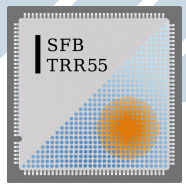
Multilevel solvers in lattice QCD

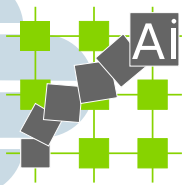
Karsten Kahl

James Brannick, Andreas Frommer, Stefan Krieg, Björn Leder,
Matthias Rottmann, Artur Strebel

Bergische Universität Wuppertal

11. April 2017





Outline

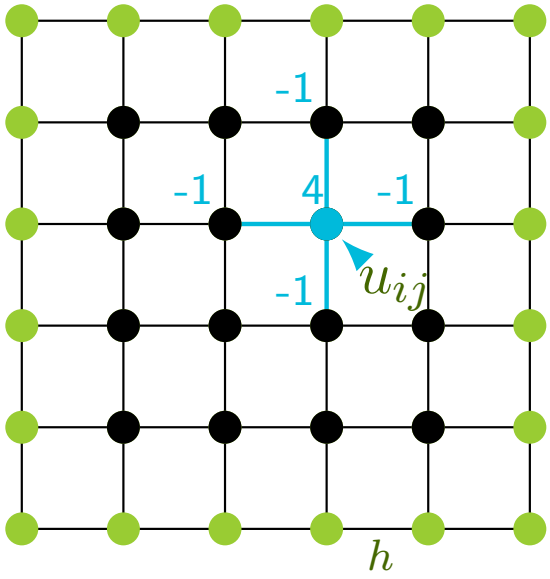
2d Laplace equation

Partial differential equation

$$\begin{aligned} -\Delta u(x,y) &= f(x,y), \quad (x,y) \in \Omega = (0,1) \times (0,1), \\ u(x,y) &= g(x,y), \quad (x,y) \in \partial\Omega \end{aligned}$$

Discretization:
grid (ih, jh), $i, j = 1, \ldots, N$, $h = \frac{1}{N+1}$

$$\begin{aligned} u_{i,j} &\approx u(ih, jh) \\ -\Delta u(ih, jh) &= \tfrac{1}{h^2} \left(4u_{i,j} - u_{i-1,j} - u_{i,j-1} \right. \\ &\quad \left. - u_{i+1,j} - u_{i,j+1}\right) + \mathcal{O}(h^2) \end{aligned}$$

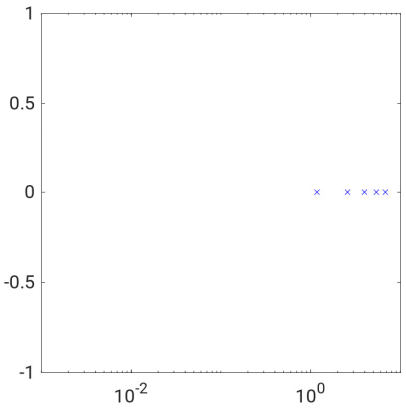


Resulting linear system: $L_h u = f$

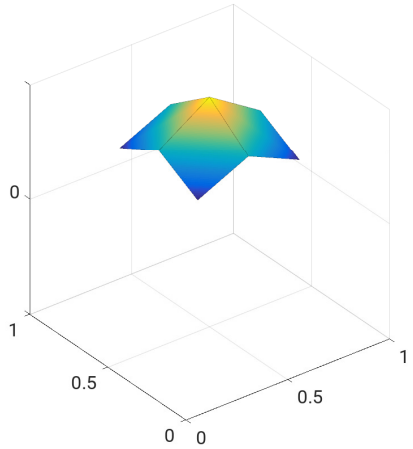
Spectral properties of L_h

Varying lattice spacing: $h = 2^{-2}$

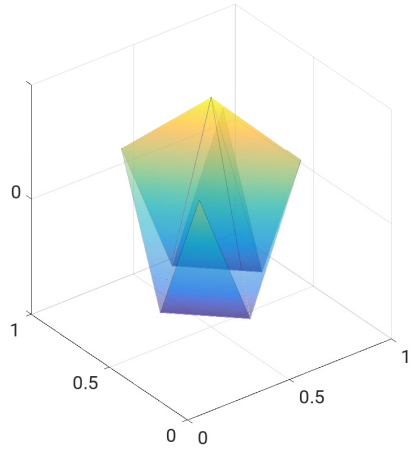
Eigenvalues



$v_{\lambda_{\min}}$



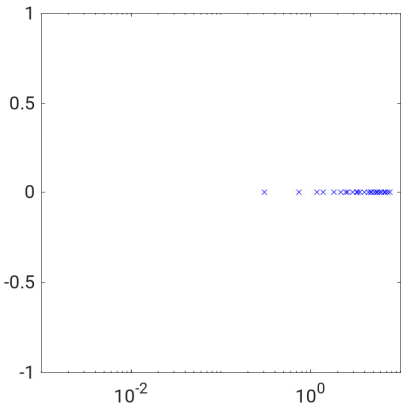
$v_{\lambda_{\max}}$



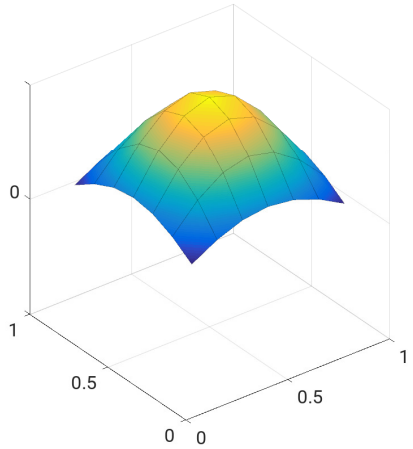
Spectral properties of L_h

Varying lattice spacing: $h = 2^{-3}$

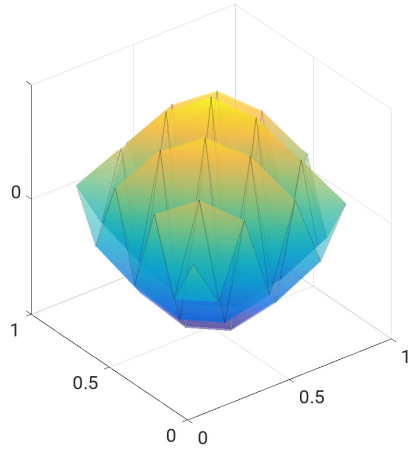
Eigenvalues



$v_{\lambda_{\min}}$



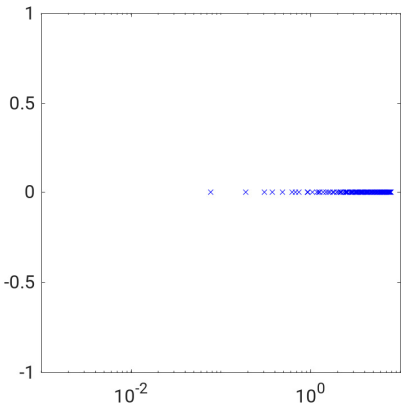
$v_{\lambda_{\max}}$



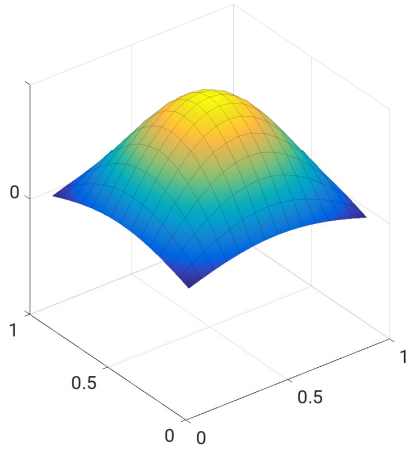
Spectral properties of L_h

Varying lattice spacing: $h = 2^{-4}$

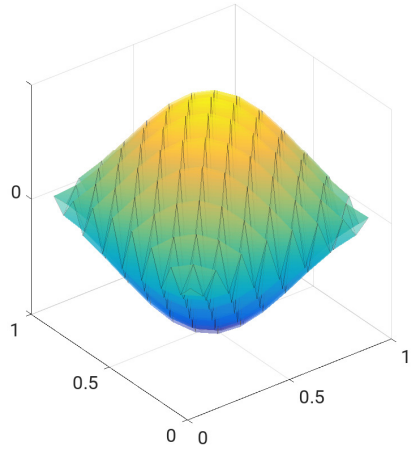
Eigenvalues



$v_{\lambda_{\min}}$



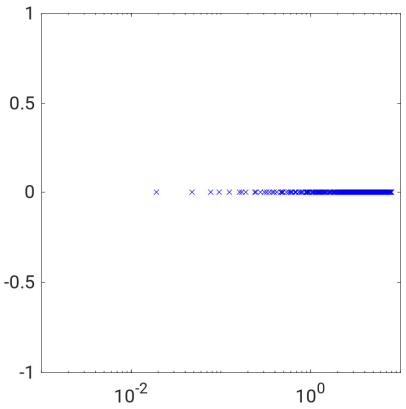
$v_{\lambda_{\max}}$



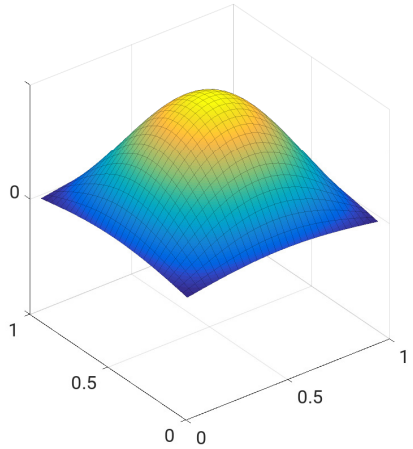
Spectral properties of L_h

Varying lattice spacing: $h = 2^{-5}$

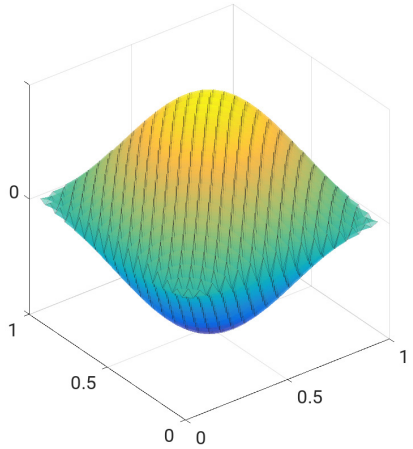
Eigenvalues



$v_{\lambda_{\min}}$



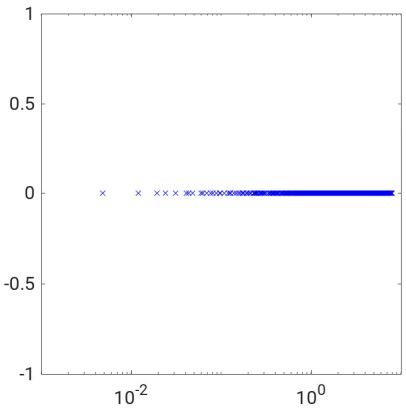
$v_{\lambda_{\max}}$



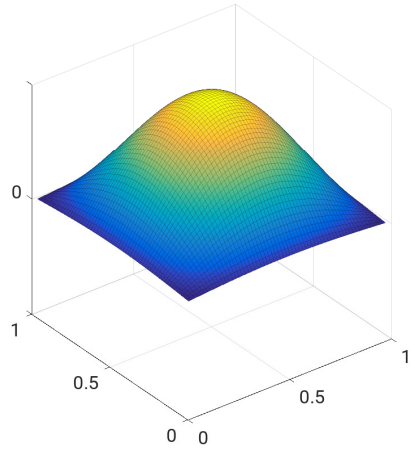
Spectral properties of L_h

Varying lattice spacing: $h = 2^{-6}$

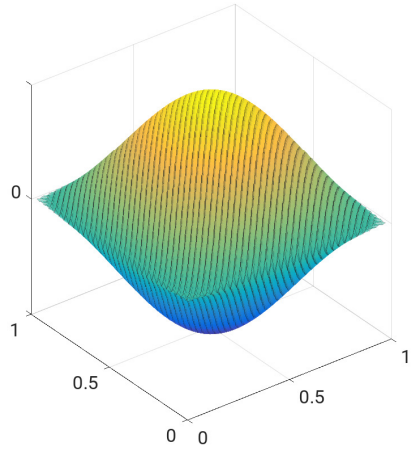
Eigenvalues



$v_{\lambda_{\min}}$



$v_{\lambda_{\max}}$



Observation:

- $\lambda_{min} \sim h^2 \quad \rightarrow \quad \kappa(L_h) \sim h^{-2}$
- $v_{\lambda_{\min}}$ "looks" the same on all lattices
  - $v^{(2h)}_{\lambda_{\min}}$ approximation of $v^{(h)}_{\lambda_{\min}}$

Relaxation: Jacobi, Gauss-Seidel

Linear system $L_h u = f$.

$$4u_{i,j} - u_{i-1,j} - u_{i,j-1} - u_{i+1,j} - u_{i,j+1} = h^2 f_{ij},\ i,j = 1,\ldots,N$$

- Jacobi iteration

$$u_{i,j}^{(k+1)} = \tfrac{1}{4}\left(h^2 f_{ij} + u_{i-1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j+1}^{(k)}\right)$$

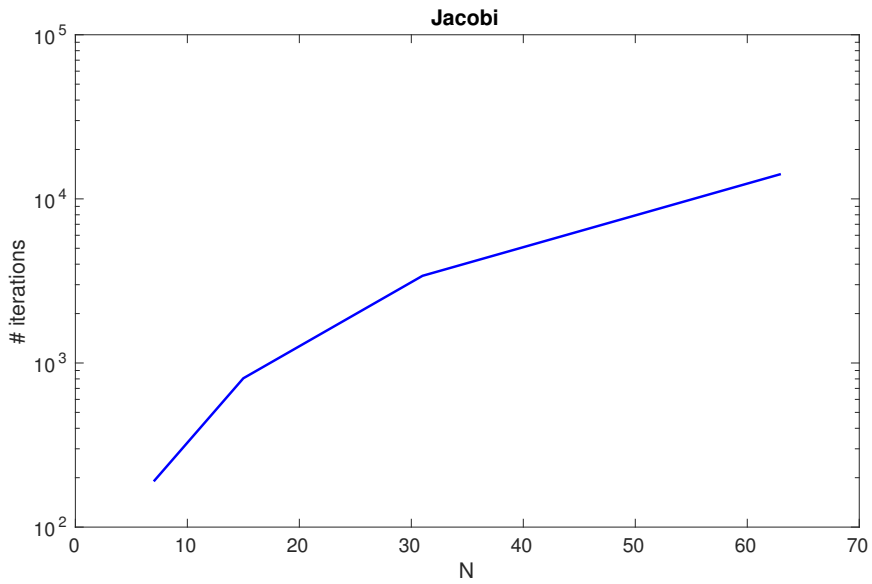
- Gauss-Seidel iteration

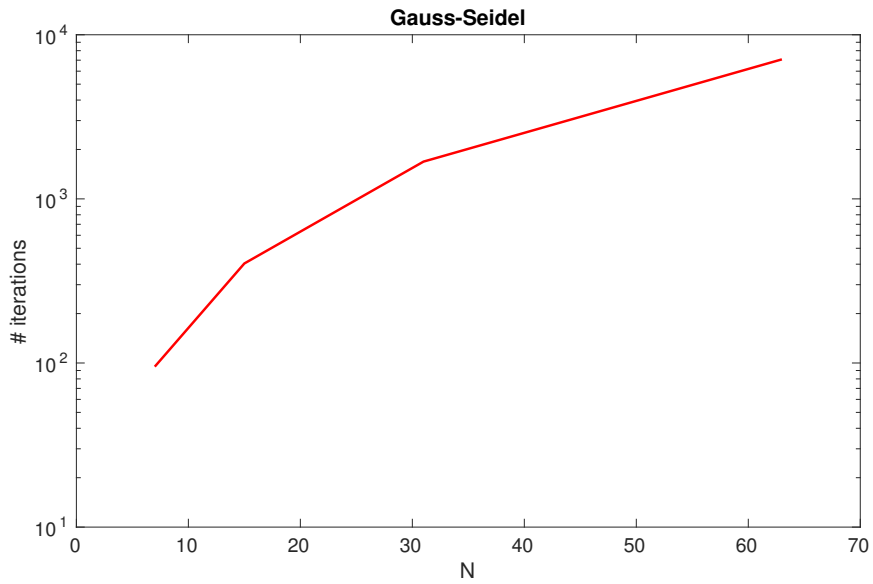
$$u_{i,j}^{(k+1)} = \tfrac{1}{4}\left(h^2 f_{ij} + u_{i-1,j}^{(k+1)} + u_{i,j-1}^{(k+1)} + u_{i+1,j}^{(k)} + u_{i,j+1}^{(k)}\right)$$

$$\begin{aligned} u &\leftarrow (I - ML_h)u + Mf \\ e &\leftarrow (I - ML_h)e \end{aligned}$$

