Application of multigrid solvers in lattice QCD simulations at physical quark masses

02. October 2023

Seminar

RIKEN Kobe

Jacob Finkenrath

BERGISCHE UNIVERSITÄT WUPPERTAL

FOR 5269

Table of Content

1. Physics
- ❖ Short motivation

2. Hardware
- ❖ Setting the stage

3. Multigrid solvers
- ❖ Introduction
- ❖ Acceleration of the HMC
- ❖ Coarse-grid improvements

Wuppertal *2010 - 2015 PhD*
DDalphaAMG

Andreas Frommer | Matthias Rottmann | Karsten Kahl | Bjoern Leder | Francesco Knechli

A. Frommer et. al., SIAM J.Sci.Comput. 36 (2014)

A. Frommer et. al., arXiv:1307.6101

The Cyprus Institute *2015 - 2022*
Adaptation to twisted mass
Acceleration of HMC

Constantia Alexandrou | Simone Bacchio | Giannis Koutsou | Shuhei Yamamoto

C. Alexandrou et al., PRD 94 (2016) 11, 114509

C. Alexandrou et al., CPC. 236 (2019), 51-64

S. Yamamoto et al. , PoS LATTICE2021 (2022) 536

Wuppertal *2022 - now*
Coarse grid improvements
Multi-level trace estimation

Gustavo Ramírez | Jesus Espinoza-Valverde | Artur Strobel | Jose Jimenez | Henning Leemhuis

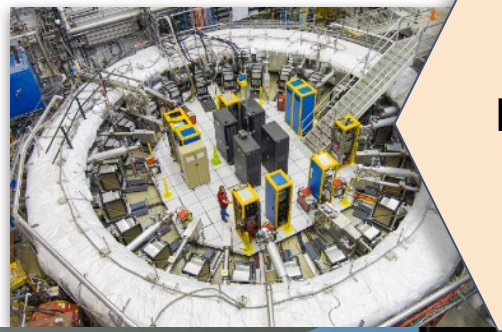
J. Espinoza-Valverde et al.,CPC 292 (2023) 108869

A. Frommer et al.,*SIAM J.Sci.Comput.* 44 (2022) 4, A2536-A2556

BERGISCHE UNIVERSITÄT WUPPERTAL

Understanding the building blocks of our world

Particle Collider: like LHC or Muon g-2

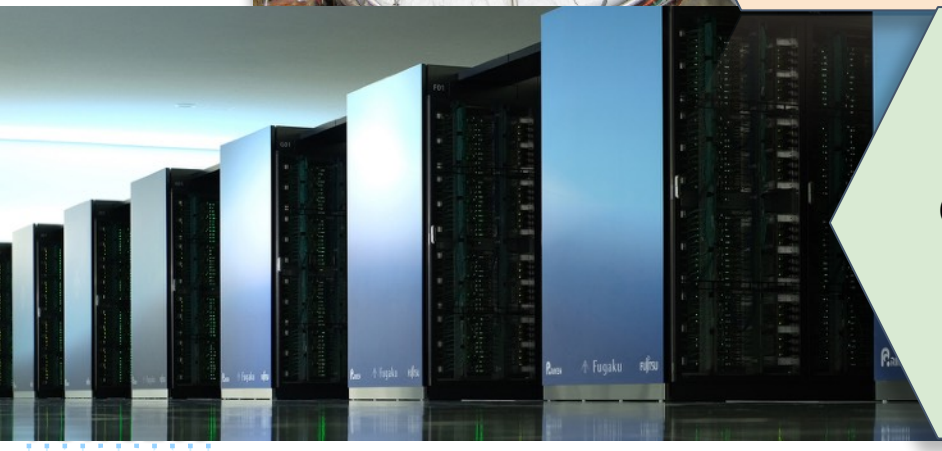


Experiment

Theory

Lagrangian of Standard model - QED, QCD, Weak Interactions - solvable via perturbation theory at high energy

$$\begin{aligned}\mathcal{L} = & -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} \\ & + i\bar{\Psi}\not{D}\psi \\ & + D_\mu\Phi^\dagger D^\mu\Phi - V(\Phi) \\ & + \bar{\Psi}_L\hat{Y}\Phi\Psi_R + h.c.\end{aligned}$$



Computation

At low energies QCD is non-perturbative

⇒ Lattice QCD only known ab initio method to predict

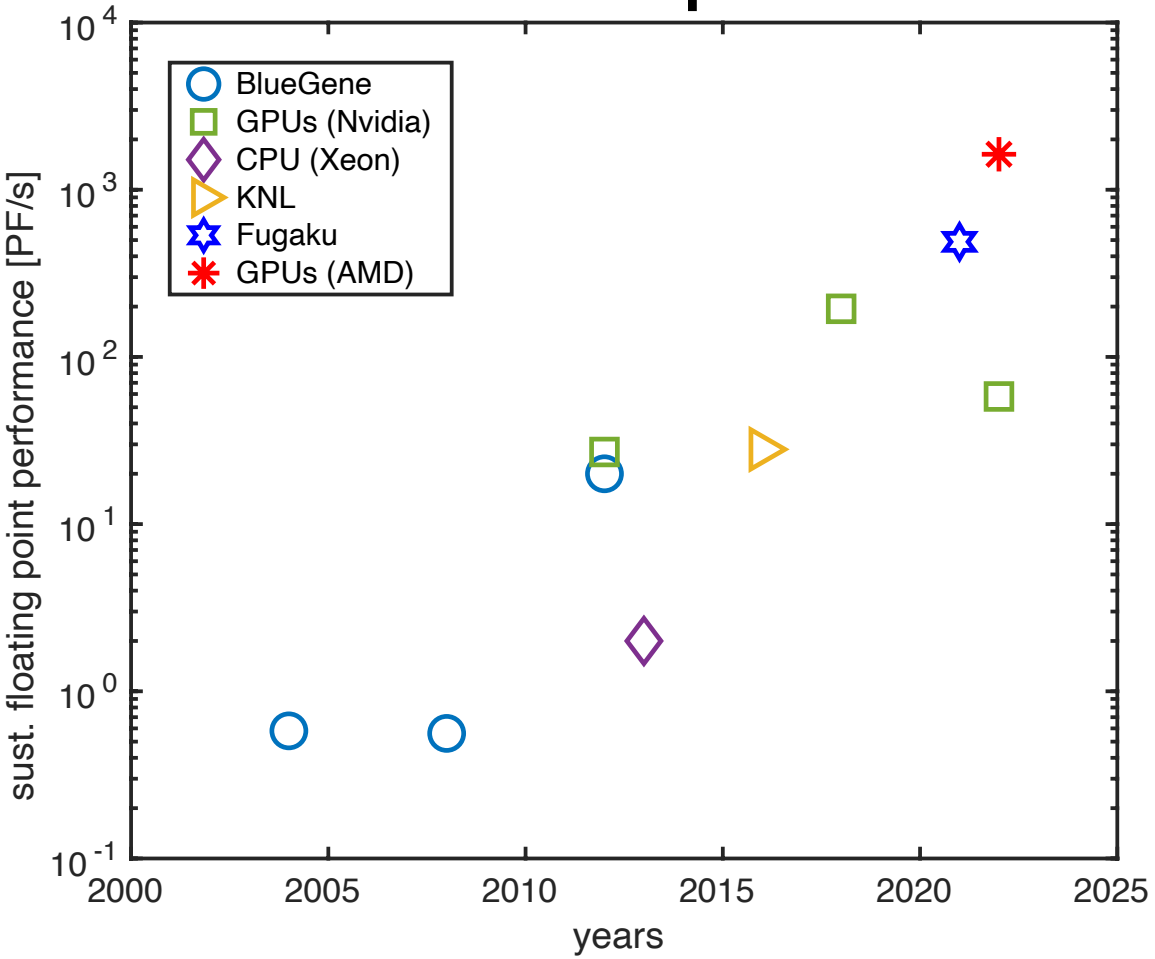
- Constituents of matter
- Zoo of baryonic states

