BLOCK ITERATIVE METHODS AND RECYCLING FOR IMPROVED SCALABILITY OF LINEAR SOLVERS

Pierre Jolivet, IRIT–CNRS
Pierre-Henri Tournier, UPMC/Inria Alpines

Sparse Days 2016
June 30

Frédéric Nataf, LJLL–CNRS/Inria Alpines
Xavier Vasseur, CERFACS

INTRODUCTION

Subspace recycling

Keep information between restart or when solving sequences of linear systems:

$$A_i x_i = b_i \qquad \forall i = 1, 2, \ldots$$

STATE OF THE ART I

Subspace recycling

Keep information between restart or when solving sequences of linear systems:

$$A_i x_i = b_i \qquad \forall i = 1, 2, \ldots$$

Block methods

Treat multiple right-hand sides simultaneously for faster convergence:

$$AX = B \qquad B \in \mathbb{K}^{n \times p}$$

State of the art II

Available options

- *hypre*, DUNE, PARALUTION, SciPy: nothing
- PETSc: Loose GMRES and Deflated GMRES
- Trilinos (Belos): Block GMRES, Block GCRO-DR

STATE OF THE ART II

Available options

- *hypre*, DUNE, PARALUTION, SciPy: nothing
- PETSc: Loose GMRES and Deflated GMRES
- Trilinos (Belos): Block GMRES, Block GCRO-DR

Why not use Belos?

- no support for variable preconditioning
- no support for languages other than C++

CONTRIBUTIONS

- implementation of (pseudo-)Block CG/GMRES/GCRO-DR
- support for left/right/variable preconditioning
- large-scale results for three different physics

CONTRIBUTIONS

- implementation of (pseudo-)Block CG/GMRES/GCRO-DR
- support for left/right/variable preconditioning
- large-scale results for three different physics

HPDDM

- open-source, `https://github.com/hpddm/hpddm`
- usable in C++, C, Python, or Fortran

SUBSPACE RECYCLING

GCRO-DR

Generalized Conjugate Residual method with inner Orthogonalization and Deflated Restarting

Proposed by Parks et al. (2007)

Closely related to GMRES-DR by Morgan (2002)

GCRO-DR

Generalized Conjugate Residual method with inner Orthogonalization and Deflated Restarting

Proposed by Parks et al. (2007)

Closely related to GMRES-DR by Morgan (2002)

Main idea

1. end of GMRES cycle: compute Ritz eigenpairs
2. next restart: use 1. to generate k vectors for Arnoldi basis
3. perform extra orthogonalizations with k vectors

GCRO-DR

Generalized Conjugate Residual method with inner Orthogonalization and Deflated Restarting

Proposed by Parks et al. (2007)

Closely related to GMRES-DR by Morgan (2002)

Main idea

1. end of GMRES cycle: compute Ritz eigenpairs
2. next restart: use 1. to generate k vectors for Arnoldi basis
3. perform extra orthogonalizations with k vectors

Overhead:

- persistent storage between cycles/solves
- one additional synchronization per cycle
- small dense (generalized) eigenvalue problem

BLOCK ITERATIVE METHODS

WHY USE BLOCK METHODS?

Numerical aspects

enlarged Krylov subspace $\Longrightarrow$ faster convergence

Performance

- higher arithmetic intensity
- fewer synchronizations with more data

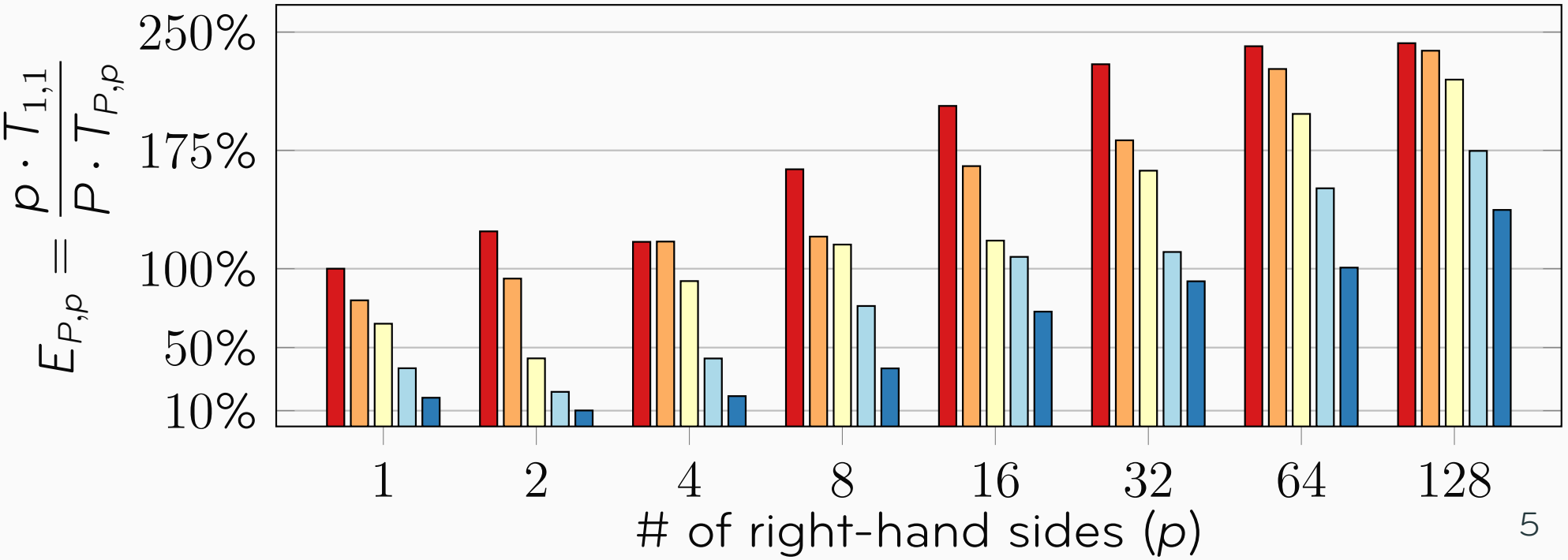
WHY USE BLOCK METHODS?

