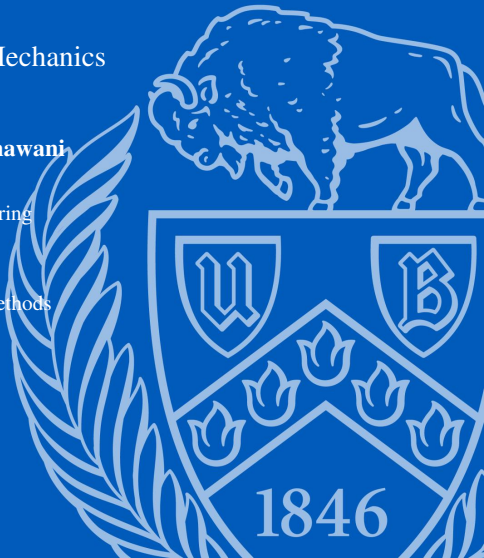
Solving Systems in Droplet Mechanics

Matt Knepley and Darsh Nathawani

Computer Science and Engineering
University at Buffalo

Acceleration and Extrapolation Methods
ICERM, Providence, RI
July 26, 2023



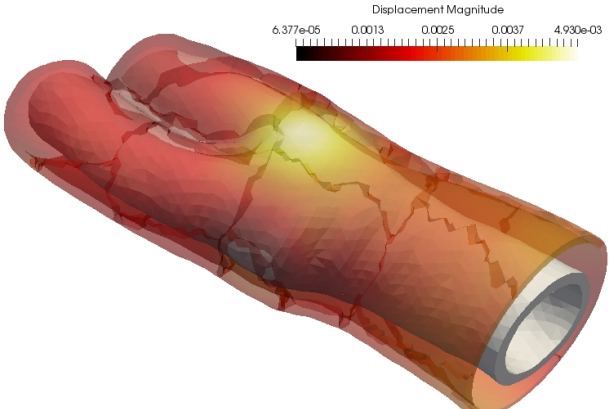
University at Buffalo
Center for Hybrid Rocket
Exascale Simulation Technology

Never believe anything

until you run it.

Pioneering Work

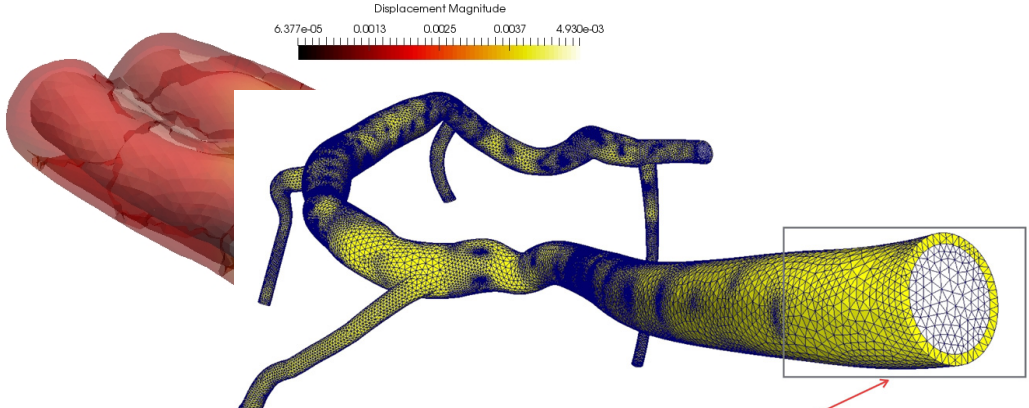
Nonlinear Elimination for Computational Hyperelasticity
Shihua Gong and Xiao-Chuan Cai



Pioneering Work

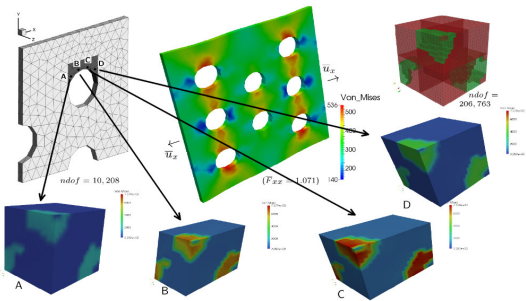
Nonlinear Elimination for Computational Hyperelasticity
Shihua Gong and Xiao-Chuan Cai

Nonlinear Elimination for Fluid-Structure Interaction
Fande Kong and Xiao-Chuan Cai



