Domain Decomposition Multigrid for Lattice QCD

Andreas Frommer, Karsten Kahl, Stefan Krieg, Björn Leder and **Matthias Rottmann**

Bergische Universität Wuppertal

June 19, 2014

Algebraic Multigrid

Numerical Results

Summary & Outlook

Algebraic Multigrid: DD-αAMG

Task: Solution of (Clover) Wilson Dirac systems

$$D(a, m_0)\psi = \eta$$

Problem: $D(a, m_0)$ ill-conditioned for $a \to 0,\ m_0 \to m_{crit}$

Idea: Adaptive Algebraic Multigrid Approach

Two-grid error propagator for ν steps of post-smoothing

$$E_{2g}^{(\nu)} = (1 - MD)^{\nu}(1 - PD_c^{-1}P^{\dagger}D),\ D_c := P^{\dagger}DP$$

Multigrid extension by introducing recursive construction for D_c

To Do: Define interpolation P and smoother M

DD-αAMG [ArXiv:1303.1377,1307.6101]

M: Schwarz Alternating Procedure (SAP)
[Hermann Schwarz 1870; Martin Lüscher 2003]

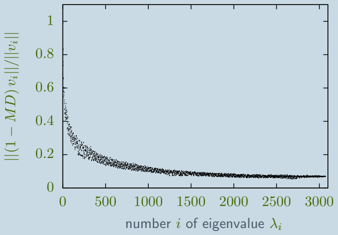
P: Aggregation Based Interpolation
[Brannick, Clark et al. 2010]

The Algebraic Multigrid Principle

Smoother: $1 - MD$

- Effective on high modes
- low modes remain

SAP



$$Dv_i = \lambda_i v_i \quad \text{with} \quad |\lambda_1| \leq \ldots \leq |\lambda_{3072}|$$

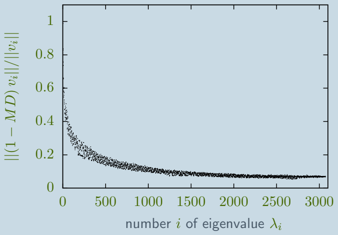
The Algebraic Multigrid Principle

Coarse-grid correction: $1 - P D_c^{-1} P^\dagger D$

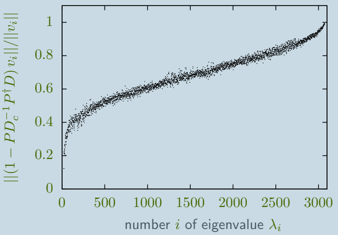
- Low modes built into interpolation P

  $\Rightarrow$ Effective on low modes

SAP



Aggregation + Low Modes



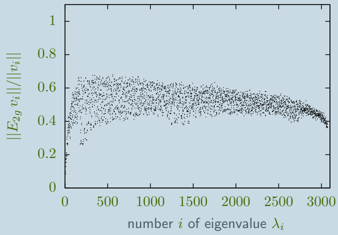
$D v_i = \lambda_i v_i$ with $|\lambda_1| \leq \ldots \leq |\lambda_{3072}|$

The Algebraic Multigrid Principle

Two-grid method: $E_{2g} = (1 - MD)(1 - PD_c^{-1}P^\dagger D)$

- Complementarity of smoother and coarse-grid correction
- Effective on all modes!

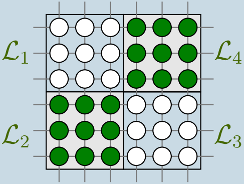
DD-αAMG



$Dv_i = \lambda_i v_i$ with $|\lambda_1| \leq \ldots \leq |\lambda_{3072}|$

SAP: Schwarz Alternating Procedure

Two color decomposition of $\mathcal{L}$



- canonical injections
  $\mathcal{I}_{\mathcal{L}_i} : \mathcal{L}_i \to \mathcal{L}$
- block restrictions
  $D_{\mathcal{L}_i} = \mathcal{I}_{\mathcal{L}_i}^\dagger D \mathcal{I}_{\mathcal{L}_i}$
- block inverses
  $B_{\mathcal{L}_i} = \mathcal{I}_{\mathcal{L}_i} D_{\mathcal{L}_i}^{-1} \mathcal{I}_{\mathcal{L}_i}^\dagger$

