomes the nonlinear iteration

# Additive Combination

The linear iteration

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

becomes the nonlinear iteration

# Additive Combination

The linear iteration

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

becomes the nonlinear iteration

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

# Nonlinear Left Preconditioning

From the additive combination, we have

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

so we define the preconditioning operation as

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

# Multiplicative Combination

The linear iteration

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

becomes the nonlinear iteration

# Multiplicative Combination

The linear iteration

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

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

becomes the nonlinear iteration



# Multiplicative Combination

The linear iteration

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

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

becomes the nonlinear iteration

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

# Nonlinear Right Preconditioning

For the linear case, we have

$$AP^{-1}\vec{y} = \vec{b} \quad (15)$$

$$\vec{x} = P^{-1}\vec{y} \quad (16)$$

so we define the preconditioning operation as

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

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

# 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.	$\setminus$	$\vec{y} = \vec{J}(\vec{x})^{-1} \vec{r}(\vec{x}) = \mathbf{K}(\vec{J}(\vec{x}), \vec{y}_0, \vec{b})$	$\mathcal{N} \setminus \mathbf{K}$

# Nonlinear Richardson

```
1: procedure NRICH( $\vec{F}$ ,  $\vec{x}_i$ ,  $\vec{b}$ )  
2:    $\vec{d} = -\vec{r}(\vec{x}_i)$   
3:    $\vec{x}_{i+1} = \vec{x}_i + \lambda\vec{d}$   
4: end procedure  
5: return  $\vec{x}_{i+1}$ 
```

▷  $\lambda$  determined by line search

L Adds line search to  $\mathcal{N}$

R Uses  $\mathcal{N}$  to improve search direction

# Nonlinear Richardson

```
1: procedure NRICH( $\vec{F}$ ,  $\vec{x}_i$ ,  $\vec{b}$ )  
2:    $\vec{d} = -\vec{r}(\vec{x}_i)$   
3:    $\vec{x}_{i+1} = \vec{x}_i + \lambda\vec{d}$   
4: end procedure  
5: return  $\vec{x}_{i+1}$ 
```

▷  $\lambda$  determined by line search

L Adds line search to  $\mathcal{N}$

R Uses  $\mathcal{N}$  to improve search direction

# Line Search

Equivalent to  $\text{NRICH} -_L \mathcal{N}$ :

$\text{NRICH} -_L \mathcal{N}$

# Line Search

Equivalent to  $\text{NRICH} -_L \mathcal{N}$ :

$\text{NRICH} -_L \mathcal{N}$

$\text{NRICH}(\vec{x} - \mathcal{N}(\mathcal{F}, \vec{x}, \vec{b}), \vec{x}, \vec{b})$

# Line Search

Equivalent to NRICH  $_{-L} \mathcal{N}$ :

NRICH  $_{-L} \mathcal{N}$

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

$$\vec{x}_{i+1} = \vec{x}_i - \lambda \vec{r}_L$$



# Line Search

Equivalent to NRICH  $_{-L} \mathcal{N}$ :

NRICH  $_{-L} \mathcal{N}$

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

$$\vec{x}_{i+1} = \vec{x}_i - \lambda \vec{r}_L$$

$$\vec{x}_{i+1} = \vec{x}_i + \lambda(\mathcal{N}(\mathcal{F}, \vec{x}_i, \vec{b}) - \vec{x}_i)$$

# Line Search

Equivalent to NRICH  ${}_{-L} \mathcal{N}$ :

$$\begin{aligned} & \text{NRICH } {}_{-L} \mathcal{N} \\ & \text{NRICH}(\vec{x} - \mathcal{N}(\mathcal{F}, \vec{x}, \vec{b}), \vec{x}, \vec{b}) \\ & \vec{x}_{i+1} = \vec{x}_i - \lambda \vec{r}_L \\ & \vec{x}_{i+1} = \vec{x}_i + \lambda(\mathcal{N}(\mathcal{F}, \vec{x}_i, \vec{b}) - \vec{x}_i) \end{aligned}$$

Let  $R_1$  be Richardson iteration with a unit step scaling (no damping). Then we have

$$\mathcal{M} {}_{-L} \mathbb{R}_1 = \mathcal{M} \mathbb{R}_1 {}_{-L} \mathcal{M} = \mathcal{M} \quad (19)$$

so that  $\mathbb{R}_1$  is the identity operation for left preconditioning, whereas for right preconditioning this is just the identity map.