Pioneering Work

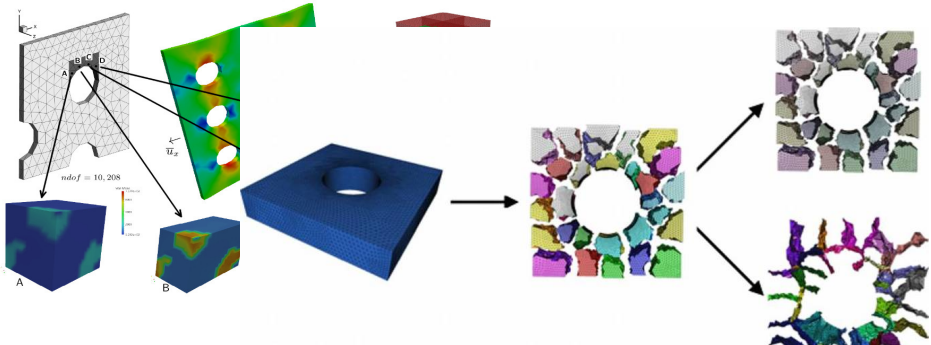
Nonlinear FETI-DP and BDDC methods: A unified framework and parallel results
Axel Klawonn, Martin Lanser, Oliver Rheinbach, Matthias Uran



Pioneering Work

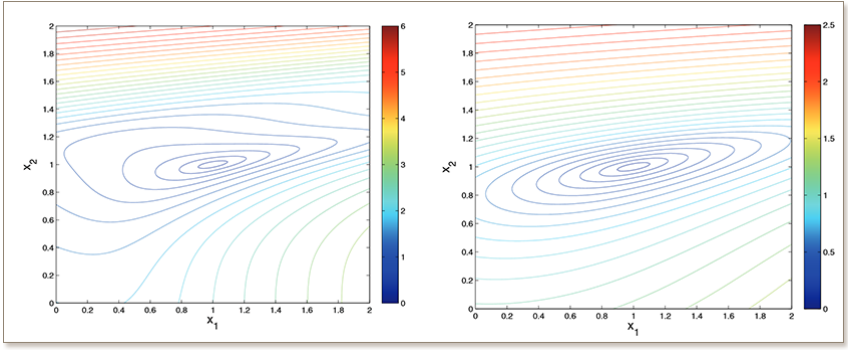
Nonlinear FETI-DP and BDDC methods: A unified framework and parallel results
Axel Klawonn, Martin Lanser, Oliver Rheinbach, Matthias Uran

A Highly Scalable Implementation of Inexact Nonlinear FETI-DP Without Sparse Direct Solvers
Axel Klawonn, Martin Lanser, Oliver Rheinbach



Pioneering Work

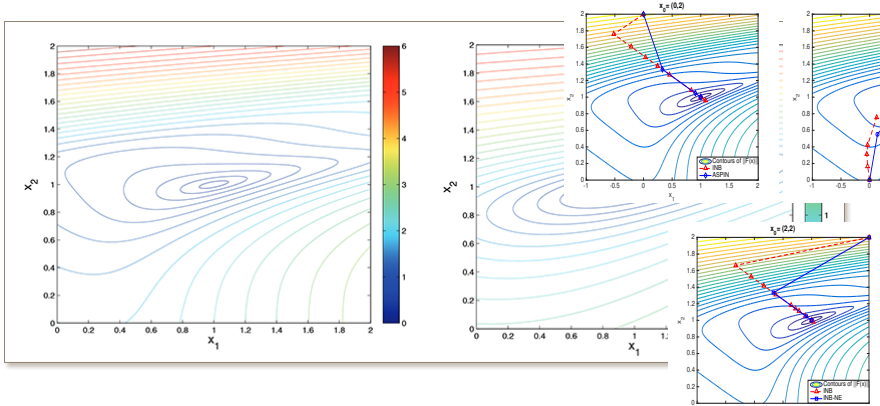
Field-split preconditioned inexact Newton algorithms
Lulu Liu and David E. Keyes



Pioneering Work

Field-split preconditioned inexact Newton algorithms
Lulu Liu and David E. Keyes

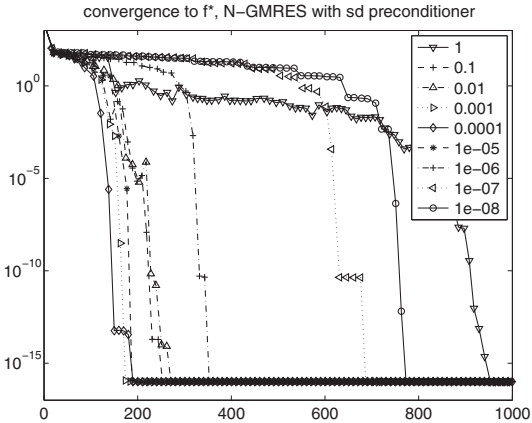
A Note on Adaptive Nonlinear Preconditioning Techniques
Lulu Liu, David E. Keyes, Rolf Krause



