An Introduction to PETSc

Matthew Knepley

Computer Science and Engineering
University at Buffalo

SIAM Geosciences
Baton Rouge, LA October 15, 2025

Outline

Libraries Hide Hardware Details

Libraries Hide Hardware Details

Libraries Hide Hardware Details

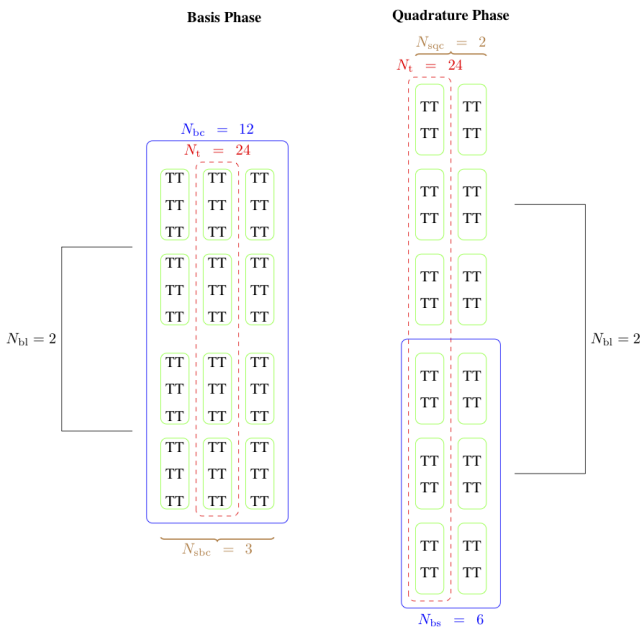
Why Libraries?

Libraries Hide Hardware Details

Libraries Hide Implementation Complexity

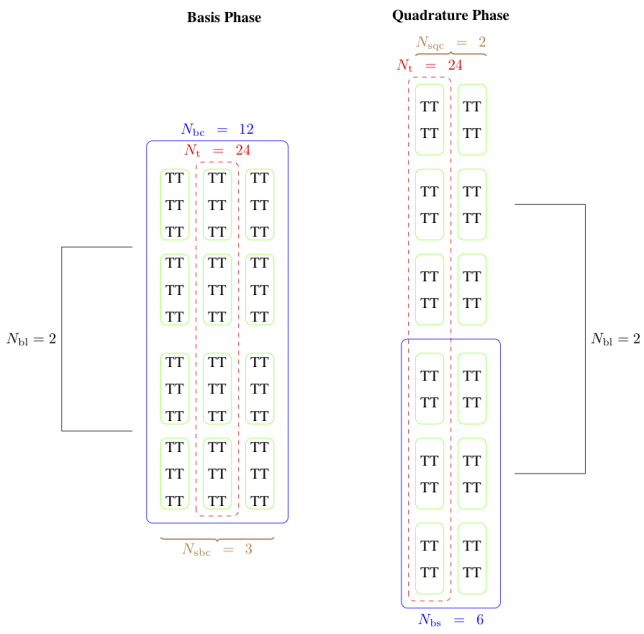
Why Libraries?

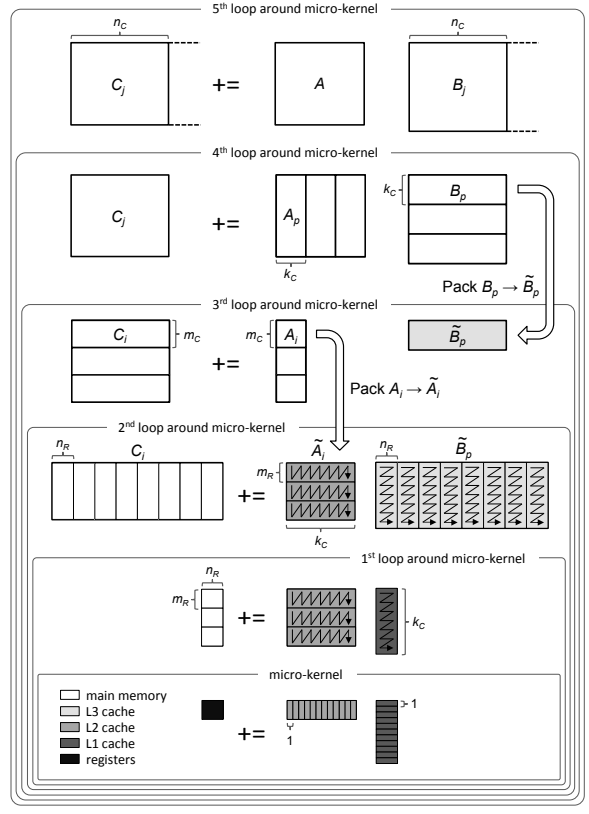
Libraries Hide Implementation Complexity



Why Libraries?

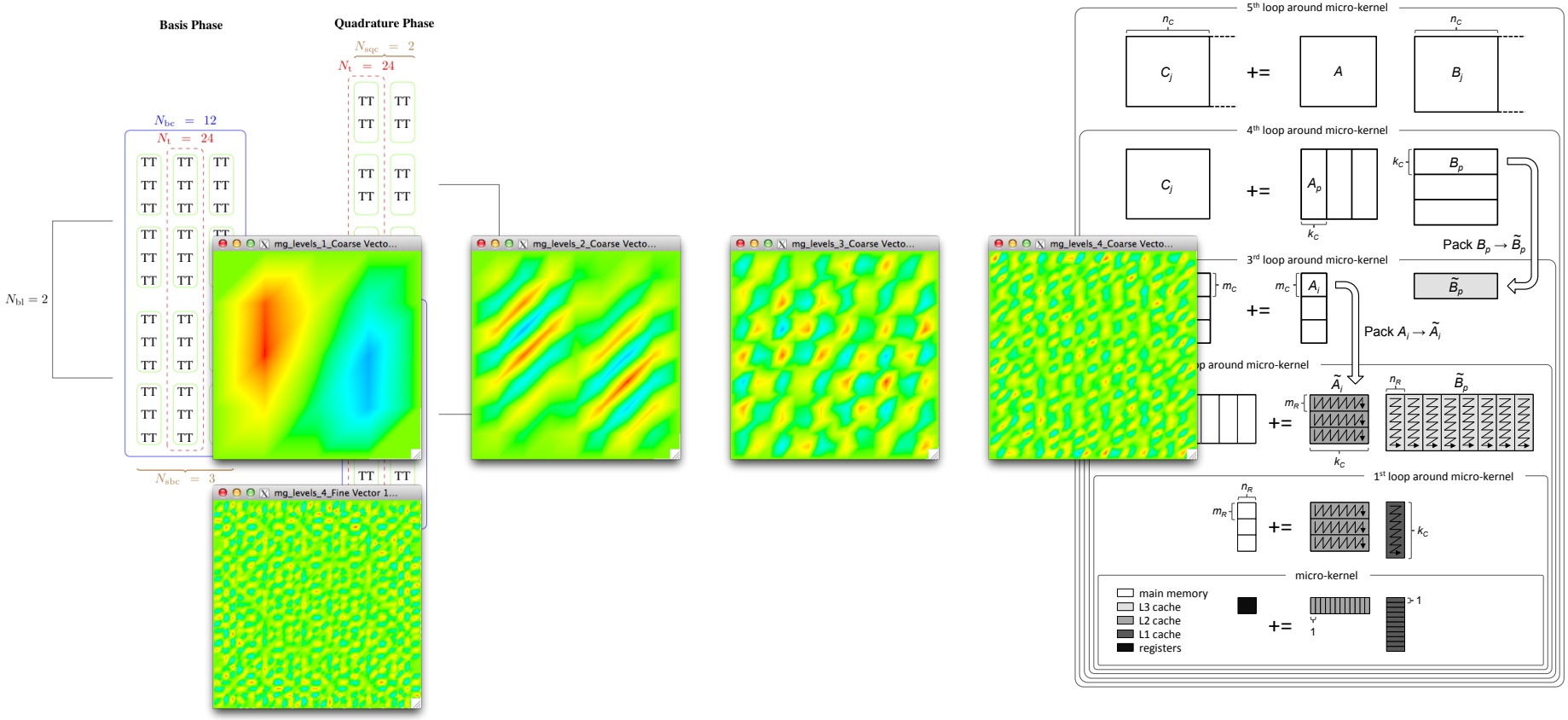
Libraries Hide Implementation Complexity





Why Libraries?

