# Computational Magma Dynamics

Matthew Knepley $\in$ PETSc Team

Computation Institute
University of Chicago

SIAM Computational Science & Engineering
Salt Lake City, UT March 16, 2015

# Collaborators

Magma Dynamics Collaboration

Richard Katz

Timestepping Collaboration

Jed Brown

# Main Point

## What am I going to talk about?

- Magma Dynamics Model
- FEM formulation and solver
- FEM+FVM formulation and solver

## Why is this important?

- It is difficult to compare meshes, discretizations, and multilevel solvers
- Comparison is essential for making informed algorithmic choices
- Comparison in a single code seems necessary

# Outline

# Dimensional Formulation

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

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

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

# Closure Conditions

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

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

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

# Nondimensional Formulation

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

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

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

# Initial and Boundary conditions

Initially

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

where

$$A \ll \phi_0$$

and on the top and bottom boundary

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

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

# Mechanical Benchmarks

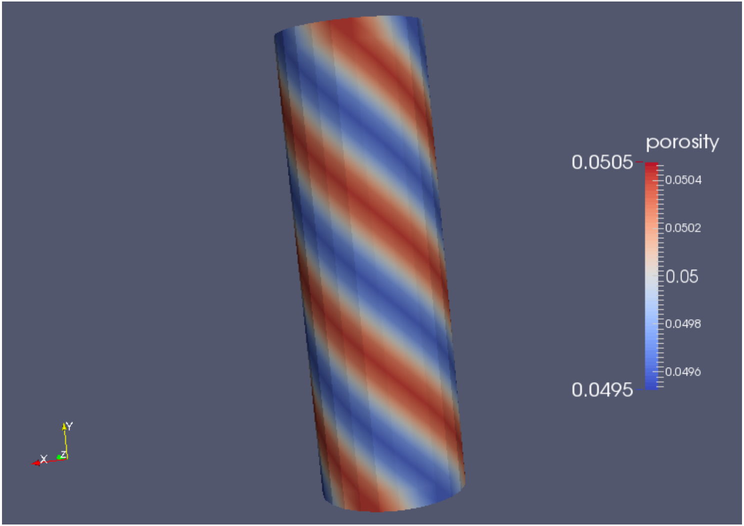
**Benchmark 0:** $\lambda = 0$
There is no porosity feedback, and the initial pattern is stably advected:

$$\vec{k}(t) = \vec{k}_0 \left( \hat{\vec{x}} \sin\theta_0 + \hat{\vec{y}} \left( \cos\theta_0 - t \sin\theta_0 \right) \right)$$

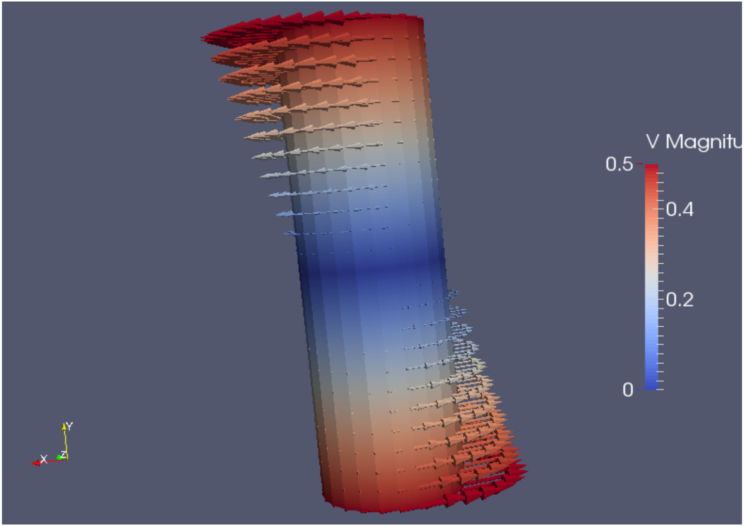
**Benchmark 1:** $\lambda > 0$
The porosity feedback causes localization, with initial compaction rate:

$$\mathcal{C} = \nabla \cdot \vec{v}_S \big|_{t=0} = \frac{A \lambda \phi_0 \sin(2\theta_0)}{r_\zeta + 4/3} \cos(\vec{k} \cdot \vec{x})$$

# Initial Porosity



# Initial Velocity



# Outline

# Mesh

With PETSc's DMPlex, we can use

- Simplices,
- Hexes,
- 2D and 3D,

changing nothing but mesh creation.