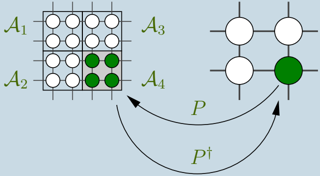
Algorithm SAP

in: ψ, η, ν – out: ψ
for $k = 1$ to ν **do**
  $r \leftarrow \eta - D\psi$
  for all *green* $\mathcal{L}_i$ **do**
    $\psi \leftarrow \psi + B_{\mathcal{L}_i} r$
  end for
  $r \leftarrow \eta - D\psi$
  for all *white* $\mathcal{L}_i$ **do**
    $\psi \leftarrow \psi + B_{\mathcal{L}_i} r$
  end for
end for

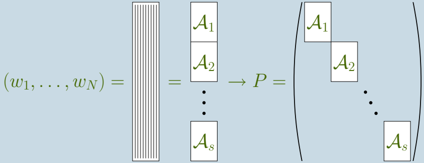
Aggregation Based Interpolation

Construction:

- Define aggregates: domain decomposition $\mathcal{A}_1, ..., \mathcal{A}_s$



- Calculate test vectors $w_1, ..., w_N$ [ArXiv:1303.1377,1307.6101]
- Decompose test vectors over aggregates $\mathcal{A}_1, ..., \mathcal{A}_s$



Algebraic Multigrid

Numerical Results

Summary & Outlook

Numerical Results: Configurations

id	lattice size $N_t \times N_s^3$	pion mass m_π [MeV]	CGNR iterations	shift m_0	clover term c_{sw}	provided by
1	48×48^3	135	53,932	-0.09933	1.00000	BMW-c [1,2]
2	64×32^3	?	11,370	-0.06430	1.00000	BMW-c [1,2]
3	64×64^3	135	84,207	-0.05294	1.00000	BMW-c [1,2]
4	128×64^3	270	45,804	-0.34262	1.75150	CLS [3,4]
5	128×64^3	190	88,479	-0.33485	1.90952	CLS [3,4]

[1]: S. Durr et al., Lattice QCD at the Physical Point: Simulation and Analysis Details, JHEP, 08(2011)148.

[2]: S. Durr et al., Lattice QCD at the Physical Point: Light Quark Masses, Phys. Lett. B701, (2011), pp. 265-268.

[3]: CLS, Coordinated Lattice Simulation. `https://twiki.cern.ch/twiki/bin/view/CLS/`.

[4]: P. Fritzsch et al., The Strange Quark Mass and Lambda Parameter of Two Flavor QCD, Nucl. Phys., B865 (2012), pp. 397-429.

Numerical Results: Parameters

	parameter		default
setup	number of iterations	n_{inv}	6
	number of test vectors	N	20
	size of lattice-blocks for aggregates/Schwarz blocks on level 1		4^4
	size of lattice-blocks for aggregates/Schwarz blocks on level $\ell > 1$		2^4
	coarse system relative residual tolerance (stopping criterion for the coarse system)$^{(*)}$	ϵ	$5 \cdot 10^{-2}$
solver	restart length of FGMRES	n_{kv}	10
	relative residual tolerance (stopping criterion)	tol	10^{-10}
	number of post-smoothing steps$^{(*)}$ on level 1	ν	2
	number of post-smoothing steps$^{(*)}$ on level $\ell > 1$		3
	number of Minimal Residual (MR) iterations to solve the local systems in SAP$^{(*)}$ on level 1		4
	number of MR iterations$^{(*)}$ on level $\ell > 1$		3
K-cycle	maximal length$^{(*)}$		5
	maximal restarts$^{(*)}$		2
	relative residual tolerance (stopping criterion)$^{(*)}$		10^{-1}

$(*)$: same in solver and setup

Comparison with biCGstab for Different Cost Measures

Configuration 3: 64×64^3

	three-level DD-αAMG	two-level DD-αAMG	odd-even MP biCGstab
processes	128	128	4,096
solve time	75.2s	130s	156s
consumed core minutes	160	277	10,600
consumed Mflop per site	1.64	2.97	69.7
Gflop/s per core	2.86	2.99	1.83

biCGstab vs. 3 Level DD-αAMG

- $66.3\times$ consumed core minutes
- $42.5\times$ consumed Mflop per lattice site