Libraries Hide Implementation Complexity



Libraries Accumulate Best Practices

Libraries Accumulate Best Practices

Classical Gram-Schmidt orthogonalization with selective reorthogonalization

Libraries Accumulate Best Practices

Classical Gram-Schmidt orthogonalization with selective reorthogonalization

Eigen-estimation for AMG with Krylov bootstrap

Libraries Accumulate Best Practices

Classical Gram-Schmidt orthogonalization with selective reorthogonalization

Eigen-estimation for AMG with Krylov bootstrap

Improvement without code changes

Libraries Simplfy Determining Provenance

Libraries Simplfy Determining Provenance

Rather than making build-time choices,

Libraries Simplfy Determining Provenance

Rather than making build-time choices,
that must be plumbed through all levels

Libraries Simplfy Determining Provenance

Rather than making build-time choices,
that must be plumbed through all levels
with another layer of workflow scripts

Libraries Simplfy Determining Provenance

Rather than making build-time choices,
that must be plumbed through all levels
with another layer of workflow scripts
and brittle top-level interfaces,

Libraries Simplfy Determining Provenance

we use packages without modification,

Libraries Simplfy Determining Provenance

we use packages without modification,
compiled in a standard way

Libraries Simplfy Determining Provenance

we use packages without modification,
compiled in a standard way
and controlled entirely via runtime options.

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
  - Nearly 30 years of continuous development
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
  - Tested at every supercomputer installation and 10,000+ users
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
  - Many Gordon Bell Winners
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
  - State-of-the-Art Linear and Nonlinear Solvers, Eigensolvers, Optimization Solvers, Timesteppers
- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness

- Flexibility & Extensibility

Why PETSc?

- Dependability & Maintainability
- Portability & Robustness
- Performance & Scalability
- Optimality & Robustness
- Flexibility & Extensibility
  - More than 11,000 citations and hundreds of application packages
  - Aerodynamics, Arterial Flow, Corrosion, Combustion, Data Mining, Earthquake Mechanics, . . .

Alternatives to PETSc

There are other packages…

Alternatives to PETSc

There are other packages…

- DUNE, Hypre

Alternatives to PETSc

There are other packages…

- DUNE, Hypre
- Firedrake/FEniCS, FreeFEM, Deal.II

Alternatives to PETSc

There are other packages…

- DUNE, Hypre
- Firedrake/FEniCS, FreeFEM, Deal.II
- PyLith, Underworld, Salvus, SPECFEM, PFLOTRAN

PETSc Capabilities

- Parallel linear algebra (sparse and dense) [1]
- Linear [2] and nonlinear solvers [3]
- Timestepping [4]
- Optimization [5]
- Eigensolvers [6]
- Meshes
  - Cartesian
  - Structured adaptive [7]
  - Unstructured adaptive [8]
- Discretization support
  - Finite Difference
  - Finite Element [9]
  - Finite Volume [10]
- Portability (GPUs) [11]
- Profiling and debugging [12]

PETSc Capabilities

- Parallel linear algebra (sparse and dense) [1]
- Linear [2] and nonlinear solvers [3]
- Timestepping [4]
- Optimization [5]
- Eigensolvers [6]
- Meshes
  - Cartesian
  - Structured adaptive [7]
  - Unstructured adaptive [8]
- Discretization support
  - Finite Difference
  - Finite Element [9]
  - Finite Volume [10]
- Portability (GPUs) [11]
- Profiling and debugging [12]

Getting More Help

- https://petsc.org [13]
- Hyperlinked documentation
  - Sphinx and PDF Manual [14]
  - Manual pages for every method
  - HTML of example code (linked to manual pages)
- New Sphinx pages for manuals and tutorials
- FAQ
- Full support at petsc-maint@mcs.anl.gov

Outline

Outline

- Hands-On Examples
  - Block Preconditioners
  - Multigrid
  - Nonlinear Preconditioning
  - Unstructured Mesh Creation*
  - ML*
  - Optimization*

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 ja
```

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

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 ga
 -fieldsplit_pressure_ksp_type preonly -fieldsplit_pressure_pc_type ja
```

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

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 multiplicativ
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type ga
 -fieldsplit_pressure_ksp_type preonly -fieldsplit_pressure_pc_type ja
```

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

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 multiplicativ
 -pc_fieldsplit_0_fields 1 -pc_fieldsplit_1_fields 0
 -fieldsplit_velocity_ksp_type preonly -fieldsplit_velocity_pc_type ga
 -fieldsplit_pressure_ksp_type preonly -fieldsplit_pressure_pc_type ja
```

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

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 ga
 -fieldsplit_pressure_ksp_type minres  -fieldsplit_pressure_pc_type no
```

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

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 ga
 -fieldsplit_pressure_ksp_type minres  -fieldsplit_pressure_pc_type no
```

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

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 ga
 -fieldsplit_pressure_ksp_type minres  -fieldsplit_pressure_pc_type no
```

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Uzawa Iteration, Uzawa, 1958

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

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

Solver Configuration: No New Code

ex62: P_2/P_1 Stokes Problem on Unstructured Mesh

Full Schur Complement, Schur, 1905.

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

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

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 jaco
 -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
```

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

Solver Configuration: No New Code

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

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

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

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

Solver Configuration: No New Code

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

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

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

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

Outline

- Hands-On Examples
  - Block Preconditioners
  - Multigrid
  - Nonlinear Preconditioning
  - Unstructured Mesh Creation*
  - ML*
  - Optimization*

Why not use AMG?

Why not use AMG?

- Of course we will try AMG
  - GAMG, `-pc_type gamg`
  - ML, `-download-ml`, `-pc_type ml`
  - BoomerAMG, `-download-hypre`, `-pc_type hypre -pc_hypre_type boomeramg`

Why not use AMG?

- Of course we will try AMG
  - GAMG, `-pc_type gamg`
  - ML, `-download-ml`, `-pc_type ml`
  - BoomerAMG, `-download-hypre`, `-pc_type hypre -pc_hypre_type boomeramg`
- Problems with
  - vector character
  - anisotropy
  - scalability of setup time

Multigrid with DM

Allows multigrid with some simple command line options

- `-pc_type mg`, `-pc_mg_levels`
- `-pc_mg_type`, `-pc_mg_cycle_type`, `-pc_mg_galerkin`
- `-mg_levels_1_ksp_type`, `-mg_levels_1_pc_type`
- `-mg_coarse_ksp_type`, `-mg_coarse_pc_type`
- `-da_refine`, `-ksp_view`

Interface also works with GAMG and 3rd party packages like ML

A 2D Problem

Problem has:

- 1,640,961 unknowns (on the fine level)
- 8,199,681 nonzeros

	Options	Explanation
`./ex5`	`-da_grid_x 21 -da_grid_y 21`	Original grid is 21x21
	`-ksp_rtol 1.0e-9`	Solver tolerance
	`-da_refine 6`	6 levels of refinement
	`-pc_type mg`	4 levels of multigrid
	`-pc_mg_levels 4`	
	`-snes_monitor -snes_view`	Describe solver

A 3D Problem

Problem has:

- 1,689,600 unknowns (on the fine level)
- 89,395,200 nonzeros

	Options	Explanation
`./ex48`	`-M 5 -N 5`	Coarse problem size
	`-da_refine 5`	5 levels of refinement
	`-ksp_rtol 1.0e-9`	Solver tolerance
	`-thi_mat_type baij`	Needs SOR
	`-pc_type mg`	4 levels of multigrid
	`-pc_mg_levels 4`	
	`-snes_monitor -snes_view`	Describe solver

Full Multigrid

The Full Multigrid algorithm (FMG)

- V-cycle at each level,
- then interpolate to the next finer grid
- Can solve to discretization error with a *single* iteration

Full Multigrid Work

$$
\begin{aligned}
C_{FMG} &= \left(1 + \frac{1}{2^d} + \frac{1}{2^{2d}} + \ldots\right) C_V \\
&= \sum_{n=0}^{\infty} \frac{1}{2^{nd}} C_V \\
&= \frac{2^d}{2^d - 1} C_V \\
&= \left(\frac{2^d}{2^d - 1}\right)^2 C_{\text{twolevel}}.
\end{aligned}
$$

Full Multigrid Work

$$
\begin{aligned}
C_{FMG} &= \left(1 + \frac{1}{2^d} + \frac{1}{2^{2d}} + \ldots\right) C_V \\
&= \sum_{n=0}^{\infty} \frac{1}{2^{nd}} C_V \\
&= \frac{2^d}{2^d - 1} C_V \\
&= \left(\frac{2^d}{2^d - 1}\right)^2 C_{\text{twolevel}}.
\end{aligned}
$$

1D FMG is $2 \times C_V$, 3D FMG is $\frac{8}{7} \times C_V$

Full Multigrid Accuracy

Suppose we have an order α method,

$$\|x - x_h\| < Ch^{\alpha}$$

FD and P_1 both have $\alpha = 2$

Full Multigrid Accuracy

E_d Discretization Error

E_a Algebraic Error

Choose iterative tolerance so that

$$E_a = rE_d \qquad r < 1$$

and

$$E \leq E_d + E_a = (1 + r)Ch^{\alpha}$$

Full Multigrid Accuracy

Suppose

- Finish V-cycle for $2h$ grid,
- Use as coarse correction for h grid
- Perform final V-cycle for h grid
- Need V-cycle error reduction factor η to get r reduction in E_a

Full Multigrid Accuracy

$$
\begin{aligned}
\eta E_a &< rCh^\alpha \\
\eta \left(E - E_d\right) &< rCh^\alpha \\
\eta \left((1+r)C(2h)^\alpha - Ch^\alpha\right) &< rCh^\alpha \\
\eta \left((1+r)2^\alpha - 1\right) &< r \\
\eta &< \frac{1}{2^\alpha + \frac{2^\alpha - 1}{r}}.
\end{aligned}
$$

If $\alpha = 2$ and $r = \frac{1}{2}$, then $\eta < \frac{1}{10}$.