I do this in SNES ex62

# Output

PETSc Viewers can output

- Meshes
- Solutions,
- Auxiliary and derived fields,

to HDF5/Xdmf using simple options.

# Output

PETSc Viewers can output

```
-dm_view hdf5:sol_fv_1.h5
-magma_view_solution hdf5:sol_fv_1.h5::append
-compaction_vec_view hdf5:sol_fv_1.h5:HDF5_VIZ:append
```

to HDF5/Xdmf using simple options.

# Discretization

Using continuous FE spaces,

which satisfy an inf-sup stability condition.

# Discretization

Using continuous FE spaces,

| | |
|---|---|
| $Q2$ | velocity |
| $Q1$ | pressure |
| $Q1$ | porosity |

which satisfy an inf-sup stability condition.

# Discretization

Using continuous FE spaces,

```
-velocity_petscspace_order 2
  -velocity_petscspace_poly_tensor
-pressure_petscspace_order 1
  -pressure_petscspace_poly_tensor
-porosity_petscspace_order 1
  -porosity_petscspace_poly_tensor
```

which satisfy an inf-sup stability condition.

## Discretization

Using continuous/discontinuous FE spaces,

$$\begin{array}{ll} Q2 & \text{velocity} \\ P1_{\mathrm{disc}} & \text{pressure} \\ Q1 & \text{porosity} \end{array}$$

which satisfy an inf-sup stability condition.

# Discretization

Using continuous/discontinuous FE spaces,

```
-velocity_petscspace_order 2
  -velocity_petscspace_poly_tensor
-pressure_petscspace_order 1
  -pressure_petscdualspace_lagrange_continuity 0
-porosity_petscspace_order 1
  -porosity_petscspace_poly_tensor
```

which satisfy an inf-sup stability condition.

# Solver Organization

## Timestepping

We will use simple Backward Euler:

$$\nabla p^{k+1} - \nabla\left(\left(\frac{\phi^{k+1}}{\phi_0}\right)^{-m}\nabla\cdot\vec{v}^{k+1}\right) - \nabla\cdot\left(2e^{-\lambda(\phi^{k+1}-\phi_0)}\dot{\epsilon}^{k+1}\right) = 0$$

$$\nabla\cdot\left(-\frac{R^2}{r_\zeta + 4/3}\left(\frac{\phi^{k+1}}{\phi_0}\right)^{n}\nabla p^{k+1} + \vec{v}^{k+1}\right) = 0$$

$$\frac{\phi^{k+1}-\phi^{k}}{\Delta t} - \nabla\cdot(1-\phi^{k+1})\vec{v}^{k+1} = 0$$

# Solver Organization

## Newton-Krylov

Begin with a Newton-Krylov solve with line search:

$$\mathcal{N} \backslash \mathrm{K} -_{L} \mathrm{NRICH}$$

Optimal linear preconditioner in
Rhebergen, Wells, Wathen, and Katz, SISC.

# Solver Organization

## Newton-Krylov without Porosity

We can separate the Stokes-like solve from the porosity advection:

$$
\left(\begin{array}{cc|c}
\multicolumn{2}{c|}{A \oplus_{\mathrm{Schur}} L} & \begin{array}{c}0\\0\end{array} \\
\hline
F & 0 & I+G
\end{array}\right)
$$

# Solver Organization

## Newton-Krylov

```
-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
```

# Solver Organization

## Newton-Krylov with Porosity

Or we can incorporate the porosity advection into the Stokes-like solve:

$$\begin{array}{c|c} A & E \\ \hline F & I{+}G \end{array} \bigoplus_{\text{Schur}} L$$

# Newton options

## Newton-Krylov with Porosity

```
-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
```

# 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
   ⋮
```

# Solver Organization

## Preconditioned Newton-Krylov

We can combine Newton-Krylov with Nonlinear CG:

$$(\mathrm{NCG} -_L \mathrm{NRICH}) * (\mathcal{N}\backslash\mathrm{K} -_L \mathrm{NRICH})$$

# 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
```

# Solver Organization

## Full Approximation Scheme

We can use Newton-Krylov as a level solver for FAS:

$$\mathrm{FAS}(\mathcal{N}\backslash \mathrm{K}, \mathcal{N}\backslash \mathrm{K})$$