Need to match experimental precision

- computational challenging

BERGISCHE UNIVERSITÄT WUPPERTAL

Advances of top machines



- Exascale is reached with Frontier: a lot of available compute time

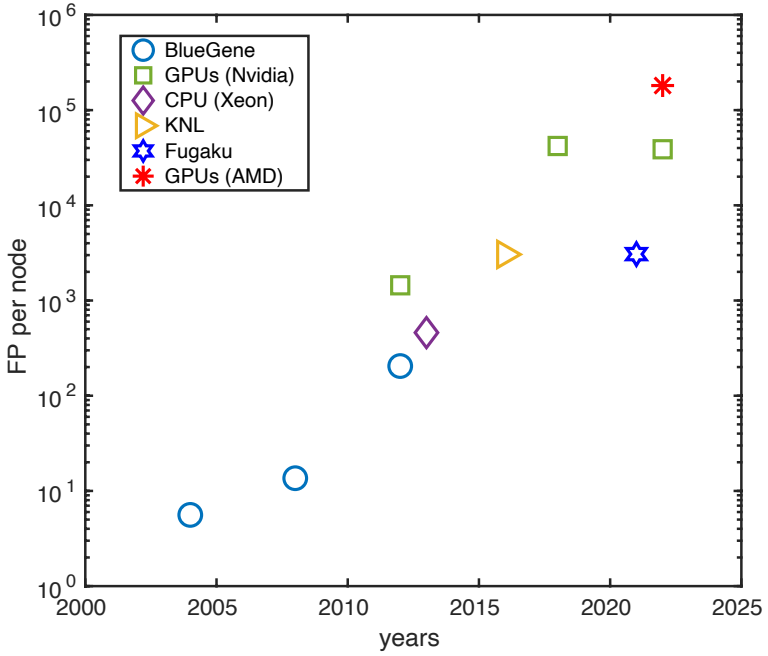
However, scaling up is non-trivial due to increase of complexity on the single but also multi-node