Numerical aspects

enlarged Krylov subspace $\Longrightarrow$ faster convergence

Performance

- higher arithmetic intensity
- fewer synchronizations with more data



Block Arnoldi

- block orthogonalization
- tall-and-skinny QR (default to CholQR $V^H V = L L^H$)

Block Arnoldi

- block orthogonalization
- tall-and-skinny QR (default to CholQR $V^H V = L L^H$)

About CholQR:

- $V = QR$ with $R = L^H$, $Q = L^{-H} V$
- BLAS 3
- single reduction, Stathopoulos (2000)
- can rank-reveal (`?pstrf`)

IMPLEMENTATION DETAILS

Block Arnoldi

- block orthogonalization
- tall-and-skinny QR (default to CholQR $V^H V = L L^H$)

About CholQR:

- $V = QR$ with $R = L^H$, $Q = L^{-H} V$
- BLAS 3
- single reduction, Stathopoulos (2000)
- can rank-reveal (`?pstrf`)

Pseudo-block methods

p subspaces, computation and communication steps fused

APPLICATIONS AND NUMERICAL RESULTS

RECYCLING

Two examples from the PETSc distribution:

1. ex32 (Poisson's equation)
2. ex56 (linear elasticity)

Geometric Algebraic MG preconditioner, Adams et al. (2004)

Two examples from the PETSc distribution:

1. ex32 (Poisson's equation)
2. ex56 (linear elasticity)

Geometric Algebraic MG preconditioner, Adams et al. (2004)

Nonlinear smoothers

$\Longrightarrow$ FGCRO-DR is mandatory

RECYCLING

For Poisson's equation I

```
mpirun -np 8192 ./ex32 -da_grid_x 4210 -da_refine 2
  -da_grid_y 4210 -ksp_rtol 1e-8 -pc_type gamg
  -pc_gamg_threshold 0.0725 -pc_gamg_square_graph 2
  -ksp_type fgmres -mg_levels_ksp_type gmres
  -mg_levels_ksp_max_it 3
```

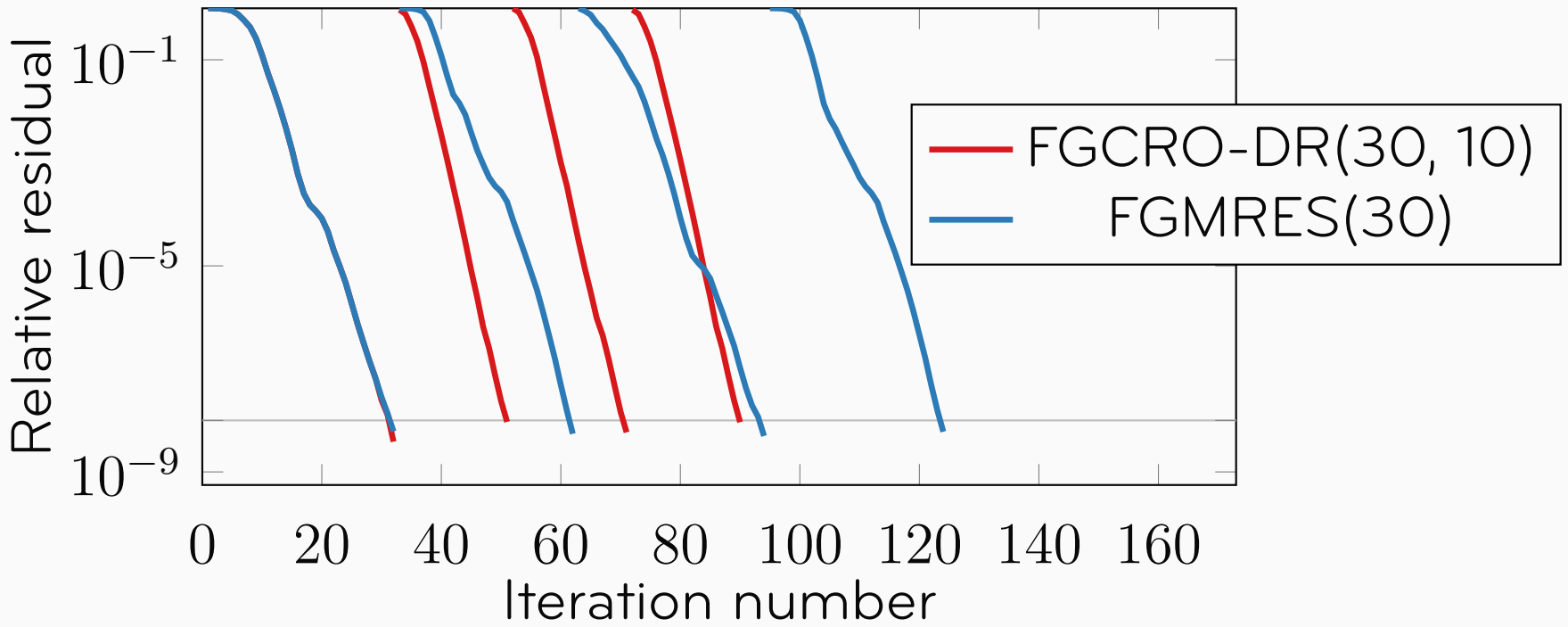
- assemble a single linear system/preconditioner
- solve a sequence with multiple RHSs:

$$Ax_i = b_i \qquad \forall i \in [\![1, 4]\!]$$

RECYCLING

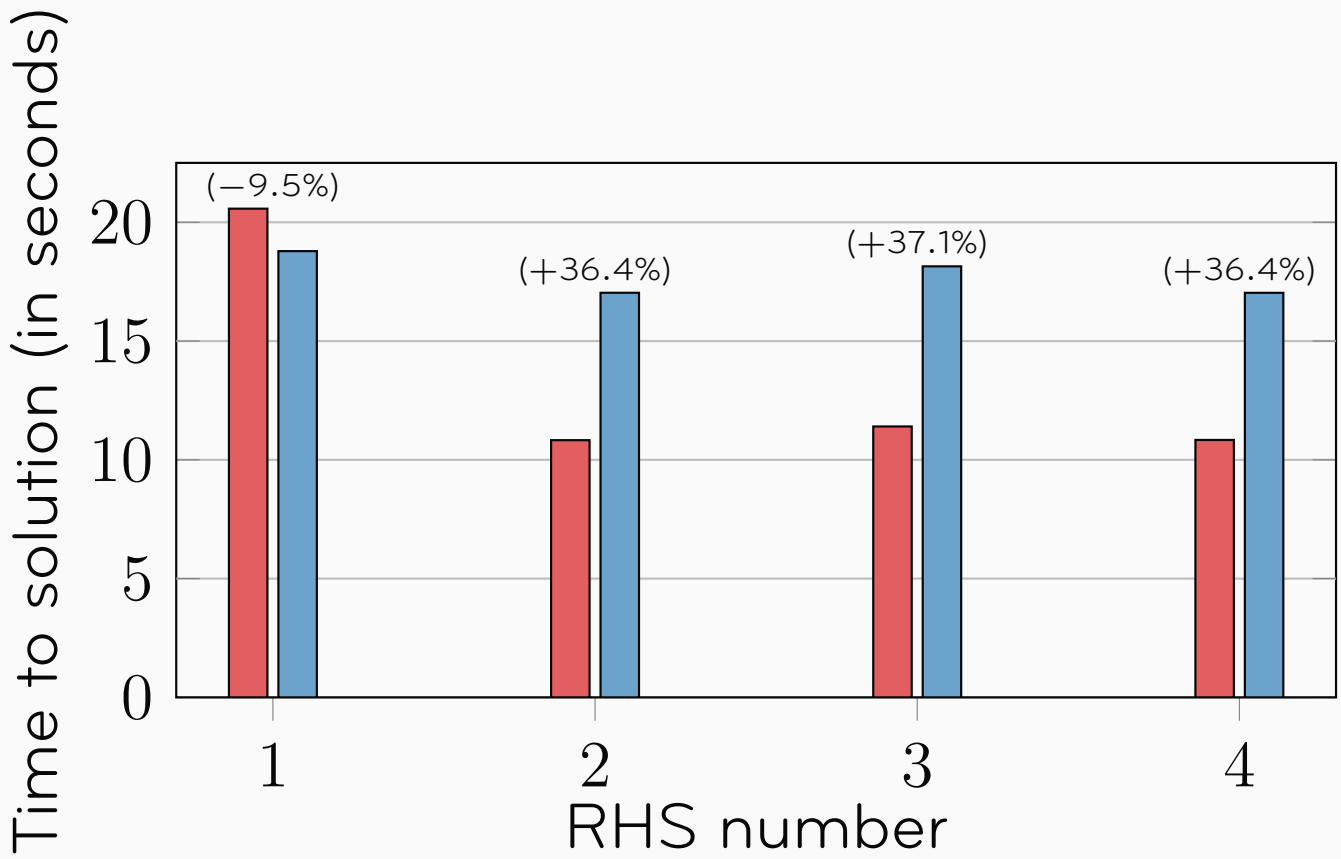
For Poisson's equation I

```
mpirun -np 8192 ./ex32 -da_grid_x 4210 -da_refine 2
  -da_grid_y 4210 -ksp_rtol 1e-8 -pc_type gamg
  -pc_gamg_threshold 0.0725 -pc_gamg_square_graph 2
  -ksp_type fgmres -mg_levels_ksp_type gmres
  -mg_levels_ksp_max_it 3
```