# FAS-Newton 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-Newton 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-Newton 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-Newton 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-Newton 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
```

# Solver Organization

## Full Approximation Scheme

On fine levels, we can replace Newton-Krylov with

Nonlinear Gauss-Siedel:

$$\mathrm{FAS}(\mathrm{NGS}, \mathcal{N}\backslash\mathrm{K})$$

# FAS-NGS 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 10
  -fas_levels_3_snes_type ngs -fas_levels_3_snes_linesearch_type nleqerr
```

# FAS-NGS convergence

```
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 9.68e-03 [1.96e-03, 1.71e-14, 9.65e-03]
      0 SNES Function norm 9.682733054059e-03
      3 SNES Function norm 9.069944580453e-01
          3 SNES Function norm 3.790367845975e-11
        0 SNES Function norm 1.884126634610e+00
        1 SNES Function norm 6.752057466899e-02
          2 SNES Function norm 3.799909413083e-11
        0 SNES Function norm 1.450032375835e-01
        1 SNES Function norm 2.567674743706e-04
      0 SNES Function norm 1.027806561203e+00
      3 SNES Function norm 1.582489644172e+00
        1 SNES Function norm 4.847533456932e-01
          3 SNES Function norm 7.366666076108e-15
        1 SNES Function norm 1.744390611632e-02
      3 SNES Function norm 1.473321454964e+00
    1 SNES Function norm 1.47e+00 [1.44e+00, 2.92e-01, 8.82e-04]
      0 SNES Function norm 9.962396109825e+03
      1 SNES Function norm 3.189537494940e+86
    Nonlinear fas_levels_2_ solve did not converge, DIVERGED_FNORM_NAN
  Nonlinear solve did not converge due to DIVERGED_INNER
```

# Outline

# Discretization

Using continuous FE spaces and simple FV,

| | |
|---|---|
| $Q2$ | velocity |
| $Q1$ | pressure |
| $FV$ | porosity |

which we connect by cell/face interpolants.

```
/* Set discretization object */
if (user->useFV) {
  PetscDSSetDiscretization(prob, 2, fv);
} else {
  PetscDSSetDiscretization(prob, 2, fe[2]);
}
/* Set pointwise residual functions */
PetscDSSetResidual(prob, 2, f0_advection, f1_scalar_zero);
PetscDSSetRiemannSolver(prob, 2, riemann_coupled_advection);
```

# Discretization

Using continuous FE spaces and simple FV,

```
-velocity_petscspace_order 2
  -velocity_petscspace_poly_tensor
-pressure_petscspace_order 1
  -pressure_petscspace_poly_tensor
-use_fv
```

which we connect by cell/face interpolants.

```
/* Set discretization object */
if (user->useFV) {
  PetscDSSetDiscretization(prob, 2, fv);
} else {
  PetscDSSetDiscretization(prob, 2, fe[2]);
}
/* Set pointwise residual functions */
PetscDSSetResidual(prob, 2, f0_advection, f1_scalar_zero);
PetscDSSetRiemannSolver(prob, 2, riemann_coupled_advection);
```

# Discretization

Using continuous FE spaces and simple FV,

$$\begin{array}{ll} Q2 & \text{velocity} \\ Q1 & \text{pressure} \\ FV & \text{porosity} \end{array}$$

which we connect by cell/face interpolants.

```
/* Set discretization object */
if (user->useFV) {
  PetscDSSetDiscretization(prob, 2, fv);
} else {
  PetscDSSetDiscretization(prob, 2, fe[2]);
}
/* Set pointwise residual functions  */
PetscDSSetResidual(prob, 2, f0_advection, f1_scalar_zero);
PetscDSSetRiemannSolver(prob, 2, riemann_coupled_advection);
```

# Solver Organization

## Timestepping

We can use a simple split scheme:

$$\nabla p^{k+1} - \nabla \left( \left( \frac{\phi^{k}}{\phi_0} \right)^{-m} \nabla \cdot \vec{v}^{k+1} \right) - \nabla \cdot \left( 2 e^{-\lambda(\phi^{k} - \phi_0)} \dot{\epsilon}^{k+1} \right) = 0$$

$$\nabla \cdot \left( -\frac{R^2}{r_\zeta + 4/3} \left( \frac{\phi^{k}}{\phi_0} \right)^{n} \nabla p^{k+1} + \vec{v}^{k+1} \right) = 0$$

$$\frac{\phi^{k+1} - \phi^{k}}{\Delta t} - \nabla \cdot (1 - \phi^{k}) \vec{v}^{k+1} = 0$$