Full Multigrid Experiment

V-cycle

```
./ex5 -mms 1 -par 0.0 -da_refine 3 -snes_type newtonls -snes_max_it 1
-ksp_rtol 1e-10 -pc_type mg -snes_monitor_short -ksp_monitor_short
```

gives

```
0 SNES Function norm 0.0287773
0 KSP Residual norm 0.793727
1 KSP Residual norm 0.00047526
2 KSP Residual norm 4.18007e-06
3 KSP Residual norm 1.1668e-07
4 KSP Residual norm 3.25952e-09
5 KSP Residual norm 7.274e-11
1 SNES Function norm 2.251e-10
N: 625 error l2 1.21529e-13 inf 9.53484e-12
```

Full Multigrid Experiment

V-cycle

```
./ex5 -mms 1 -par 0.0 -da_refine 3 -snes_type newtonls -snes_max_it 1
-ksp_rtol 1e-10 -pc_type mg -snes_monitor_short -ksp_monitor_short
```

and it changes little if we refine six more times

```
0 SNES Function norm 0.000455131
0 KSP Residual norm 50.6842
1 KSP Residual norm 0.00618427
2 KSP Residual norm 9.87833e-07
3 KSP Residual norm 2.99517e-09
1 SNES Function norm 2.83358e-09
N: 2362369 error l2 1.28677e-15 inf 7.68693e-12
```

Full Multigrid Experiment

FMG

```
./ex5 -mms 1 -par 0.0 -da_refine 3 -snes_type newtonls -snes_max_it 1
 -ksp_rtol 1e-10 -pc_type mg -snes_monitor_short -ksp_monitor_short
 -pc_mg_type full
```

We do not seem to see the convergence acceleration

```
0 SNES Function norm 0.0287773
0 KSP Residual norm 0.799687
1 KSP Residual norm 6.95292e-05
2 KSP Residual norm 1.50836e-06
3 KSP Residual norm 2.62524e-08
4 KSP Residual norm 6.184e-10
5 KSP Residual norm 1.275e-11
1 SNES Function norm 3.757e-11
N: 625 error l2 2.1428e-14 inf 1.80611e-12
```

Full Multigrid Experiment

FMG

```
./ex5 -mms 1 -par 0.0 -da_refine 3 -snes_type newtonls -snes_max_it 1
 -ksp_rtol 1e-10 -pc_type mg -snes_monitor_short -ksp_monitor_short
 -pc_mg_type full
```

although its a little more apparent as we refine,

```
0 SNES Function norm 0.000455131
0 KSP Residual norm 51.2
1 KSP Residual norm 2.92416e-06
2 KSP Residual norm 3.76404e-09
1 SNES Function norm 8.50096e-09
N: 2362369 error l2 1.70304e-15 inf 6.22476e-11
```

Full Multigrid Experiment

Script

```
#! /usr/bin/env python
import argparse
import subprocess
import numpy as np

parser = argparse.ArgumentParser(
description = 'CAAM 519 FMG',
epilog = 'For more information, visit http://www.mcs.anl.gov/petsc',
formatter_class = argparse.ArgumentDefaultsHelpFormatter)
parser.add_argument('--kmax', type=int, default=5,
                    help='The number of doublings to test')
parser.add_argument('--save', action='store_true', default=False,
                    help='Save the figures')
args = parser.parse_args()

sizesA = []
sizesB = []
errorA = []
errorB = []
```

Full Multigrid Experiment

Script

```
for k in range(args.kmax):
options = ['-snes_type', 'newtonls', '-snes_max_it', '1', '-da_refine'
           '-par', '0.0', '-ksp_atol', '1e-1', '-mms', '1',
           '-pc_type', 'mg', '-pc_mg_type', 'multiplicative',
           '-mg_levels_ksp_max_it', '5']
cmd = './ex5 '+' '.join(options)
out = subprocess.check_output(['./ex5']+options).split(' ')
# This is l_2, out[6] is l_infty
sizesA.append(int(out[1]))
errorA.append(float(out[4]))
for k in range(args.kmax):
options = ['-snes_type', 'newtonls', '-snes_max_it', '1', '-da_refine'
           '-par', '0.0', '-ksp_atol', '1e-1', '-mms', '1',
           '-pc_type', 'mg', '-pc_mg_type', 'full',
           '-mg_levels_ksp_max_it', '5']
cmd = './ex5 '+' '.join(options)
out = subprocess.check_output(['./ex5']+options).split(' ')
# This is l_2, out[6] is l_infty
sizesB.append(int(out[1]))
errorB.append(float(out[4]))
```

Full Multigrid Experiment

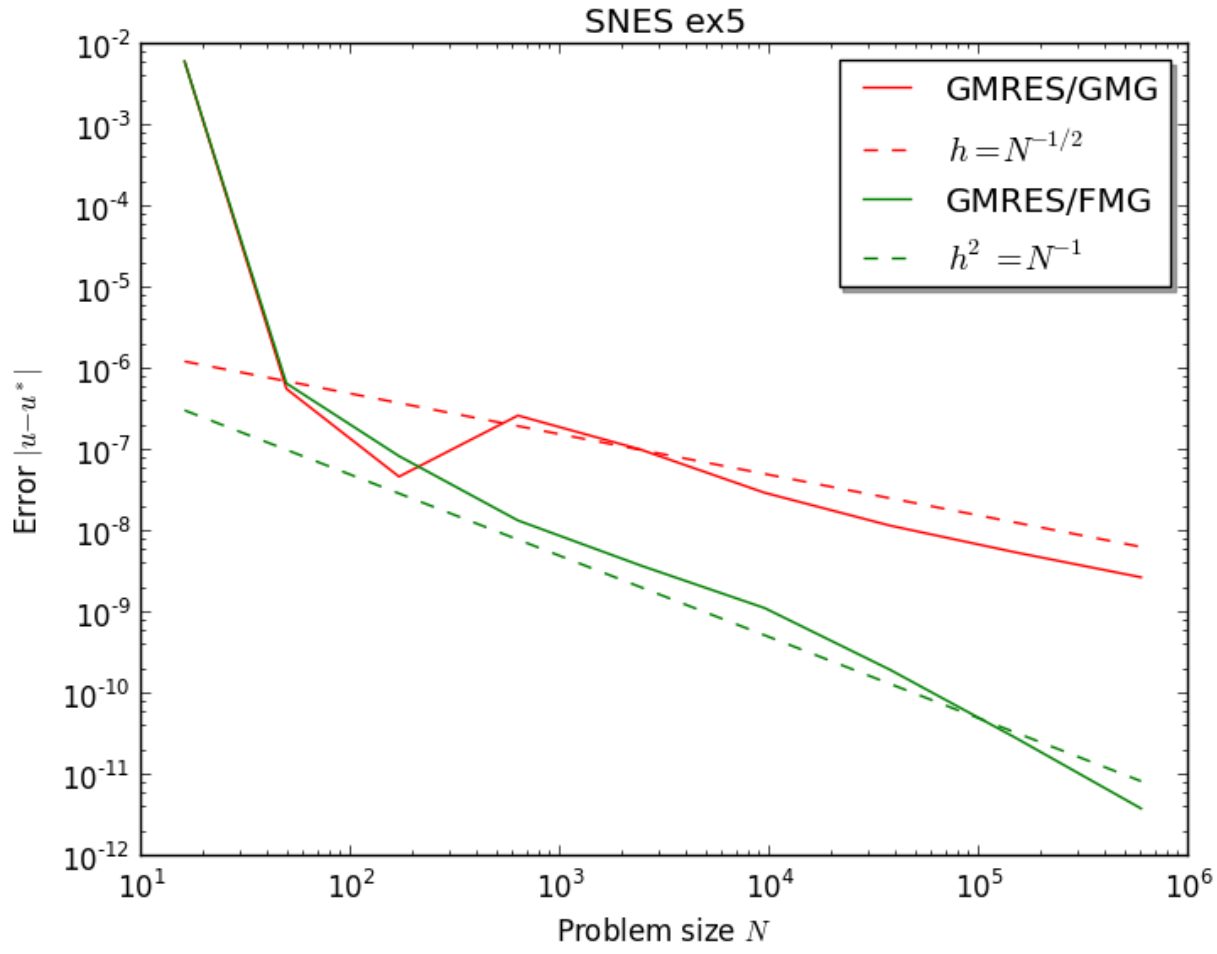
Script

```
SA = np.array(sizesA)
SB = np.array(sizesB)

from pylab import legend,plot,loglog,show,title,xlabel,ylabel,savefig
loglog(SA, errorA, 'r', SA, 5e-6 * SA ** -0.5, 'r--',
       SB, errorB, 'g', SB, 5e-6 * SB ** -1.,  'g--')
title('SNES ex5')
xlabel('Problem size $N$')
ylabel('Error $\|u - u^*\|$')
legend(['GMRES/GMG', '$h = N^{-1/2}$', 'GMRES/FMG', '$h^2 = N^{-1}$'],
       'upper right', shadow = True)
if args.save:
savefig('fmg.png')
else:
show()
```

