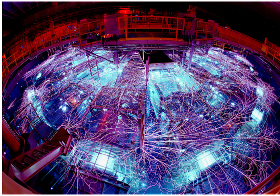
Exceptional service in the national interest

Sandia National Laboratories

SAND2017-1892C



Multiphysics Preconditioning

with the MueLu Multigrid Library

Tobias Wiesner, R.S. Tuminaro, E.C. Cyr, J. Shadid, J.J. Hu

February 20, 2017

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2016-3198 C

Motivation

Sandia National Laboratories

What is the problem?

- **Multiphysics problems:** *application-specific* with *increasing complexity* through new types of physics and/or discretizations
- No resource-efficient *general* black-box solver available

What can we do about this?

Flexible software framework for iterative solvers/preconditioners

- Modular design for application-specific solver layouts
- Usability through simplified user interfaces for non-experts

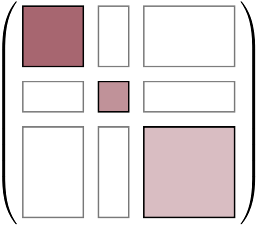
Representation of fully-coupled multiphysics problems

Blocked linear operator

- Single-field problems represented by diagonal blocks

Example: 3×3 blocked operator

$A =$



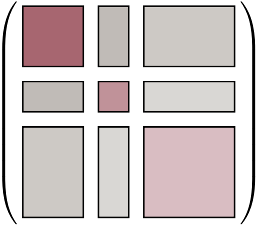
Representation of fully-coupled multiphysics problems

Blocked linear operator

- Single-field problems represented by diagonal blocks
- Coupling represented by off-diagonal blocks

Example: 3×3 blocked operator

$A =$

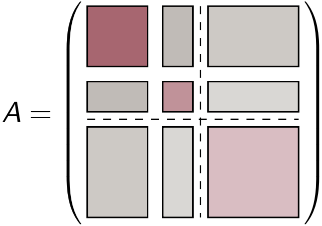


Representation of fully-coupled multiphysics problems

Blocked linear operator

- Single-field problems represented by diagonal blocks
- Coupling represented by off-diagonal blocks
- Nested blocked operators for hierarchical dependencies

Example: 2×2 blocked operator with nested 2×2 blocked operator

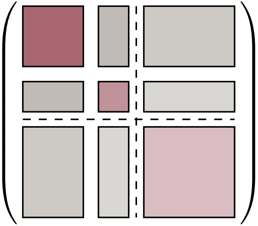


Representation of fully-coupled multiphysics problems

Blocked linear operator

- Single-field problems represented by diagonal blocks
- Coupling represented by off-diagonal blocks
- Nested blocked operators for hierarchical dependencies

Example: 2×2 blocked operator with nested 2×2 blocked operator

$A =$ 

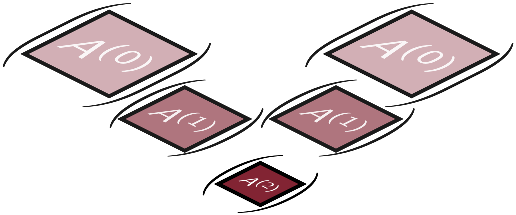
How to design efficient multigrid preconditioners for multiphysics problems?

Idea of Algebraic Multigrid Methods

Multigrid method

Transfer operators + Level smoothers

- Generate coarse representations $A^{(i)}$ of fine level problem $A^{(0)}$

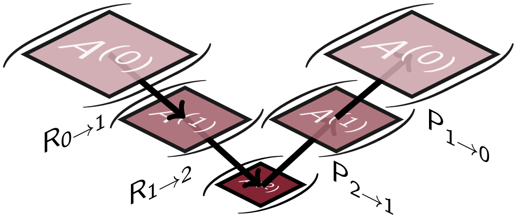


Idea of Algebraic Multigrid Methods

Multigrid method

Transfer operators + Level smoothers

- Generate coarse representations $A^{(i)}$ of fine level problem $A^{(0)}$ using $A^{(i+1)} = R_{i \to (i+1)} A^{(i)} P_{(i+1) \to i}$
- Rectangular transfer operators $P_{(i+1) \to i}$ and $R_{i \to (i+1)}$

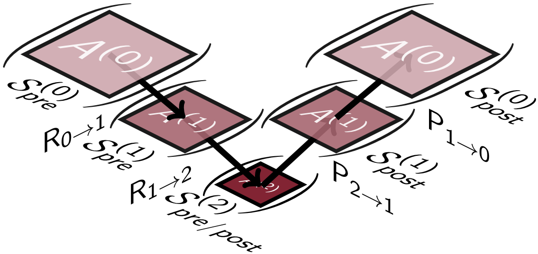


Idea of Algebraic Multigrid Methods

Multigrid method

Transfer operators + **Level smoothers**

- Generate coarse representations $A^{(i)}$ of fine level problem $A^{(0)}$ using $A^{(i+1)} = R_{i\to(i+1)}A^{(i)}P_{(i+1)\to i}$
- Rectangular transfer operators $P_{(i+1)\to i}$ and $R_{i\to(i+1)}$
- Level smoothers $\mathcal{S}^{(i)}$ damp high-oscillatory error modes



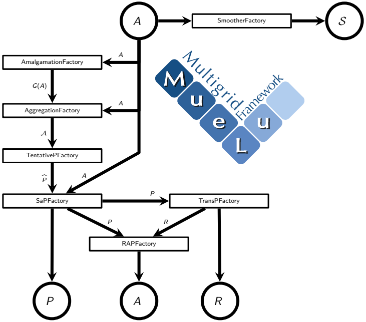
MueLu – The Trilinos Multigrid framework

MueLu multigrid framework:

- Extensible software layout
  - Modularity: Preconditioner layout defined by small building blocks
  - Logic: Building blocks connected through logical data dependencies
- Flexible user input system through XML files
- Designed for next-generation HPC systems

www.trilinos.org/packages/muelu

Example: Building blocks for transfer operators and level smoother

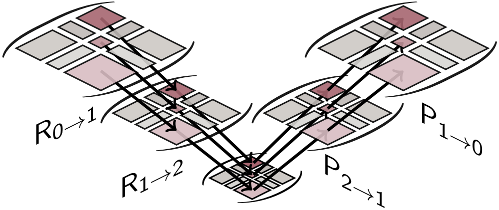


Multigrid preconditioner layout for multiphysics

Multigrid for multiphysics

Transfer operators + Level smoothers + Coupling

- Segregated transfer operators P and R to keep algebraic blocks separate on coarse levels

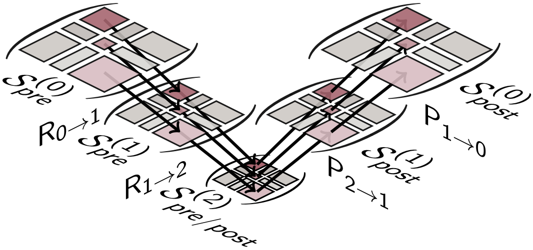


Multigrid preconditioner layout for multiphysics

Multigrid for multiphysics

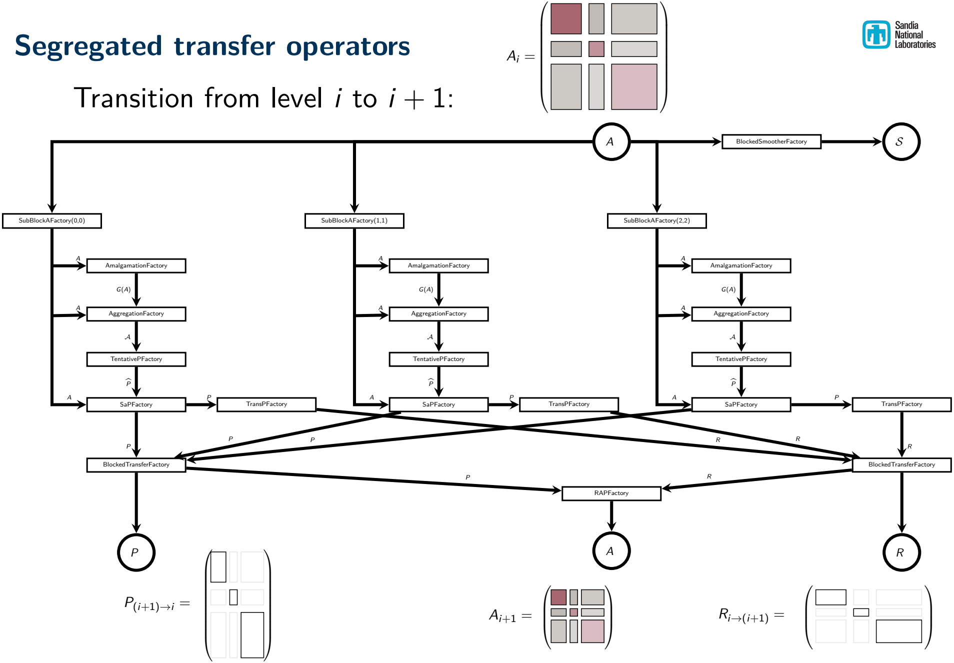
Transfer operators + **Level smoothers + Coupling**

- Segregated transfer operators P and R to keep algebraic blocks separate on coarse levels
- Nested block smoothers consider coupling of different fields



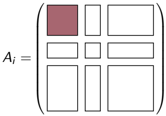
Segregated transfer operators

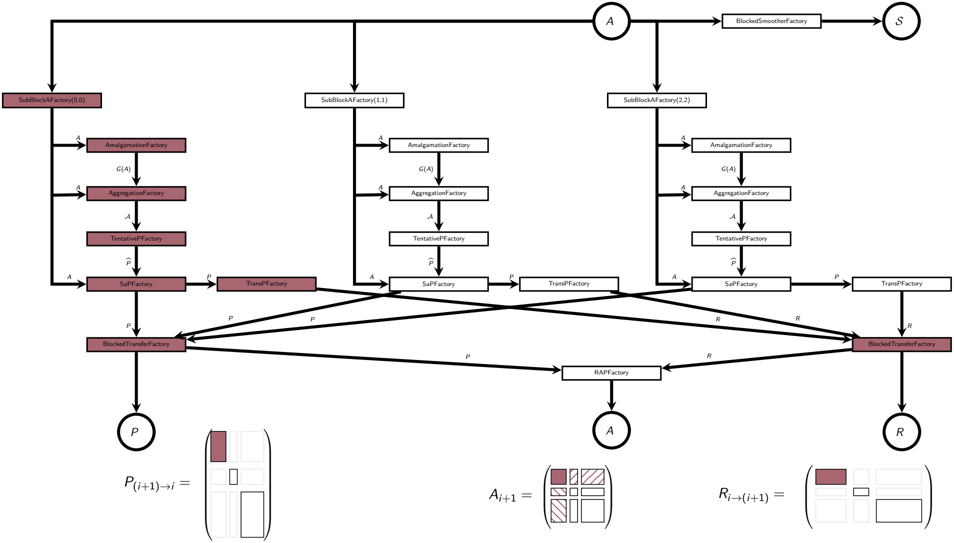
Transition from level i to $i+1$:



Segregated transfer operators

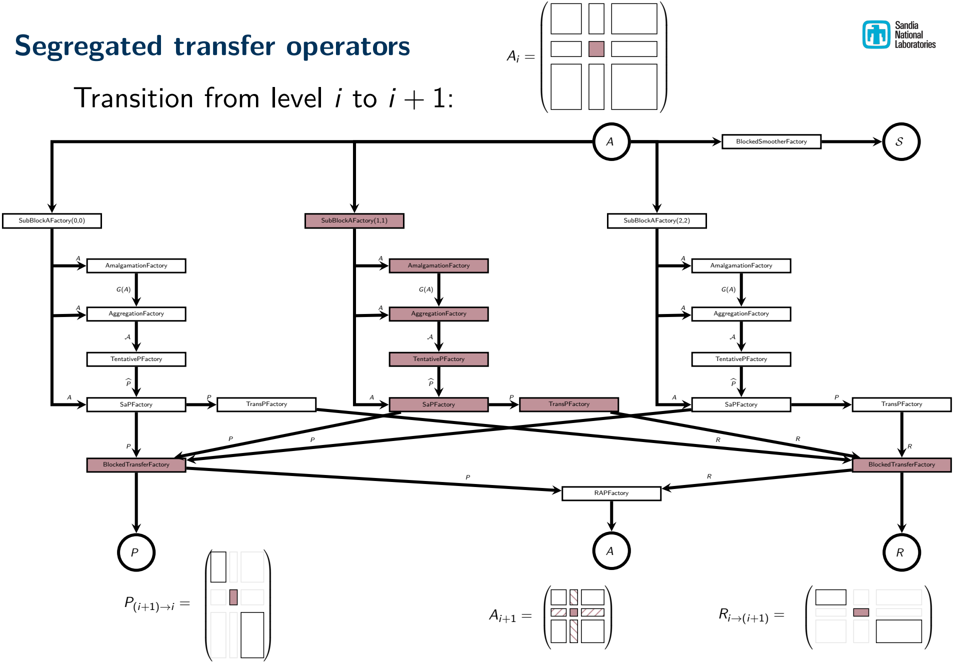
Transition from level i to $i+1$:





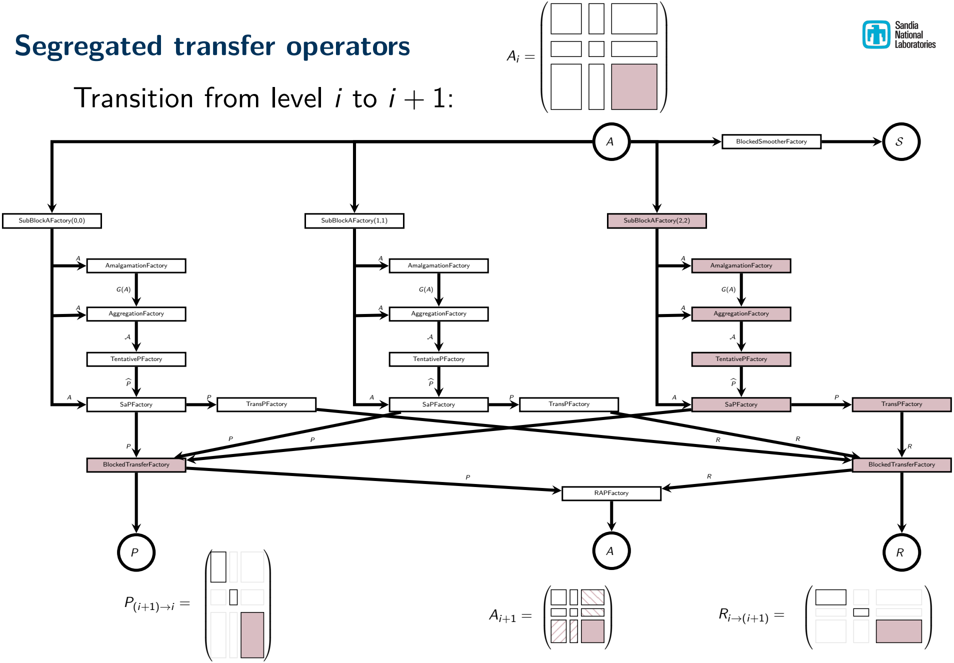
Segregated transfer operators

Transition from level i to $i+1$:



Segregated transfer operators

Transition from level i to $i+1$:



Block smoothers

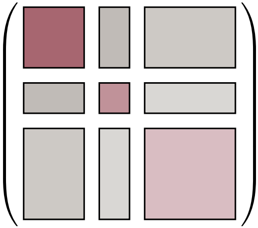
Pool of block smoothers

- General $n \times n$ block systems: Blocked Gauss-Seidel smoother
- General 2×2 block systems: SIMPLE, Uzawa, Braess-Sarazin
- Physics-based block smoothers from the Teko package

Build your application-specific block smoother

- Consider the coupling blocks when designing the block smoother

$A =$



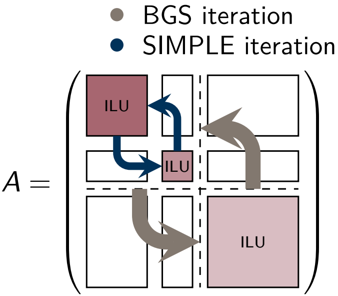
Block smoothers

Pool of block smoothers

- General $n \times n$ block systems: Blocked Gauss-Seidel smoother
- General 2×2 block systems: SIMPLE, Uzawa, Braess-Sarazin
- Physics-based block smoothers from the Teko package

Build your application-specific block smoother

- Consider the coupling blocks when designing the block smoother
- Use nested block smoothers:
  1 BGS (0.5)
  - 1 SIMPLE (0.8)
    - ILU(0), ov=1
    - ILU(0), ov=1
  - ILU(0), ov=1



Resistive Magnetohydrodynamics

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot \nu \nabla \boldsymbol{u} + \nabla p + \nabla \cdot \left(-\frac{1}{\mu_0} \boldsymbol{B} \otimes \boldsymbol{B} + \frac{1}{2\mu_0} ||\boldsymbol{B}||^2 \boldsymbol{I} \right) = \boldsymbol{0}$$

$$\nabla \cdot \boldsymbol{u} = 0$$

$$\frac{\partial \boldsymbol{B}}{\partial t} + \nabla \times (\boldsymbol{u} \times \boldsymbol{B}) - \nabla \times \frac{\eta}{\mu_0} \nabla \times \boldsymbol{B} + \nabla r = 0$$

$$\nabla \cdot \boldsymbol{B} = 0$$

with appropriate initial and boundary conditions.