FOR 5269

Challenges

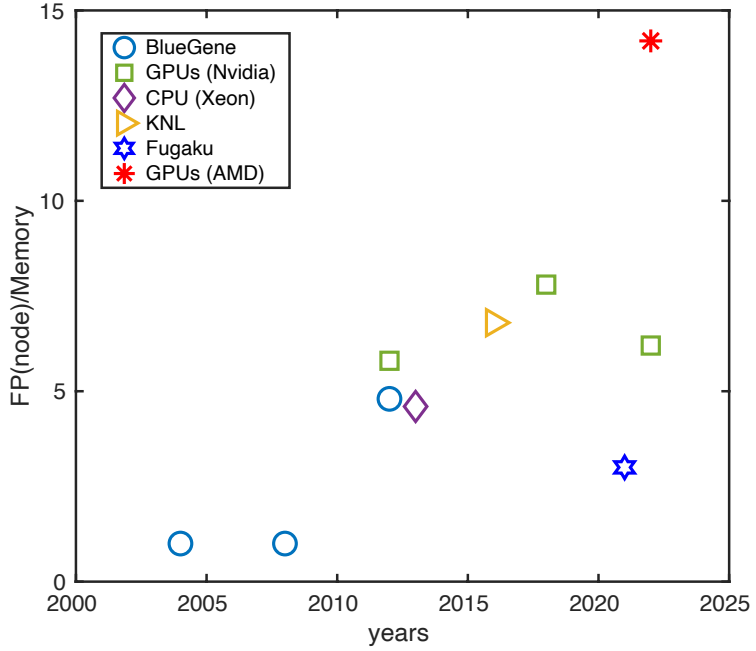
Best = 1 , worse >> 1

Single node peak performance



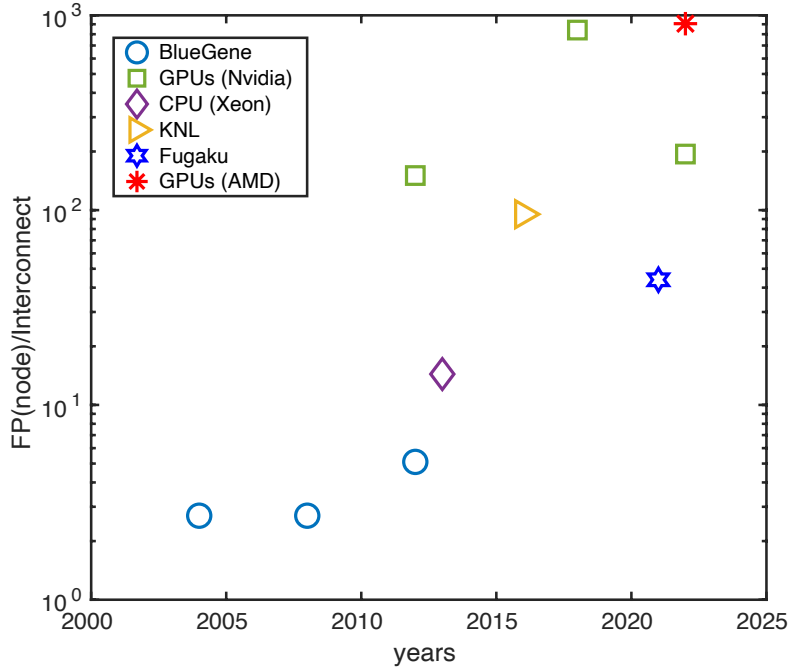
- Node level performance is increasing
- Single Nodes become a power house usually equipped with several accelerators (higher complexity to optimize)

Flops over bandwidth



- On node bandwidth does not grow as fast as FP
- Limiting factor for lattice QCD kernels
- Scalable algorithms are becoming more complex

Flops over interconnect



- Interconnect bandwidth is way behind FP node performance
- Extrem scaling is very likely limited by interconnect

P. Boyle, Lattice 2023, Plenary, Advances in algorithms for solvers and gauge generation

BERGISCHE UNIVERSITÄT WUPPERTAL

Multigrid solvers

From BlueGene to IntelXeon

- higher performance on single node

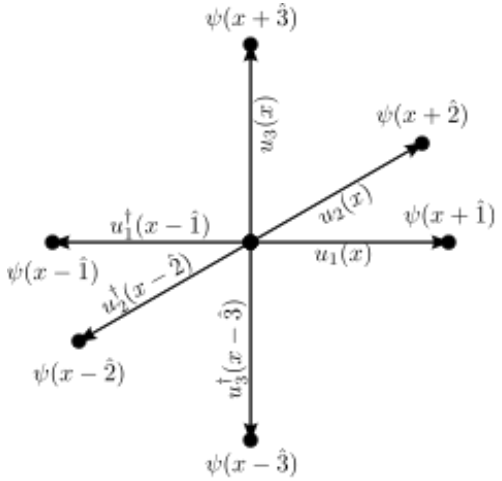
BERGISCHE UNIVERSITÄT WUPPERTAL

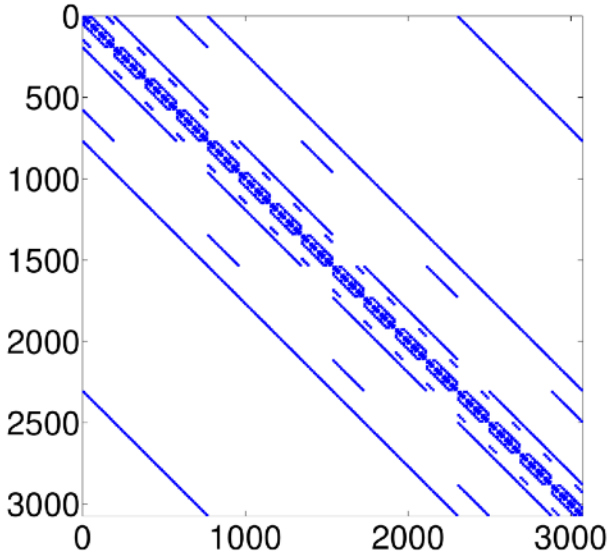
Linear equation

✤ Computational challenge solving:

$$Dx = b \qquad D \in \mathbb{C}^{n \times n} \text{ and } x, b \in \mathbb{C}^n$$

❖ 4-dim matrix stencil with $n \sim 10^9$





❖ matrix vector stencil with nearest neighbour connections
❖ Iterative linear solver approach requires matrix-vector product
❖ Matrix vector product bandwidth bounded:

Flops : Bytes = $1.3 : 1.1$

FOR 5269

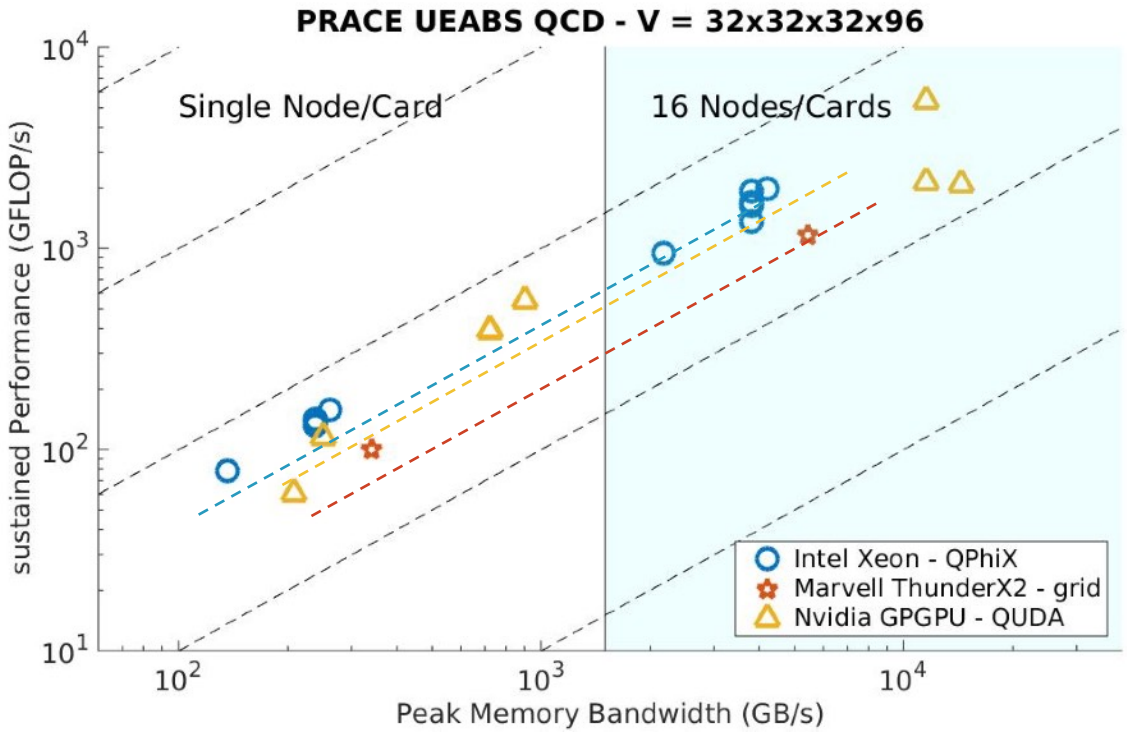
BERGISCHE UNIVERSITÄT WUPPERTAL

Performance of matrix vector product

UEABS Benchmark-kernels on PRACE Tier 0 systems (PRACE 4/5IP)