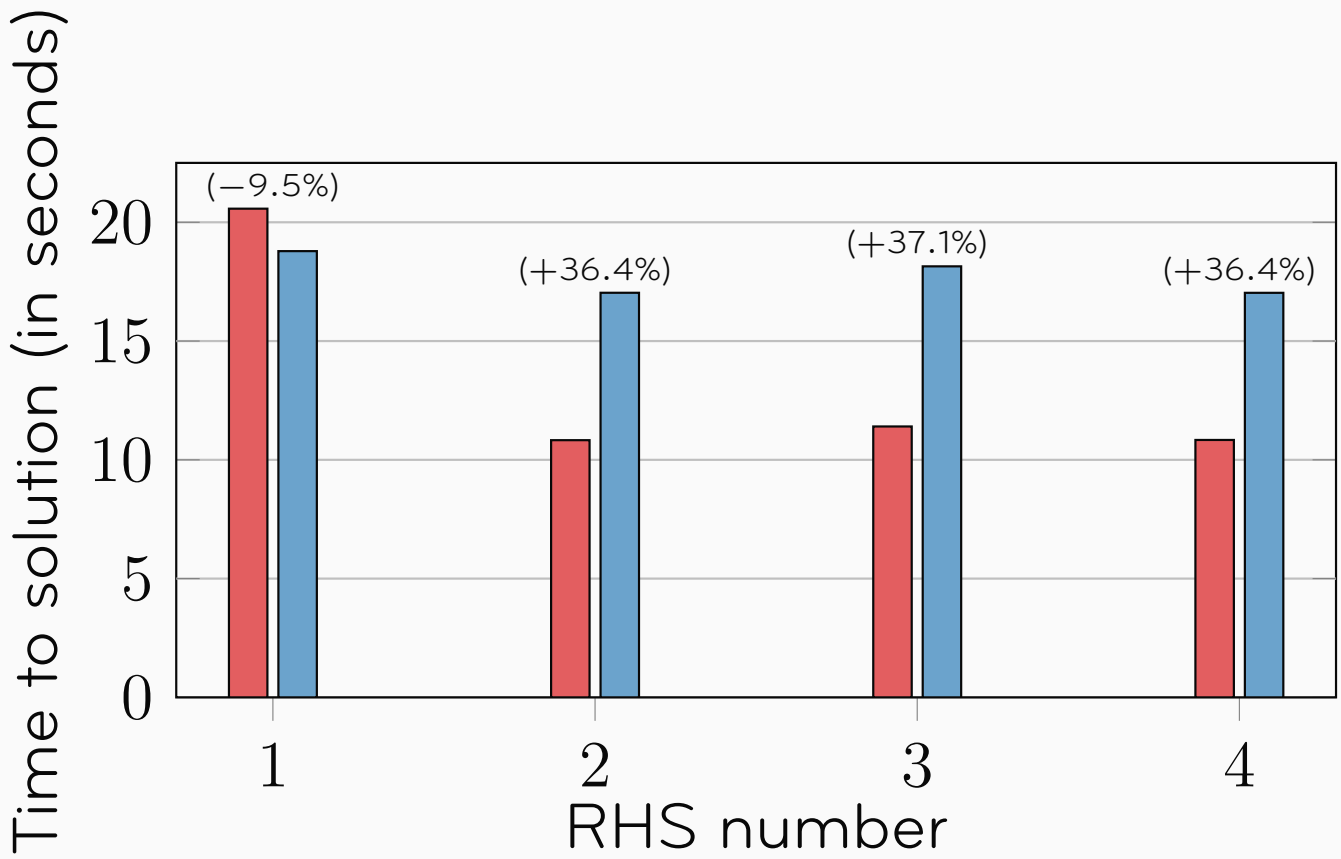
RECYCLING

For Poisson's equation II



RECYCLING

For Poisson's equation II



recycling $\Longrightarrow$ relax preconditioner setup parameters

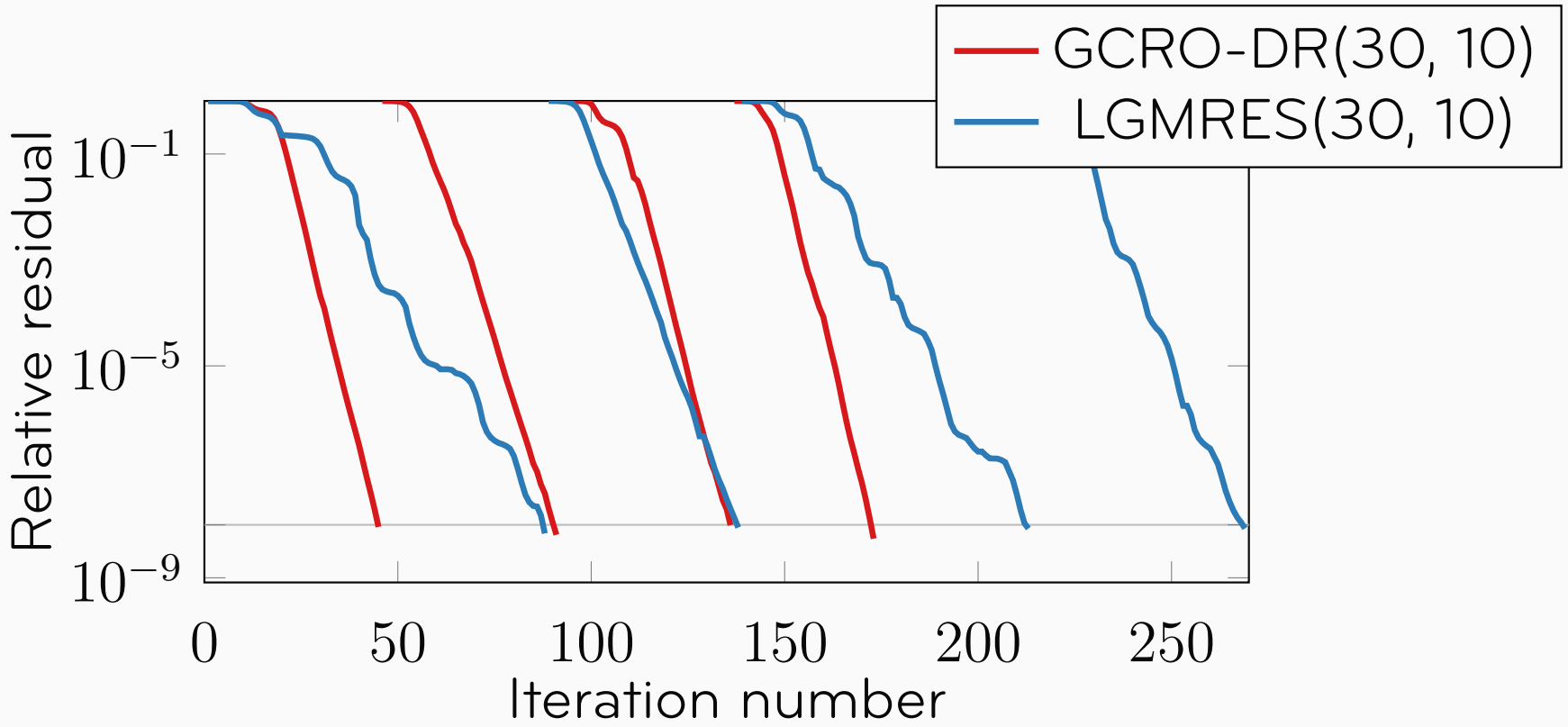
Comparison with Loose GMRES by Baker et al. (2004)

```
mpirun -np 8000 ./ex56 -ne 399 -ksp_rtol 1e-8
  -ksp_type lgmres -ksp_pc_side right -pc_type gamg
  -ksp_lgmres_augment 10
```