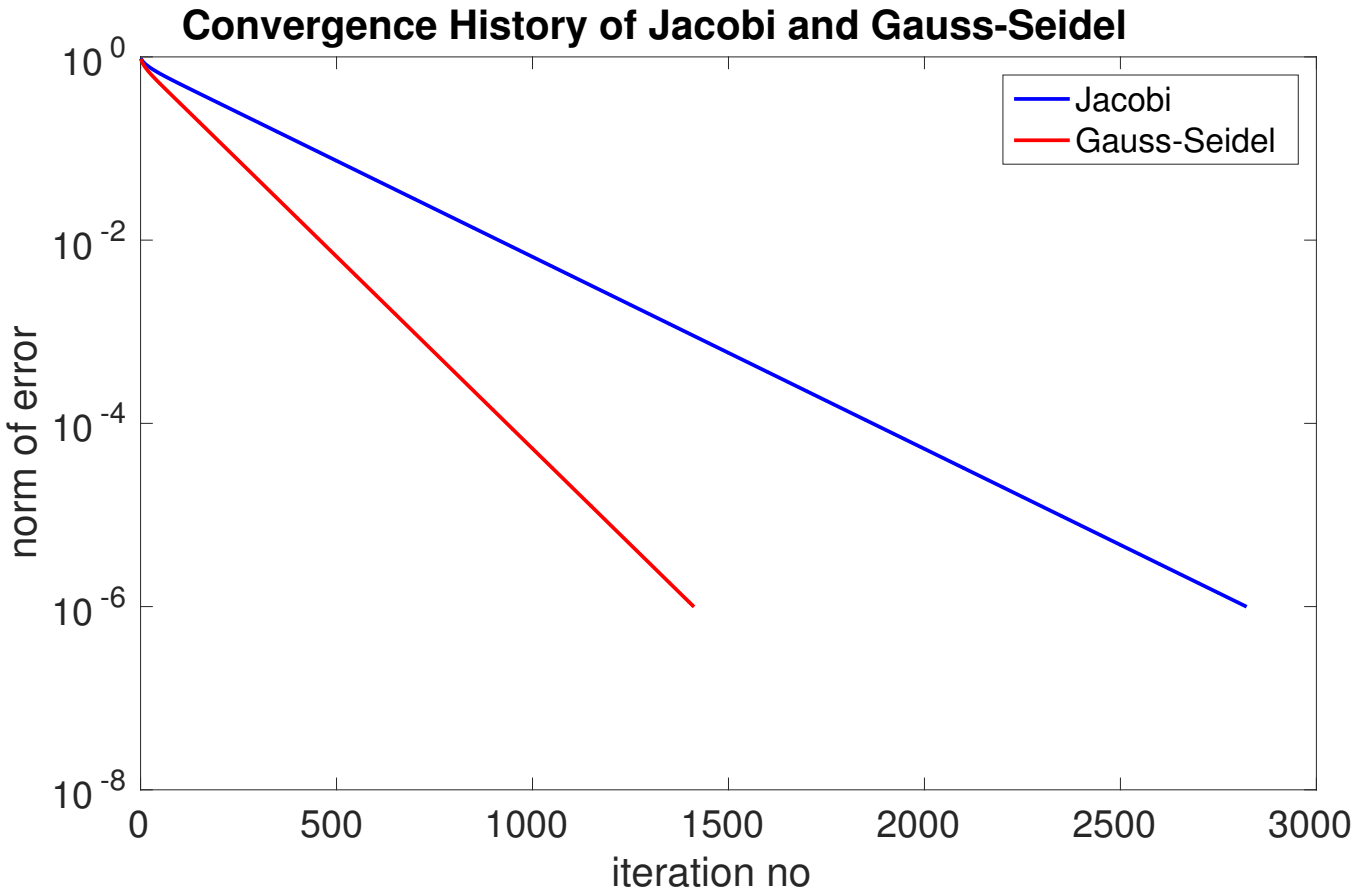
Numerical experiments

Relative tolerance 10^{-6}

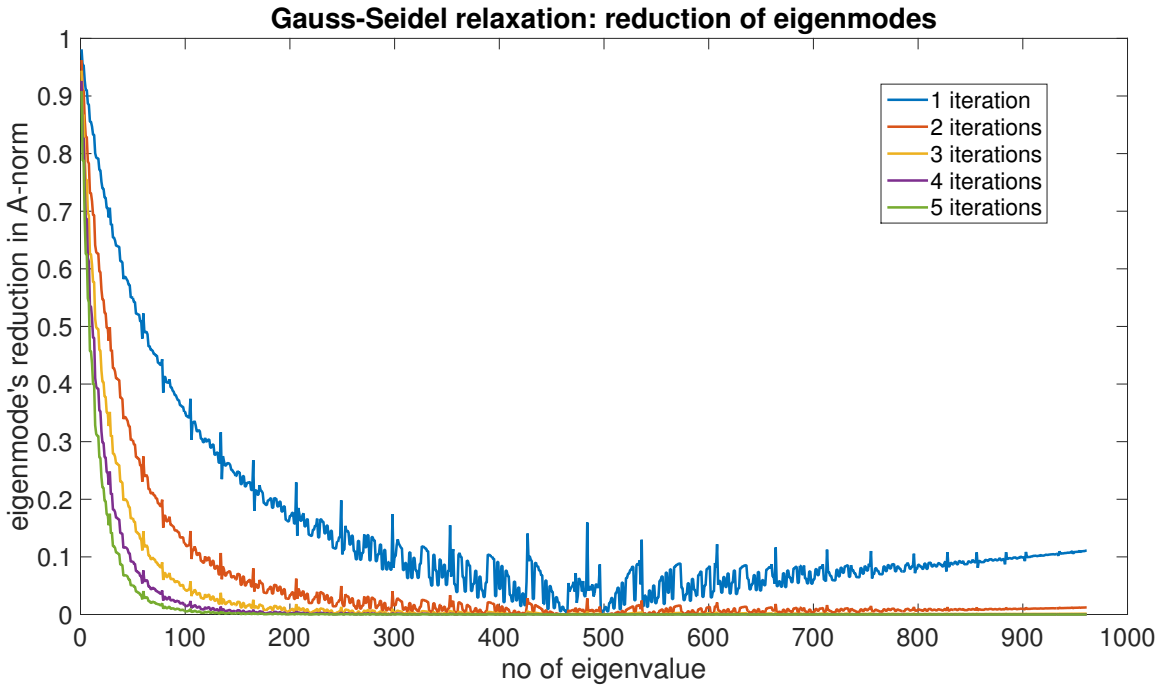




Convergence history

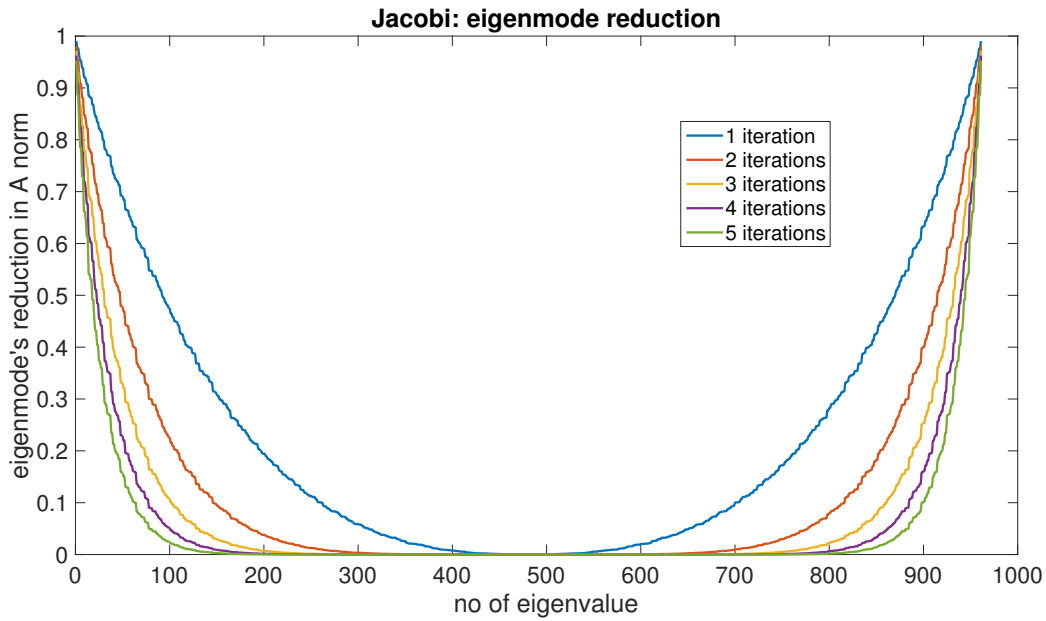


Details: action on eigenmodes – Gauss-Seidel

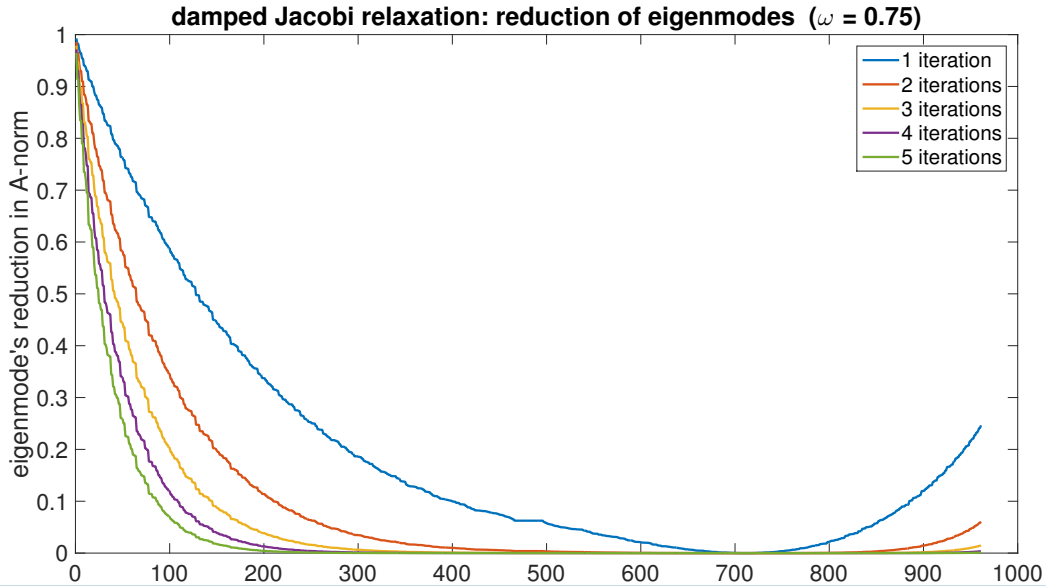


Details: action on eigenmodes – Jacobi

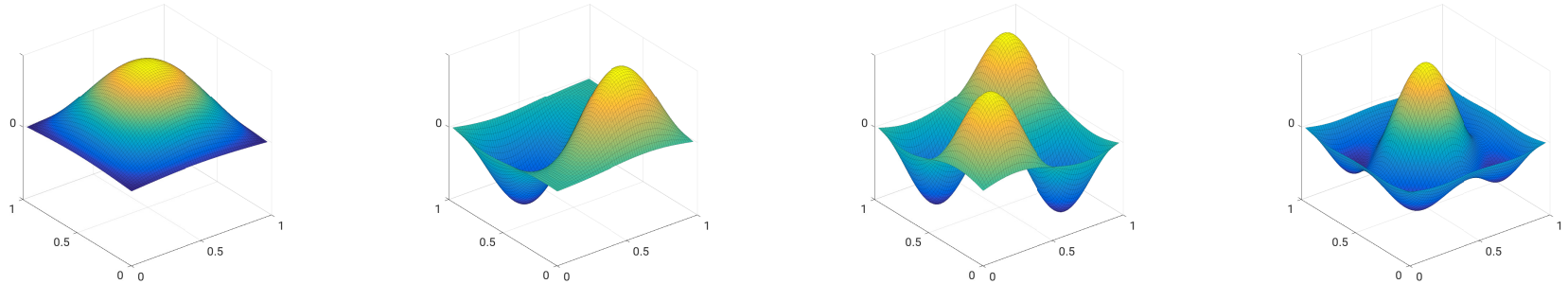
plain Jacobi



damped Jacobi



Small eigenmodes are smooth



Observation: If the error e is composed of small eigenmodes, it is geometrically smooth.

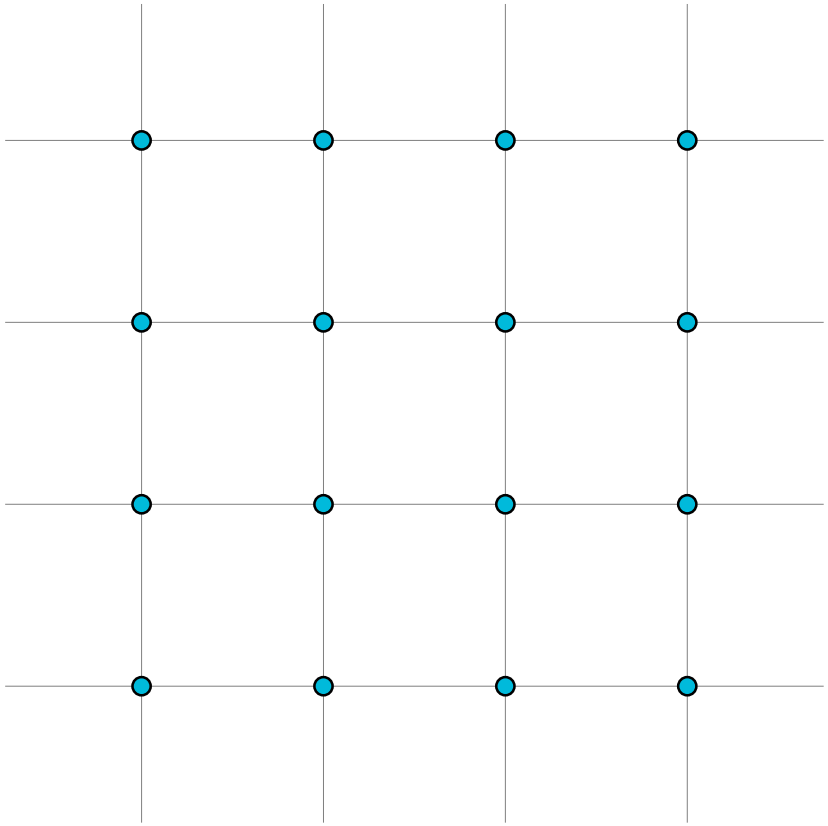
Consequence: Smooth error can be approximated from coarser discretization.