Pioneering Work

Steepest Descent Preconditioning for NGMRES Optimization

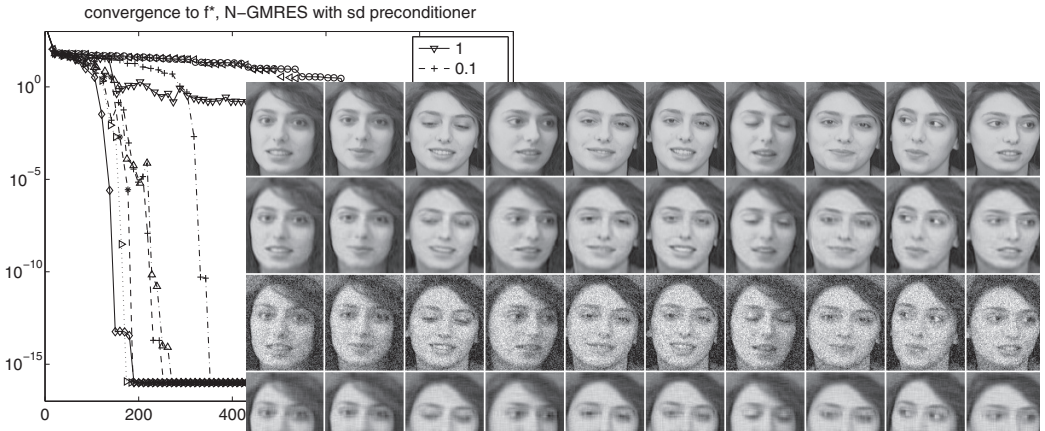
Hans De Sterck



Pioneering Work

Steepest Descent Preconditioning for NGMRES Optimization
Hans De Sterck

Nonlinear Preconditioning for Tucker Decomposition
Alexander Howse and Hans De Sterck



Outline

Abstract System

Out prototypical nonlinear equation is:

$$\mathbf{F}(\mathbf{x}) = \mathbf{b}$$

and we define the residual as

$$\mathbf{r}(\mathbf{x}) = \mathbf{F}(\mathbf{x}) - \mathbf{b}$$

Abstract System

Out prototypical nonlinear equation is:

$$\mathbf{F}(\mathbf{x}) = \mathbf{b}$$

and we define the (linear) residual as

$$\mathbf{r}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$$

Linear Left Preconditioning

The modified equation becomes

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

Linear Left Preconditioning

The modified defect correction equation becomes

$$P^{-1}\left(A\mathbf{x}_i - \mathbf{b}\right) = \mathbf{x}_{i+1} - \mathbf{x}_i \tag{2}$$

Additive Combination

The linear iteration

$$\mathbf{x}_{i+1} = \mathbf{x}_i - (\alpha P^{-1} + \beta Q^{-1})(A\mathbf{x}_i - \mathbf{b}) \tag{3}$$

becomes the nonlinear iteration

Additive Combination

The linear iteration

$$\mathbf{x}_{i+1} = \mathbf{x}_i - (\alpha P^{-1} + \beta Q^{-1})\mathbf{r}_i \tag{4}$$

becomes the nonlinear iteration

Additive Combination

The linear iteration

$$\mathbf{x}_{i+1} = \mathbf{x}_i - (\alpha P^{-1} + \beta Q^{-1})\mathbf{r}_i \tag{4}$$

becomes the nonlinear iteration

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha(\mathcal{N}(\mathbf{F}, \mathbf{x}_i, \mathbf{b}) - \mathbf{x}_i) + \beta(\mathcal{M}(\mathbf{F}, \mathbf{x}_i, \mathbf{b}) - \mathbf{x}_i) \tag{5}$$

Nonlinear Left Preconditioning

From the additive combination, we have

$$P^{-1}\mathbf{r} \Longrightarrow \mathbf{x}_i - \mathcal{N}(\mathbf{F}, \mathbf{x}_i, \mathbf{b}) \tag{6}$$

so we define the preconditioning operation as

$$\mathbf{r}_L \equiv \mathbf{x} - \mathcal{N}(\mathbf{F}, \mathbf{x}, \mathbf{b}) \tag{7}$$

Multiplicative Combination

The linear iteration

$$\mathbf{x}_{i+1} = \mathbf{x}_i - (P^{-1} + Q^{-1} - Q^{-1}AP^{-1})\mathbf{r}_i \tag{8}$$

becomes the nonlinear iteration

Multiplicative Combination

The linear iteration

$$\mathbf{x}_{i+1/2} = \mathbf{x}_i - P^{-1}\mathbf{r}_i \tag{9}$$

$$\mathbf{x}_i = \mathbf{x}_{i+1/2} - Q^{-1}\mathbf{r}_{i+1/2} \tag{10}$$

becomes the nonlinear iteration

Multiplicative Combination

The linear iteration

$$\mathbf{x}_{i+1/2} = \mathbf{x}_i - P^{-1}\mathbf{r}_i \tag{9}$$

$$\mathbf{x}_i = \mathbf{x}_{i+1/2} - Q^{-1}\mathbf{r}_{i+1/2} \tag{10}$$

becomes the nonlinear iteration

$$\mathbf{x}_{i+1} = \mathcal{M}(\mathbf{F},\, \mathcal{N}(\mathbf{F}, \mathbf{x}_i, \mathbf{b}),\, \mathbf{b}) \tag{11}$$