# Newton-Krylov

```
1: procedure  $\mathcal{N}\backslash\mathcal{K}(\vec{F}, \vec{x}_i, \vec{b})$   
2:    $\vec{d} = \vec{J}(\vec{x}_i)^{-1}\vec{r}(\vec{x}_i, \vec{b})$   
3:    $\vec{x}_{i+1} = \vec{x}_i + \lambda\vec{d}$   
4: end procedure  
5: return  $\vec{x}_{i+1}$ 
```

- ▷ solve by Krylov method
- ▷  $\lambda$  determined by line search

# Left Preconditioned Newton-Krylov

- 1: **procedure**  $\mathcal{N}\backslash\mathcal{K}(\vec{x} - \vec{M}(\mathcal{F}, \vec{x}, \vec{b}), \vec{x}_i, 0)$
- 2:      $\vec{d} = \frac{\partial(\vec{x}_i - \mathcal{M}(\mathcal{F}, \vec{x}_i, \vec{b}))}{\partial \vec{x}_i}^{-1} (\vec{x}_i - \mathcal{M}(\mathcal{F}, \vec{x}_i, \vec{b}))$
- 3:      $\vec{x}_{i+1} = \vec{x}_i + \lambda \vec{d}$
- 4: **end procedure**
- 5: **return**  $\vec{x}_{i+1}$

# Jacobian Computation

$$\frac{\partial(\vec{x} - \mathcal{M}(\mathcal{F}, \vec{x}_i, \vec{b}))}{\vec{x}_i} = I - \frac{\partial \mathcal{M}(\mathcal{F}, \vec{x}_i, \vec{b})}{\partial \vec{x}_i},$$

Direct differencing would require

- one inner nonlinear iteration per **Krylov** iteration.

# Jacobian Computation

$$\frac{\partial(\vec{x} - \mathcal{M}(\mathcal{F}, \vec{x}_i, \vec{b}))}{\vec{x}_i} = I - \frac{\partial \mathcal{M}(\mathcal{F}, \vec{x}_i, \vec{b})}{\partial \vec{x}_i},$$

Direct differencing would require

- one inner nonlinear iteration per **Krylov** iteration.

# Jacobian Computation

## Impractical!

$$\frac{\partial(\vec{x} - \mathcal{M}(\mathcal{F}, \vec{x}_i, \vec{b}))}{\vec{x}_i} = I - \frac{\partial \mathcal{M}(\mathcal{F}, \vec{x}_i, \vec{b})}{\partial \vec{x}_i},$$

Direct differencing would require

- one inner nonlinear iteration per **Krylov** iteration.

# Jacobian Computation

Approximation for NASM

$$\frac{\partial(\vec{x} - \mathcal{M}(\mathcal{F}, \vec{x}, \vec{b}))}{\partial \vec{x}} = \frac{\partial(\vec{x} - (\vec{x} - \sum_b \mathbf{J}_b(\vec{x}_b)^{-1} \mathcal{F}_b(\vec{x}_b)))}{\partial \vec{x}}$$

$$\approx \sum_b \mathbf{J}_b(\vec{x}_{b*})^{-1} \mathbf{J}(\vec{x})$$

This would require

- one inner nonlinear iteration
- small number of block solves

per **outer nonlinear** iteration.