Application of lattice codes:
- ❖ QUDA for Nvidia GPUs
- ❖ QPhiX for Intel CPUs
- ❖ Grid for Arm

Computation cost dominated by matrix vector application



- ❖ Computational performance is bounded by available bandwidth
- ❖ Towards 16 nodes, Nvidia GPU deviates from perfect scaling

FOR 5269

Krylov Subspace Methods

Iterative solver: Krylov space

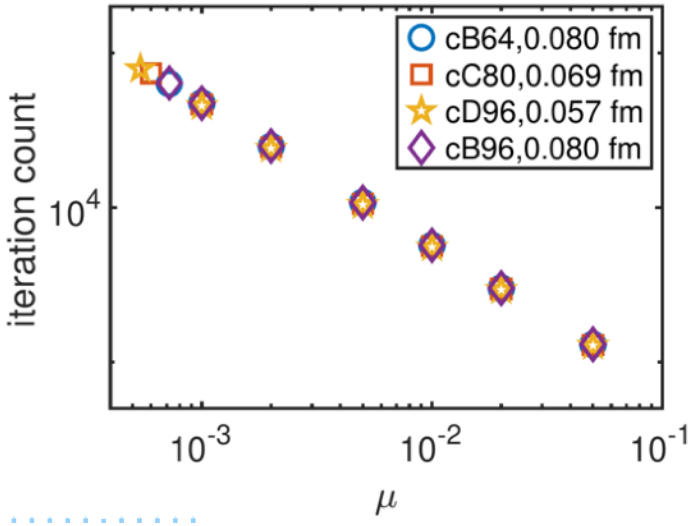
$$x \in \mathscr{K}_n(D,b) = \{b, Db, D^2b, \ldots, D^{n-1}b\}$$

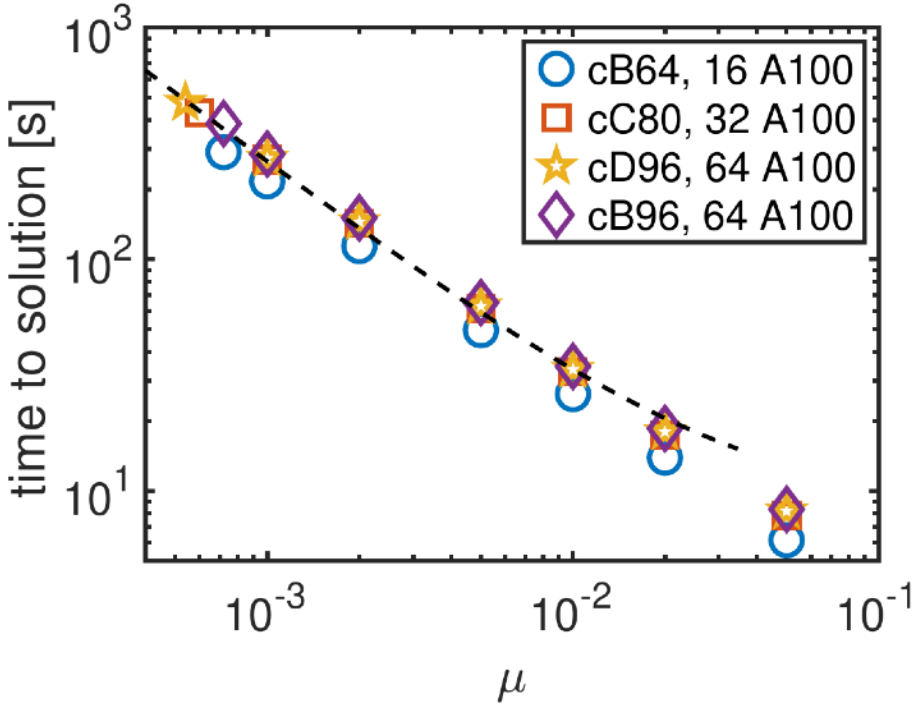
And solve subsystem

Cost for Conjugate gradient solver:

$$cost \approx V \cdot \left(\frac{b}{\mu} + a\right) \approx V\frac{b}{\mu}$$

with $b/a \sim 0.04$





QUDA mixCG: (Juwels Booster)

- 64^3x128: 16 A100 GPUs
- 80^3x160: 32 A100 GPUs
- 96^3x196: 64 A100 GPUs

Iteration proportional to condition number

- ❖ only depend on smallest mode
- ❖ iteration count increases drastically at physical point

Computational kernel

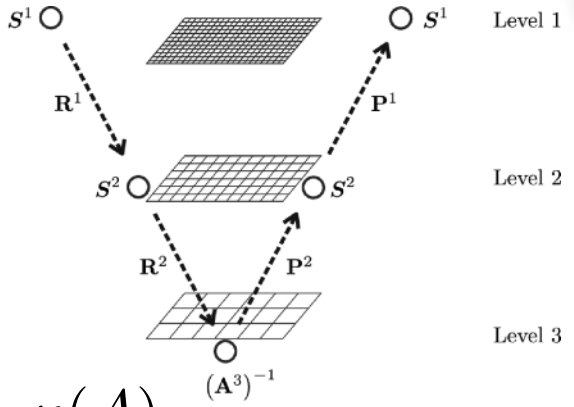
- ❖ Simple, mainly $D \cdot x_i$

Increase convergence by preconditioning

Idea: Preconditioning IR and UV modes

❖ Error can be split:

$$\epsilon = \epsilon_{low} + \epsilon_{high}$$



❖ A good smoother eliminates ϵ_{high} high effectively

❖ Multigrid solver can lead to convergence independent of $\kappa(A)$

A good smoother: apply block inverse in alternating procedure

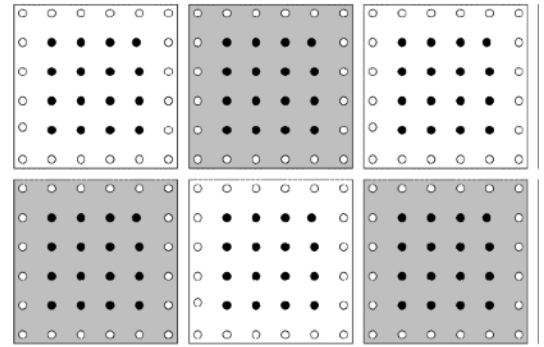
Precondition iterate in Krylov space via:

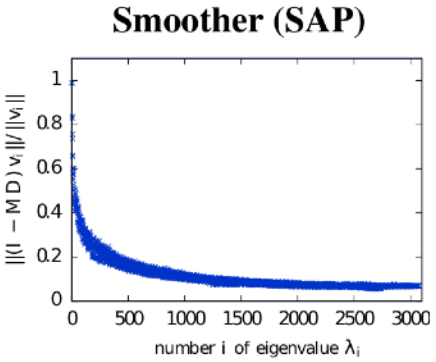
$$x_i \rightarrow (1 - KD)x_i + Kb$$

With $K = D_w^{-1} + D_b^{-1} - D_b^{-1} D_{bw} D_w^{-1}$

And error:

$$\epsilon_{SAP} = 1 - KD = (1 - D_b^{-1} D)(1 - D_w^{-1} D)$$





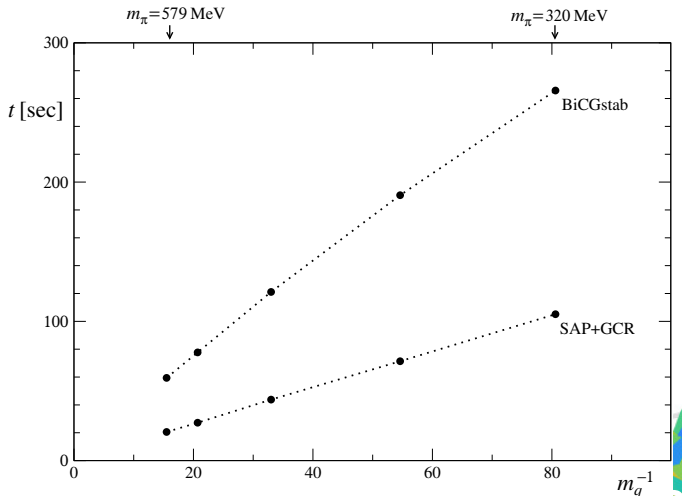
Assign at least two block per node (or core)

❖ no (inter-node) communication is needed while inverting blocks

Smoother

❖ simple to implement

❖ acts on the large eigenvalues



M. Luscher, Comput.Phys.Commun. 156 (2004) 209-220

BERGISCHE UNIVERSITÄT WUPPERTAL

Algebraic multigrid: non-geometric

Two-level AMG

- Petrov-Galerkin projection

$$D_c = RDP$$

- Coarse-grid correction:

$$\psi \leftarrow \psi + PD_c^{-1}Rr$$

with $r = \eta - D\psi$

- Error propagation

$$\epsilon = I - PD_c^{-1}RD$$

- Capture sufficient low modes within P

Algorithm Two-level V-cycle

Input : ψ, η, ν

Output : ψ

$r \leftarrow \eta - D\psi$

$\psi \leftarrow \psi + PD_c^{-1}Rr$

$r \leftarrow \eta - D\psi$

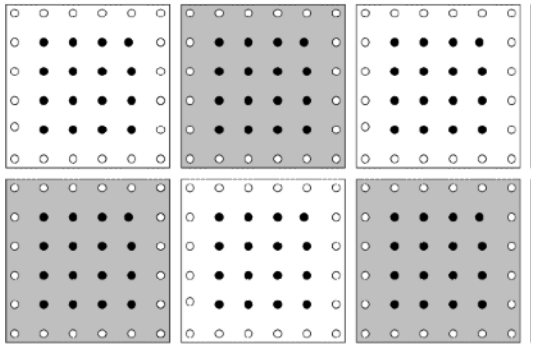
$\psi \leftarrow \psi + M_{SAP}^{(\nu)} r$

- Extension to Multilevel AMG

Extension is possible: apply decomposition to the second level

FOR 5269

Aggregation



Aggregation blocks:

$$\mathcal{A}_i = \mathcal{L}_{j(i)} \times \mathcal{W}_i$$

❖ standard aggregation

$$\{\mathcal{A}_{i,\tau} : j = 1, \ldots, n_{\mathcal{L}_c}, \tau = 0, 1\}$$

Where

$$\mathcal{A}_{i,0} = \mathcal{L}_j \times \{0, 1\} \times \mathcal{C}$$
$$\mathcal{A}_{i,1} = \mathcal{L}_j \times \{2, 3\} \times \mathcal{C}$$

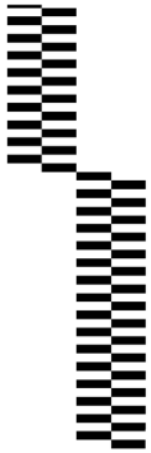
❖ this implies $\Gamma_5 P = P\Gamma_5^c$

❖ use similar restriction like projection $R = P^\dagger$

$$D_c = P^\dagger D P$$

$$(v_1 \mid \ldots \mid v_N) = \longrightarrow P = \begin{matrix} \mathcal{A}_1 \\ \mathcal{A}_2 \\ \vdots \\ \mathcal{A}_s \end{matrix}$$

Test vector Aggregation:



Building up the coarse grid space

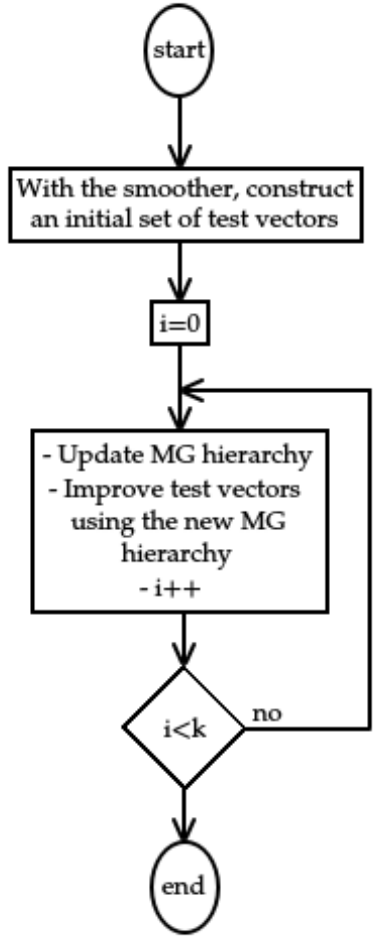
Test vectors: use approximate eigenvectors in the construction of P
An ideal approach: Take eigenspace of small eigenvalues
❖ done in setup phase

Idea: use multigrid hierarchy to approximated D^{-1}

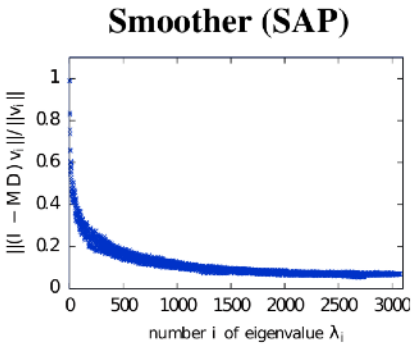
❖ Update multigrid hierarchy as the test vectors improve

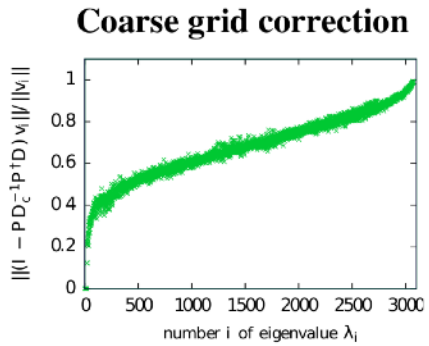
$$v_{i+1} \leftarrow \text{"}D^{-1}\text{"} v_i$$

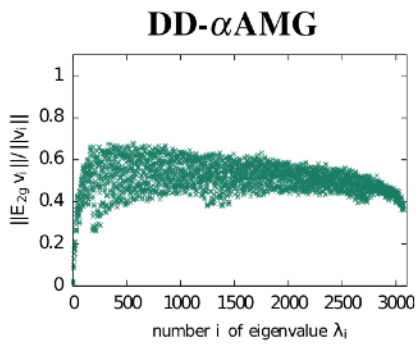
M. Luscher, JHEP 12 (2007) 011, JHEP 07 (2007) 081

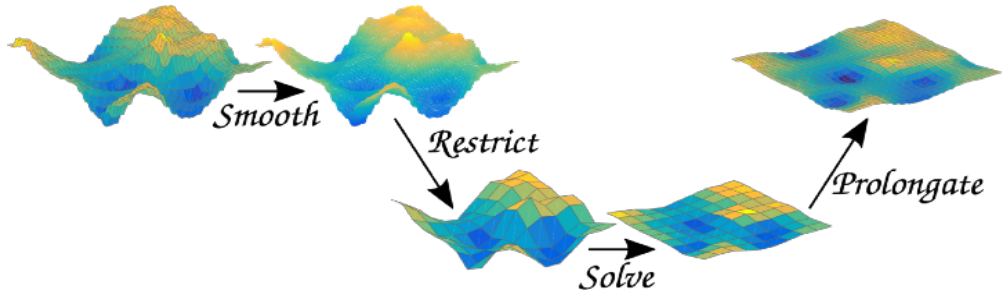


DDalphaAMG









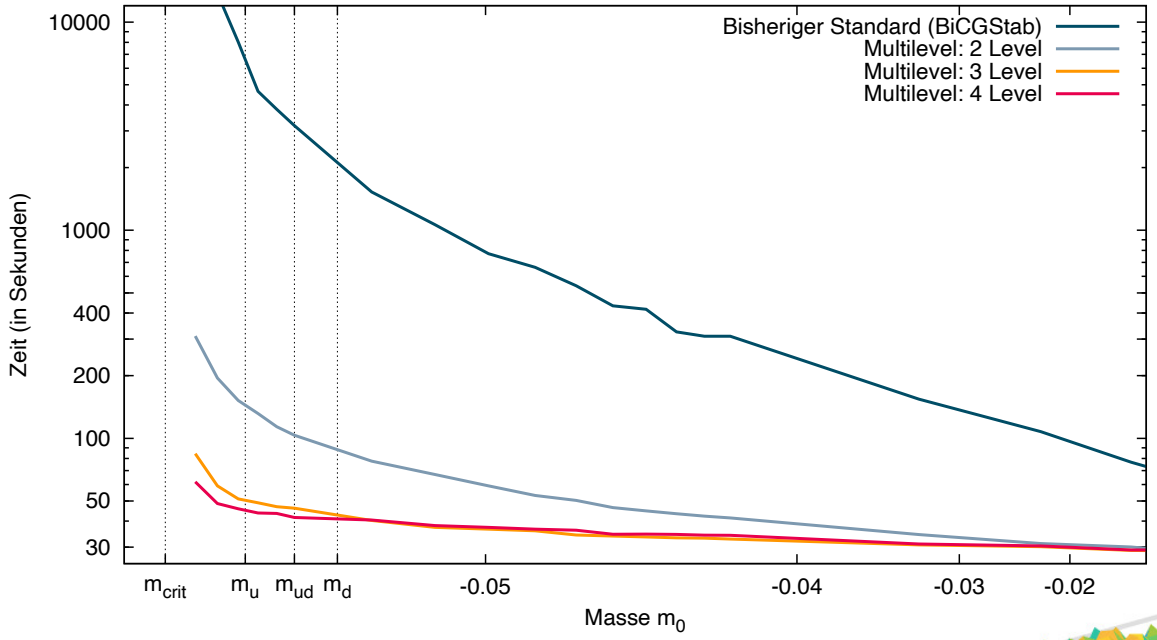
Multigrid solver is almost independent of condition number

DDalphaAMG
Domain Decomposed adaptive algebraic Multigrid solver

- Up to two magnitude faster at the physical point compare to BiCGstab or CG
- Comes with an additional overhead, coarse grid projection has to be build