Nonlinear Right Preconditioning

For the linear case, we have

$$AP^{-1}\mathbf{y} = \mathbf{b} \tag{12}$$

$$\mathbf{x} = P^{-1}\mathbf{y} \tag{13}$$

so we define the preconditioning operation as

$$\mathbf{y} = \mathcal{M}(\mathbf{F}(\mathcal{N}(\mathcal{F}, \cdot, \mathbf{b})),\ \mathbf{x}_i, \mathbf{b}) \tag{14}$$

$$\mathbf{x} = \mathcal{N}(\mathbf{F}, \mathbf{y}, \mathbf{b}) \tag{15}$$

Nonlinear Preconditioning

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

Composing Scalable Nonlinear Algebraic Solvers (**bruneknepleysmithtu15**)

Newton

```
./bubble
-h_0 0.00135 -u_0 0.01 -rho_d 1.20 -nu_d 1.5E-05 -gamma 0.0728 -rho_c 1000.00
-nu_c 1.0E-06 -R 0.0254 -G 0.00 -gr -9.81
-cells 100 -dm_plex_transform_type refine_1d -dm_plex_hash_location
-vel_petscspace_degree 3 -rad_petscspace_degree 3 -slope_petscspace_degree 2
-ts_dt 1e-6 -ts_type beuler -ts_max_reject 20 -ts_max_steps 1 -ts_monitor
-snes_converged_reason -snes_max_funcs 1000000 -snes_mf_operator -snes_view
-ksp_rtol 1e-10 -pc_type lu
```

Newton

```
SNES Object: 1 MPI process
  type: newtonls
  maximum iterations=50, maximum function evaluations=1000000
  tolerances: relative=1e-08, absolute=1e-50, solution=1e-08
  total number of linear solver iterations=6
  total number of function evaluations=20
  norm schedule ALWAYS
  Jacobian is applied matrix-free with differencing
  SNESLineSearch Object: 1 MPI process
    type: bt
      interpolation: cubic
      alpha=1.000000e-04
    maxstep=1.000000e+08, minlambda=1.000000e-12
    tolerances: relative=1.000000e-08, absolute=1.000000e-15, lambda=1.000000e-08
    maximum iterations=40
```

Newton

```
KSP Object: 1 MPI process
  type: gmres
    restart=30, using Classical (unmodified) Gram-Schmidt Orthogonalization with no ite
    happy breakdown tolerance 1e-30
  maximum iterations=10000, initial guess is zero
  tolerances:  relative=1e-05, absolute=1e-50, divergence=10000.
  left preconditioning
  using PRECONDITIONED norm type for convergence test
PC Object: 1 MPI process
  type: lu
    out-of-place factorization
    tolerance for zero pivot 2.22045e-14
    matrix ordering: nd
    factor fill ratio given 5., needed 1.14047
      Factored matrix follows:
        Mat Object: 1 MPI process
          type: seqaij
          rows=800, cols=800
          package used to perform factorization: petsc
          total: nonzeros=12714, allocated nonzeros=12714
            using I-node routines: found 201 nodes, limit used is 5
```

Newton

```
linear system matrix followed by preconditioner matrix:
Mat Object: 1 MPI process
  type: mffd
  rows=800, cols=800
    Matrix-free approximation:
      err=1.49012e-08 (relative error in function evaluation)
      Using wp compute h routine
          Does not compute normU
Mat Object: 1 MPI process
  type: seqaij
  rows=800, cols=800
  total: nonzeros=11148, allocated nonzeros=11148
  total number of mallocs used during MatSetValues calls=0
    using I-node routines: found 201 nodes, limit used is 5
```

Additive NPC

```
./bubble
...
-snes_type composite -snes_composite_type additiveoptimal
  -snes_composite_sneses nrichardson,newtonls
  -sub_0_snes_linesearch_type cp
  -sub_1_snes_mf_operator -sub_1_ksp_rtol 1e-10 -sub_1_pc_type lu -sub_1_snes_max_it 1
-snes_converged_reason -snes_max_funcs 1000000 -snes_view
```

Additive NPC

```
SNES Object: 1 MPI process
  type: composite
    type - ADDITIVEOPTIMAL
    SNESes on composite preconditioner follow
    ---------------------------------
    SNES Object: (sub_0_) 1 MPI process
      type: nrichardson
      maximum iterations=1, maximum function evaluations=1000000
      tolerances: relative=1e-08, absolute=1e-50, solution=1e-08
      total number of function evaluations=2
      norm schedule FINALONLY
      SNESLineSearch Object: (sub_0_) 1 MPI process
        type: cp
        maxstep=1.000000e+08, minlambda=1.000000e-12
        tolerances: relative=1.000000e-08, absolute=1.000000e-15, lambda=1.000000e-08
        maximum iterations=1
    SNES Object: (sub_1_) 1 MPI process
      type: newtonls
      maximum iterations=1, maximum function evaluations=1000000
      tolerances: relative=1e-08, absolute=1e-50, solution=1e-08
      total number of linear solver iterations=2
      total number of function evaluations=4
      norm schedule FINALONLY
      Jacobian is applied matrix-free with differencing
```