- small, moving inclusion (high contrast in E)
- assemble multiple linear systems/preconditioners
- PETSc doesn't implement flexible LGMRES

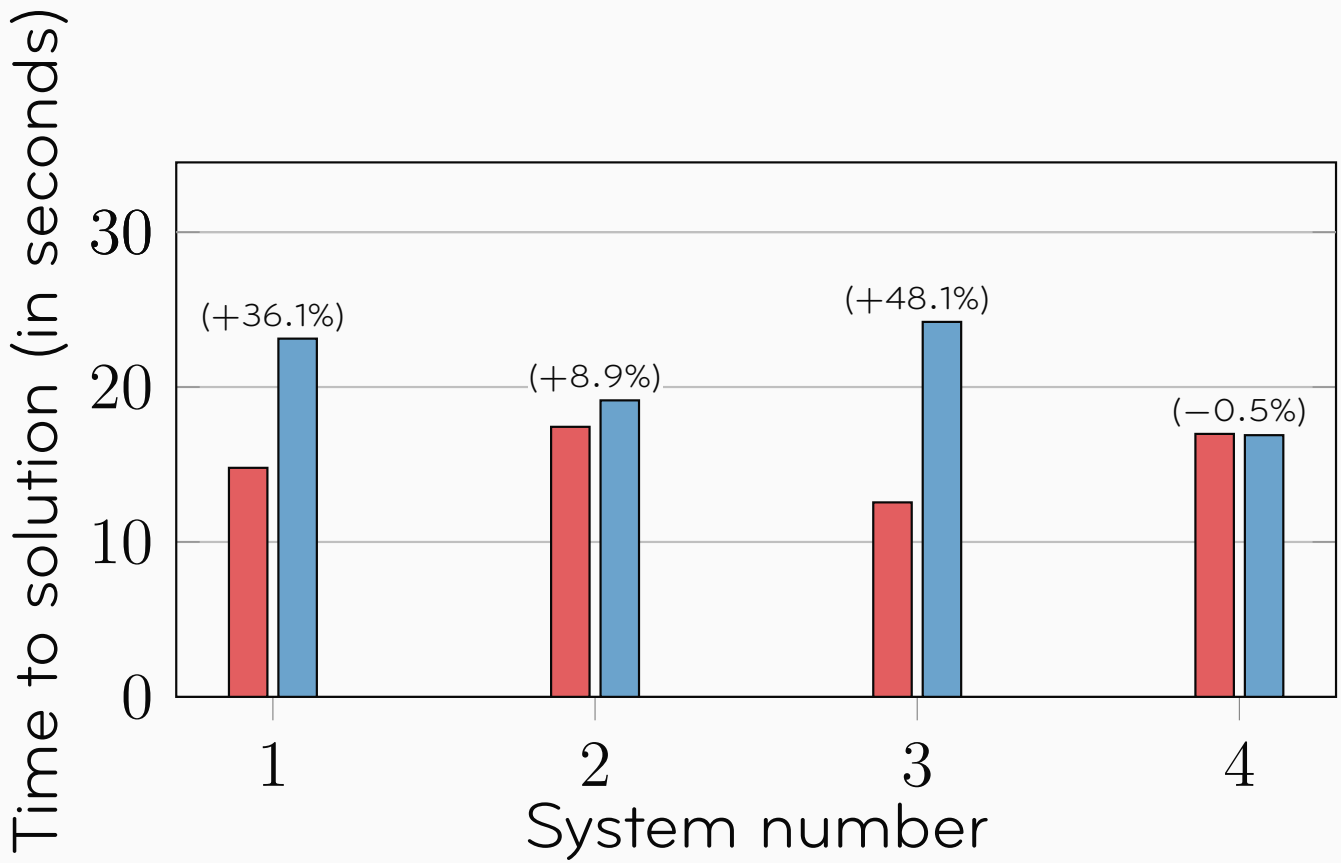
Comparison with Loose GMRES by Baker et al. (2004)

```
mpirun -np 8000 ./ex56 -ne 399 -ksp_rtol 1e-8
  -ksp_type lgmres -ksp_pc_side right -pc_type gamg
  -ksp_lgmres_augment 10
```