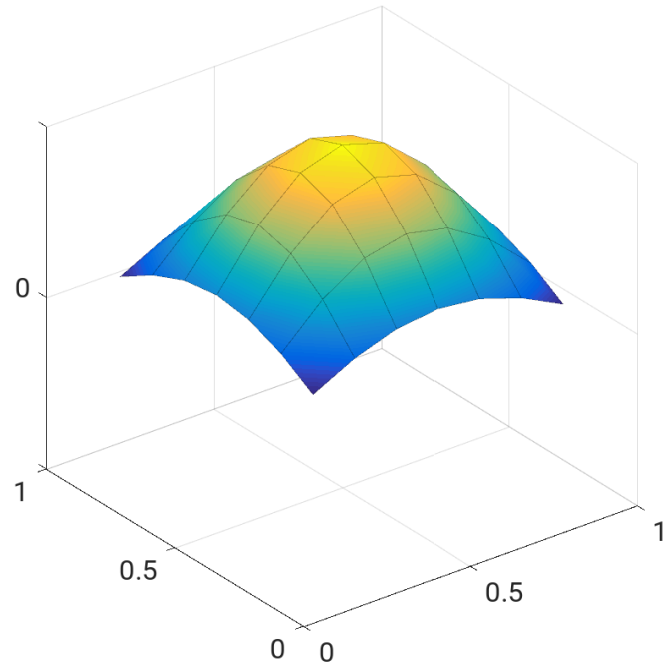
- On fine grid: $L_h e = r \ (= f - L_h u)$
- On coarse grid $L_{2h} e_c = Rr$, $e \approx P e_c$, where:

  R: restriction operator: fine grid to coarse grid
  P: prolongation operator: coarse grid to fine grid