Multiplicative NPC

```
./bubble
...
-snes_type composite -snes_composite_type multiplicative
  -snes_composite_sneses nrichardson,newtonls
  -sub_0_snes_linesearch_type cp
  -sub_1_snes_mf_operator -sub_1_ksp_rtol 1e-10 -sub_1_pc_type lu -sub_1_snes_max_it 1
-snes_converged_reason -snes_max_funcs 1000000 -snes_view
```

Multiplicative NPC

```
SNES Object: 1 MPI process
  type: composite
    type - MULTIPLICATIVE
    SNESes on composite preconditioner follow
    ---------------------------------
    SNES Object: (sub_0_) 1 MPI process
      type: nrichardson
      maximum iterations=1, maximum function evaluations=1000000
      tolerances: relative=1e-08, absolute=1e-50, solution=1e-08
      total number of function evaluations=2
      norm schedule FINALONLY
      SNESLineSearch Object: (sub_0_) 1 MPI process
        type: cp
        maxstep=1.000000e+08, minlambda=1.000000e-12
        tolerances: relative=1.000000e-08, absolute=1.000000e-15, lambda=1.000000e-08
        maximum iterations=1
    SNES Object: (sub_1_) 1 MPI process
      type: newtonls
      maximum iterations=1, maximum function evaluations=1000000
      tolerances: relative=1e-08, absolute=1e-50, solution=1e-08
      total number of linear solver iterations=2
      total number of function evaluations=4
      norm schedule FINALONLY
      Jacobian is applied matrix-free with differencing
```

Left NPC

```
./bubble
...
-snes_mf_operator -ksp_rtol 1e-10 -pc_type lu
  -npc_snes_type nrichardson -snes_npc_side left -npc_snes_linesearch_type cp
-snes_converged_reason -snes_max_funcs 1000000 -snes_view
```

Left NPC

```
SNES Object: 1 MPI process
  type: newtonls
  maximum iterations=50, maximum function evaluations=1000000
  tolerances: relative=1e-08, absolute=1e-50, solution=1e-08
  total number of linear solver iterations=0
  total number of function evaluations=0
  norm schedule ALWAYS
  Jacobian is applied matrix-free with differencing, no explicit Jacobian
  SNESLineSearch Object: 1 MPI process
    type: bt
      interpolation: cubic
      alpha=1.000000e-04
    maxstep=1.000000e+08, minlambda=1.000000e-12
    tolerances: relative=1.000000e-08, absolute=1.000000e-15, lambda=1.000000e-08
    maximum iterations=40
```

Left NPC

```
NPC side LEFT
SNES Object: (npc_) 1 MPI process
  type: nrichardson
  maximum iterations=1, maximum function evaluations=30000
  tolerances: relative=0., absolute=0., solution=0.
  total number of function evaluations=3
  Jacobian is applied matrix-free with differencing
  SNESLineSearch Object: (npc_) 1 MPI process
    type: cp
    maxstep=1.000000e+08, minlambda=1.000000e-12
    tolerances: relative=1.000000e-08, absolute=1.000000e-15, lambda=1.000000e-08
    maximum iterations=1
```

Right NPC

```
./bubble
...
-snes_mf_operator -ksp_rtol 1e-10 -pc_type lu
  -npc_snes_type nrichardson -snes_npc_side right -npc_snes_linesearch_type cp
-snes_converged_reason -snes_max_funcs 1000000 -snes_view
```

Right NPC

```
NPC side RIGHT
SNES Object: (npc_) 1 MPI process
  type: nrichardson
  maximum iterations=1, maximum function evaluations=30000
  tolerances: relative=0., absolute=0., solution=0.
  total number of function evaluations=3
  Jacobian is applied matrix-free with differencing
  SNESLineSearch Object: (npc_) 1 MPI process
    type: cp
    maxstep=1.000000e+08, minlambda=1.000000e-12
    tolerances: relative=1.000000e-08, absolute=1.000000e-15, lambda=1.000000e-08
    maximum iterations=1
```

Outline

Subproblems

- Field subset
- Subdomain
- Coarse mesh
- Approximate equations

Subproblems

PETSc provides DMCreateSubDM():

- Creates embedding of subproblem (IS)
- Parallel data layout for subproblem
- Subsets discretizations and equations
- Subsets boundary conditions