Full Multigrid Experiment

Comparison



Outline

Hands-On Examples

- Block Preconditioners
- Multigrid
- Nonlinear Preconditioning
- Unstructured Mesh Creation*
- ML*
- Optimization*

Alternate Version

Excellent version by
Richard Mills

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
```

$$
\begin{aligned}
-\Delta U - \partial_y \Omega &= 0 \\
-\Delta V + \partial_x \Omega &= 0 \\
-\Delta \Omega + \nabla \cdot ([U\Omega, V\Omega]) - \mathrm{Gr}\, \partial_x T &= 0 \\
-\Delta T + \mathrm{Pr}\, \nabla \cdot ([UT, VT]) &= 0
\end{aligned}
$$

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 iteratio
```

Driven Cavity Problem

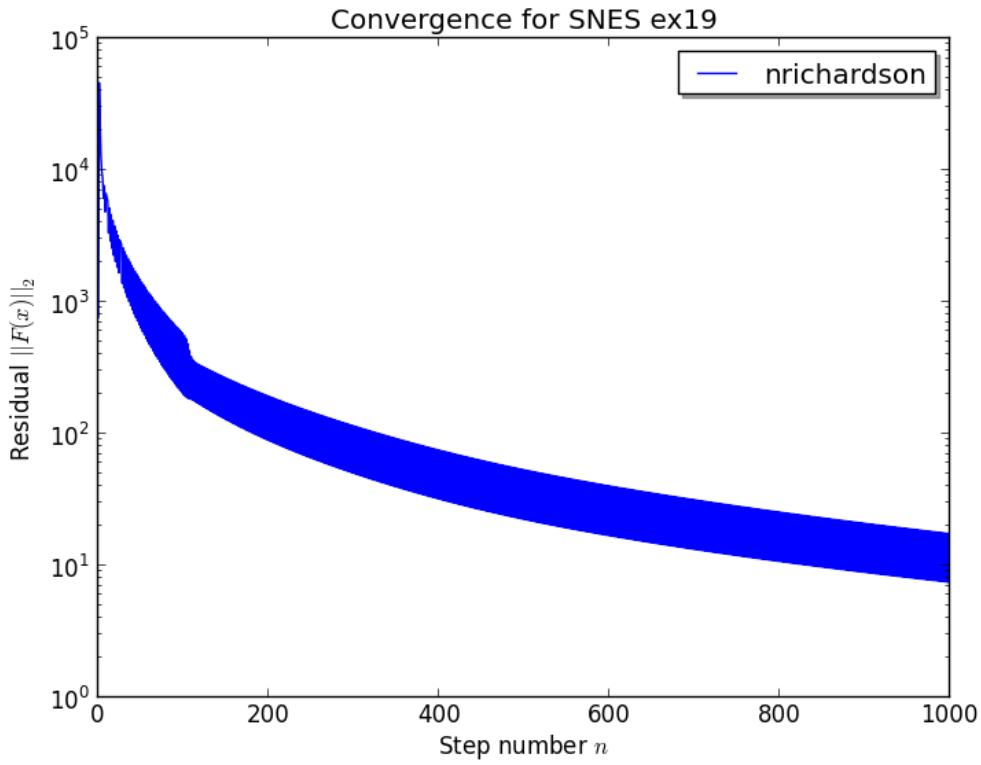
SNES ex19.c

```
./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
 ...
```

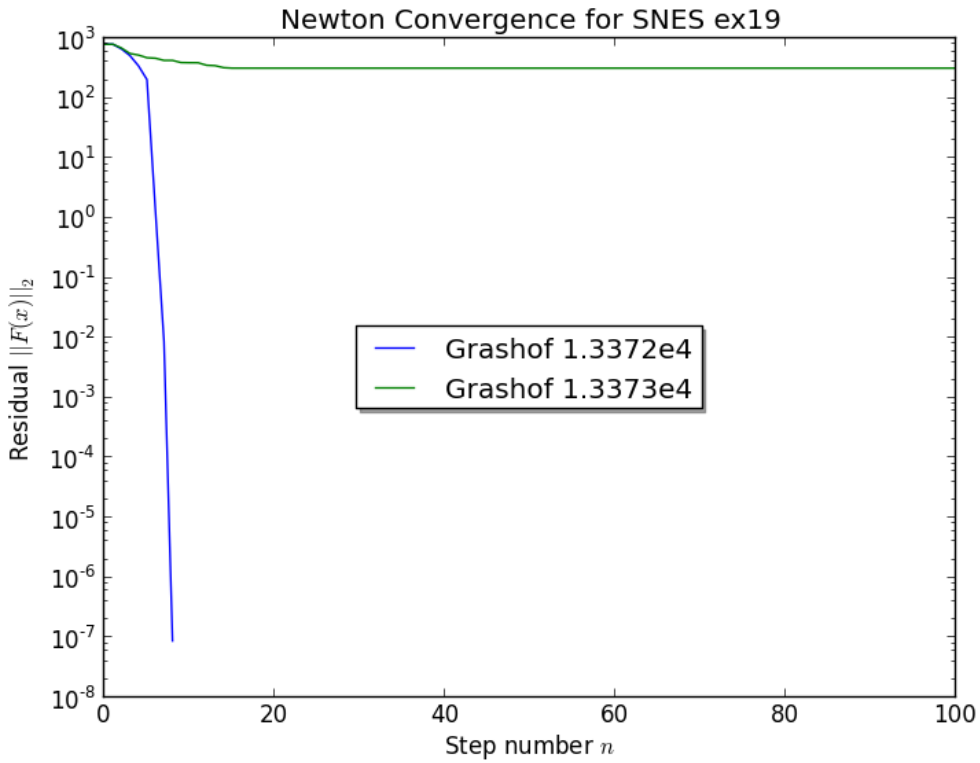
Deceleration of Convergence

```
./ex19 -lidvelocity 100 -grashof 1.3372e2
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type nrichardson -snes_linesearch_type cp -snes_max_it 10000
  -snes_monitor draw::draw_lg
```