Discretization

Stabilized discretization of MHD equations using equal order piecewise bilinear elements on hexahedrons

Reference solver

Discretization

Stabilized discretization of MHD equations using equal order piecewise bilinear elements on hexahedons

⇒ Collocated solution unknowns $(\boldsymbol{u}_x, \boldsymbol{u}_y, \boldsymbol{u}_z, p, \boldsymbol{B}_x, \boldsymbol{B}_y, \boldsymbol{B}_z, r)$ on each mesh node

Reference solver

- Preconditioned GMRES (from Belos or AztecOO package)
- Fully-coupled MueLu multigrid preconditioner
  - 8 DOFs per node
  - Level smoother: Additive Schwarz (overlap=1) with ILU(0)
  - Non-smoothed transfer operators

P.T. Lin, J.N. Shadid, R.S. Tuminaro, M. Sala, G.L. Hennigan, R.P. Pawlowski; *A parallel fully coupled algebraic multilevel preconditioner applied to multiphysics PDE applications: Drift-diffusion, flow/transport/reaction, resistive MHD*; Int. J. Numer. Meth. Fluids, 64,1148-1179; 2010

J.N. Shadid, R.P. Pawlowski, E.C. Cyr, R.S. Tuminaro, L. Chacon, P.D. Weber; *Scalable implicit incompressible resistive MHD with stabilized FE and fully-coupled Newton-Krylov-AMG*; Comput. Methods Appl. Mech. Engrg., 304, 1-25; 2016

Block multigrid preconditioner

Incompressible resistive MHD equations:

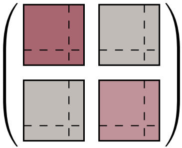
$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot \nu \nabla \boldsymbol{u} + \nabla p + \nabla \cdot \left(-\frac{1}{\mu_0} \boldsymbol{B} \otimes \boldsymbol{B} + \frac{1}{2\mu_0} ||\boldsymbol{B}||^2 \boldsymbol{I} \right) = \boldsymbol{0}$$

$$\nabla \cdot \boldsymbol{u} = 0$$

$$\frac{\partial \boldsymbol{B}}{\partial t} + \nabla \times (\boldsymbol{u} \times \boldsymbol{B}) - \nabla \times \frac{\eta}{\mu_0} \nabla \times \boldsymbol{B} + \nabla r = 0$$

$$\nabla \cdot \boldsymbol{B} = 0$$

Block structure of linear systems after discretization:

$A =$ 

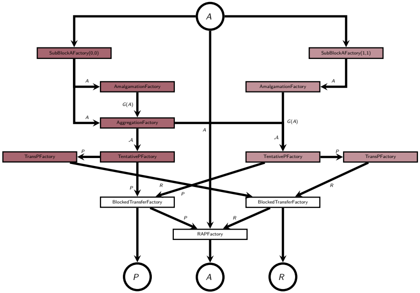
- 2×2 block system with 4 DOFs per node each block
- Solution variables: $(\boldsymbol{u}_x, \boldsymbol{u}_y, \boldsymbol{u}_z, p)$ and $(\boldsymbol{B}_x, \boldsymbol{B}_y, \boldsymbol{B}_z, r)$

Block multigrid preconditioner for MHD

Design principles:

- Preserve coincidence of MHD unknowns on coarse levels
- Reduce memory footprint by avoiding global ILU smoothers
- Performance through ILU as single-field smoothers

Solver layout:



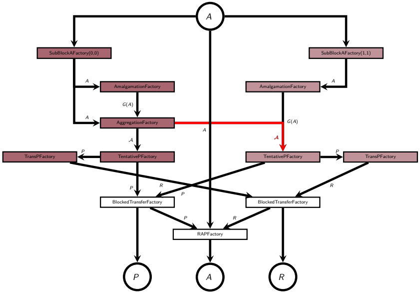
- Reuse aggregates $\mathcal{A}$ from Navier-Stokes part for magnetics part
- Non-smoothed transfer ops.
- Block smoother:
  n BGS(ω)
  - ILU(0), ov=1
  - ILU(0), ov=1

Block multigrid preconditioner for MHD

Design principles:

- Preserve coincidence of MHD unknowns on coarse levels
- Reduce memory footprint by avoiding global ILU smoothers
- Performance through ILU as single-field smoothers

Solver layout:



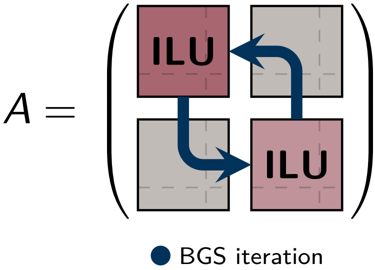
- Reuse aggregates $\mathcal{A}$ from Navier-Stokes part for magnetics part
- Non-smoothed transfer ops.
- Block smoother: n BGS(ω)
  - ILU(0), ov=1
  - ILU(0), ov=1

Block multigrid preconditioner for MHD

Design principles:

- Preserve coincidence of MHD unknowns on coarse levels
- Reduce memory footprint by avoiding global ILU smoothers
- Performance through ILU as single-field smoothers

Block smoother:



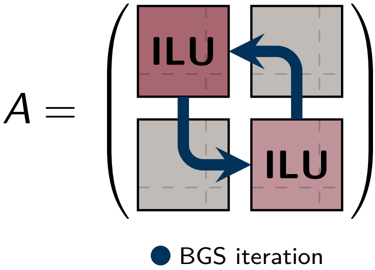
- Reuse aggregates $\mathcal{A}$ from Navier-Stokes part for magnetics part
- Non-smoothed transfer ops.
- Block smoother:
  n BGS(ω)
  - ILU(0), ov=1
  - ILU(0), ov=1

Block multigrid preconditioner for MHD

Design principles:

- Preserve coincidence of MHD unknowns on coarse levels
- Reduce memory footprint by avoiding global ILU smoothers
- Performance through ILU as single-field smoothers

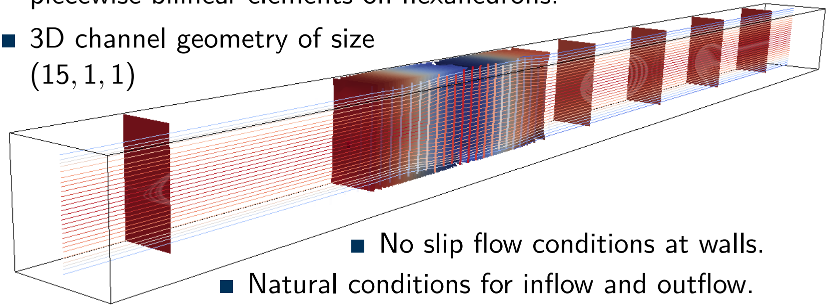
Block smoother:



- Reuse aggregates $\mathcal{A}$ from Navier-Stokes part for magnetics part
- Non-smoothed transfer ops.
- Block smoother:
  n BGS(ω)
  - ILU(0), ov=1
  - ILU(0), ov=1

MHD generator problem

- Stabilized discretization of MHD equations using equal order piecewise bilinear elements on hexahedrons.
- 3D channel geometry of size $(15, 1, 1)$



- No slip flow conditions at walls.
- Natural conditions for inflow and outflow.
- Dirichlet BCs for magnetic field at top and bottom

$$\begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{2} B_0(\tanh(x - x_0)/\Delta - \tanh(x - x_f)/\Delta) \end{bmatrix}$$

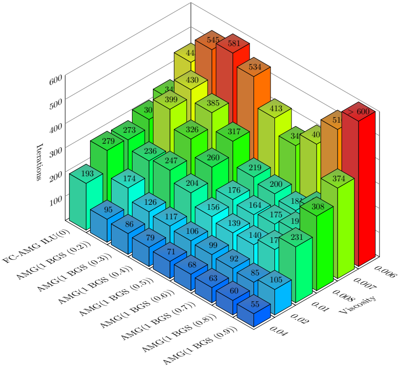
with $x_0 = 4.0, x_f = 6.0, \Delta = 0.5$ and $B_0 = 3.354$.

MHD generator problem – results

Sandia National Laboratories

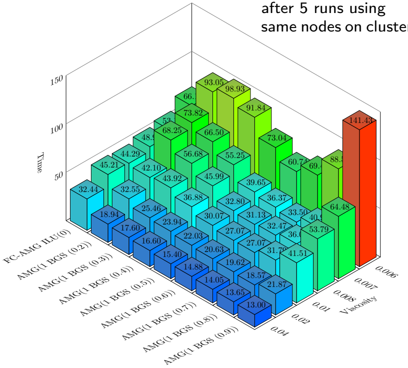
$240 \times 16 \times 16$ mesh on 32 processors

Iterations:



Timings:

Averaged timings after 5 runs using same nodes on cluster



Relative linear solver tolerance: $\varepsilon = 1 \cdot 10^{-5}$

Multigrid hierarchy:

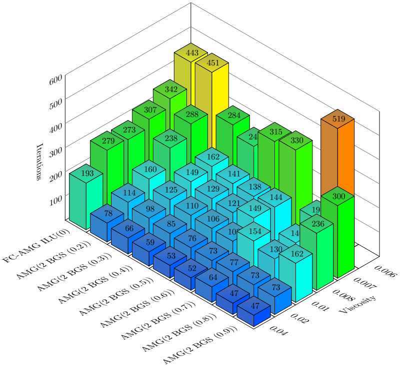
ℓ	rows	nnz	nnz/row	c ratio	procs	smoother
0	491,520	97,234,432	197.82		32	1 BGS (ω)
1	23,040	3,024,896	131.29	21.33	32	1 BGS (ω)
2	1,280	95,872	74.90	18.00	5	direct

MHD generator problem – results

Sandia National Laboratories

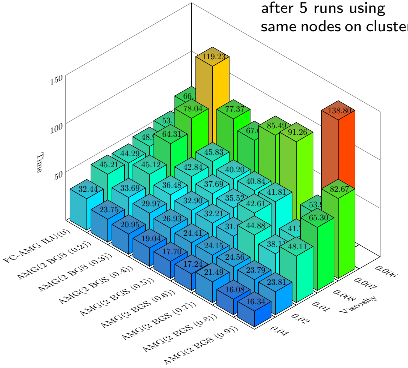
240 × 16 × 16 mesh on 32 processors

Iterations:



Timings:

Averaged timings after 5 runs using same nodes on cluster



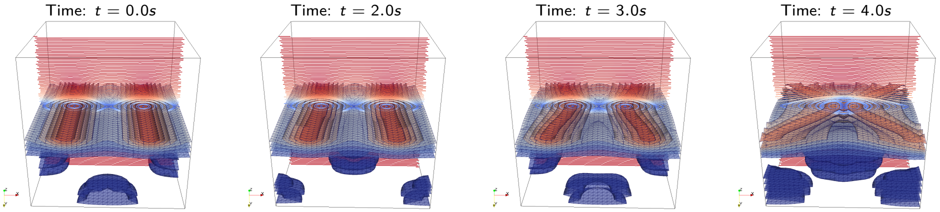
Relative linear solver tolerance: $\varepsilon = 1 \cdot 10^{-5}$