RECYCLING

For linear elasticity II



MAXWELL'S EQUATION

$$\nabla \times (\nabla \times \mathbf{E}) - \mu_0 \Big(\omega^2 \varepsilon + \mathrm{i}\omega\sigma \Big) \mathbf{E} = 0$$

AMS and MueLu:

1. only support eddy current formulation
2. are not trivial to use with high-order edge elements

AMS cannot deal with multiple RHSs

MAXWELL'S EQUATION

$$\nabla \times (\nabla \times \mathbf{E}) - \mu_0 \left(\cancel{\omega^2 \varepsilon} + \mathrm{i}\omega\sigma \right) \mathbf{E} = 0$$

AMS and MueLu:

1. only support eddy current formulation
2. are not trivial to use with high-order edge elements

AMS cannot deal with multiple RHSs

MAXWELL'S EQUATION

$$\nabla \times (\nabla \times \mathbf{E}) - \mu_0 \left(\cancel{\omega^2 \varepsilon} + \mathrm{i}\omega\sigma \right) \mathbf{E} = 0$$

AMS and MueLu:

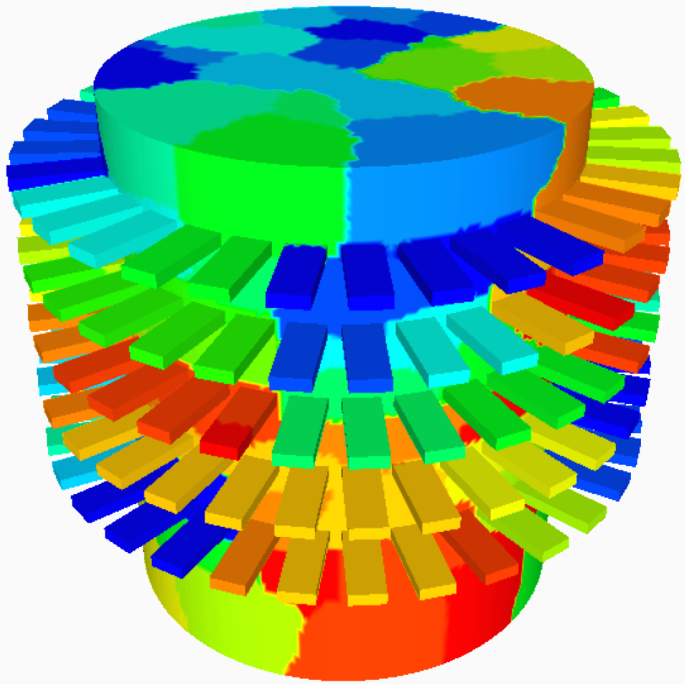
1. only support eddy current formulation
2. are not trivial to use with high-order edge elements

AMS cannot deal with multiple RHSs

$$\mathcal{M}_{\text{ORAS}}^{-1} = \sum_{i=1}^{N} R_i^T D_i B_i^{-1} R_i,$$

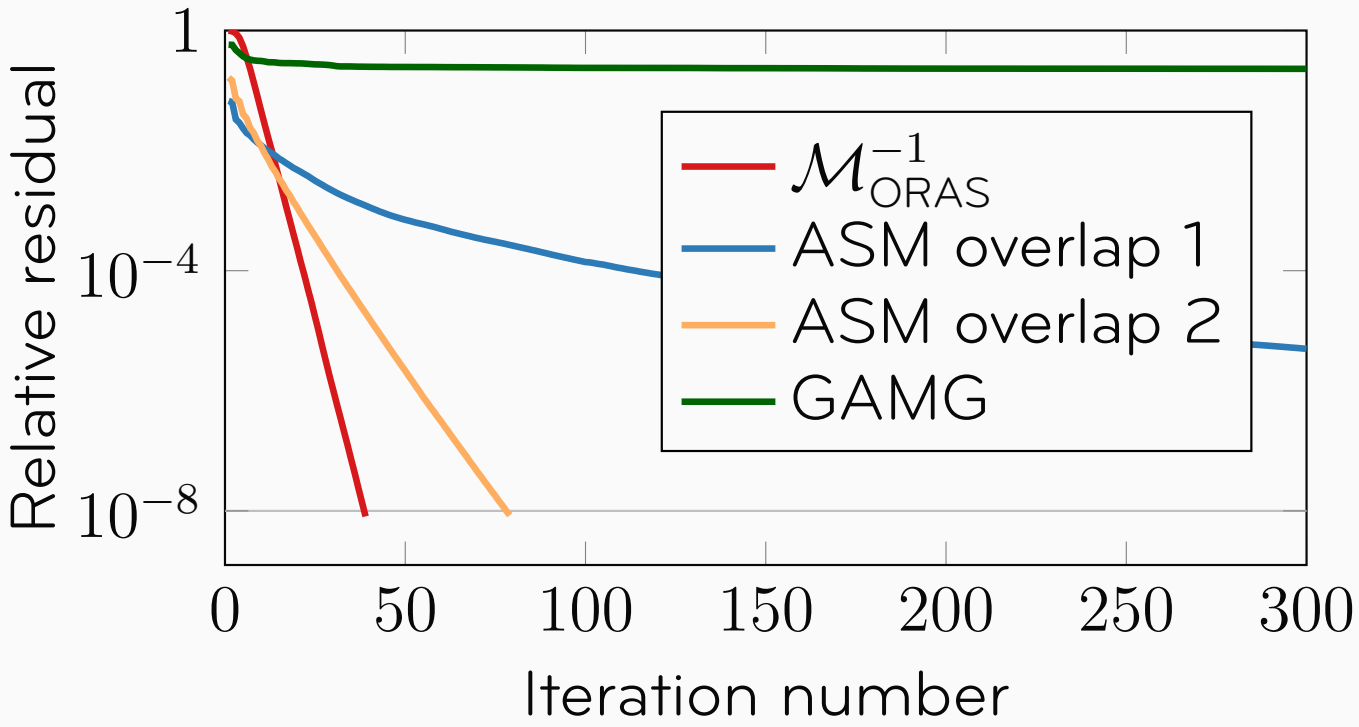
cf. Gander (2009).