Prolongation and Restriction

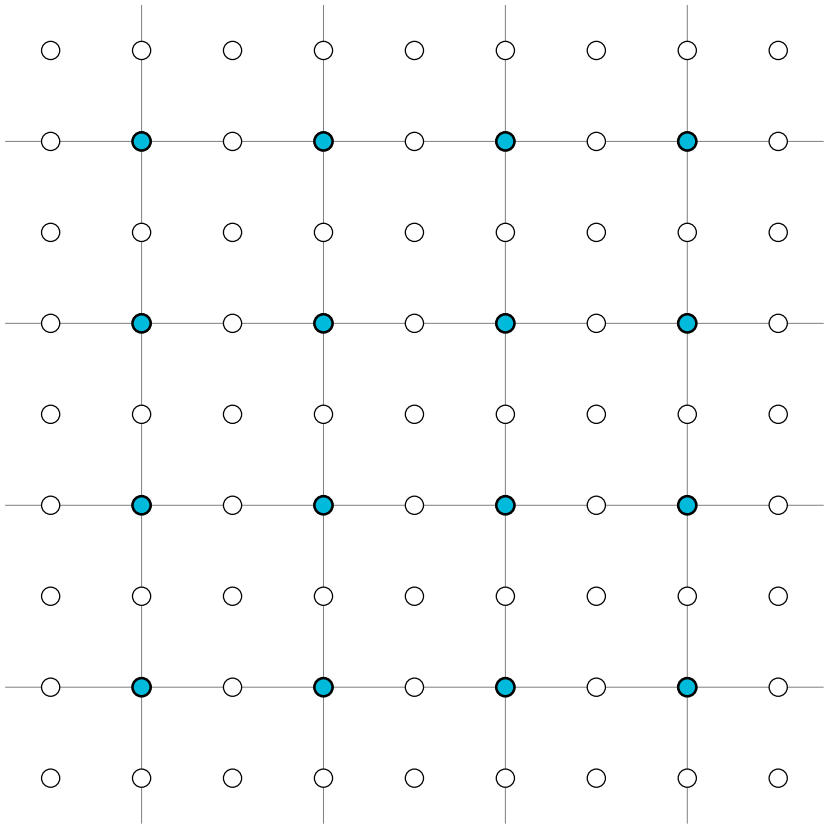
Linear interpolation

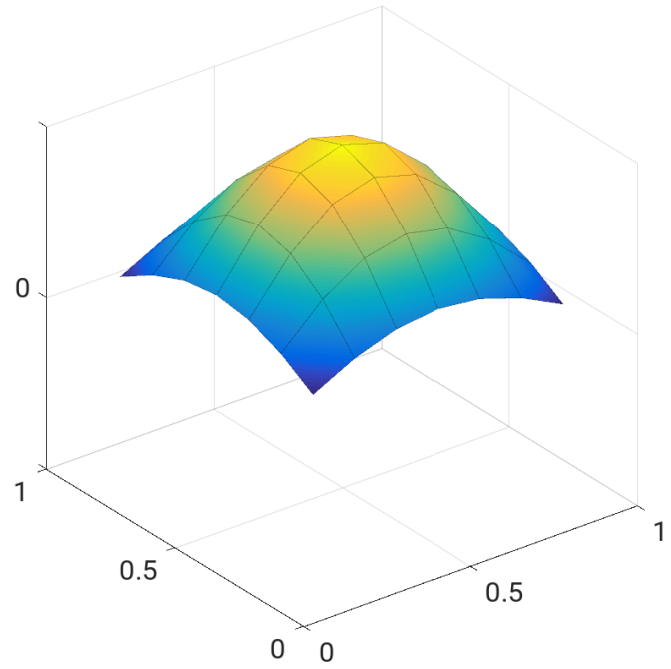




Prolongation and Restriction

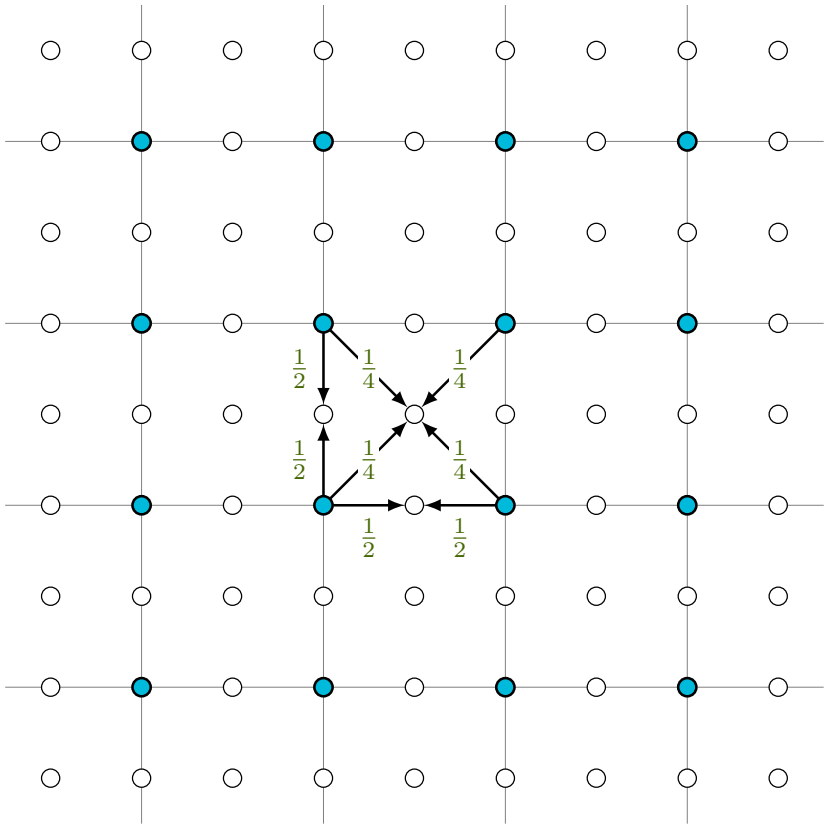
Linear interpolation

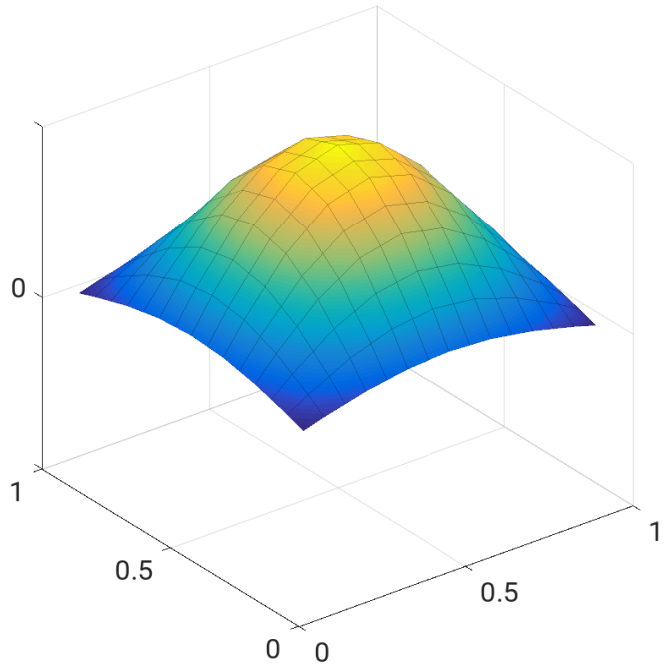




Prolongation and Restriction

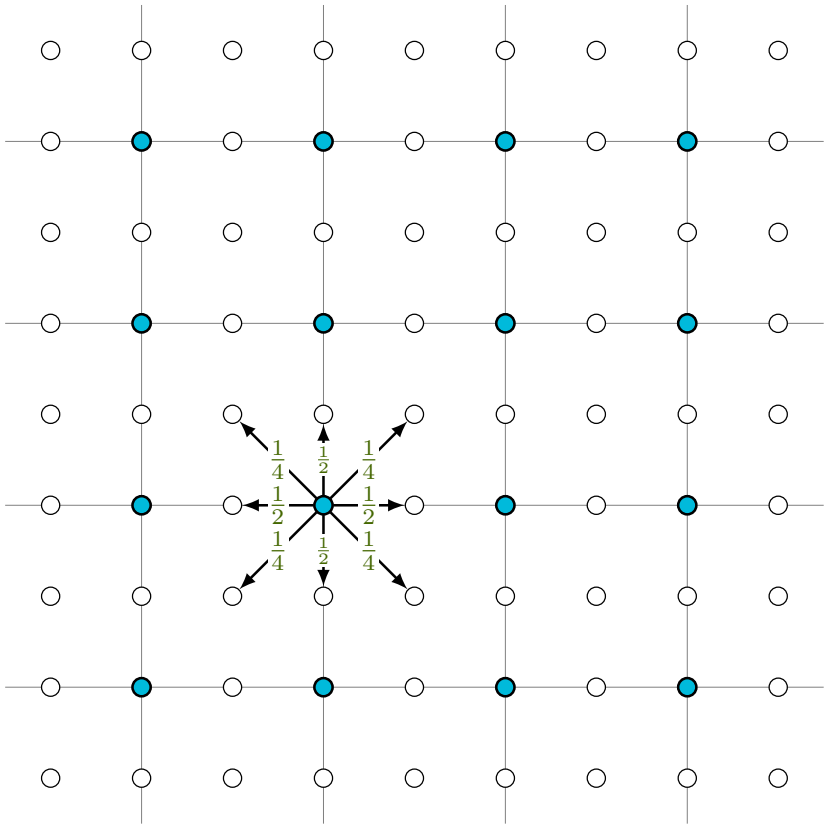
Linear interpolation

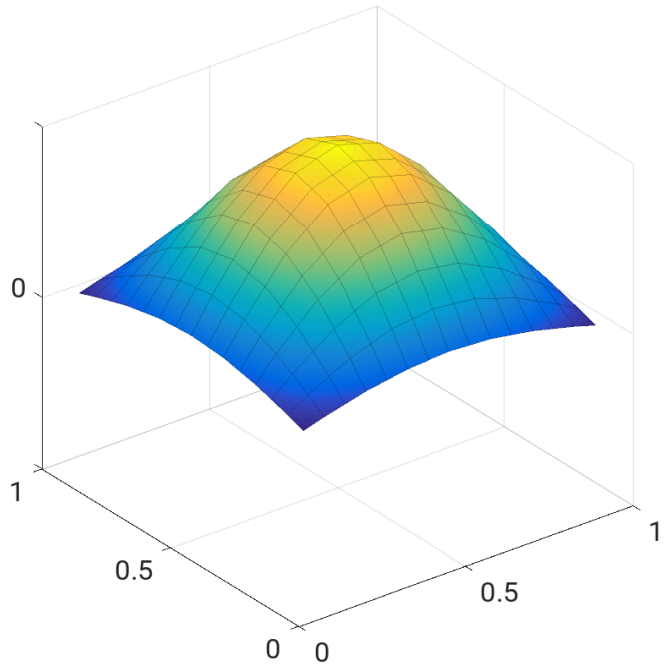




Prolongation and Restriction

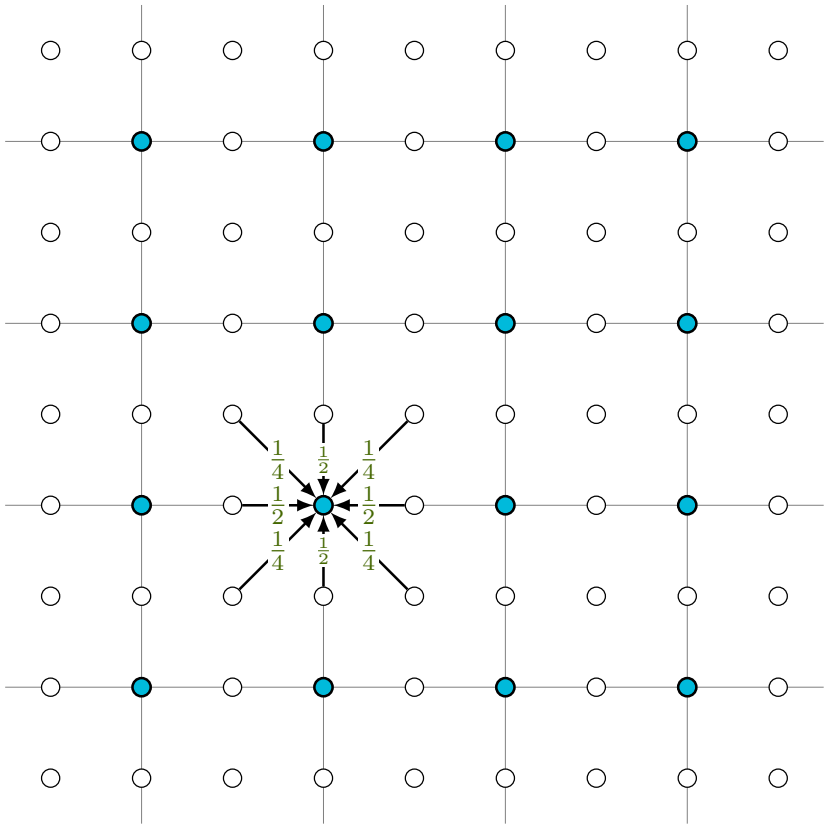
Linear interpolation

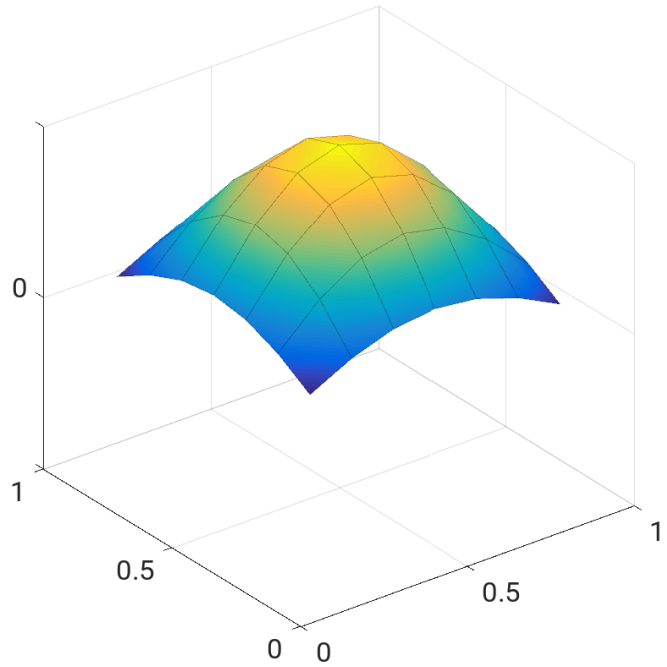




Prolongation and Restriction

Local averaging

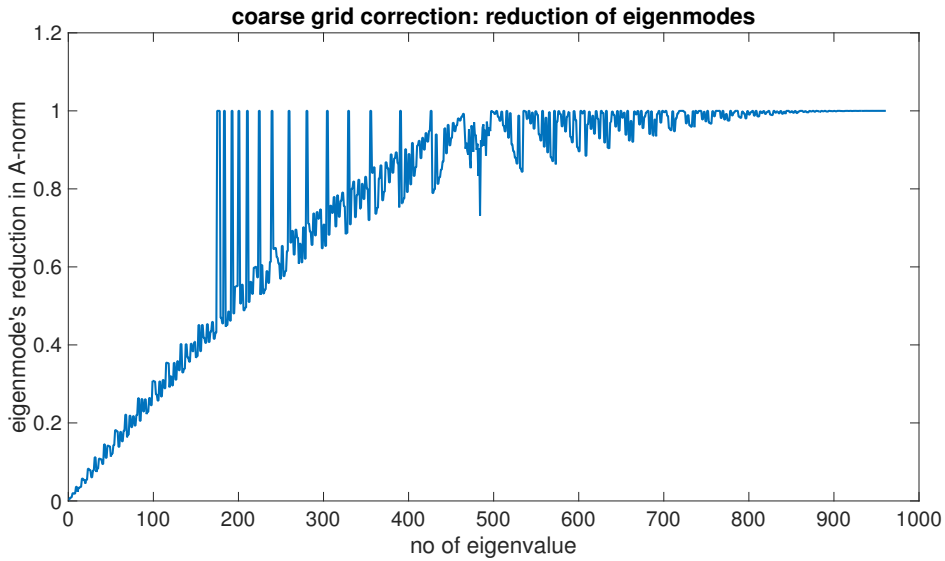




The coarse grid correction

Coarse grid correction for current iterate x:

- Compute $r = f - L_h u$
- Restrict $r_c = Rr$
- Solve $L_{2h} e_c = r_c$
- correct $u \leftarrow u + P e_c$



$$
\begin{aligned}
u &\leftarrow u + P L_{2h}^{-1} R (f - L_h u) \\
e &\rightarrow (I - P L_{2h}^{-1} R L_h) e
\end{aligned}
$$

The Galerkin coarse grid operator

So far: L_{2h} obtained via same discretization scheme, but on coarser grid

Alternative: The (Petrov-)Galerkin coarse grid operator is obtained as

$$L_{2h}^G = RL_hP$$

Consequences:

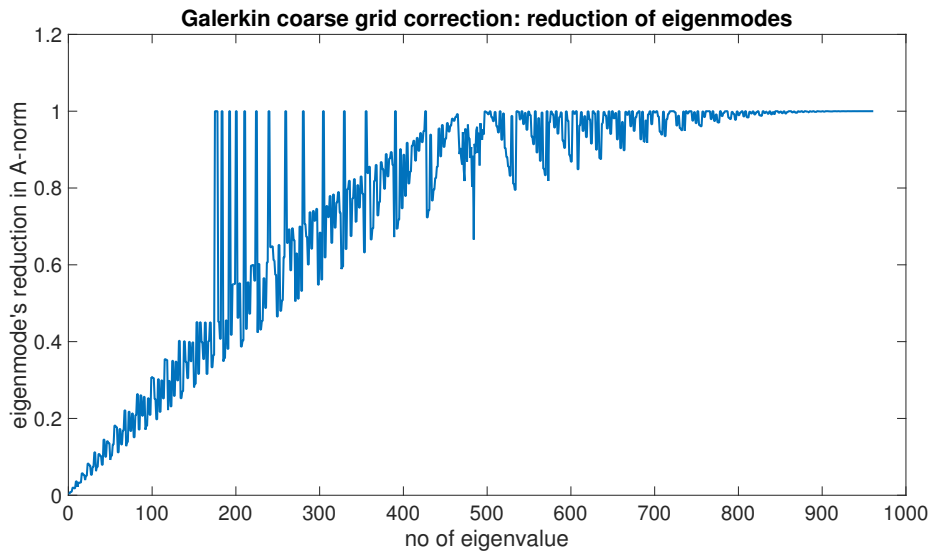
- Coarse grid projection operator $(I - P(L_{2h}^G)^{-1}RL_h)$ is indeed a projection:

$$P(L_{2h}^G)^{-1}RL_h \cdot P(L_{2h}^G)^{-1}RL_h = P(L_{2h}^G)^{-1}RL_h$$

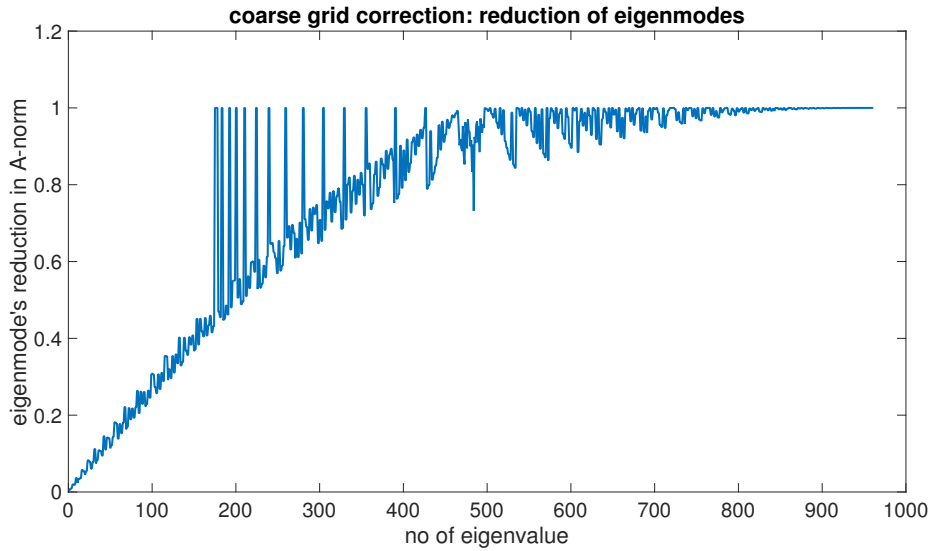
- Since $P(L_{2h}^G)^{-1}RL_h$ projects on $\mathrm{range}(P)$, $\mathrm{range}(P)$ should well approximate small eigenmodes.
- If $R = P^\dagger$, then $P(L_{2h}^G)^{-1}RL_h$ is an L_h-orthogonal projection (L_h is spd)

Coarse grid correction with Galerkin operator

Galerkin



L_{2h}



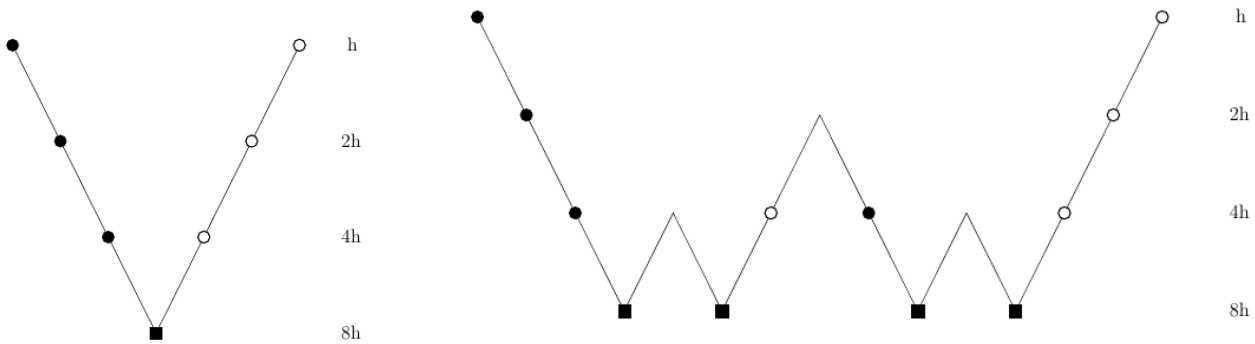
Two-grid and multigrid

Two-grid method

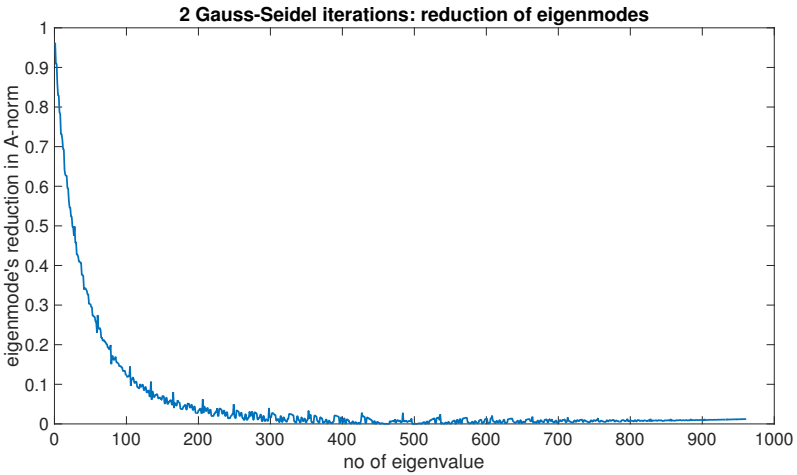
- alternate relaxation and coarse grid correction
- solve coarse system (L_{2h}) exactly

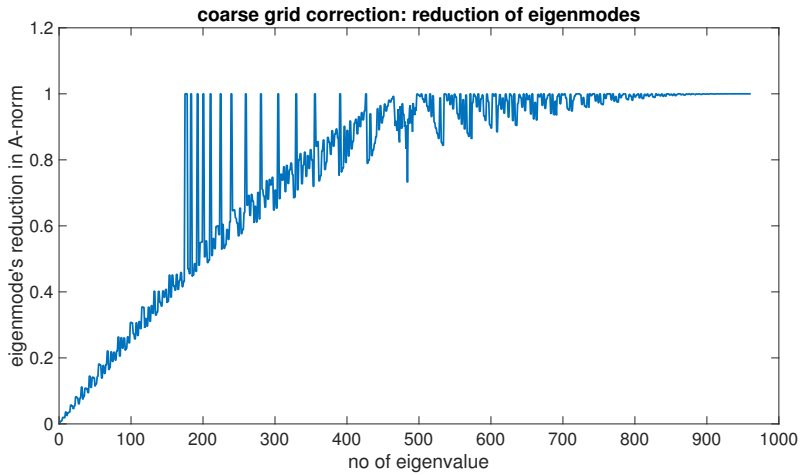
Multigrid

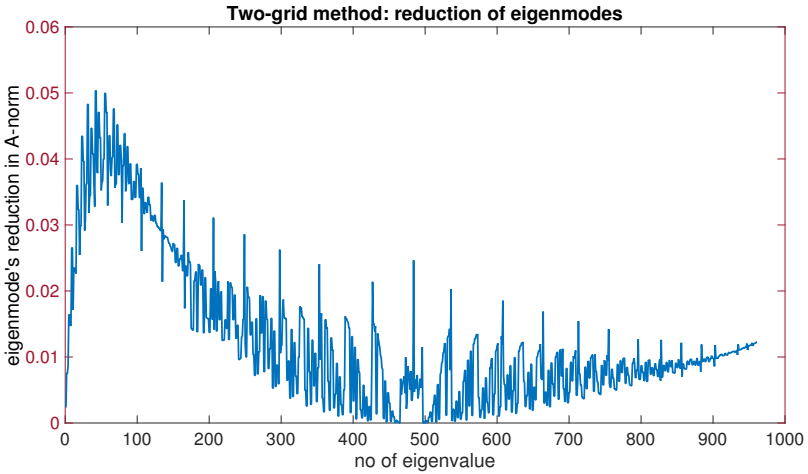
- use multigrid recursively to solve coarse system in coarse grid equation
- different cycling strategies