Block Nonlinear Gauss-Seidel

```
./bubble
...
-snes_type multiblock -snes_multiblock_0_fields 0 -snes_multiblock_1_fields 1,2
  -multiblock_0_snes_mf_operator -multiblock_0_pc_type lu
  -multiblock_1_snes_mf_operator -multiblock_1_pc_type lu
-snes_converged_reason -snes_max_funcs 1000000 -snes_view
```

Block Nonlinear Gauss-Seidel

```
SNES Object: 1 MPI process
  type: multiblock
  SNES Multiblock with MULTIPLICATIVE composition: total blocks = 2
  Solver info for each split is in the following SNES objects:
    Block 0 Fields  0
    SNES Object: (multiblock_0_) 1 MPI process
      type: newtonls
      maximum iterations=50, maximum function evaluations=10000
      tolerances: relative=1e-08, absolute=1e-50, solution=1e-08
      total number of linear solver iterations=1
      total number of function evaluations=4
      norm schedule ALWAYS
      Jacobian is applied matrix-free with differencing
      SNESLineSearch Object: (multiblock_0_) 1 MPI process
        type: bt
          interpolation: cubic
          alpha=1.000000e-04
        maxstep=1.000000e+08, minlambda=1.000000e-12
        tolerances: relative=1.000000e-08, absolute=1.000000e-15, lambda=1.000000e-08
        maximum iterations=40
```

Block Nonlinear Gauss-Seidel

```
KSP Object: (multiblock_0_) 1 MPI process
  type: gmres
    restart=30, using Classical (unmodified) Gram-Schmidt Orthogonalization with no
    happy breakdown tolerance 1e-30
  maximum iterations=10000, initial guess is zero
  tolerances:  relative=1e-05, absolute=1e-50, divergence=10000.
  left preconditioning
  using PRECONDITIONED norm type for convergence test
PC Object: (multiblock_0_) 1 MPI process
  type: lu
    out-of-place factorization
    tolerance for zero pivot 2.22045e-14
    matrix ordering: nd
    factor fill ratio given 5., needed 1.11647
      Factored matrix follows:
        Mat Object: (multiblock_0_) 1 MPI process
          type: seqaij
          rows=300, cols=300
          package used to perform factorization: petsc
          total: nonzeros=1668, allocated nonzeros=1668
            using I-node routines: found 199 nodes, limit used is 5
```

Block Nonlinear Gauss-Seidel

```
linear system matrix followed by preconditioner matrix:
Mat Object: 1 MPI process
  type: mffd
  rows=300, cols=300
    Matrix-free approximation:
      err=1.49012e-08 (relative error in function evaluation)
      Using wp compute h routine
          Does not compute normU
Mat Object: 1 MPI process
  type: seqaij
  rows=300, cols=300
  total: nonzeros=1494, allocated nonzeros=1494
  total number of mallocs used during MatSetValues calls=0
    using I-node routines: found 200 nodes, limit used is 5
```

Block Nonlinear Gauss-Seidel

```
Block 1 Fields  1, 2
SNES Object: (multiblock_1_) 1 MPI process
  type: newtonls
  maximum iterations=50, maximum function evaluations=10000
  tolerances: relative=1e-08, absolute=1e-50, solution=1e-08
  total number of linear solver iterations=1
  total number of function evaluations=4
  norm schedule ALWAYS
  Jacobian is applied matrix-free with differencing
  SNESLineSearch Object: (multiblock_1_) 1 MPI process
    type: bt
      interpolation: cubic
      alpha=1.000000e-04
    maxstep=1.000000e+08, minlambda=1.000000e-12
    tolerances: relative=1.000000e-08, absolute=1.000000e-15, lambda=1.000000e-08
    maximum iterations=40
```

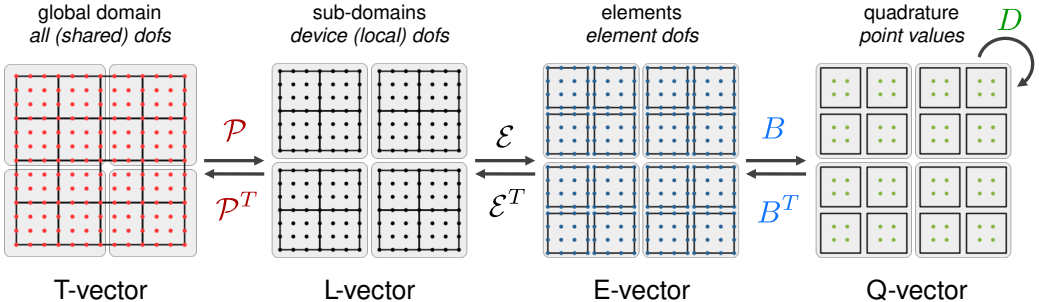
Block Nonlinear Gauss-Seidel

```
KSP Object: (multiblock_1_) 1 MPI process
  type: gmres
    restart=30, using Classical (unmodified) Gram-Schmidt Orthogonalization with no
    happy breakdown tolerance 1e-30
  maximum iterations=10000, initial guess is zero
  tolerances:  relative=1e-05, absolute=1e-50, divergence=10000.
  left preconditioning
  using PRECONDITIONED norm type for convergence test
PC Object: (multiblock_1_) 1 MPI process
  type: lu
    out-of-place factorization
    tolerance for zero pivot 2.22045e-14
    matrix ordering: nd
    factor fill ratio given 5., needed 1.15543
      Factored matrix follows:
        Mat Object: (multiblock_1_) 1 MPI process
          type: seqaij
          rows=500, cols=500
          package used to perform factorization: petsc
          total: nonzeros=5174, allocated nonzeros=5174
            using I-node routines: found 199 nodes, limit used is 5
```