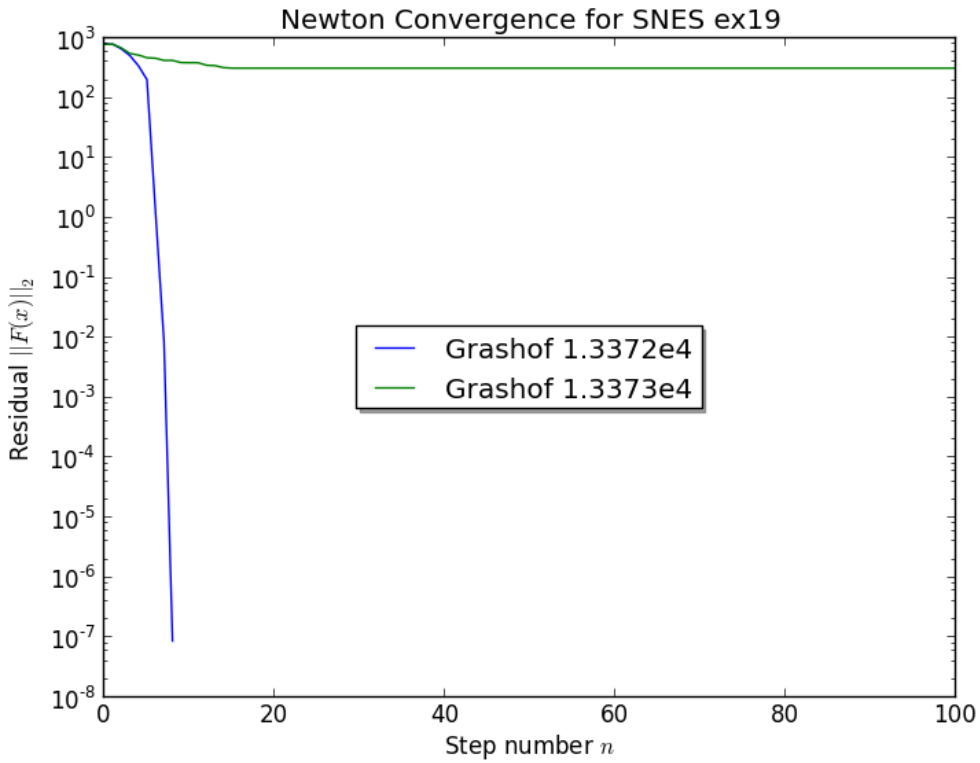
Stagnation of Newton

```
./ex19 -lidvelocity 100 -grashof 1.3372e4
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type newtonls -snes_max_it 100 -pc_type lu
```



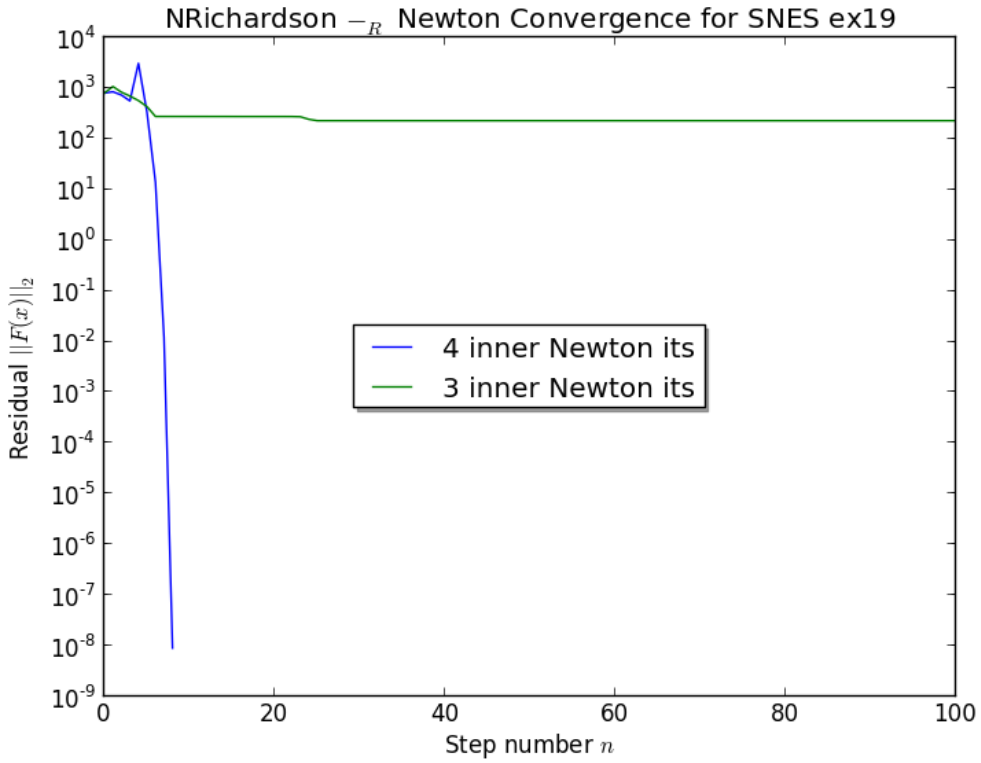
Stagnation of Newton

```
./ex19 -lidvelocity 100 -grashof 1.3373e4
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type newtonls -snes_max_it 100 -pc_type lu
```



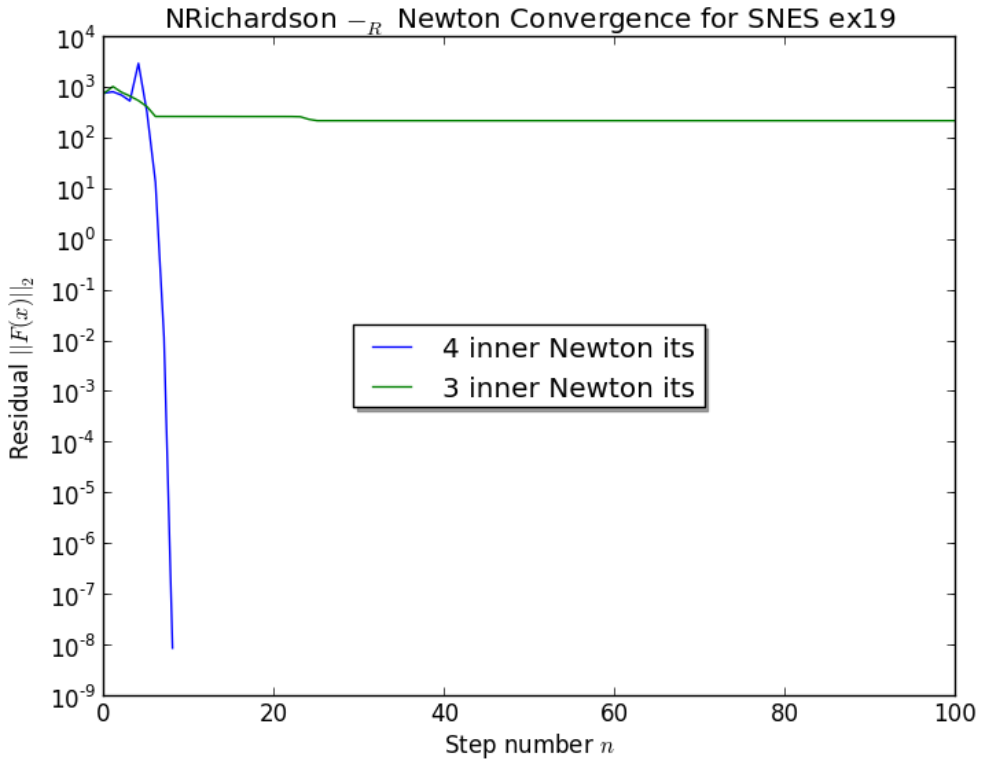
Preconditioning NRichardson with Newton

```
./ex19 -lidvelocity 100 -grashof 1.35e4
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type nrichardson -snes_max_it 200
  -npc_snes_type newtonls -npc_snes_max_it 1 -npc_pc_type lu
  -snes_monitor draw::draw_lg -snes_monitor_pause_final
```