2 Level DD-αAMG vs. 3 Level DD-αAMG

- $1.73\times$ consumed core minutes
- $1.81\times$ consumed Mflop per lattice site

Comparison with biCGstab for Different Cost Measures

Configuration 3: 64×64^3

	three-level DD-αAMG	two-level DD-αAMG	odd-even MP biCGstab
processes	128	128	$4{,}096$
setup+solve time	819s	866s	156s
consumed core minutes	$1{,}744$	$1{,}856$	$10{,}600$
consumed Mflop per site	17.9	20.0	69.7
Gflop/s per core	2.86	2.99	1.83

biCGstab vs. 3 Level DD-αAMG

- $6.08\times$ consumed core minutes
- $3.89\times$ consumed Mflop per lattice site

Note: Lattice QCD applications require

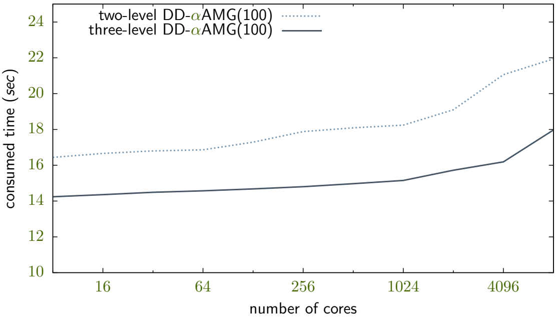
- at least 2 solves/cnfg (generation of configurations)
- mostly $\mathcal{O}(100)$ solves/cnfg (calculation of observables)

Comparison of 2, 3 & 4 Level DD-αAMG

	configuration	1	3	4	5
	lattice size	48×48^3	64×64^3	128×64^3	128×64^3
	pion mass m_π	135 MeV	135 MeV	270 MeV	190 MeV
two levels	setup time	316s	736s	630s	701s
	solve time	48.6s	130s	113s	141s
three levels	setup time	374s	744s	719s	948s
	solve time	42.6s	75.2s	74.0s	79.0s
four levels	setup time	–	806s	755s	$1{,}004$s
	solve time	–	79.8s	75.7s	79.1s
	processes	81	128	256	256
local lattice	level 1	16×16^3	32×16^3	32×16^3	32×16^3
	level 2	4×4^3	8×4^3	8×4^3	8×4^3
	level 3	2×2^3	4×2^3	4×2^3	4×2^3
	level 4	–	2×1^3	2×1^3	2×1^3

- Conf. 1 slight speed up by using 3rd level
- Conf. 2, 3, 4 significant speed up by using 3rd level
- No use for 4th level yet

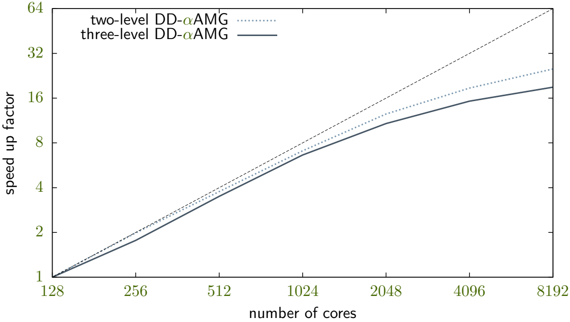
Weak Scaling: 2 and 3 Levels



- 100 V-Cycles for 2 and 3 levels
- 16×8^3 local lattice per process
- Scaling behavior almost independent of number of levels
- Coarse solver: GMRES(32); profiling shows: slow down mainly caused by allrecude operations $\rightarrow \log(p)$ scaling

Strong Scaling: 2 and 3 Levels

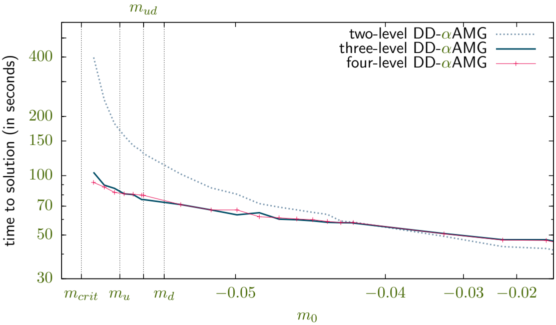
Configuration 3: 64×64^3



- Scaling behavior only slightly corrupted by adding an additional level, although there are
  - Idle times on third level
  - Worse data/comm ratio on third level