- B_i^{-1} may be applied to multiple vectors at once
- B_i makes "more sense"?



MAXWELL'S EQUATION

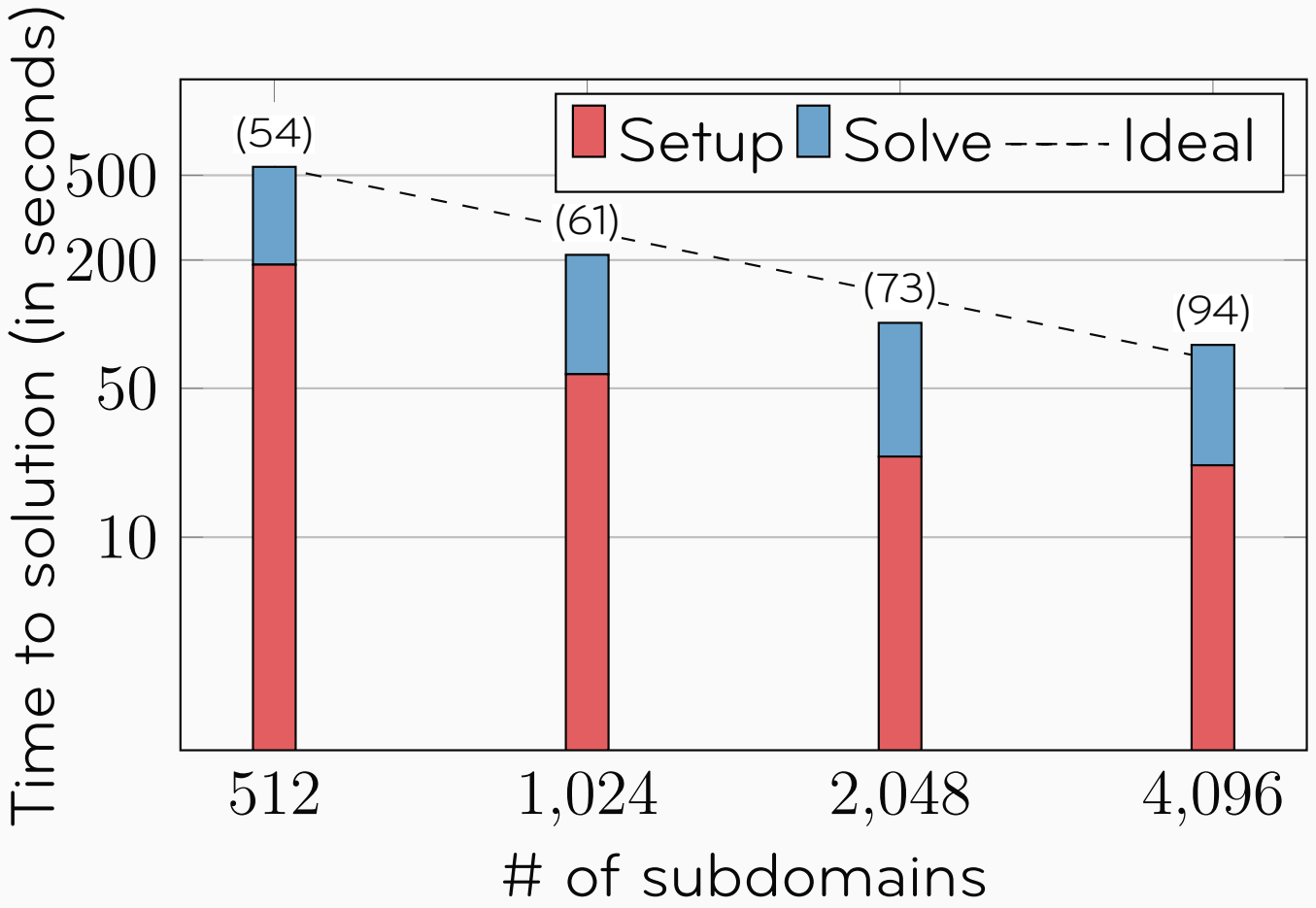
Robustness of the preconditioner



- 50 million double-precision complex unknowns
- degree 3 edge elements
- 512 subdomains, 1 thread per subdomain

MAXWELL'S EQUATION

Scalability of the preconditioner



- 119 million double-precision complex unknowns
- degree 2 edge elements

BLOCK METHODS FOR MAXWELL'S EQUATION

alternative	p	solve	# of it.	per RHS	eff.
GMRES	1				
GCRO-DR	1				

- $(m, k) = (50, 10)$ for solving 32 RHSs
- 2,048 subdomains and 2 threads per subdomain