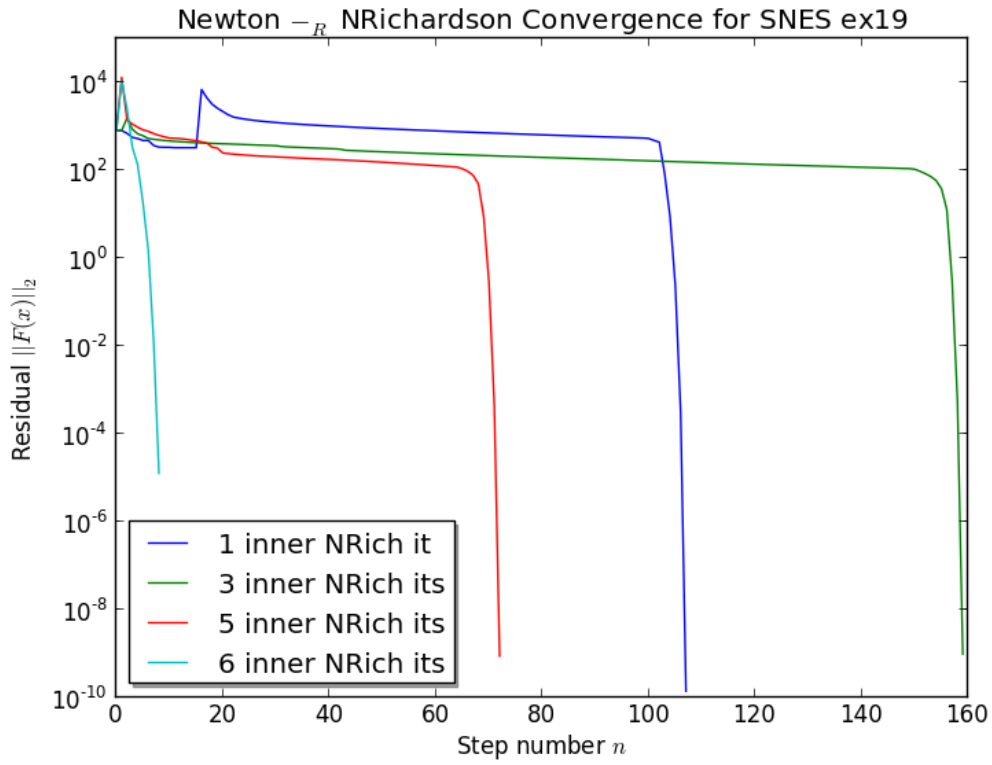
Preconditioning NRichardson with Newton

```
./ex19 -lidvelocity 100 -grashof 1.35e4
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type nrichardson -snes_max_it 200
  -npc_snes_type newtonls -npc_snes_max_it 2 -npc_pc_type lu
  -snes_monitor draw::draw_lg -snes_monitor_pause_final
```



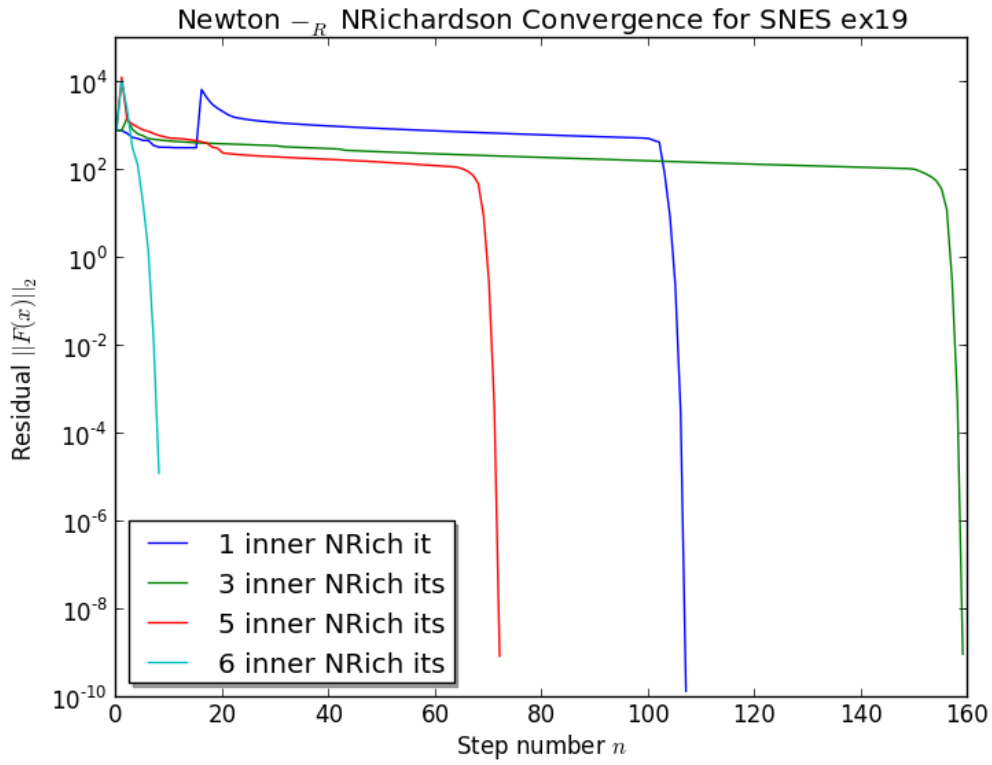
Preconditioning Newton with NRichardson

```
./ex19 -lidvelocity 100 -grashof 1.35e4
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type newtonls -snes_max_it 1000 -pc_type lu
  -npc_snes_type nrichardson -npc_snes_max_it 1
```



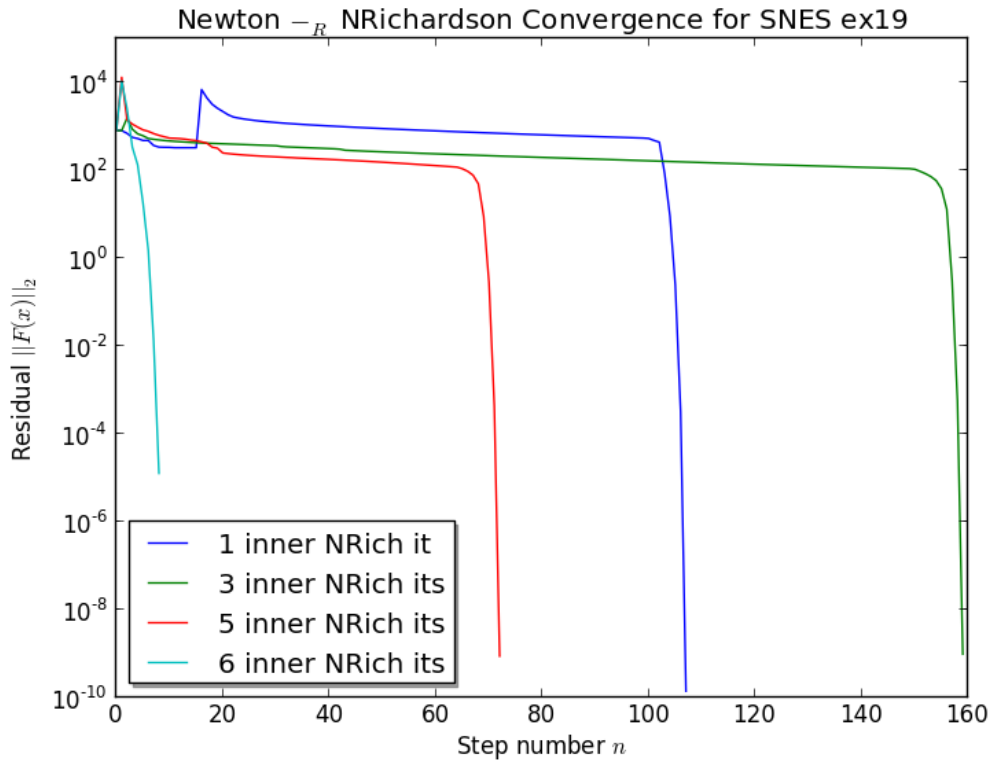
Preconditioning Newton with NRichardson

```
./ex19 -lidvelocity 100 -grashof 1.35e4
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type newtonls -snes_max_it 1000 -pc_type lu
  -npc_snes_type nrichardson -npc_snes_max_it 3
```



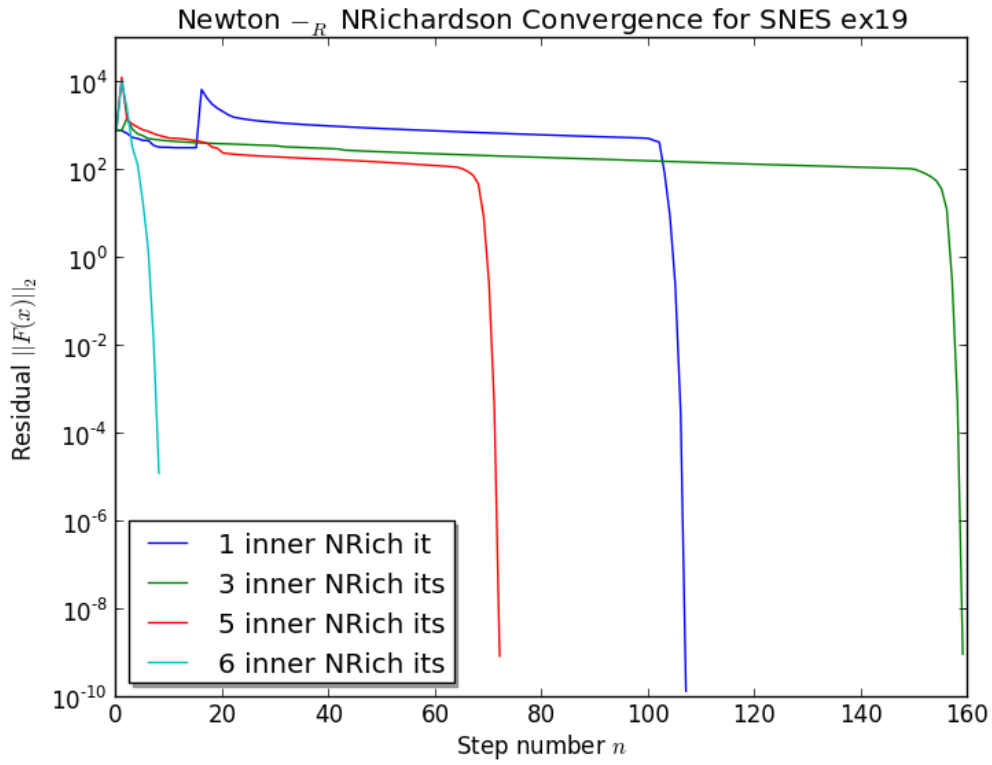
Preconditioning Newton with NRichardson

```
./ex19 -lidvelocity 100 -grashof 1.35e4
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type newtonls -snes_max_it 1000 -pc_type lu
  -npc_snes_type nrichardson -npc_snes_max_it 5
```