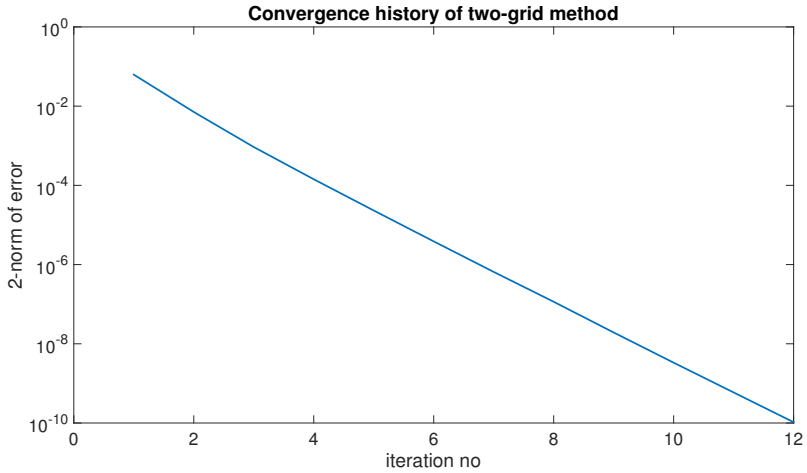


Eigenmode reduction of the two-grid scheme

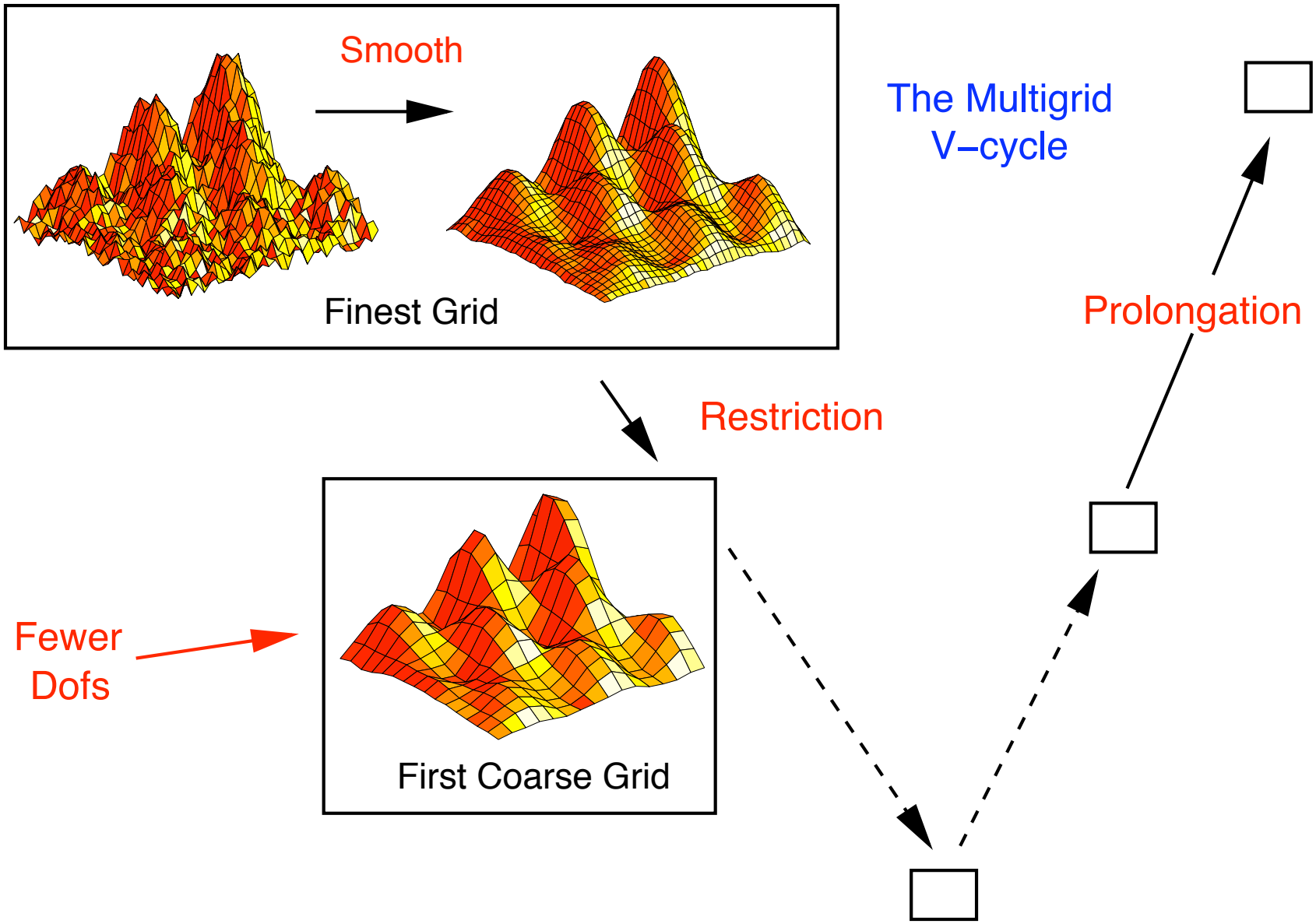








In a nutshell



Convergence theory: an example

Let A be Hermitian and positive definite. Let S be the smoothing operator. Assume that

- $P = R^\dagger$,
- $A_c = RAP$ (Galerkin operator),
- $\|Se\|_A^2 \leq \|e\|_A^2 - \alpha_1 \|e\|_{AD^{-1}A}^2$ "smoothing property",
- $\min_{v_c} \|u - Pv_c\|_D \leq \beta \|u\|_A^2$ "weak approximation prop.".

Then

$$\|S(I - PA_c^1 RA)\|_A \leq \sqrt{1 - \frac{\alpha_1}{\beta}}.$$

Framework

- D: Wilson Dirac operator
- Background gauge fields introduce randomness in couplings
- There is no "natural" $D_{2h} \rightarrow$ Galerkin construction
- Generic two-grid iteration for $D\psi = \eta$
  - current iterate ψ^k
  - apply smoother: $\psi^k \leftarrow (I - MD)\psi^k + M\eta$
  - compute residual: $r^k = \eta - D\psi^k$
  - apply coarse grid correction: $e^k = PD_c^{-1}Rr^k$
  - next iterate: $\psi^{k+1} = \psi^k + e^k$

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

$$E_{2g}^{(\nu)} = \underbrace{(I - PD_c^{-1}P^\dagger D)}_{\text{coarse grid correction}} \underbrace{(I - MD)^\nu}_{\text{smoother}}, \quad \underbrace{D_c := P^\dagger DP}_{\text{coarse operator}}$$

- low accuracy for D_c^{-1} and M is sufficient
- introduce recursive construction for $D_c \rightarrow$ multigrid

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]

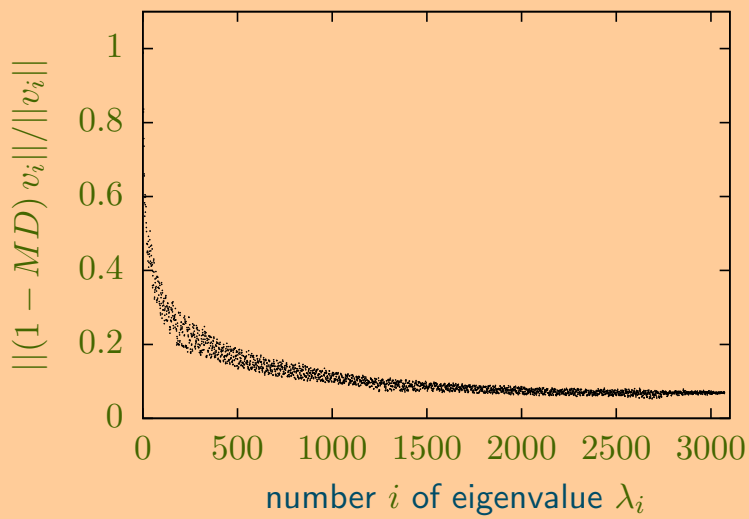
P: Aggregation Based Interpolation
[Brannick, Clark et al. 2010]

Preview: The multigrid principle for Dirac-Wilson

Smoother: $I - MD$

- Effective on "large" eigenvectors
- "small" eigenvectors remain

SAP



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

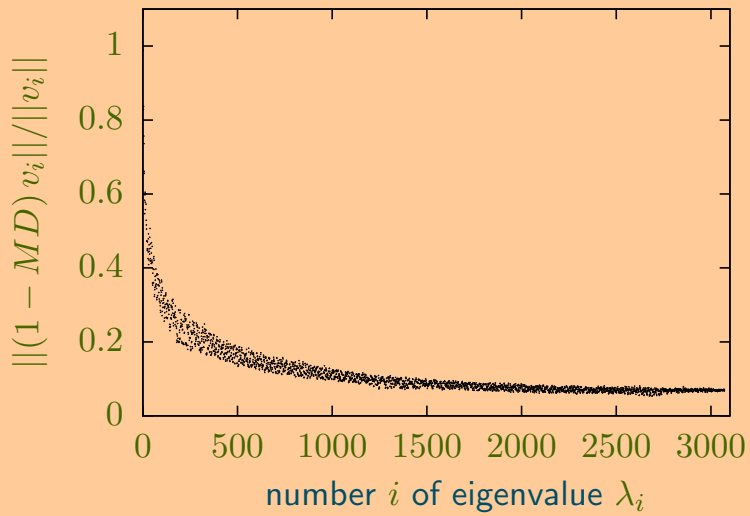
Preview: The multigrid principle for Dirac-Wilson

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

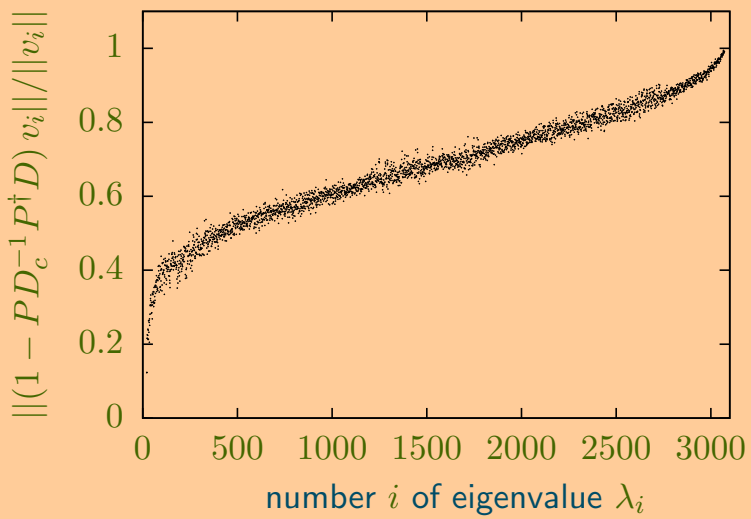
- small eigenvectors built into interpolation P

  $\Rightarrow$ Effective on small eigenvectors

SAP



Aggregation + Low Modes



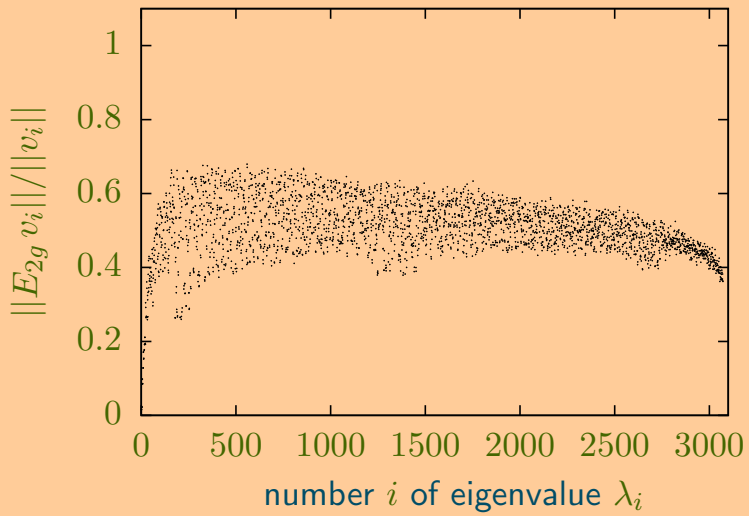
$Dv_i = \lambda_i v_i$ with $|\lambda_1| \leq \ldots \leq |\lambda_{3072}|$

Preview: The multigrid principle for Dirac-Wilson

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

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

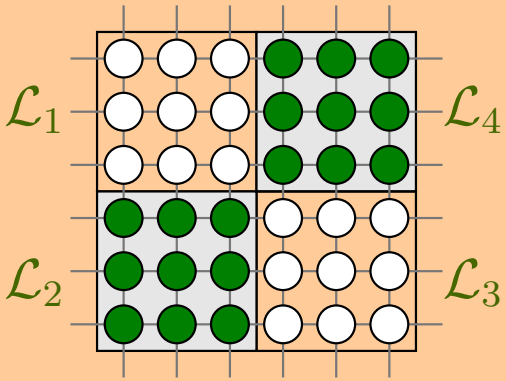
DD-αAMG



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

SAP: Schwarz Alternating Procedure

Two color decomposition of $\mathcal{L}$



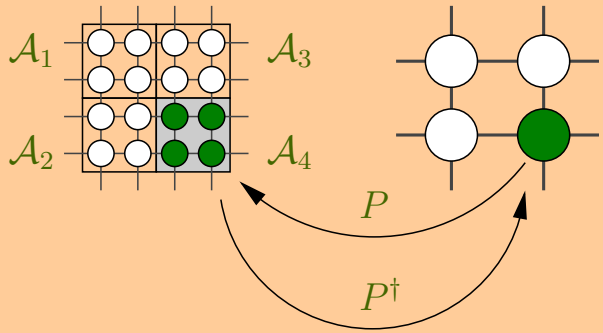
- canonical injections
  $$\mathcal{I}_{\mathcal{L}_i} : \mathcal{L}_i \to \mathcal{L}$$
- block restrictions
  $$D_{\mathcal{L}_i} = \mathcal{I}^\dagger_{\mathcal{L}_i} D \mathcal{I}_{\mathcal{L}_i}$$
- block inverses
  $$B_{\mathcal{L}_i} = \mathcal{I}_{\mathcal{L}_i} D^{-1}_{\mathcal{L}_i} \mathcal{I}^\dagger_{\mathcal{L}_i}$$

1: in: ψ, η, ν – out: ψ
2: **for** $k = 1$ to ν **do**
3: $\quad r \leftarrow \eta - D\psi$
4: $\quad$ **for all** *green* $\mathcal{L}_i$ **do**
5: $\quad\quad \psi \leftarrow \psi + B_{\mathcal{L}_i} r$
6: $\quad$ **end for**
7: $\quad r \leftarrow \eta - D\psi$
8: $\quad$ **for all** *white* $\mathcal{L}_i$ **do**
9: $\quad\quad \psi \leftarrow \psi + B_{\mathcal{L}_i} r$
10: $\quad$ **end for**
11: **end for**

Aggregation Based Interpolation

Construction:

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

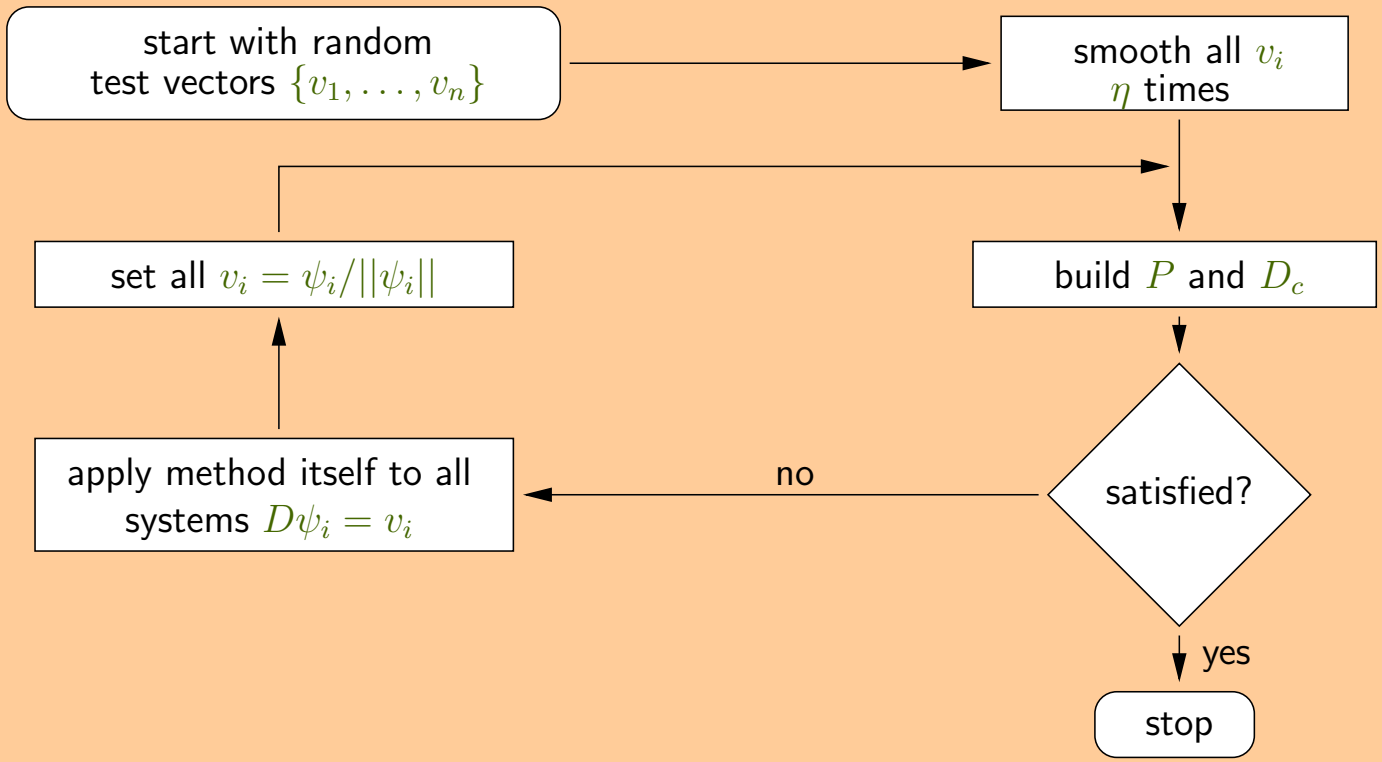


- Calculate test vectors $w_1, ..., w_N$ [ArXiv:1303.1377,1307.6101]
- Decompose test vectors over aggregates $\mathcal{A}_1, ..., \mathcal{A}_s$

$$(v^{(1)}, \ldots, v^{(k)}) = \left|\begin{matrix}\\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix}\right| = \begin{matrix}\mathcal{A}_1\\ \mathcal{A}_2\\ \vdots\\ \mathcal{A}_s\end{matrix} \rightarrow P = \begin{pmatrix}\mathcal{A}_1 & & \\ & \mathcal{A}_2 & \\ & & \ddots \\ & & & \mathcal{A}_s\end{pmatrix}$$

Setup Procedure: How to Obtain Test Vectors

Bootstrapping process



Krylov acceleration

Recall:

$$\psi^k - \psi = (I - MD)(\psi^{k-1} - \psi) = (I - MD)^k(\psi^0 - \psi)$$
$$\Rightarrow \quad \psi^k - \psi = p_k(MD)(\psi^0 - \psi), \qquad p_k(t) = (1-t)^k$$

Note: $\lim_{k\to\infty} \psi^k - \psi = 0$ for any $\psi^0 \;\Leftrightarrow\; \rho(I - MD) < 1$.

Krylov acceleration

Krylov method "chooses" ploynomial p_k better than $(1-t)^k$:

- CG: minimizes $\langle p_k(MD)(\psi^0 - \psi) \,|\, D \,|\, p_k(MD)(\psi^0 - \psi)\rangle$
- GMRES minimizes $\|MD\big(p_k(MD)(\psi^0 - \psi)\big)\|_2$
- BiCG, QMR, BiCGStab

Terminology: M preconditioner

Some history

- **Adaptive algebraic multigrid αAMG:** Brezina, Falgout, Manteuffel, MacLachlan, McCormick, Ruge 2004
- **Inexact deflation method:** Lüscher 2007.
  Solves
  $$D(I - PD_c^{-1}P^\dagger D)\psi = \eta$$
  using SAP as a preconditioner.
- **αAMG for lattice QCD:** Babich, Brannick, Brower, Clark, Manteuffel, McCormick, Osborn, Rebbi 2010.
- **DD-αAMG:** F., Kahl, Leder, Krieg, Rottmann 2013

Some history

- **Adaptive algebraic multigrid αAMG:** Brezina, Falgout, Manteuffel, MacLachlan, McCormick, Ruge 2004
- **Inexact deflation method:** Lüscher 2007.
  Solves
  $$D(I - PD_c^{-1}P^\dagger D)\psi = \eta$$
  using SAP as a preconditioner.
- **αAMG for lattice QCD:** Babich, Brannick, Brower, Clark, Manteuffel, McCormick, Osborn, Rebbi 2010.
- **DD-αAMG:** F., Kahl, Leder, Krieg, Rottmann 2013

- 2013: "Inexact deflation with inexact projection"
- QPACE2: Targeted implementation of DD-αAMG
- Targeted implementations within BMWc

Current AMG solvers for D_W

	QOPQDP	OpenQCD	DD-αAMG
clover term	included	included	included
mixed precision	yes	yes	yes
smoother	GMRES	SAP	SAP
aggregation	γ_5-comp.	arbitrary	γ_5-comp.
setup	1)	2)	3)
typ. # test vecs (N)	20	30	20
# vars / coarse site	$2N$	N	$2N$
cycling	K-cycle	n.a.	K-cycle

1) inverse iterations with GMRES on sequence of test vecs
2) repeated inverse iteration with emerging solver on all test vecs at once
3) modification of 2)

Snapshots on performance

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	48×16^3	250	7,055	-0.095300	1.00000	BMW-c
2	48×24^3	250	11,664	-0.095300	1.00000	BMW-c
3	48×32^3	250	15,872	-0.095300	1.00000	BMW-c
4	48×48^3	135	53,932	-0.099330	1.00000	BMW-c
5	64×64^3	135	84,207	-0.052940	1.00000	BMW-c
6	128×64^3	270	45,804	-0.342623	1.75150	CLS

Tabelle: Ensembles used.

Snapshots on performance

Setup time vs. solve time

number of setup steps n_{inv}	average setup timing	average iteration count	lowest iteration count	highest iteration count	average solver timing	average total timing
1	2.08	149	144	154	6.42	8.50
2	3.06	59.5	58	61	3.42	6.48
3	4.69	34.5	33	36	2.37	7.06
4	7.39	27.2	27	28	1.95	9.34
5	10.8	24.1	24	25	1.82	12.6
6	14.1	23.0	23	23	1.89	16.0
8	19.5	22.0	22	22	2.02	21.5
10	24.3	22.5	22	23	2.31	26.6

Tabelle: Evaluation of DD-αAMG-setup($n_{inv}, 2$), 48^4 lattice, configuration id 4), 2,592 cores, averaged over 20 runs.

Snapshots on performance

oe-BiCGStab vs. DD-αAMG

	BiCGStab	DD-αAMG	speed-up factor	coarse grid
setup time		22.9s		
solve iter	13,450	21		$3{,}716^{(*)}$
solve time	91.2s	3.15s	29.0	2.43s
total time	91.2s	26.1s	3.50	

Tabelle: BiCGStab vs. DD-αAMG with default parameters, configuration id 5, $8{,}192$ cores, $(*)$: coarse grid iterations summed up over all iterations on the fine grid.

Snapshots on performance

Mass scaling and levels

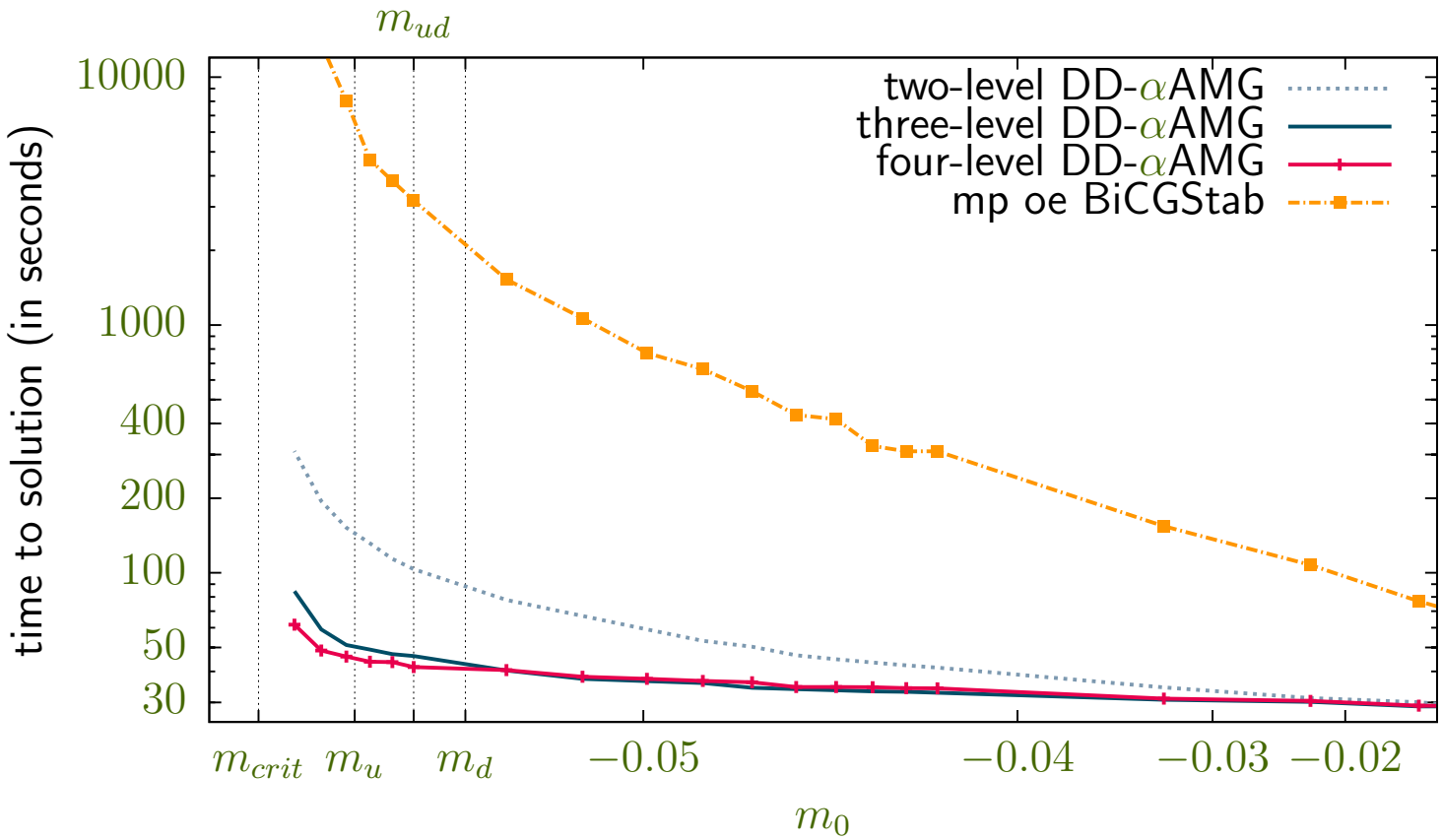


Abbildung: Mass scaling of 2, 3 and 4 level DD-αAMG, 64^4 lattice, configuration id 5, restart length $n_{kv} = 10$, 128 cores

Snapshots on performance

Mass scaling and levels

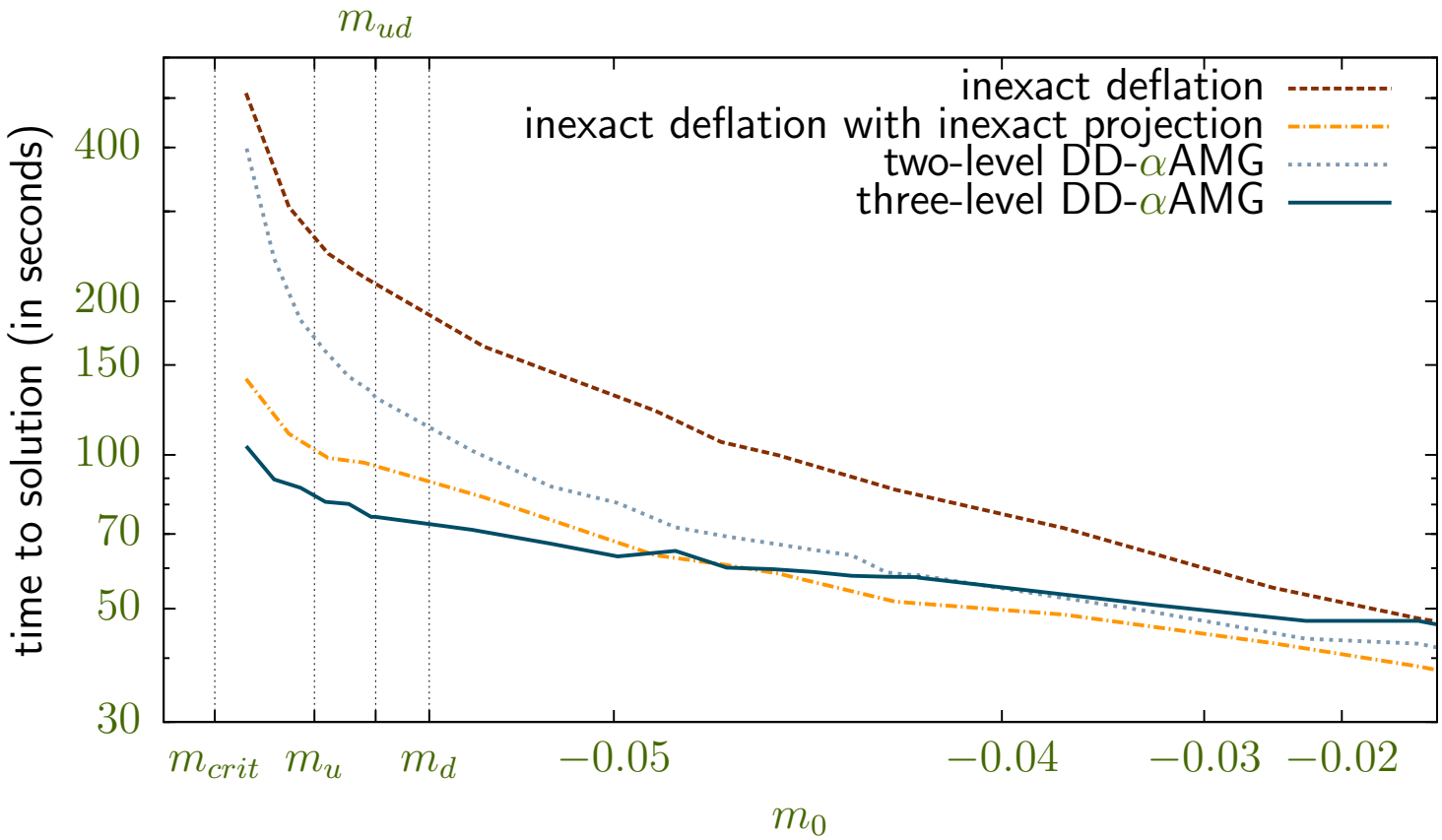
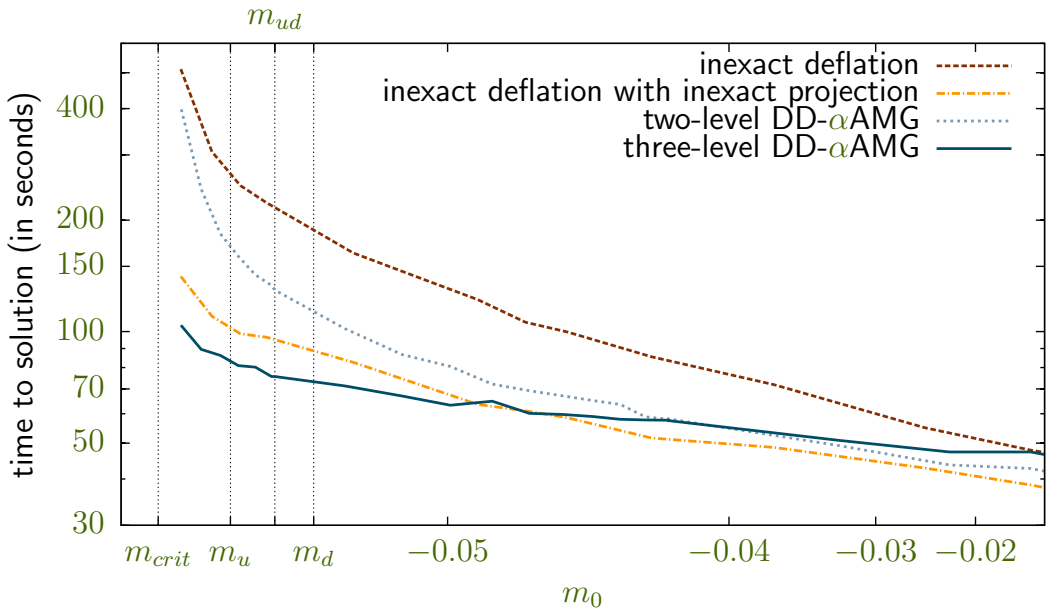