# Solver Organization

## Timestepping

Or one that couples the algebraic and evolution equations:

$$\nabla p^{k+1} - \nabla \left( \left( \frac{\phi^{k+1}}{\phi_0} \right)^{-m} \nabla \cdot \vec{v}^{k+1} \right) - \nabla \cdot \left( 2 e^{-\lambda(\phi^{k+1} - \phi_0)} \dot{\epsilon}^{k+1} \right) = 0$$

$$\nabla \cdot \left( -\frac{R^2}{r_\zeta + 4/3} \left( \frac{\phi^{k+1}}{\phi_0} \right)^{n} \nabla p^{k+1} + \vec{v}^{k+1} \right) = 0$$

$$\frac{\phi^{k+1} - \phi^k}{\Delta t} - \nabla \cdot (1 - \phi^k) \vec{v}^{k+1} = 0$$

# Newton options

```
-snes_atol 1.0e-10 -snes_monitor_field -snes_converged_reason
-snes_linesearch_type basic -snes_fd_color -snes_fd_color_use_mat
-mat_coloring_type greedy -mat_coloring_greedy_symmetric 0
 -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_ksp_rtol 1.0e-8  -fieldsplit_0_pc_type lu
  -fieldsplit_pressure_ksp_rtol 1.0e-9 -fieldsplit_pressure_pc_type svd
```

# Early Newton convergence

```
5 TS dt 0.005 time 0.025
    0 SNES Function norm 6.52e-02 [1.46e-14, 4.91e-16, 6.52e-02]
      0 KSP Residual norm 4.26e-04
      1 KSP Residual norm 1.78e-17
    1 SNES Function norm 2.19e-03 [2.96e-08, 3.91e-09, 2.19e-03]
    2 SNES Function norm 7.51e-05 [3.40e-11, 4.55e-12, 7.51e-05]
    3 SNES Function norm 2.58e-06 [2.46e-13, 1.28e-14, 2.58e-06]
    4 SNES Function norm 8.86e-08 [1.39e-14, 6.64e-16, 8.86e-08]
6 TS dt 0.005 time 0.03
```

# Late Newton convergence

```
0 TS dt 0.005 time 0.825
    0 SNES Function norm 2.14e+00 [  1.40e-14,   3.67e-16,   2.14e+00]
      0 KSP Residual norm 3.53e-01
      1 KSP Residual norm 1.03e-10
      2 KSP Residual norm 2.82e-16
    1 SNES Function norm 5.13e-02 [  2.01e-04,   1.47e-04,   5.13e-02]
    2 SNES Function norm 2.47e-02 [  9.20e-06,   7.73e-06,   2.47e-02]
    3 SNES Function norm 7.81e-03 [  2.13e-06,   1.67e-06,   7.81e-03]
    4 SNES Function norm 2.12e-03 [  1.81e-07,   1.41e-07,   2.12e-03]
    5 SNES Function norm 4.72e-04 [  1.08e-08,   8.28e-09,   4.72e-04]
    6 SNES Function norm 1.12e-04 [  5.76e-10,   4.41e-10,   1.12e-04]
    7 SNES Function norm 2.63e-05 [  3.21e-11,   2.50e-11,   2.63e-05]
    8 SNES Function norm 6.17e-06 [  1.77e-12,   1.26e-12,   6.17e-06]
    9 SNES Function norm 1.45e-06 [  1.07e-13,   9.84e-14,   1.45e-06]
   10 SNES Function norm 3.40e-07 [  1.78e-14,   4.74e-15,   3.40e-07]
   11 SNES Function norm 7.99e-08 [  1.36e-14,   1.88e-15,   7.99e-08]
   12 SNES Function norm 1.88e-08 [  1.34e-14,   5.72e-16,   1.88e-08]
1 TS dt 0.005 time 0.83
```

# FAS-Newton options

## Top level

```
-snes_atol 1.0e-9 -snes_monitor_field -snes_converged_reason
-snes_type fas -snes_fas_type full -snes_fas_levels 3
 -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 basic
  -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 svd
     -fas_levels_2_fieldsplit_pressure_ksp_gmres_restart 100
     -fas_levels_2_fieldsplit_pressure_ksp_max_it 200
```

# FAS-Newton options

## Coarse level

```
-fas_coarse_snes_max_it 10 -fas_coarse_snes_max_linear_solve_fail 10
-fas_coarse_snes_atol 1.0e-9
-fas_coarse_snes_monitor -fas_coarse_snes_converged_reason
-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 svd
```