Block Nonlinear Gauss-Seidel

```
      linear system matrix followed by preconditioner matrix:
      Mat Object: 1 MPI process
        type: mffd
        rows=500, cols=500
          Matrix-free approximation:
            err=1.49012e-08 (relative error in function evaluation)
            Using wp compute h routine
                Does not compute normU
      Mat Object: 1 MPI process
        type: seqaij
        rows=500, cols=500
        total: nonzeros=4478, allocated nonzeros=4478
        total number of mallocs used during MatSetValues calls=0
          using I-node routines: found 201 nodes, limit used is 5
maximum iterations=50, maximum function evaluations=1000000
tolerances: relative=1e-08, absolute=1e-50, solution=1e-08
total number of function evaluations=51
norm schedule ALWAYS
```

$$A = \mathcal{P}^T \mathcal{E}^T B^T D B \mathcal{E} \mathcal{P}$$



Assembly

- In order not to change kernels on subsetting, they need to
  - receive the full space
  - output for a single test space
- PETSc handles restriction ($\mathcal{P}$ and $\mathcal{E}$)
- Need restrictions for full space and subset for each kernel
- Propagate adapted mesh to all subsolvers

Outer Timestepping

- Create a subTS for each block, copy state and adjust callbacks
- Callbacks need to receive both the subspace and fullspace inputs
  - Have to preserve $\mathbf{u}^n$ while updating copy of state
- For FEM, we need to handle the mass matrix

Outline

Mathematical Modeling: NS equations

The Navier-Stokes equations in cylindrical coordinates:

$$\frac{\partial u_r}{\partial r} + \frac{\partial u_z}{\partial z} + \frac{u_r}{r} = 0$$

$$\frac{\partial u_r}{\partial t} + u_r \frac{\partial u_r}{\partial r} + u_z \frac{\partial u_r}{\partial z} + \frac{1}{\rho}\frac{\partial p}{\partial r} - \nu\left(\frac{\partial^2 u_r}{\partial r^2} + \frac{\partial^2 u_r}{\partial z^2} + \frac{1}{r}\frac{\partial u_r}{\partial r} - \frac{u_r}{r^2}\right) = 0$$

$$\frac{\partial u_z}{\partial t} + u_r \frac{\partial u_z}{\partial r} + u_z \frac{\partial u_z}{\partial z} + \frac{1}{\rho}\frac{\partial p}{\partial z} - \nu\left(\frac{\partial^2 u_z}{\partial r^2} + \frac{\partial^2 u_z}{\partial z^2} + \frac{1}{r}\frac{\partial u_z}{\partial r}\right) - g = 0$$

Mathematical Modeling: Interface tracking

The motion is axisymmetric and the interface moves with the fluid velocity (**EggersDupont1994**)

$$\frac{\partial h}{\partial t} + u_z \frac{\partial h}{\partial z} = u_r\Big|_{r=h}$$

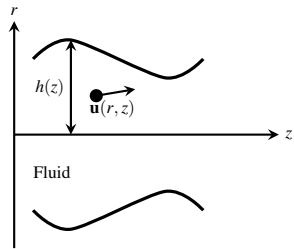


Figure: Generic flow geometry describing the interface $h(z)$ and rotational symmetry around the z axis.

Mathematical Modeling: Asymptotic expansion

- The radius of the fluid column is much smaller than the length.
- The radial contraction is faster than the elongation.

Mathematical Modeling: Asymptotic expansion

- The radius of the fluid column is much smaller than the length.
- The radial contraction is faster than the elongation.
- Therefore, using r as the asymptotic parameter, we expand the field variables.

$$u_z^d = u_0^d + u_2^d r^2 + \ldots \tag{16}$$

$$p^d = p_0^d + p_2^d r^2 + \ldots \tag{17}$$

Mathematical Modeling: Asymptotic expansion

- The radius of the fluid column is much smaller than the length.
- The radial contraction is faster than the elongation.
- Therefore, using r as the asymptotic parameter, we expand the field variables.

$$u_z^d = u_0^d + u_2^d r^2 + \ldots \tag{16}$$

$$p^d = p_0^d + p_2^d r^2 + \ldots \tag{17}$$

and using u_z^d in continuity equation, we get u_r^d.