A. Frommer et al., SIAM J.Sci.Comput. 36 (2014)

A. Frommer et. al., arXiv:1307.6101



Acceleration of HMC

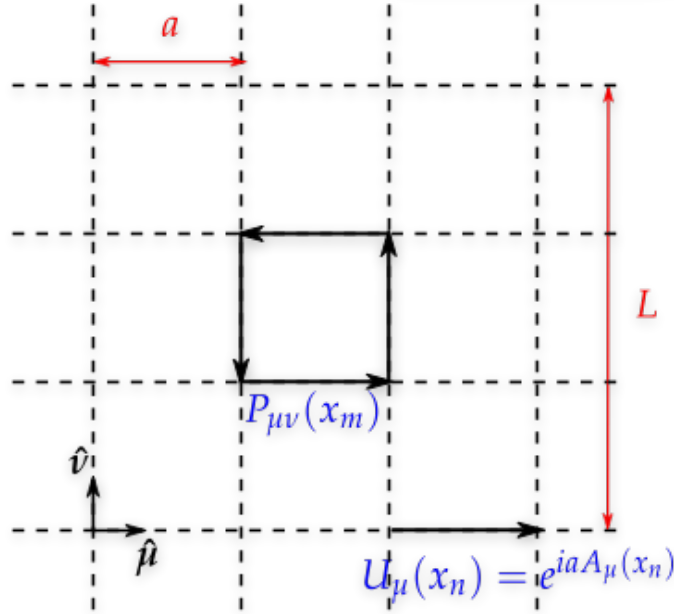
From IntelXeon to GPUs

BERGISCHE UNIVERSITÄT WUPPERTAL

Evaluation of observables via the path integral:

$$\langle \mathcal{O} \rangle = Z^{-1} \int D[U,P,\phi]\, \mathcal{O}(U) \cdot e^{-H(U,P,\phi)}$$

a high dimensional integral of up to
5 500 000 000 degrees of freedom in U



1. Hybrid Monte Carlo (MCMC)

Importance sampling via Boltzmann weight by evolving (U,P) via Hamilton's Equations:

$$\dot{P} = -\frac{\partial H}{\partial U} \quad \text{and} \quad \dot{U} = \frac{\partial H}{\partial P}$$

Force calculation
> 10 000 per Trajectory

Inversion of Dirac operator

$$D(U) \cdot x = b$$

2. Observable calculation

Primary observables are calculated on a set of configurations:

$$\langle \mathcal{O} \rangle = \frac{1}{N} \sum_i^N \mathcal{O}(U) + O\left(\sqrt{2\tau_{int}/N}\right)$$

Fermion propagators
> 1 Mi per measurement

FOR 5269

Work with Simone Bacchio

BERGISCHE UNIVERSITÄT WUPPERTAL

Hybrid Monte Carlo with IR-UV preconditioning

Target: sample configuration distributed with the Boltzmann weight

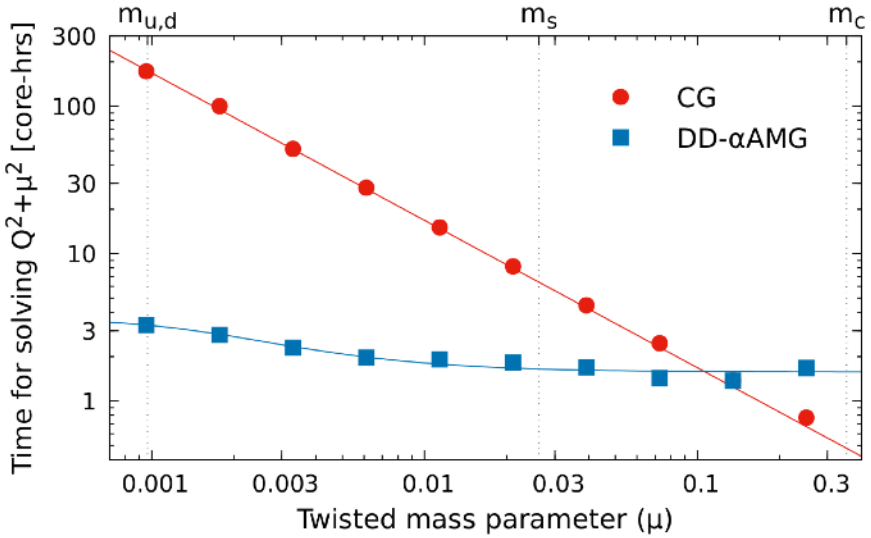
$$P(U) \propto \det[D^\dagger D(U)] \exp\{-\beta S_g(U)\}$$

Idea: use IR+UV filtering via:

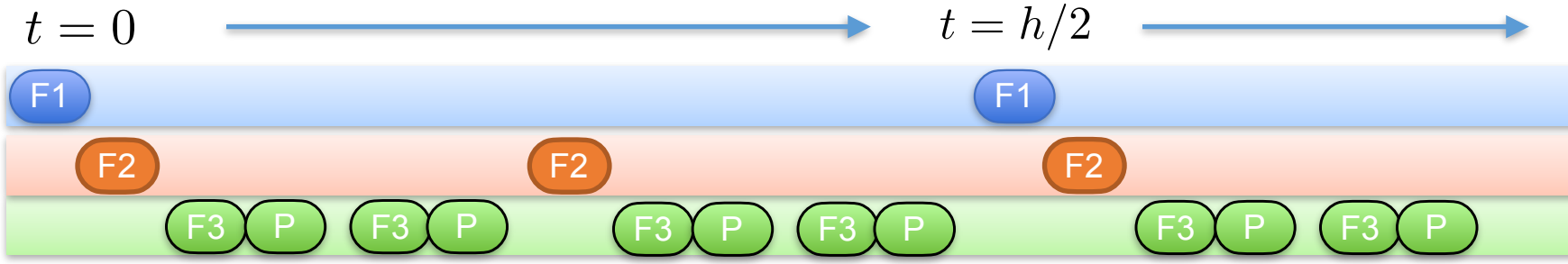
- ❖ Hasenbusch mass - preconditioning
- ❖ Nested integrators