Abbildung: Mass scaling of 2, 3 and 4 level DD-αAMG, 64^4 lattice, configuration id 5, restart length $n_{kv} = 10$, 128 cores

2 & 3 Level DD-αAMG, Inexact Deflation

Configuration 5: 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

Wuppertal und das Bergische Land

Home of the tools industry

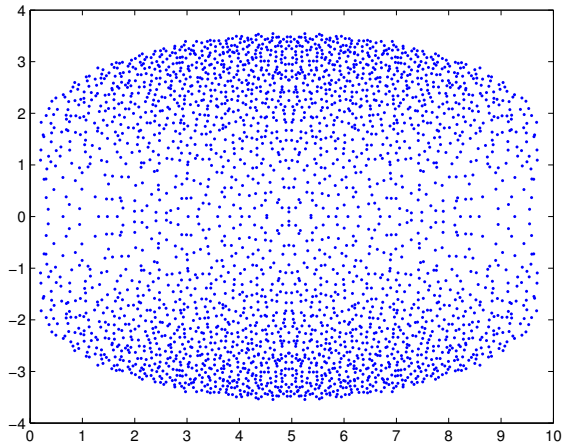
Wuppertal und das Bergische Land

Home of the tools industry

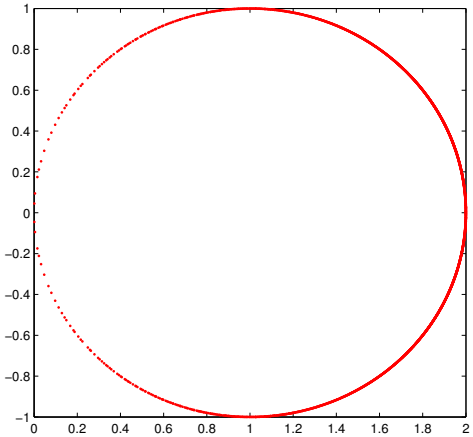
We want many nails!

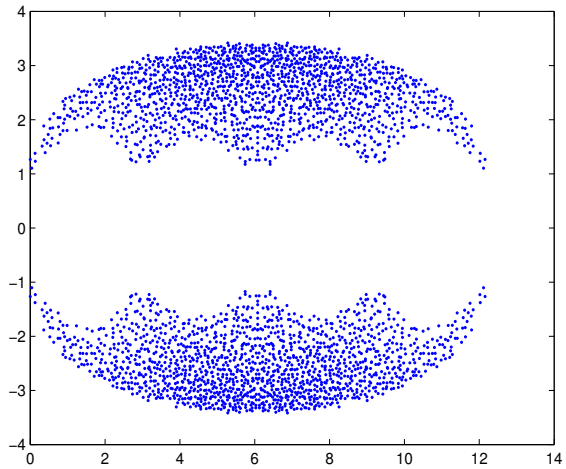
The overlap operator

$$D_N = I + \rho\gamma_5 \operatorname{sign}(\underbrace{\gamma_5 D_W}_{:=Q})$$

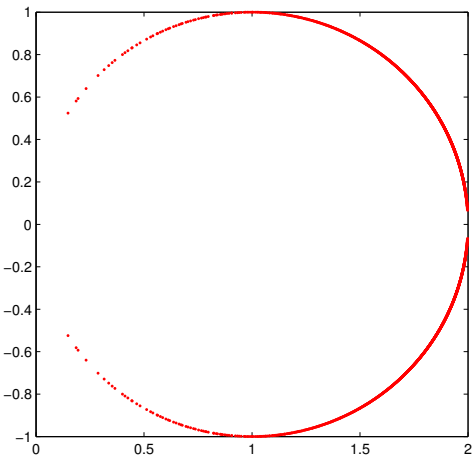


$\rightarrow$





$\rightarrow$



Solving $D_N\psi = \eta$

$$D_N = I + \rho \cdot \gamma_5 \cdot \mathrm{sign}(Q(m_k))$$

Generic Krylov subspace iteration for $D_N\psi = \eta$

1: **while** error too large **do**
2: compute next basis vector (involves computation $D_N v$)
3: update current iterate
4: **end while**

Solving $D_N\psi = \eta$

$$D_N = I + \rho \cdot \gamma_5 \cdot \mathrm{sign}(Q(m_k))$$

Generic Krylov subspace iteration for $D_N\psi = \eta$

1: **while** error too large **do**
2: compute next basis vector (involves computation $D_N v$)
3: update current iterate
4: **end while**

Challenges:

i) Evaluating $\mathrm{sign}(Q(m_k))v$ is quite costly within $D_N v$

ii) Iteration counts of $\mathcal{O}(1000) \rightarrow$ preconditioning

Current approach: recursive preconditioning

Idea: Preconditioner = "inner" iteration with GMRES for D_N

Consequences:

- inner iteration requires low accuracy only
- needs "flexible" outer iteration (FGMRES, GCR)
- requires low accuracy for $\mathrm{sign}(Q(m_k))c$ only
- accury for sign in outer iteration can be decreased as iteration proceeds
  Simoncini, Szyld [03], van den Eshof, Sleijpen [04]

Cundy, van den Eshof, F., Krieg, Schäfer 2005

New approach: use multigrid solver for D_W

Definition: D_W is normal if $D_W^\dagger D_w = D_W D_W^\dagger$

Equivalently: D_W admits an orthonormal basis of eigenvectors

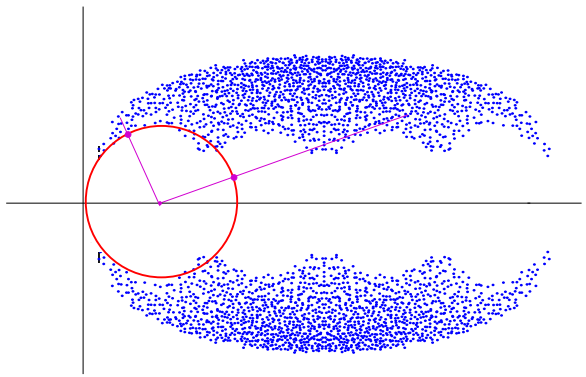
Proposition

Assume $D_W(0)$ is normal. Then

$$D_W(0)x = \lambda x$$

$$\Longleftrightarrow$$

$$D_N(m_k)x = \big(\rho + \mathrm{csign}(\lambda + m_k)\big)x$$



New approach: use multigrid solver for D_W

Definition: D_W is normal if $D_W^\dagger D_w = D_W D_W^\dagger$

Equivalently: D_W admits an orthonormal basis of eigenvectors

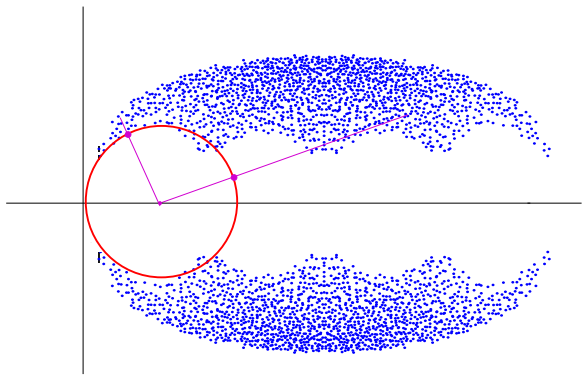
Proposition

Assume $D_W(0)$ is normal. Then

$$D_W(0)x = \lambda x$$

$$\Longleftrightarrow$$

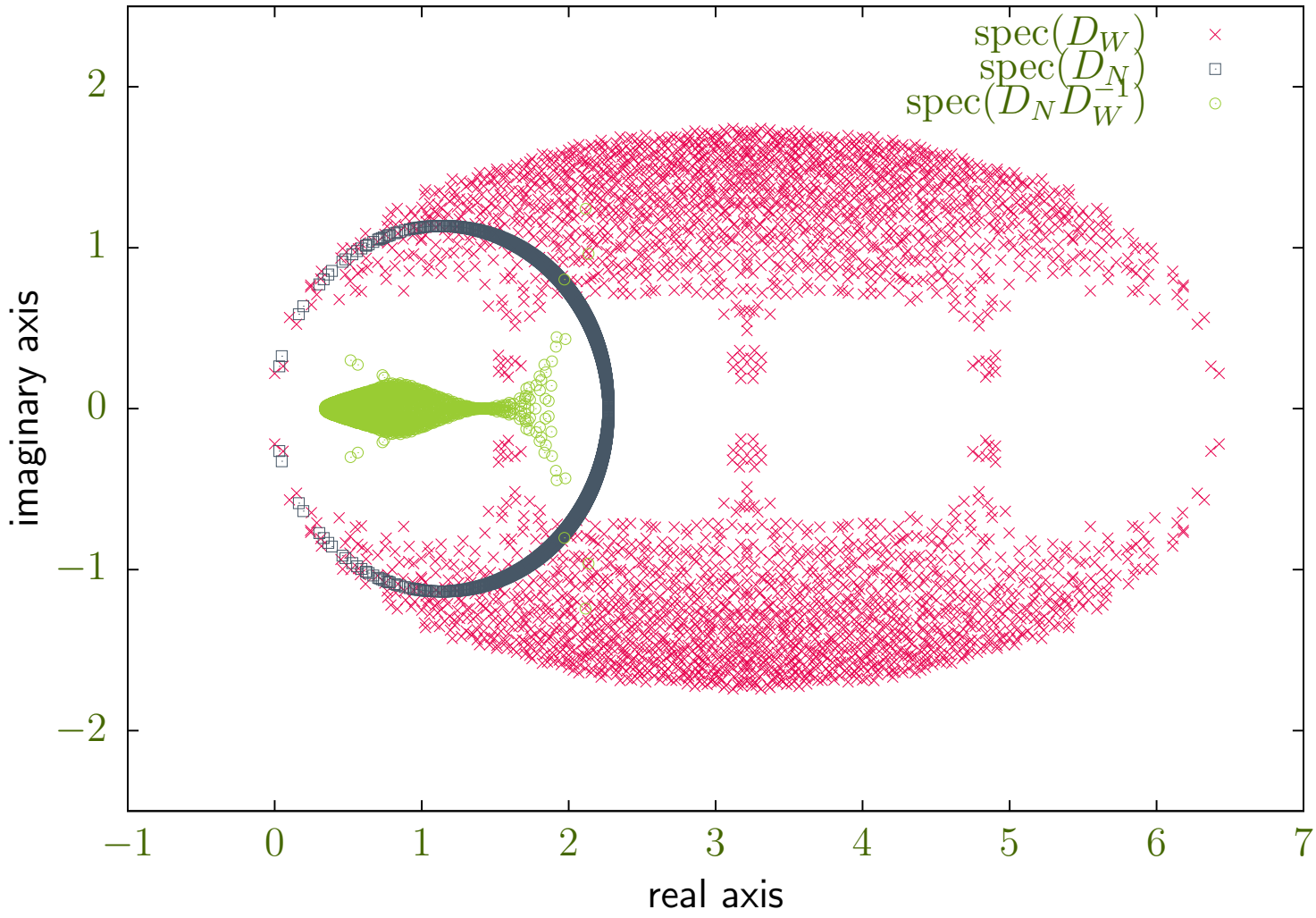
$$D_N(m_k)x = \big(\rho + \mathrm{csign}(\lambda + m_k)\big)x$$



- adapt α, m_0 s.t.small evs of $\alpha D_W(m_0)$ and D_N match.

Brannick, Frommer, Kahl, Leder, Rottmann, Strebel arXiv:1410.7170

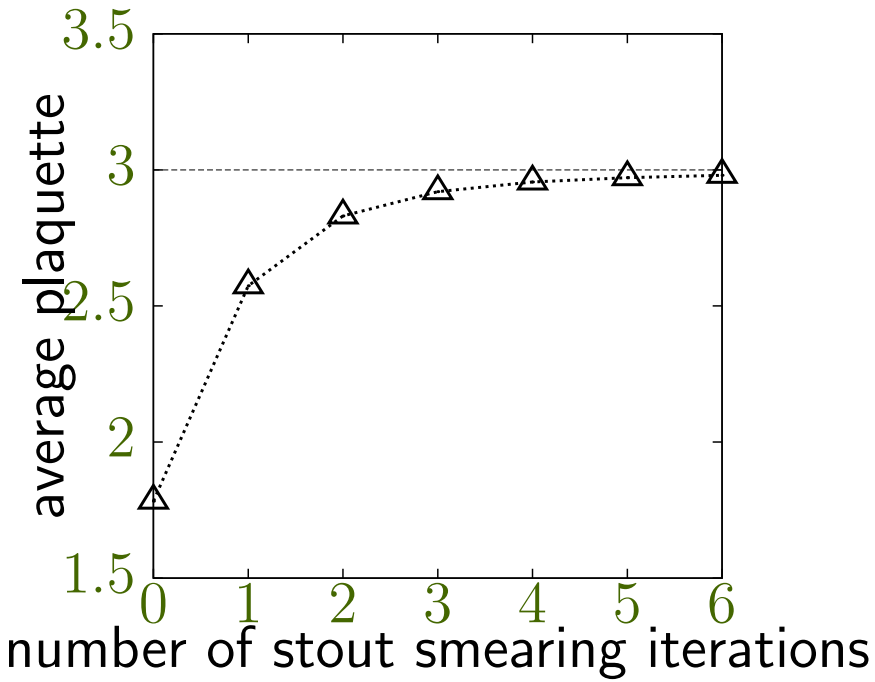
Spectra


spec(D_W)
spec(D_N)
spec($D_N D_W^{-1}$)
imaginary axis
real axis
2
1
0
−1
−2
−1
0
1
2
3
4
5
6
7

Smearing drives towards normality I

Fact 1: We have

$$\|D_W^\dagger D_W - D_W D_W^\dagger\|_F = 16 N_Q (3 - Q_{avg})$$



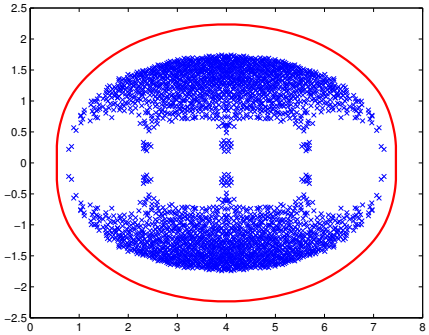
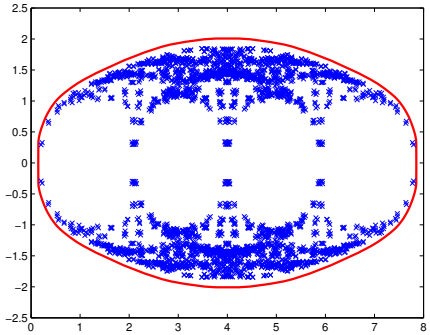
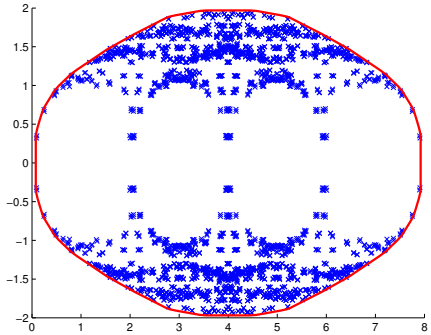
Smearing drives towards normality II

Fact 2

If D is normal, its field of values

$$\mathcal{F}(D) = \{\frac{\langle x, Dx\rangle}{\langle x, x\rangle}, x \neq= 0\}$$

is the convex hull of the spectrum.