$$u_r^d = -\frac{\partial u_0^d}{\partial z}\frac{r}{2} - \frac{\partial u_2^d}{\partial z}\frac{r^3}{4} - \ldots \tag{18}$$

Mathematical Modeling: Continuous phase flow

The velocity profile in a co-flow type environment is assumed to be parabolic (**Hua2007**)

$$u_z^c(r) = \frac{r^2}{4\mu^c}\frac{dp^c}{dz} + C_1 \ln(r) + C_2$$

Mathematical Modeling: Continuous phase flow

The velocity profile in a co-flow type environment is assumed to be parabolic (**Hua2007**)

$$u_z^c(r) = \frac{r^2}{4\mu^c}\frac{dp^c}{dz} + C_1 \ln(r) + C_2$$

The constants are defined using following conditions.

$$r = h \qquad u_z^c = u_z^d$$

$$r = Ch \qquad u_z^c = \frac{1}{4\mu^c}\frac{dp}{dz}\left(C^2h^2 - R^2\right)$$

Here, the parameter C must be greater than 1.

Mathematical Modeling: Continuous phase flow

- The continuous phase velocity profile

$$u_z^c = u_0^d + u_0^d \frac{\ln(h/r)}{\ln(C)} + \frac{\left(r^2 - h^2\right)}{4\mu^c}\frac{dp}{dz} - \frac{\left(h^2 - R^2\right)}{4\mu^c}\frac{dp}{dz}\frac{\ln(h/r)}{\ln(C)}$$

Mathematical Modeling: Continuous phase flow

- The continuous phase velocity profile

$$u_z^c = u_0^d + u_0^d \frac{\ln(h/r)}{\ln(C)} + \frac{\left(r^2 - h^2\right)}{4\mu^c}\frac{dp}{dz} - \frac{\left(h^2 - R^2\right)}{4\mu^c}\frac{dp}{dz}\frac{\ln(h/r)}{\ln(C)}$$

- The force balance on the interface.

$$\hat{\mathbf{n}}\sigma^d\hat{\mathbf{n}} + \hat{\mathbf{n}}\sigma^c\hat{\mathbf{n}} = -\gamma\mathcal{K}$$

$$\hat{\mathbf{n}}\sigma^d\hat{\mathbf{t}} + \hat{\mathbf{n}}\sigma^c\hat{\mathbf{t}} = 0$$

where, σ^c and σ^d represents the stress tensor for the continuous phase and diffuse phase respectively.

Mathematical Modeling: Governing equations

- The advecting boundary equation

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial z} + \frac{1}{2}\frac{\partial u}{\partial z}h = 0$$

- The momentum equation in case of no outer flow,

$$\frac{\partial u_0^d}{\partial t} + u_0^d\frac{\partial u_0^d}{\partial z} + \frac{\gamma}{\rho}\frac{\partial \mathcal{K}}{\partial z} - \frac{6\nu^d}{h}\frac{\partial u_0^d}{\partial z}\frac{\partial h}{\partial z} - 3\nu^d\frac{\partial^2 u_0^d}{\partial z^2} - g = 0$$

Mathematical Modeling: Governing equations

- The advecting boundary equation

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial z} + \frac{1}{2}\frac{\partial u}{\partial z}h = 0$$

- The momentum equation in case of no outer flow,

$$\frac{\partial u_0^d}{\partial t} + u_0^d\frac{\partial u_0^d}{\partial z} + \frac{\gamma}{\rho}\frac{\partial \mathcal{K}}{\partial z} - \frac{6\nu^d}{h}\frac{\partial u_0^d}{\partial z}\frac{\partial h}{\partial z} - 3\nu^d\frac{\partial^2 u_0^d}{\partial z^2} - g = 0$$

- The momentum equation in case of outer flow,

$$\frac{\partial u_0^d}{\partial t} + u_0^d\frac{\partial u_0^d}{\partial z} + \frac{\gamma}{\rho}\frac{\partial \mathcal{K}}{\partial z} - \frac{6\nu^d}{h}\frac{\partial u_0^d}{\partial z}\frac{\partial h}{\partial z}\left(1 + \frac{\mu^c}{\mu^d}\right) - 3\nu^d\frac{\partial^2 u_0^d}{\partial z^2}\left(1 + \frac{2}{3}\frac{\mu^c}{\mu^d}\right)$$
$$+\frac{2}{\rho^d}\frac{dp^c}{dz} + \frac{1}{2\rho^d\ln(C)}\frac{dp^c}{dz} - \left(1 - \frac{\rho^c}{\rho^d}\right)g = 0$$

where,

$$\mathcal{K} = \left[\frac{1}{h\left(1 + \frac{\partial h}{\partial z}^2\right)^{1/2}} - \frac{\frac{\partial^2 h}{\partial z^2}}{\left(1 + \frac{\partial h}{\partial z}^2\right)^{3/2}}\right]$$