X.-C. Cai and D. E. Keyes, *SIAM J. Sci. Comput.*, 24 (2002), pp. 183–200



# Right Preconditioned Newton-Krylov

- 1: **procedure** NK( $\mathcal{F}(\mathcal{M}(\vec{F}, \cdot, \vec{b})), \vec{y}_i, \vec{b}$ )
- 2:      $\vec{x}_i = \mathcal{M}(\mathcal{F}, \vec{y}_i, \vec{b})$
- 3:      $\vec{d} = \vec{J}(\vec{x})^{-1} \vec{r}(\vec{x}_i)$
- 4:      $\vec{x}_{i+1} = \vec{x}_i + \lambda \vec{d}$
- 5: **end procedure**
- 6: **return**  $\vec{x}_{i+1}$

▷  $\lambda$  determined by line search

# Jacobian Computation

## First-Order Approximation

Only the action of the original Jacobian is needed at first order:

$$\vec{y}_{i+1} = \vec{y}_i - \lambda \frac{\partial \mathcal{M}(\mathcal{F}, \vec{y}_i)}{\partial \vec{y}_i}^{-1} J(\mathcal{M}(\mathcal{F}, \vec{y}_i))^{-1} \mathcal{F}(\mathcal{M}(\mathcal{F}, \vec{y}_i))$$

$$\mathcal{M}(\mathcal{F}, \vec{y}_{i+1}) = \mathcal{M}(\mathcal{F}, \vec{y}_i - \lambda \frac{\partial \mathcal{M}(\mathcal{F}, \vec{y}_i)}{\partial \vec{y}_i}^{-1} J(\mathcal{M}(\mathcal{F}, \vec{y}_i))^{-1} \mathcal{F}(\mathcal{M}(\mathcal{F}, \vec{y}_i)))$$

$$\approx \mathcal{M}(\mathcal{F}, \vec{y}_i)$$

$$- \lambda \frac{\partial \mathcal{M}(\mathcal{F}, \vec{y}_i)}{\partial \vec{y}_i} \frac{\partial \mathcal{M}(\mathcal{F}, \vec{y}_i)}{\partial \vec{y}_i}^{-1} J(\mathcal{M}(\mathcal{F}, \vec{y}_i))^{-1} \mathcal{F}(\mathcal{M}(\mathcal{F}, \vec{y}_i))$$

$$= \mathcal{M}(\mathcal{F}, \vec{y}_i) - \lambda J(\mathcal{M}(\mathcal{F}, \vec{y}_i))^{-1} \mathcal{F}(\mathcal{M}(\mathcal{F}, \vec{y}_i))$$

$$\vec{x}_{i+1} = \vec{x}_i - \lambda J(\vec{x}_i)^{-1} \mathcal{F}(\vec{x}_i)$$

$\mathcal{N} \setminus K -_R \vec{M}$  is equivalent to  $\mathcal{N} \setminus K * \vec{M}$  at first order

# Jacobian Computation

## First-Order Approximation

Only the action of the original Jacobian is needed at first order:

$$\vec{y}_{i+1} = \vec{y}_i - \lambda \frac{\partial \mathcal{M}(\mathcal{F}, \vec{y}_i)^{-1}}{\partial \vec{y}_i} J(\mathcal{M}(\mathcal{F}, \vec{y}_i))^{-1} \mathcal{F}(\mathcal{M}(\mathcal{F}, \vec{y}_i))$$

$$\mathcal{M}(\mathcal{F}, \vec{y}_{i+1}) = \mathcal{M}(\mathcal{F}, \vec{y}_i - \lambda \frac{\partial \mathcal{M}(\mathcal{F}, \vec{y}_i)^{-1}}{\partial \vec{y}_i} J(\mathcal{M}(\mathcal{F}, \vec{y}_i))^{-1} \mathcal{F}(\mathcal{M}(\mathcal{F}, \vec{y}_i)))$$

$$\approx \mathcal{M}(\mathcal{F}, \vec{y}_i)$$

$$- \lambda \frac{\partial \mathcal{M}(\mathcal{F}, \vec{y}_i)}{\partial \vec{y}_i} \frac{\partial \mathcal{M}(\mathcal{F}, \vec{y}_i)^{-1}}{\partial \vec{y}_i} J(\mathcal{M}(\mathcal{F}, \vec{y}_i))^{-1} \mathcal{F}(\mathcal{M}(\mathcal{F}, \vec{y}_i))$$