Preconditioning Newton with NRichardson

```
./ex19 -lidvelocity 100 -grashof 1.35e4
  -da_grid_x 16 -da_grid_y 16 -da_refine 2
  -snes_type newtonls -snes_max_it 1000 -pc_type lu
  -npc_snes_type nrichardson -npc_snes_max_it 6
```



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
  :
 61 SNES Function norm 320.325
 62 SNES Function norm 320.325
 63 SNES Function norm 320.325 21 SNES Function norm 585.143
  :
```

Nonlinear Preconditioning

```
./ex19 -lidvelocity 100 -grashof 5e4 -da_refine 4 -snes_monitor_short
 -snes_type fas -snes_converged_reason
 -fas_levels_snes_type ngs -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
  ...cycles without convergence...
```

Nonlinear Preconditioning

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


lid velocity = 100, prandtl # = 1, grashof # = 50000
  0 SNES Function norm 1228.95
  1 SNES Function norm 552.271
    Nonlinear npc_fas_coarse_ solve did not converge due to DIVERGED_M
  2 SNES Function norm 355.543
    Nonlinear npc_fas_coarse_ solve converged due to CONVERGED_FNORM_R
  :
  :
  38 SNES Function norm 3.74123e-05
    Nonlinear npc_fas_coarse_ solve converged due to CONVERGED_SNORM_R
  39 SNES Function norm 2.61273e-05
    Nonlinear npc_fas_coarse_ solve converged due to CONVERGED_SNORM_R
  40 SNES Function norm 1.16528e-05
Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE iterations 4
```

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 npc_fas_coarse_ solve converged due to CONVERGED_FNORM_R
  1 SNES Function norm 538.605
    Nonlinear npc_fas_coarse_ solve did not converge due to DIVERGED_M
  2 SNES Function norm 418.638
    Nonlinear npc_fas_coarse_ solve converged due to CONVERGED_FNORM_R
⋮


  24 SNES Function norm 0.000119837
    Nonlinear npc_fas_coarse_ solve converged due to CONVERGED_FNORM_R
  25 SNES Function norm 2.77111e-05
    Nonlinear npc_fas_coarse_ solve converged due to CONVERGED_SNORM_R
  26 SNES Function norm 1.11942e-05
Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE iterations 2
```

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_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 npc_fas_levels_4_ solve converged due to CONVERGED_ITS i
    :
        Nonlinear npc_fas_coarse_ solve converged due to CONVERGED_SNO
    :
  1 SNES Function norm 2.88944
  2 SNES Function norm 0.0803355
  3 SNES Function norm 0.0131489
  4 SNES Function norm 0.00118131
  5 SNES Function norm 1.65056e-05
  6 SNES Function norm 9.6292e-09
Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE iterations 6
```