Scaling with the Mass Parameter: 2, 3 & 4 Levels

Configuration 3: 64×64^3, 128 cores



- 3 & 4 levels clearly show improved scaling
- Speed up factor roughly 2 at m_u
- 4 levels best beyond m_u
- Consider using 4th level for future calculations

The Inexact Deflation Method [Lüscher 2007]

Let $\pi_L = 1 - DPD_c^{-1}P^\dagger$ and $\pi_R = 1 - PD_c^{-1}P^\dagger D$.
Using these projections and $D\pi_R = \pi_L D$, we decompose $D\psi = \eta$ into

$$D\pi_R\psi = \pi_L\eta \quad \text{and} \tag{1}$$

$$D(1-\pi_R)\psi = (1-\pi_L)\eta. \tag{2}$$

(2) is simplified to $(1-\pi_R)\psi = PD_c^{-1}P^\dagger\eta$.
(1) is solved via GCR preconditioned with SAP.

Notes:

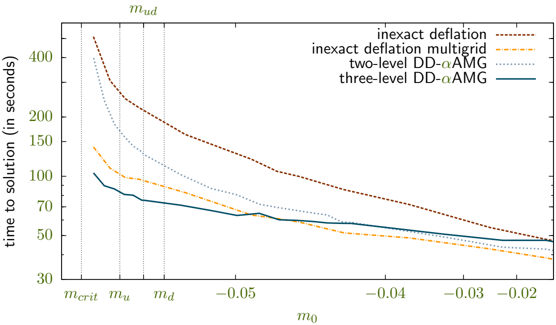
- Occurrence of D_c^{-1} in (1) and (2) yields that

$$||\eta - D\psi|| < tol \Rightarrow ||\eta_c - D_c\psi_c|| < tol.$$

- One step multigrid method due to exact complementarity
  - disadvantage: costly smoother

2 & 3 Level DD-αAMG, Inexact Deflation

Configuration 3: 64×64^3, 128 cores



- 32 test vectors for inex. defl. w. inex. proj.[OpenQCD 1.2]
- Inex. defl. w. inex. proj. scales better than ordinary inex. defl.[DD-HMC 1.2.2] and 2 level DD-αAMG
- 3 level DD-αAMG shows best scaling behavior
- 3 levels perform best in range of m_u and m_d

Intermediate Summary

Intermediate Summary:

- Multigrid beneficial for lattice QCD
- Outperforms oe-mp-biCGstab and inexact deflation
- 3 level DD-αAMG shows
  - pay off for large lattice sizes and smaller masses m_0
  - up to factor 2 speed up over 2 level DD-αAMG
  - great potential for future calculations

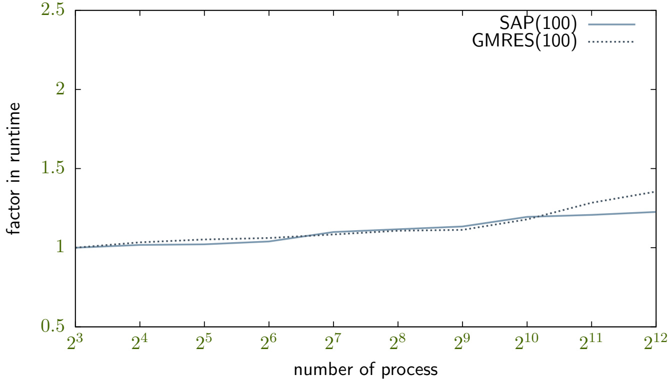
Boston & Boulder AMG

First AMG for LQCD by Brannick, Brower, Clark, Osborn et al.
[ArXiv:0707.4018,0710.3612,0912.2186,1005.3043,1011.2775]

Comparison of ingredients:

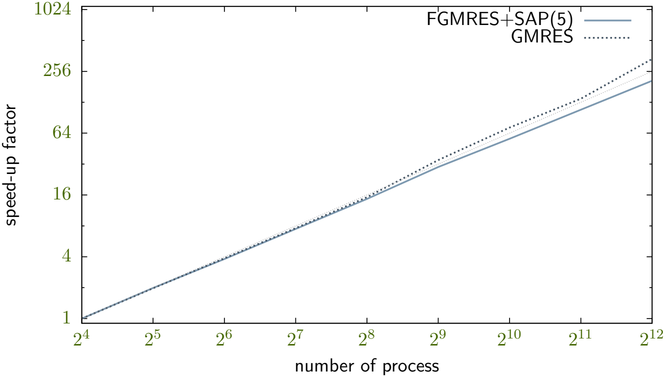
	DD-αAMG	AMG
interpolation	γ_5-compatible aggregation	γ_5-compatible aggregation
smoother	SAP ⊕ data locality / arithmetic intensity ⊕ communication avoidance	GCR ⊕ convergent method
setup	inverse iteration using DD-αAMG ⊕ bootstrapping $\rightarrow$ speed up	inverse iteration using GCR/BiCGStab ⊕ convergent inverse iteration
implementation	⊕ fast coarse grid setup	⊕ SSE-optimization ⊕ publicly available

Weak Scaling SAP(100) and GMRES(100)



- 8×4^3 local lattice
- beyond 2^{10} cores all-to-all communication becomes visible

Strong Scaling FGMRES+SAP(5) and GMRES



- 64×32^3 lattice (conf id 2)
- GMRES suffers from cache effects for large local volumes

Comparison Within Common Code Framework: 2 Levels

Comparison of DD-αAMG with AMG for cnfg 3 (64×64^3 lat)

- independently tuned parameters for both methods, $8{,}192$ cores
- Bootstrap setup for both approaches

	AMG	DD-αAMG	speed-up factor
smoother iter	6	2	
setup iter	5	5	
setup time	19.7s	17.7s	1.11
smoother time	0.60s	0.40s	1.50
solve iter	23	23	
solve time	3.76s	3.07s	1.22
total time	23.5s	20.8s	1.13

- Speed-up factor for smoother: $1.5\times$
- two grid method, $8{,}192$ cores $\rightarrow$ time spent on coarse grid predominates

Comparison Within Common Code Framework: 4 Levels

Comparison of DD-αAMG with AMG for cnfg 3 (64×64^3 lat)

- independently tuned parameters for both methods, 128 cores
- Bootstrap setup for both approaches

	4 level AMG	4 level DD-αAMG	speed-up factor
smoother iter	6	1	
setup iter	6	6	
setup time	815s	541s	1.51
solve iter	23	27	
solve time	60.6s	41.4s	1.46
total time	876s	582s	1.51

- Smoother now used on coarser grids as well
  $\longrightarrow$ speed-up factor of $1.5\times$ carried over to total time

Comparison with official AMG implementation in QOPQDP: 2 Levels

Comparison for different cnfgs and parallelizations, tol=1E-5

- AMG-d uses default parameter settings
- AMG-k uses k iterations of GCR per inverse iteration step

	cnfg 3, 128 cores			cnfg 4, 256 cores		
	AMG-d	AMG-20	DD-αAMG	AMG-d	AMG-10	DD-αAMG
setup time	2451s	838s	896s	2552s	629s	656s
solve iter	14	22	10	13	21	11
solve time	44.8s	65.7s	57.1s	36.3s	50.6s	37.3s
	cnfg 3, 8192 cores			cnfg 4, 8192 cores		
	AMG-d	AMG-40	DD-αAMG	AMG-d	AMG-20	DD-αAMG
setup time	66.0s	30.5s	27.7s	102s	30.9s	32.3s
solve iter	14	16	10	13	16	11
solve time	5.76s	7.34s	1.82s	4.35s	5.01s	1.86s

- For smaller parallelizations AMG setup can be comparatively expensive but yielding faster solver
- DD-αAMG superior for larger parallelizations

Algebraic Multigrid

Numerical Results

Summary & Outlook

Summary & Outlook

Summary:

Schwarz smoother

- Slight speed up for 2 levels
- $1.5\times$ speed up for 4 levels

DD-αAMG Code

- large parallelizations: faster than QOPQDP
- small parallelizations: setup cheaper but solver slower
- Iteration count always lower than for QOPQDP

Outlook:

- Improve setup procedure
- SSE/AVX optimization
- Open source

Acknowledgments

All results computed on JUROPA at
Jülich Supercomputing Centre (JSC)

Funded by Deutsche Forschungsgemeinschaft (DFG),
Transregional Collaborative Research Centre 55
(SFB TR 55)

All configurations provided by BMW-c & CLS