$$= \mathcal{M}(\mathcal{F}, \vec{y}_i) - \lambda J(\mathcal{M}(\mathcal{F}, \vec{y}_i))^{-1} \mathcal{F}(\mathcal{M}(\mathcal{F}, \vec{y}_i))$$

$$\vec{x}_{i+1} = \vec{x}_i - \lambda J(\vec{x}_i)^{-1} \mathcal{F}(\vec{x}_i)$$

$\mathcal{N} \setminus \mathbb{K} -_R \vec{M}$  is equivalent to  $\mathcal{N} \setminus \mathbb{K} * \vec{M}$  at first order

# Jacobian Computation

## Direct Approximation

$$\begin{aligned} \mathcal{F}(\mathcal{M}(\mathcal{F}, \vec{y}_i, \vec{b})) &= \mathbf{J}(\mathcal{M}(\mathcal{F}, \vec{y}_i, \vec{b})) \frac{\partial \mathcal{M}(\mathcal{F}, \vec{y}_i, \vec{b})}{\partial \vec{y}_i} (\vec{y}_{i+1} - \vec{y}_i) \\ &\approx \mathbf{J}(\mathcal{M}(\mathcal{F}, \vec{y}_i, \vec{b})) (\mathcal{M}(\mathcal{F}, \vec{y}_i + \vec{d}, \vec{b}) - \vec{x}_i) \end{aligned}$$

- Solve for  $\vec{d}$
- Requires inner nonlinear solve for each Krylov iterate
- Needs FGMRES

P. Birken and A. Jameson, *J. Num. Meth. in Fluids*, 62 (2010), pp. 565–573

# Outline

## 1 Algorithmics

## 2 Experiments

- Composition
- Multilevel
- Magma Dynamics

# Outline

- 2 Experiments
  - Composition
  - Multilevel
  - Magma Dynamics

## SNES ex16

## 3D Large Deformation Elasticity

$$\int_{\Omega} \mathbf{F} \cdot \mathbf{S} : \nabla \mathbf{v} \, d\Omega + \int_{\Omega} \text{loading} \, \mathbf{e}_y \cdot \mathbf{v} \, d\Omega = 0 \quad (20)$$

**F** Deformation gradient

**S** Second Piola-Kirchhoff tensor

Saint Venant-Kirchhoff model of hyperelasticity

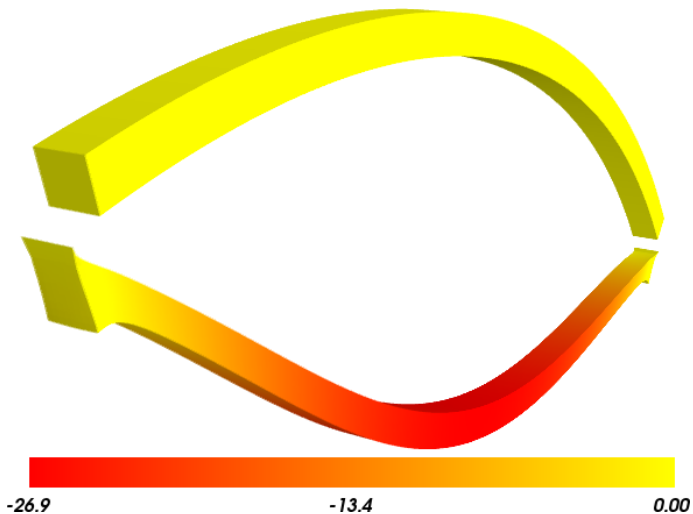
$\Omega$  -arc *angle* subsection of a cylindrical shell