Multigrid hierarchy:

ℓ	rows	nnz	nnz/row	c ratio	procs	smoother
0	491,520	97,234,432	197.82		32	2 BGS (ω)
1	23,040	3,024,896	131.29	21.33	32	2 BGS (ω)
2	1,280	95,872	74.90	18.00	5	direct

Island coalescence problem

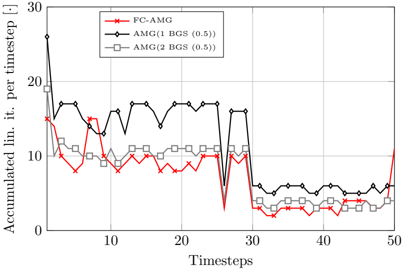
- Stabilized discretization of MHD equations using equal order piecewise bilinear elements on hexahedrons.
- 3D cube geometry of size $(2, 2, 2)$
- 2 magnetic islands as initial condition
- Viscosity: $\nu = 10^{-4}$
- Timestep size: $\Delta t \in \{0.05, 0.025, 0.0125\}$
- Relative linear solver tolerance: $\varepsilon = 10^{-5}$



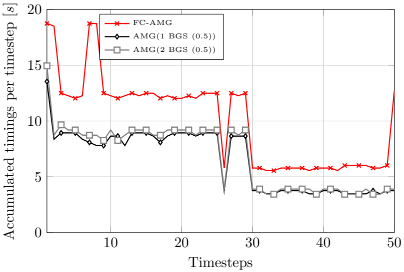
Island coalescing example

CFL = 3.2	$32 \times 32 \times 32$ mesh	$\Delta t = 0.05s$

Iterations:



Timings:



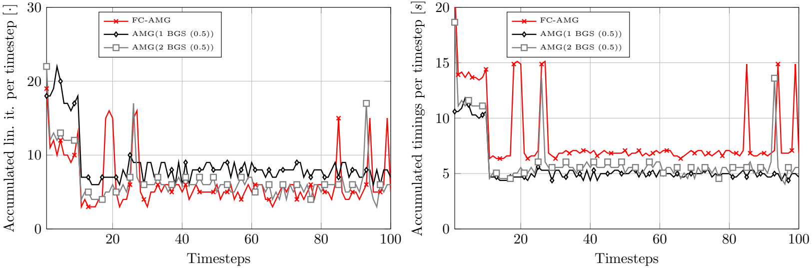
Multigrid hierarchy:

ℓ	rows	nnz	nnz/row	c ratio	procs	smoother
0	246,016	52,032,384	211.50		8	n BGS(0.5)
1	10,736	1,855,392	172.82	22.92	8	n BGS(0.5)
2	560	80,976	144.60	19.17	2	direct

Island coalescing example

CFL = 3.2	$64 \times 64 \times 64$ mesh	$\Delta t = 0.025s$

Iterations: **Timings:**



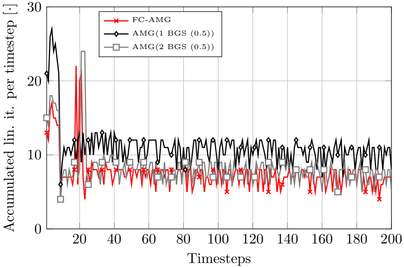
Multigrid hierarchy:

ℓ	rows	nnz	nnz/row	c ratio	procs	smoother
0	2,032,128	434,367,360	213.75		64	n BGS(0.5)
1	90,528	15,788,304	174.40	22.45	64	n BGS(0.5)
2	4,768	721,728	151.37	18.99	18	n BGS(0.5)
3	416	48,720	117.12	11.46	1	direct

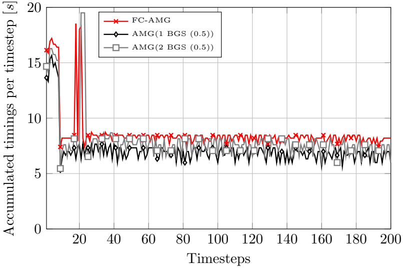
Island coalescing example

CFL = 3.2	$128 \times 128 \times 128$ mesh	$\Delta t = 0.0125s$

Iterations:



Timings:



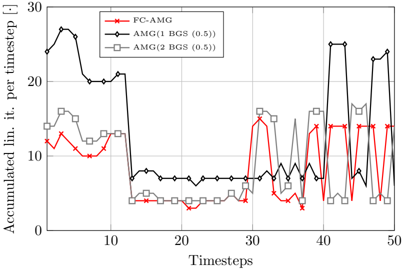
Multigrid hierarchy:

ℓ	rows	nnz	nnz/row	c ratio	procs	smoother
0	16,516,096	3,548,896,128	214.88		512	n BGS(0.5)
1	742,976	130,015,536	174.99	22.23	512	n BGS(0.5)
2	39,488	6,090,336	154.23	18.82	154	n BGS(0.5)
3	4,000	511,056	127.76	9.87	15	n BGS(0.5)
4	384	42,784	111.42	10.42	1	direct

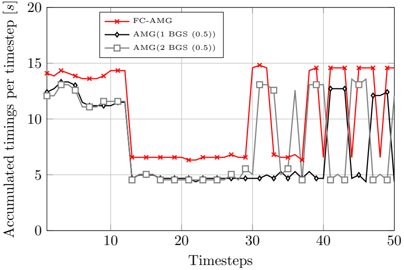
Island coalescing example

CFL = 6.4	$64 \times 64 \times 64$ mesh	$\Delta t = 0.05s$

Iterations:



Timings:



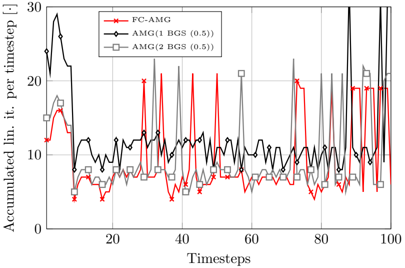
Multigrid hierarchy:

ℓ	rows	nnz	nnz/row	c ratio	procs	smoother
0	2,032,128	434,367,360	213.75		64	n BGS(0.5)
1	90,528	15,788,304	174.40	22.45	64	n BGS(0.5)
2	4,768	721,728	151.37	18.99	18	n BGS(0.5)
3	416	48,720	117.12	11.46	1	direct

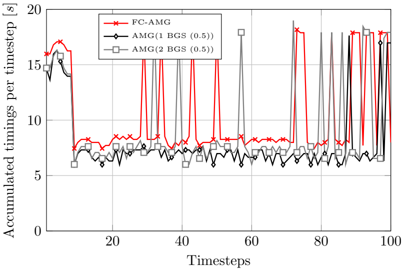
Island coalescing example

CFL = 6.4	128 × 128 × 128 mesh	$\Delta t = 0.025s$

Iterations:



Timings:



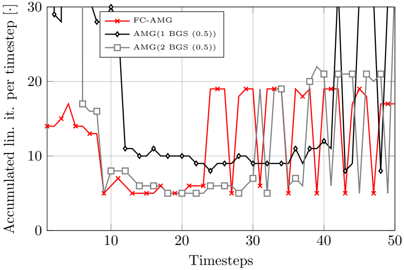
Multigrid hierarchy:

ℓ	rows	nnz	nnz/row	c ratio	procs	smoother
0	16,516,096	3,548,896,128	214.88		512	n BGS(0.5)
1	742,976	130,015,536	174.99	22.23	512	n BGS(0.5)
2	39,488	6,090,336	154.23	18.82	154	n BGS(0.5)
3	4,000	511,056	127.76	9.87	15	n BGS(0.5)
4	384	42,784	111.42	10.42	1	direct

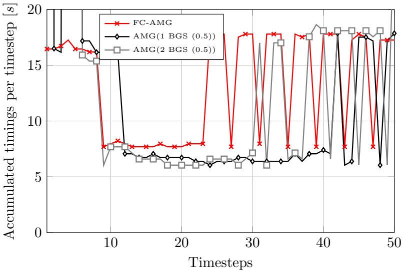
Island coalescing example

CFL = 12.8	$128 \times 128 \times 128$ mesh	$\Delta t = 0.05s$

Iterations:



Timings:

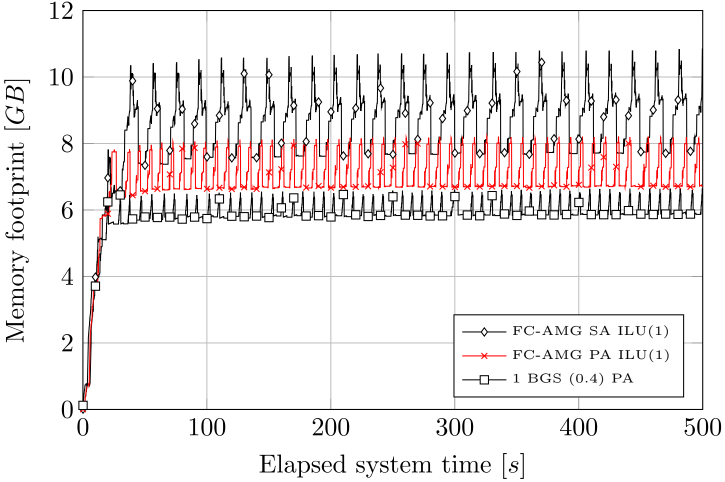


Multigrid hierarchy:

ℓ	rows	nnz	nnz/row	c ratio	procs	smoother
0	16,516,096	3,548,896,128	214.88		512	n BGS(0.5)
1	742,976	130,015,536	174.99	22.23	512	n BGS(0.5)
2	39,488	6,090,336	154.23	18.82	154	n BGS(0.5)
3	4,000	511,056	127.76	9.87	15	n BGS(0.5)
4	384	42,784	111.42	10.42	1	direct

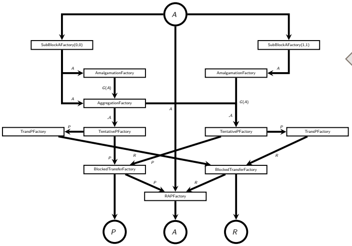
Island coalescing − memory

$64 \times 64 \times 64$ mesh on 64 procs



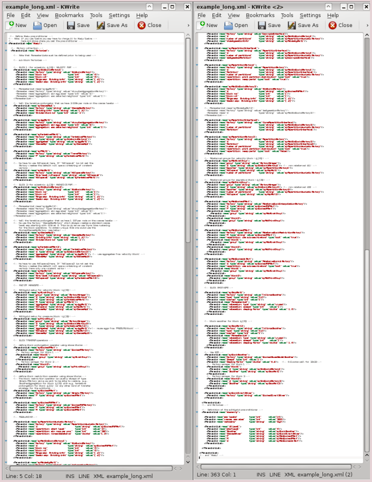
User interface

- Flexible framework for multiphysics preconditioners
  - Modular through building blocks
  - XML based input deck for defining preconditioner layout
- Flexible modular input deck **not** user friendly



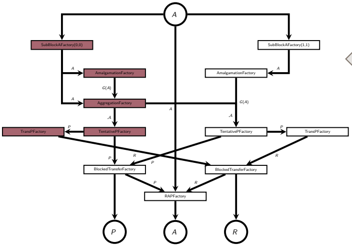
$\Rightarrow$ **Facade Classes:**
application-specific simplified user interfaces

MueLu input deck



User interface

- Flexible framework for multiphysics preconditioners
  - Modular through building blocks
  - XML based input deck for defining preconditioner layout
- Flexible modular input deck **not** user friendly