# FAS-Newton convergence

```
0 TS dt 0.005 time 0.825
    0 SNES Function norm 2.14e+00 [1.40e-14, 3.67e-16, 2.14e+00]
      0 SNES Function norm 2.136811983007e+00
      2 SNES Function norm 2.467490038458e-02
          0 SNES Function norm 2.892788645925e-02
          5 SNES Function norm 6.686368379854e-11
        0 SNES Function norm 5.034219273717e-02
        1 SNES Function norm 1.054842559307e-03
          0 SNES Function norm 1.663080254945e-03
          4 SNES Function norm 2.126370356882e-10
        0 SNES Function norm 2.599480303180e-03
        1 SNES Function norm 9.990047497418e-05
      0 SNES Function norm 4.798584798600e-02
      2 SNES Function norm 1.288870672992e-03
        1 SNES Function norm 3.770621359658e-05
          2 SNES Function norm 1.127970439777e-08
        1 SNES Function norm 1.008431552413e-06
      0 SNES Function norm 2.502531975042e-03
      2 SNES Function norm 4.730240156687e-05
    1 SNES Function norm 4.73e-05 [1.04e-10, 7.85e-11, 4.73e-05]
    2 SNES Function norm 3.98e-09 [1.38e-14, 4.18e-16, 3.98e-09]
1 TS dt 0.005 time 0.83
```

# FAS-NGS options

## Top level

```
-snes_atol 1.0e-9 -snes_monitor_field -snes_converged_reason
-snes_type fas -snes_fas_type full -snes_fas_levels 3
  -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 10
  -fas_levels_2_snes_type ngs -fas_levels_2_snes_linesearch_type bt
```

# FAS-NGS convergence

```
0 TS dt 0.005 time 0.825
    0 SNES Function norm 2.14e+00 [1.39e-14, 3.66e-16, 2.13e+00]
    1 SNES Function norm 2.87e-04 [2.08e-04, 4.80e-06, 1.98e-04]
    2 SNES Function norm 3.15e-05 [2.30e-05, 9.56e-07, 2.19e-05]
    3 SNES Function norm 1.65e-05 [1.14e-05, 5.44e-07, 1.21e-05]
    4 SNES Function norm 1.07e-05 [7.38e-06, 3.48e-07, 7.89e-06]
    5 SNES Function norm 7.06e-06 [4.85e-06, 2.26e-07, 5.20e-06]
    6 SNES Function norm 4.67e-06 [3.20e-06, 1.48e-07, 3.44e-06]
    7 SNES Function norm 3.09e-06 [2.12e-06, 9.82e-08, 2.27e-06]
    8 SNES Function norm 2.05e-06 [1.40e-06, 6.52e-08, 1.50e-06]
    9 SNES Function norm 1.36e-06 [9.35e-07, 4.34e-08, 1.00e-06]
   10 SNES Function norm 9.03e-07 [6.21e-07, 2.89e-08, 6.64e-07]
   11 SNES Function norm 6.00e-07 [4.13e-07, 1.94e-08, 4.41e-07]
   12 SNES Function norm 3.99e-07 [2.75e-07, 1.30e-08, 2.94e-07]
   13 SNES Function norm 2.67e-07 [1.84e-07, 8.84e-09, 1.96e-07]
   14 SNES Function norm 1.78e-07 [1.23e-07, 6.01e-09, 1.31e-07]
   15 SNES Function norm 1.20e-07 [8.31e-08, 4.12e-09, 8.80e-08]
   16 SNES Function norm 8.12e-08 [5.64e-08, 2.85e-09, 5.94e-08]
   17 SNES Function norm 5.55e-08 [3.87e-08, 1.99e-09, 4.05e-08]
   18 SNES Function norm 3.86e-08 [2.70e-08, 1.41e-09, 2.80e-08]
   19 SNES Function norm 2.74e-08 [1.93e-08, 1.01e-09, 1.97e-08]
   20 SNES Function norm 2.00e-08 [1.43e-08, 7.48e-10, 1.44e-08]
1 TS dt 0.005 time 0.83
```

# What does this mean?

- PETSc allows comparsion between
  Meshes,
  Discretizations, and
  Solvers
- Can allow more robust, scalable solves
- Can allow better physical fidelity

http://www.mcs.anl.gov/petsc

# What does this mean?

There are a bunch of unanswered analytical questions:

- Is the hybrid method stable? Working on a proof.
- Can we use the Implicit-Input/Explicit-Output scheme to increase the timestep?
- What is the accuracy/degree of freedom?

http://www.mcs.anl.gov/petsc