-height *thickness*

-rad *inner radius*

-width *width*

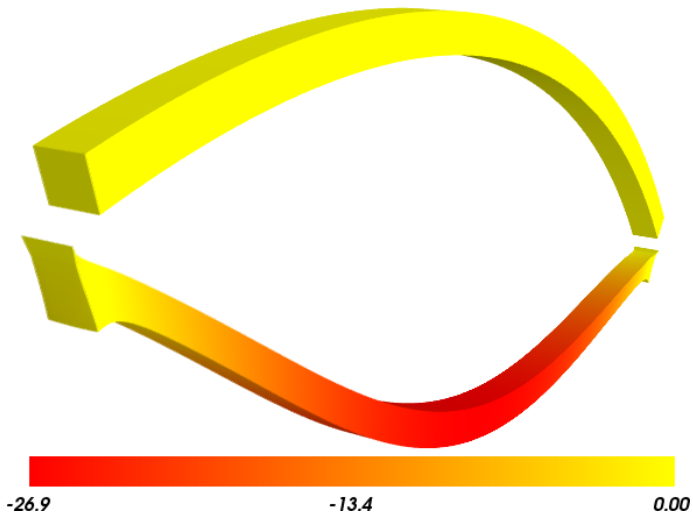
# Large Deformation Elasticity



Unstressed and stressed configurations for the elasticity test problem.

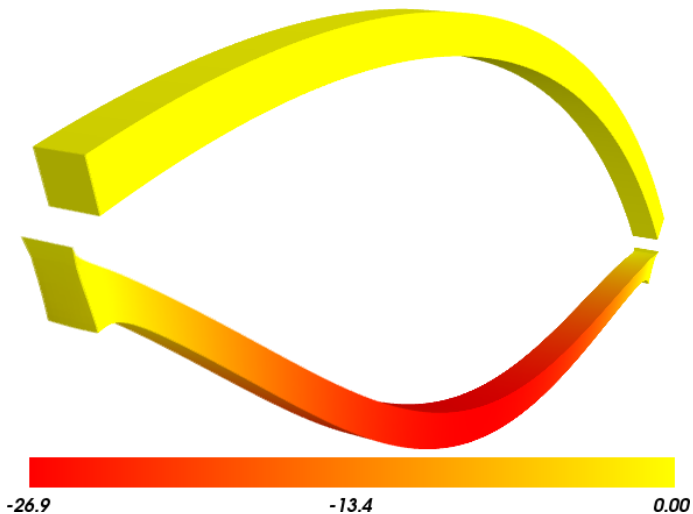


# Large Deformation Elasticity



Coloration indicates vertical displacement in meters.