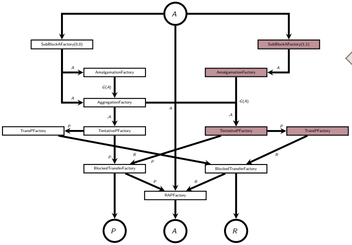
MueLu input deck

⇒ **Facade Classes:**

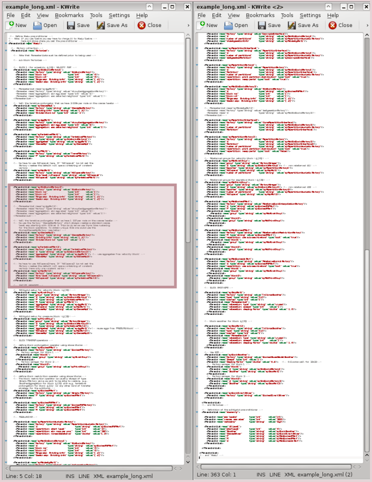
application-specific simplified user interfaces

User interface

- Flexible framework for multiphysics preconditioners
  - Modular through building blocks
  - XML based input deck for defining preconditioner layout
- Flexible modular input deck **not** user friendly



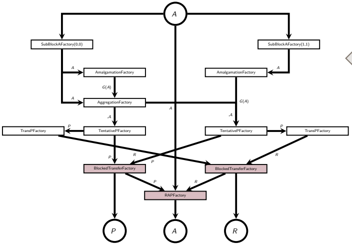
MueLu input deck



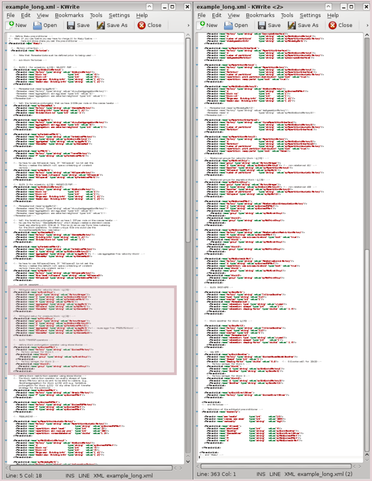
⇒ **Facade Classes:**
application-specific simplified user interfaces

User interface

- Flexible framework for multiphysics preconditioners
  - Modular through building blocks
  - XML based input deck for defining preconditioner layout
- Flexible modular input deck **not** user friendly



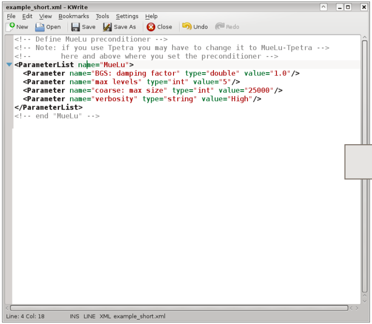
MueLu input deck



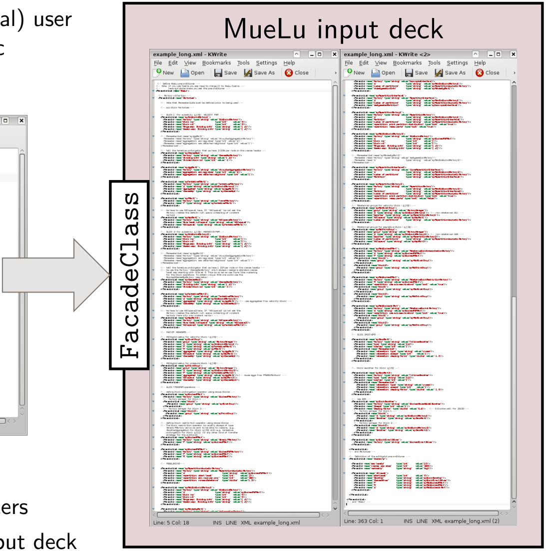
⇒ **Facade Classes:**
application-specific simplified user interfaces

User interface

- Expert provides simplified (minimal) user interface in an application-specific `FacadeClass`



```xml
<!-- Define MueLu preconditioner -->
<!-- Note: if you use Tpetra you may have to change it to MueLu-Tpetra -->
<!--       here and above where you set the preconditioner -->
<ParameterList name="MueLu">
  <Parameter name="BGS: damping factor" type="double" value="1.0"/>
  <Parameter name="max levels" type="int" value="5"/>
  <Parameter name="coarse: max size" type="int" value="25000"/>
  <Parameter name="verbosity" type="string" value="High"/>
</ParameterList>
<!-- end "MueLu" -->
```



The `FacadeClass`

- takes the simpflified user parameters
- expands them to a full MueLu input deck

Questions?

In collaboration with

- John N. Shadid (SNL)
- Eric C. Cyr (SNL)
- Jonathan J. Hu (SNL)
- Raymond S. Tuminaro (SNL)

References

- A. Prokopenko, J.J. Hu, T.A. Wiesner, C.M. Siefert, R.S. Tuminaro; *MueLu User's Guide*; SAND2014-18874; www.trilinos.org/packages/muelu; 2014
- P.T. Lin, J.N. Shadid, R.S. Tuminaro, M. Sala, G.L. Hennigan, R.P. Pawlowski; *A parallel fully coupled algebraic multilevel preconditioner applied to multiphysics PDE applications: Drift-diffusion, flow/transport/reaction, resistive MHD*; Int. J. Numer. Meth. Fluids, 64,1148-1179; 2010
- J.N. Shadid, R.P. Pawlowski, E.C. Cyr, R.S. Tuminaro, L. Chacon, P.D. Weber; *Scalable implicit incompressible resistive MHD with stabilized FE and fully-coupled Newton-Krylov-AMG*; Comput. Methods Appl. Mech. Engrg., 304, 1-25; 2016

Acknowledgement

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program.