BLOCK METHODS FOR MAXWELL'S EQUATION

alternative	p	solve	# of it.	per RHS	eff.
GMRES	1	3,078.4	20,068	627	–
GCRO-DR	1	1,836.9	10,701	334	1.7

- $(m, k) = (50, 10)$ for solving 32 RHSs
- 2,048 subdomains and 2 threads per subdomain

BLOCK METHODS FOR MAXWELL'S EQUATION

alternative	p	solve	# of it.	per RHS	eff.
GMRES	1	3,078.4	20,068	627	–
GCRO-DR	1	1,836.9	10,701	334	1.7
pseudo-BGMRES	32	1,577.9	653	–	2.0
BGMRES	32	724.8	158	–	4.2

- $(m, k) = (50, 10)$ for solving 32 RHSs
- 2,048 subdomains and 2 threads per subdomain

BLOCK METHODS FOR MAXWELL'S EQUATION

alternative	p	solve	# of it.	per RHS	eff.
GMRES	1	3,078.4	20,068	627	–
GCRO-DR	1	1,836.9	10,701	334	1.7
pseudo-BGMRES	32	1,577.9	653	–	2.0
BGMRES	32	724.8	158	–	4.2
pseudo-BGCRO-DR	8	1,357.8	1,508	377	2.3
pseudo-BGCRO-DR	32	1,376.1	469	–	2.2

- $(m, k) = (50, 10)$ for solving 32 RHSs
- 2,048 subdomains and 2 threads per subdomain

BLOCK METHODS FOR MAXWELL'S EQUATION

alternative	p	solve	# of it.	per RHS	eff.
GMRES	1	3,078.4	20,068	627	–
GCRO-DR	1	1,836.9	10,701	334	1.7
pseudo-BGMRES	32	1,577.9	653	–	2.0
BGMRES	32	724.8	158	–	4.2
pseudo-BGCRO-DR	8	1,357.8	1,508	377	2.3
pseudo-BGCRO-DR	32	1,376.1	469	–	2.2
BGCRO-DR	8	677.6	524	131	4.5
BGCRO-DR	32	992.3	127	–	3.1

- $(m, k) = (50, 10)$ for solving 32 RHSs
- 2,048 subdomains and 2 threads per subdomain

BLOCK METHODS FOR MAXWELL'S EQUATION

alternative	p	solve	# of it.	per RHS	eff.
GMRES	1	3,078.4	20,068	627	–
GCRO-DR	1	1,836.9	10,701	334	1.7
pseudo-BGMRES	32	1,577.9	653	–	2.0
BGMRES	32	724.8	158	–	4.2
pseudo-BGCRO-DR	8	1,357.8	1,508	377	2.3
pseudo-BGCRO-DR	32	1,376.1	469	–	2.2
BGCRO-DR	8	677.6	524	131	4.5
BGCRO-DR	32	992.3	127	–	3.1

- $(m, k) = (50, 10)$ for solving 32 RHSs
- 2,048 subdomains and 2 threads per subdomain
- alternative #1 to #8 $\implies$ 158$\times$ fewer iterations

BLOCK METHODS FOR MAXWELL'S EQUATION

alternative	p	solve	# of it.	per RHS	eff.
GMRES	1	3,078.4	20,068	627	–
GCRO-DR	1	1,836.9	10,701	334	1.7
pseudo-BGMRES	32	1,577.9	653	–	2.0
BGMRES	32	724.8	158	–	4.2
pseudo-BGCRO-DR	8	1,357.8	1,508	377	2.3
pseudo-BGCRO-DR	32	1,376.1	469	–	2.2
BGCRO-DR	8	677.6	524	131	4.5
BGCRO-DR	32	992.3	127	–	3.1

- $(m, k) = (50, 10)$ for solving 32 RHSs
- 2,048 subdomains and 2 threads per subdomain
- alternative #1 to #8 $\implies$ 158$\times$ fewer iterations
- GCRO-DR always performs fewer iterations than GMRES

BLOCK METHODS FOR MAXWELL'S EQUATION

alternative	p	solve	# of it.	per RHS	eff.
GMRES	1	3,078.4	20,068	627	–
GCRO-DR	1	1,836.9	10,701	334	1.7
pseudo-BGMRES	32	1,577.9	653	–	2.0
BGMRES	32	724.8	158	–	4.2
pseudo-BGCRO-DR	8	1,357.8	1,508	377	2.3
pseudo-BGCRO-DR	32	1,376.1	469	–	2.2
BGCRO-DR	8	677.6	524	131	4.5
BGCRO-DR	32	992.3	127	–	3.1

- $(m, k) = (50, 10)$ for solving 32 RHSs
- 2,048 subdomains and 2 threads per subdomain
- alternative #1 to #8 $\implies$ 158$\times$ fewer iterations
- GCRO-DR always performs fewer iterations than GMRES
- working on all 32 RHSs is costly (#5/#7 vs. #6/#8)

CONCLUSION

FINAL WORDS

Summary:

- various strategies for block methods and recycling
- large-scale experiments
- problem-specific performance but worth the try

Future work:

- other applications (inverse problems)
- block size reduction

For more: Jolivet and Tournier, SC16

FINAL WORDS

Summary:

- various strategies for block methods and recycling
- large-scale experiments
- problem-specific performance but worth the try

Future work:

- other applications (inverse problems)
- block size reduction

For more: Jolivet and Tournier, SC16

Thank you!