# Large Deformation Elasticity



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

# Large Deformation Elasticity

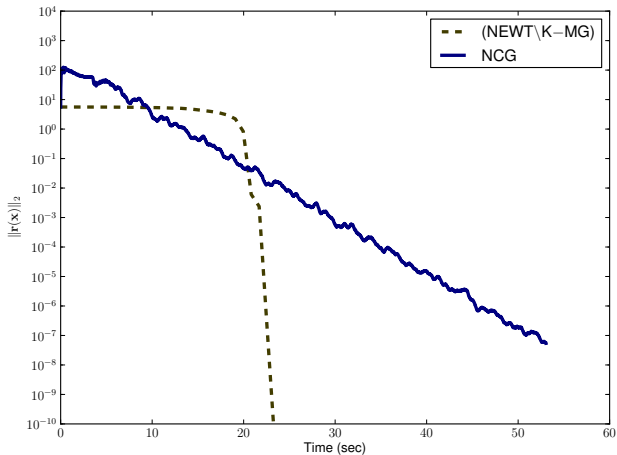
Running

## 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

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

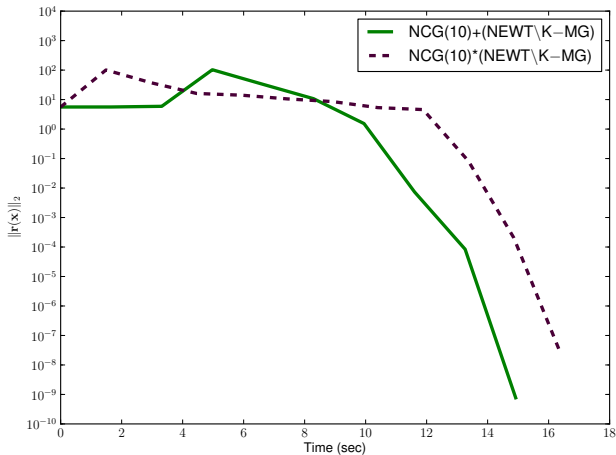


# Plain SNES Convergence

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

# Composed SNES Convergence

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

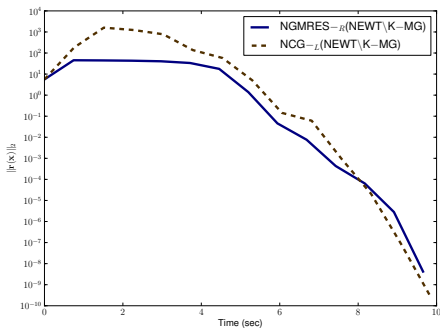


# Composed SNES Convergence

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

# Preconditioned SNES Convergence

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





# Preconditioned SNES Convergence

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

# Outline

## 2 Experiments

- Composition
- **Multilevel**
- Magma Dynamics

# 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
$(\mathcal{N} \setminus \mathcal{K} - \text{MG})$	9.83	17	352	34	85	370	–
NGMRES $_{-R}$ $(\mathcal{N} \setminus \mathcal{K} - \text{MG})$	7.48	10	220	21	50	231	10
FAS	6.23	162	0	2382	377	754	–
FAS + $(\mathcal{N} \setminus \mathcal{K} - \text{MG})$	8.07	10	197	232	90	288	–
FAS * $(\mathcal{N} \setminus \mathcal{K} - \text{MG})$	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 solvers,**

Peter Brune, Matthew Knepley, Barry Smith, and Xuemin Tu,

ANL/MCS-P2010-0112, Argonne National Laboratory, 2012.

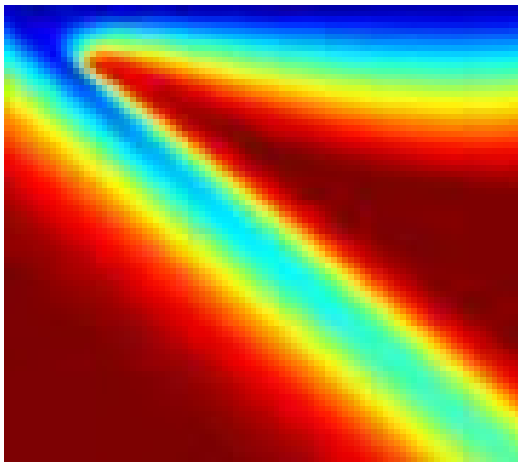
<http://www.mcs.anl.gov/uploads/cels/papers/P2010-0112.pdf>