Numerical results

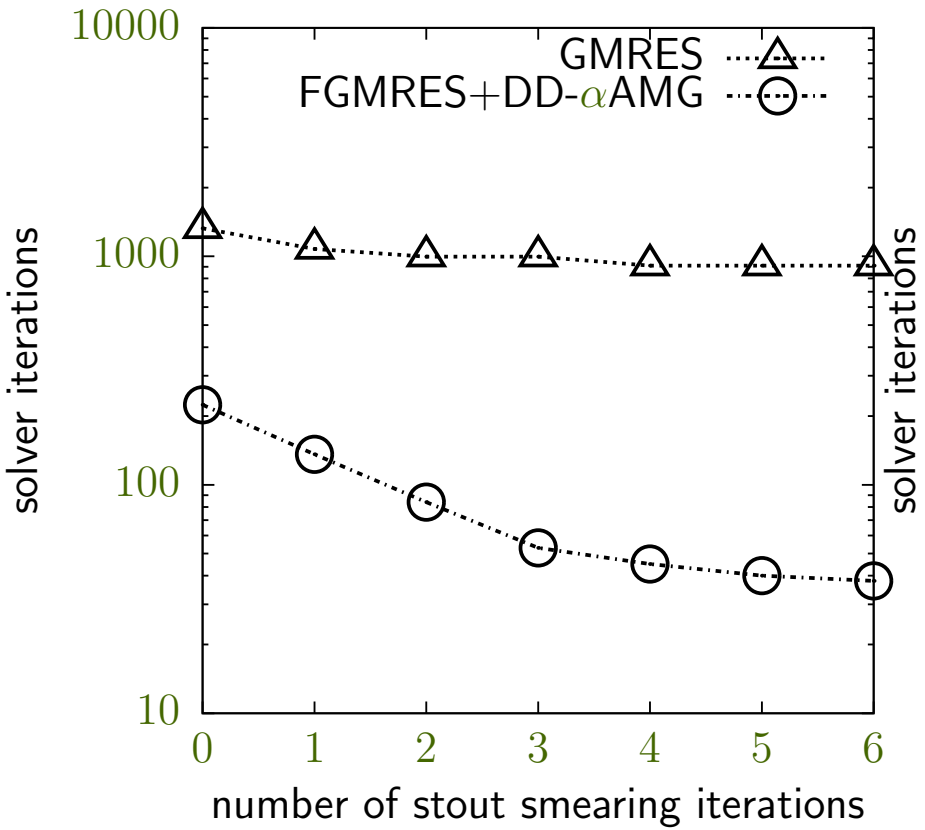
Configurations

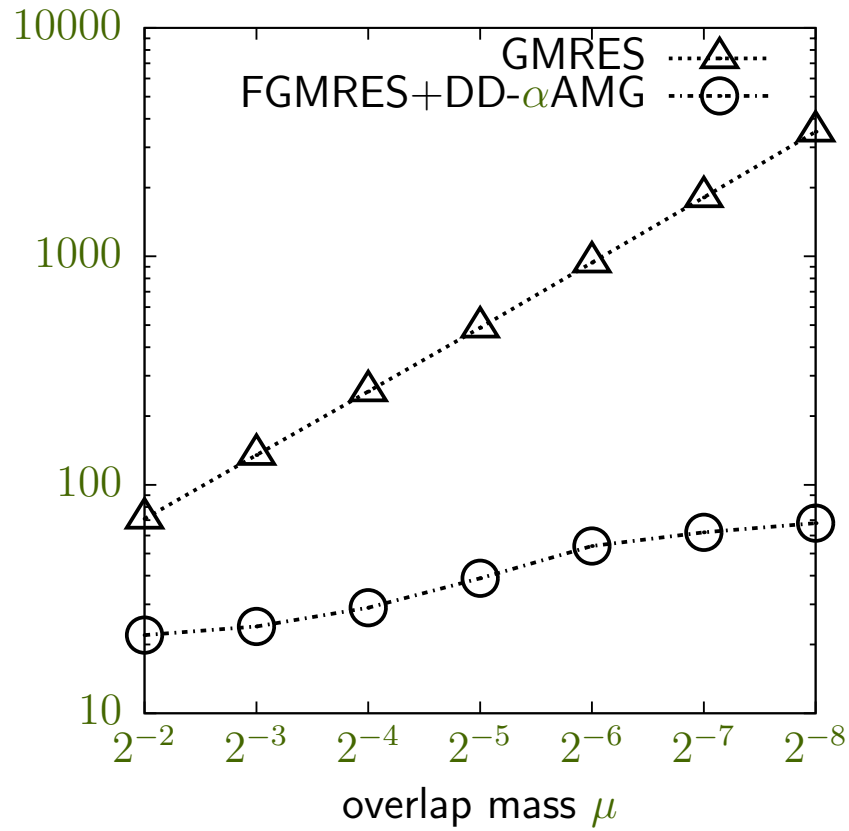
ID	lattice size $N_t \times N_s^3$	kernel mass m_0^{ker}	default overlap mass μ	smearing s	provided by
1	32×32^3	$-1 - \frac{3}{4}\sigma_{\min}$	0.0150000	$\{0, \ldots, 6\}$-stout	J. Finkenrath
2	32×32^3	-1.3	0.0135778	3HEX	BMW-c 2013

- We used 1024 processors of Juropa@FZ-Jülich
- $\rho = \frac{-\mu/2+m_0^{ker}}{\mu/2+m_0^{ker}}$

Comparison of iterations

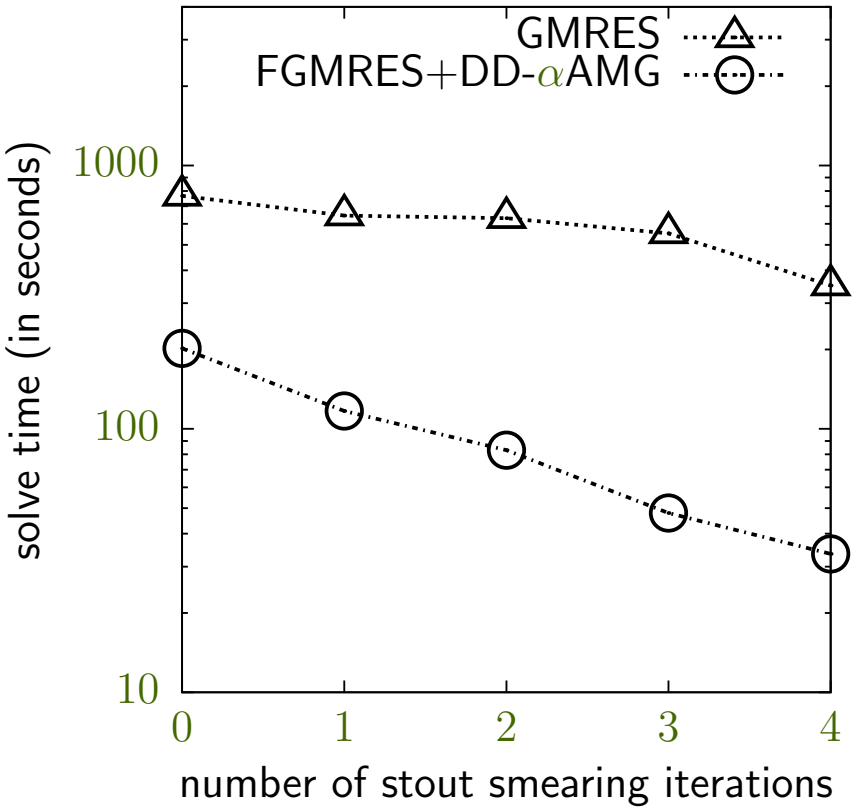
Configuration 1

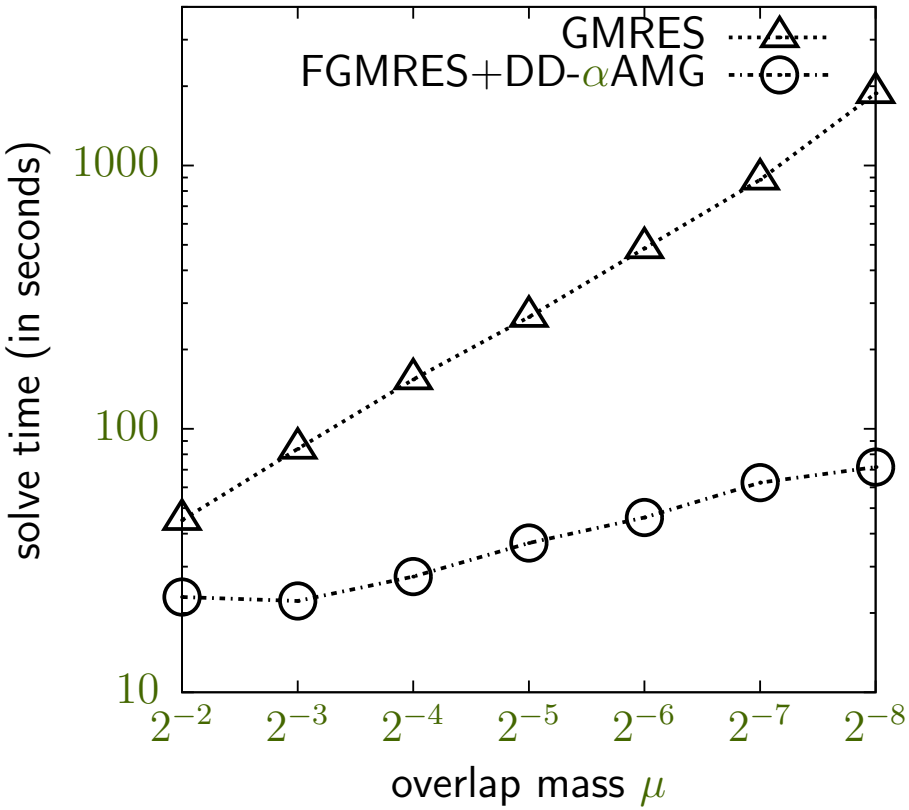




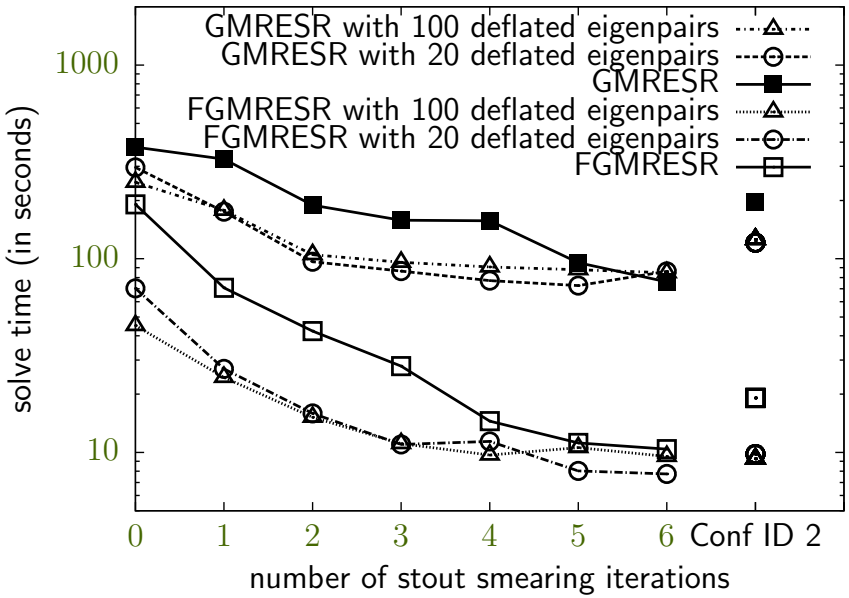
Comparison of time to solution

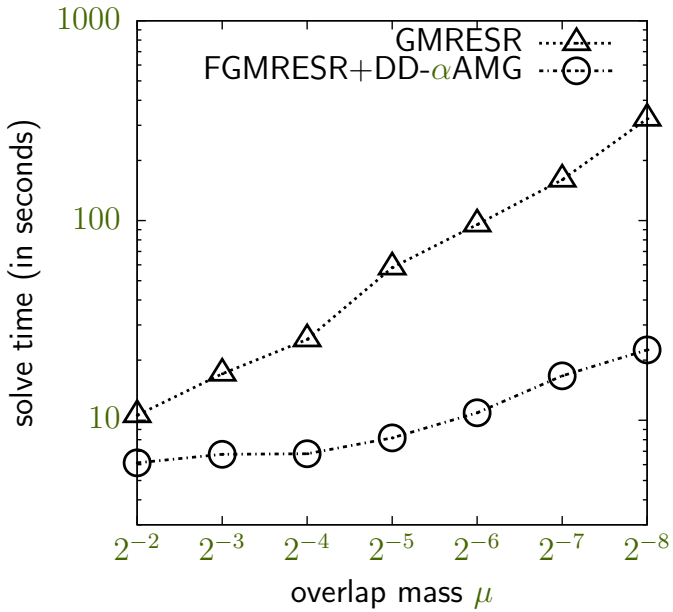
Configuration 1





Comparison of time to solution, play every trick





Conclusions

- Adaptivity is the key to success in AMG for LQCD
- Setup in AMG is expensive
- More levels require coarse grain parallelism
- AMG outperforms other solvers, especially for multiple sources
- AMG allows to use D_W as a preconditioner for D_N
- Performance gains increase as D_W gets more normal through smearing