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 ngs -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 538.539
  2 SNES Function norm 163.178
  3 SNES Function norm 49.5499
  4 SNES Function norm 11.5314
  5 SNES Function norm 0.182861
  6 SNES Function norm 0.00148081
  7 SNES Function norm 1.49034e-07
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 544.417
  2 SNES Function norm 18.2329
  3 SNES Function norm 1.05348
  4 SNES Function norm 0.00460528
  5 SNES Function norm 1.40734e-05
  6 SNES Function norm 3.07181e-08
  Nonlinear solve converged due to CONVERGED_FNORM_RELATIVE iterations
```

Nonlinear Preconditioning

Solver	T	N. It	L. It	Func	Jac	PC	NPC
$(\mathcal{N}\backslash\mathrm{K}-\mathrm{MG})$	9.83	17	352	34	85	370	–
NGMRES $-_R$ $(\mathcal{N}\backslash\mathrm{K}-\mathrm{MG})$	7.48	10	220	21	50	231	10
FAS	6.23	162	0	2382	377	754	–
FAS $+$ $(\mathcal{N}\backslash\mathrm{K}-\mathrm{MG})$	8.07	10	197	232	90	288	–
FAS $*$ $(\mathcal{N}\backslash\mathrm{K}-\mathrm{MG})$	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

- Hands-On Examples
  - Block Preconditioners
  - Multigrid
  - Nonlinear Preconditioning
  - Unstructured Mesh Creation*
  - ML*
  - Optimization*

Plex Tutorial ex2: source

Outline

- Hands-On Examples
  - Block Preconditioners
  - Multigrid
  - Nonlinear Preconditioning
  - Unstructured Mesh Creation*
  - ML*
  - Optimization*

PetscRegressor Tutorial and Notebook

Outline

- Hands-On Examples
  - Block Preconditioners
  - Multigrid
  - Nonlinear Preconditioning
  - Unstructured Mesh Creation*
  - ML*
  - Optimization*

Optimization Tutorial

Outline

Parallel Computing is Not Magic

Performance controlled by bandwidth and latency

- Bandwidth is determined by nodes, not cores
- Usually saturates with less than half the cores
- STREAM benchmark is a good measure

Parallel Computing is Not Magic

Most effective for

- increasing problem size (weak scaling),
- not faster solutions (strong scaling).

Strong scalability is limited by

- Ahmdahl's Law, but
- bandwidth saturation is often confused with this.

GPUs are Not Magic

Performance improvement is often quite limited:

- Bandwidth $<$ 10x of a top end CPU

  AMD EPYC 9000 1 TB/s, NVIDIA Grace 800 GB/s
  AMD MI325X 6 TB/s, NVIDIA Blackwell 8 TB/s

- Very high memory latency,
- Steep fall off for smaller loads

Toolchain is rapidly evolving and not robust.

Machine Learning is Not Magic

The sweet spot for Deep Neural Networks is

- many parameters (millions), and
- no knowledge of what is important.

Machine Learning is Not Magic

	Few Parameters	Many Parameters
Some idea	Logistic Reg.	Bayesian Inference
No idea	Logistic Reg.	Neural Networks

References I

[1] Junchao Zhang, Jed Brown, Satish Balay, Jacob Faibussowitsch, Matthew Knepley, Oana Marin, Richard Tran Mills, Todd Munson, Barry F. Smith, and Stefano Zampini. "The PetscSF Scalable Communication Layer". In: IEEE Transactions on Parallel and Distributed Systems 33.4 (2022), pp. 842–853. DOI: 10.1109/TPDS.2021.3084070.

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

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

[4] Shrirang Abhyankar, Jed Brown, Emil Constantinescu, Debojyoti Ghosh, and Barry F. Smith. PETSc/TS: A Modern Scalable DAE/ODE Solver Library. Preprint ANL/MCS-P5061-0114. ANL, Jan. 2014.

[5] Steven J Benson and Todd S Munson. "Flexible complementarity solvers for large-scale applications". In: Optimization Methods and Software 21.1 (2006), pp. 155–168.

[6] Carmen Campos, José E. Román, Eloy Romero, Andrés Tomás, Vicente Hernández, and Vicent Vidal. "SLEPc Website". Scalable Library for Eigenvalue Problem Computations. URL: http://slepc.upv.es/.

[7] Tobin Isaac and Matthew G. Knepley. "Support for Non-conformal Meshes in PETSc's DMPlex Interface". In: ArXiv e-prints (2017). http://arxiv.org/abs/1508.02470. eprint: 1508.02470.

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

[9] Toby Isaac. Unifying the geometric decompositions of full and trimmed polynomial spaces in finite element exterior calculus. 2022. arXiv: 2112.02174 [math.NA]. URL: https://arxiv.org/abs/2112.02174.

References II

[10] Gautam Bisht, Donghui Xu, Jeffrey Johnson, Jed Brown, Matthew G. Knepley, Mark F Adams, Dongyu Feng, Dalei Hao, Darren Engwirda, Mukesh Kumar, and Zeli Tan. "Development of a River Dynamical Core for E3sm to Simulate Compound Flooding on Exascale-Class Heterogeneous Supercomputers". In: Environmental Modelling & Software (2025). in review.

[11] Richard Tran Mills, Mark F. Adams, Satish Balay, Jed Brown, Alp Dener, Matthew Knepley, Scott E. Kruger, Hannah Morgan, Todd Munson, Karl Rupp, Barry F. Smith, Stefano Zampini, Hong Zhang, and Junchao Zhang. "Toward performance-portable PETSc for GPU-based exascale systems". In: Parallel Computing 108 (2021), p. 102831. ISSN: 0167-8191. DOI: https://doi.org/10.1016/j.parco.2021.102831. URL: https://www.sciencedirect.com/science/article/pii/S016781912100079X.

[12] Justin Chang, Maurice S. Fabien, Matthew G. Knepley, and Richard T. Mills. "Comparative study of finite element methods using the Time-Accuracy-Size (TAS) spectrum analysis". In: SIAM Journal on Scientific Computing 40.6 (2018), pp. C779–C802. DOI: 10.1137/18M1172260. eprint: 1802.07832.

[13] Satish Balay et al. PETSc Web page. https://petsc.org/. 2025. URL: https://petsc.org/.

[14] Satish Balay et al. PETSc/TAO Users Manual. Tech. rep. ANL-21/39 - Revision 3.23. Argonne National Laboratory, 2025. DOI: 10.2172/2565610.

Outline

Outline

Downloading PETSc

- There is a Git repository
- The latest tarball is on the PETSc site: https://web.cels.anl.gov/projects/petsc/download/release-snapshots/
- There is a pip package (`pip install petsc petsc4py`)
- There is a Debian package (`aptitude install petsc-dev`)

Cloning PETSc

- The full development repository is open to the public
  - https://gitlab.com/petsc/petsc/
- Why is this better?
  - You can clone to any release (or any specific ChangeSet)
  - You can easily rollback changes (or releases)
  - You can get fixes from us the same day
  - You can easily submit changes using a pull request
- All releases are just tags:
  - Source at tag v3.18.0

Unpacking PETSc

- Just clone development repository
  - `git clone http://gitlab.com/petsc/petsc.git`
  - `git checkout -rv3.24.0`

or

- Unpack the tarball
  - `tar xzf petsc.tar.gz`

Exercise 1

Download and Unpack PETSc!

Outline

Configuring PETSc

- Set `$PETSC_DIR` to the installation root directory
- Run the configuration utility
  - `$PETSC_DIR/configure`
  - `$PETSC_DIR/configure --help`
  - `$PETSC_DIR/configure --download-mpich`
  - `$PETSC_DIR/configure --prefix=/usr`
- There are many examples in `$PETSC_DIR/config/examples`
- Config files in `$PETSC_DIR/$PETSC_ARCH/lib/petsc/conf`
  - Config header in `$PETSC_DIR/$PETSC_ARCH/include`
  - `$PETSC_ARCH` has a default if not specified

Configuring PETSc

- You can easily reconfigure with the same options
  -
    ```
    ./$PETSC_ARCH/lib/petsc/conf/reconfigure-$PETSC_ARCH.py
    ```
- Can maintain several different configurations
  -
    ```
    ./configure -PETSC_ARCH=arch-linux-opt --with-debugging=0
    ```
- All configuration information is in the logfile
  - `./$PETSC_ARCH/lib/petsc/conf/configure.log`
  - ALWAYS send this file with bug reports

Configuring PETSc for FEM

$PETSC_DIR/configure

–download-triangle –download-ctetgen
–download-p4est –download-egads

Configuring PETSc for FEM

$PETSC_DIR/configure

–download-triangle –download-ctetgen
–download-p4est –download-egads

–download-eigen –download-pragmatic
–download-metis –download-parmetis

Configuring PETSc for FEM

$PETSC_DIR/configure

–download-triangle –download-ctetgen
–download-p4est –download-egads

–download-eigen –download-pragmatic
–download-metis –download-parmetis

–download-hdf5 –download-netcdf
–download-pnetcdf –download-exodusii

Configuring PETSc for FEM

$PETSC_DIR/configure

–download-triangle –download-ctetgen
–download-p4est –download-egads

–download-eigen –download-pragmatic
–download-metis –download-parmetis

–download-hdf5 –download-netcdf
–download-pnetcdf –download-exodusii

–download-ml –download-superlu –download-superlu_dist
–download-scalapack –download-mumps

Configuring PETSc for FEM

$PETSC_DIR/configure

–download-triangle –download-ctetgen
–download-p4est –download-egads

–download-eigen –download-pragmatic
–download-metis –download-parmetis

–download-hdf5 –download-netcdf
–download-pnetcdf –download-exodusii

–download-ml –download-superlu –download-superlu_dist
–download-scalapack –download-mumps

–download-fftw –download-slepc –download-bamg

Configuring PETSc for Acclerators

```
$PETSC_DIR/configure
        –with-precision=single
```

Configuring PETSc for Acclerators

$PETSC_DIR/configure

–with-precision=single

–with-cuda

Configuring PETSc for Acclerators

$PETSC_DIR/configure

–with-precision=single

–with-cuda

–with-cudac=’nvcc -m64’ –with-cuda-arch=sm_10

Configuring PETSc for Acclerators

$PETSC_DIR/configure

–with-precision=single

–with-opencl

Configuring PETSc for Acclerators

$PETSC_DIR/configure

–with-precision=single

–with-opencl

–with-opencl-include=/System/Library/Frameworks/OpenCL.framework/Headers/

–with-opencl-lib=/System/Library/Frameworks/OpenCL.framework/OpenCL

Configuring PETSc for Acclerators

$PETSC_DIR/configure

–with-precision=single

–download-kokkos –download-kokkos-kernels

Configuring PETSc for Acclerators

$PETSC_DIR/configure

–with-precision=single

–with-cuda

–download-kokkos –download-kokkos-kernels

Configuring PETSc for Acclerators

$PETSC_DIR/configure

–with-precision=single

–download-kokkos –download-kokkos-kernels

–with-hip

Configuring PETSc for Acclerators

$PETSC_DIR/configure

–with-precision=single

–download-kokkos –download-kokkos-kernels

–with-sycl

Configuring PETSc for Acclerators

$PETSC_DIR/configure

–with-precision=single

–download-kokkos –download-kokkos-kernels

–with-openmp

Automatic Downloads

- Starting in 2.2.1, some packages are automatically
  - Downloaded
  - Configured and Built (in `$PETSC_DIR/externalpackages`)
  - Installed with PETSc
- Currently works for
  - petsc4py, mpi4py
  - PETSc documentation utilities (Sowing, c2html)
  - BLAS, LAPACK, Elemental, ScaLAPACK
  - GMP, MPFR
  - ConcurrencyKit, hwloc
  - MPICH, OpenMPI
  - ParMetis, Chaco, Jostle, Party, Scotch, Zoltan
  - SuiteSparse, MUMPS, SuperLU, SuperLU_Dist, PaStiX, Pardiso
  - SLEPc, HYPRE, ML
  - BLOPEX, FFTW, STRUMPACK, SPAI, CUSP, Sundials
  - Triangle, TetGen, p4est, Pragmatic
  - HDF5, NetCDF, ExodusII
  - AfterImage, gifLib, libjpeg, opengl

Exercise 2

Configure your downloaded PETSc.

Outline

Building PETSc

- There is now One True Way to build PETSc:
  - `make`
  - `make install` if you configured with `--prefix`
  - Check build when done with `make check`
- Can build multiple configurations
  - `PETSC_ARCH=arch-linux-opt make`
  - Libraries are in `$PETSC_DIR/$PETSC_ARCH/lib/`
- Complete log for each build is in logfile
  - `./$PETSC_ARCH/lib/petsc/conf/make.log`
  - ALWAYS send this with bug reports

Exercise 3

Build your configured PETSc.

Exercise 4

Reconfigure PETSc to use ParMetis.

1. `linux-debug/lib/petsc/conf/reconfigure-linux-debug.py`
   - `--PETSC_ARCH=arch-linux-parmetis`
   - `--download-metis --download-parmetis`
2. `PETSC_ARCH=linux-parmetis make`
3. `PETSC_ARCH=linux-parmetis make check`