# Outline

- 2 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



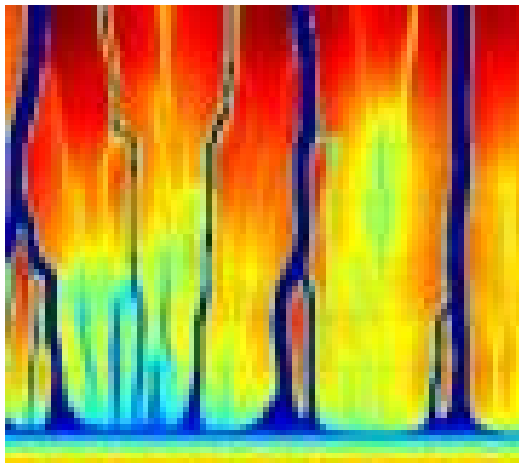
---

<sup>a</sup>Katz, Speigelman



# 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



<sup>a</sup>Katz, Spiegelman

# Dimensional Formulation

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

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

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

# Closure Conditions

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

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

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

# Nondimensional Formulation

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

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

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

# 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
```

See discussion in:

**Composing scalable nonlinear solvers,**

Peter Brune, Matthew Knepley, Barry Smith, and Xuemin Tu,  
ANL/MCS-P2010-0112, Argonne National Laboratory, 2012.

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

# What Are We Missing?

We need a short time convergence theory:

- Most iterations never enter the asymptotic regime
- Most complex solvers are composed

We need a viable nonlinear smoother:

- GS is too expensive for FEM
- NASM is a possibility,

# Nonlinear GMRES

- 1: **procedure** NGMRES( $\mathcal{F}, \vec{x}_j \cdots \vec{x}_{j-m+1}, \vec{b}$ )
- 2:      $\vec{d}_j = -\vec{r}(\vec{x}_j)$
- 3:      $\vec{x}_j^M = \vec{x}_j + \lambda \vec{d}_j$
- 4:      $\mathcal{F}_j^M = \vec{r}(\vec{x}_j^M)$
- 5:     **minimize**  $\|\vec{r}((1 - \sum_{k=i-m}^{i-1} \alpha_k) \vec{x}_j^M + \sum_{k=i-m}^{i-1} \alpha_k \vec{x}_k)\|_2$  over  
        $\{\alpha_{i-m} \cdots \alpha_{i-1}\}$
- 6:      $\vec{x}_j^A = (1 - \sum_{k=i-m}^{i-1} \alpha_k) \vec{x}_j^M + \sum_{k=i-m}^{i-1} \alpha_k \vec{x}_k$
- 7:      $\vec{x}_{j+1} = \vec{x}_j^A$  or  $\vec{x}_j^M$  if  $\vec{x}_j^A$  is insufficient.
- 8: **end procedure**
- 9: **return**  $\vec{x}_{j+1}$

Can emulate Anderson mixing and DIIS

# Nonlinear GMRES

- 1: **procedure** NGMRES( $\mathcal{F}, \vec{x}_i \cdots \vec{x}_{i-m+1}, \vec{b}$ )
- 2:      $\vec{d}_i = -\vec{r}(\vec{x}_i)$
- 3:      $\vec{x}_i^M = \vec{x}_i + \lambda \vec{d}_i$
- 4:      $\mathcal{F}_i^M = \vec{r}(\vec{x}_i^M)$
- 5:     **minimize**  $\|\vec{r}((1 - \sum_{k=i-m}^{i-1} \alpha_k) \vec{x}_i^M + \sum_{k=i-m}^{i-1} \alpha_k \vec{x}_k)\|_2$  over  
        $\{\alpha_{i-m} \cdots \alpha_{i-1}\}$
- 6:      $\vec{x}_i^A = (1 - \sum_{k=i-m}^{i-1} \alpha_k) \vec{x}_i^M + \sum_{k=i-m}^{i-1} \alpha_k \vec{x}_k$
- 7:      $\vec{x}_{i+1} = \vec{x}_i^A$  or  $\vec{x}_i^M$  if  $\vec{x}_i^A$  is insufficient.
- 8: **end procedure**
- 9: **return**  $\vec{x}_{i+1}$

Can emulate Anderson mixing and DIIS

# Full Approximation Scheme (FAS)

## Nonlinear Multigrid

- 1: **procedure** FAS( $\vec{F}, \vec{x}_i, \vec{b}$ )
- 2:      $\vec{x}_s = \mathcal{M}_s(\mathcal{F}, \vec{x}_i, \vec{b})$
- 3:      $\vec{x}_c = \widehat{\mathbf{R}}\vec{x}_s$
- 4:      $\vec{b}_c = \mathcal{F}_c(\vec{x}_c) - \mathbf{R}[\mathcal{F}(\vec{x}_s) - \vec{b}]$
- 5:      $\vec{x}_s = \vec{x}_s + \mathbf{P}[\text{FAS}(\vec{F}_c, \vec{x}_c, \vec{b}_c) - \vec{x}_c]$
- 6:      $\vec{x}_{i+1} = \mathcal{M}_s(\mathcal{F}, \vec{x}_s, \vec{b})$
- 7: **end procedure**
- 8: **return**  $\vec{x}_{i+1}$

# Other Nonlinear Solvers

**NASM** Nonlinear Additive Schwarz

**NGS** Nonlinear Gauss-Siedel

**NCG** Nonlinear Conjugate Gradients

**QN** Quasi-Newton methods