Re-write the determinants:

$$\det[D^\dagger D] = \det\left[\frac{D^\dagger D}{D^\dagger D + \mu^2}\right] \cdot \det[D^\dagger D + \mu^2]$$



Nested leap-frog:



Outer level: $F_1 = -\frac{\partial}{\partial U}\eta^\dagger[1 + \mu^2(D^\dagger D)^{-1}]\eta$

Middle level: $F_2 = -\frac{\partial}{\partial U}\eta^\dagger(D^\dagger D + \mu^2)^{-1}\eta$

Finest level: $F_3 = -\frac{\partial}{\partial U}\beta S_g(U)$

Fermion force requires inversions

- ❖ integrate all terms with mu<0.01 with MG

FOR 5269

Enabling MG solver in Molecular Dynamics

- Requires single solves:

$$(DD^\dagger + \mu^2)^{-1} = \gamma_5 (D - i\gamma_5\mu)^{-1} \gamma_5 (D + i\gamma_5\mu)^{-1}$$

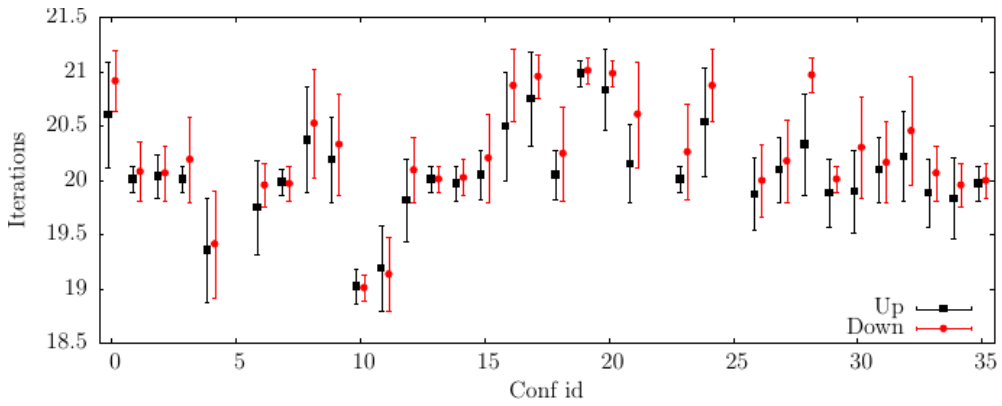
Cost efficient if one can

- re-use setup phase, adequate update procedure during trajectory
- re-use coarse levels for all operators
  - for different flavors, up, down, strange and charm quarks
  - with different shifts

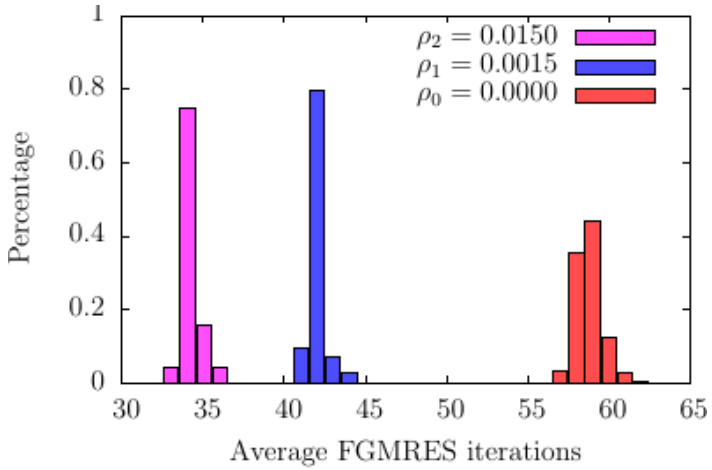
by ensuring stability of method during molecular dynamics

Outcome of DDalphaAMG adaptation in HMC:

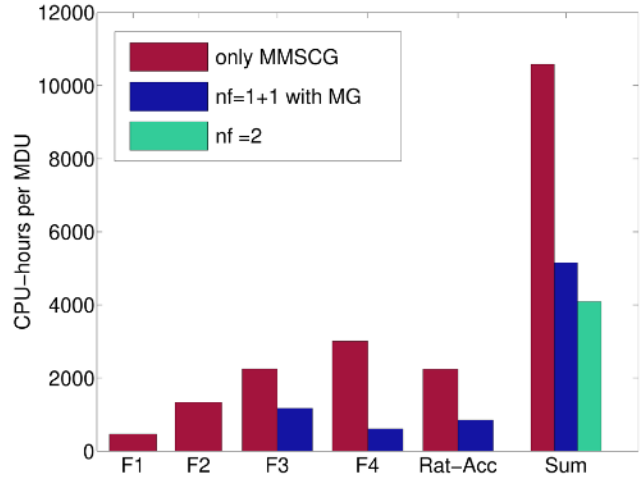
- very stable
- need one update iteration for setup every light quark force
  (20 - 50 updates during 1 MDU)



Stable under Sign-flips



Stable under shifts



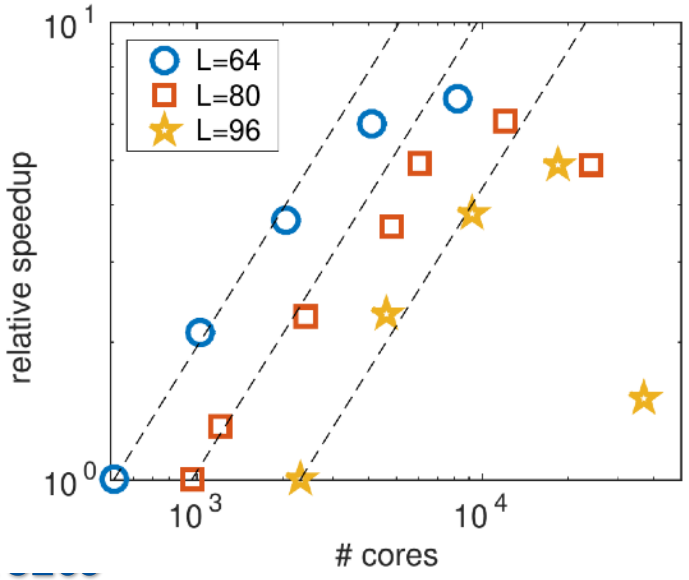
Acceleration in Rational approximation

Multigrid solver in HMC

Scalability:

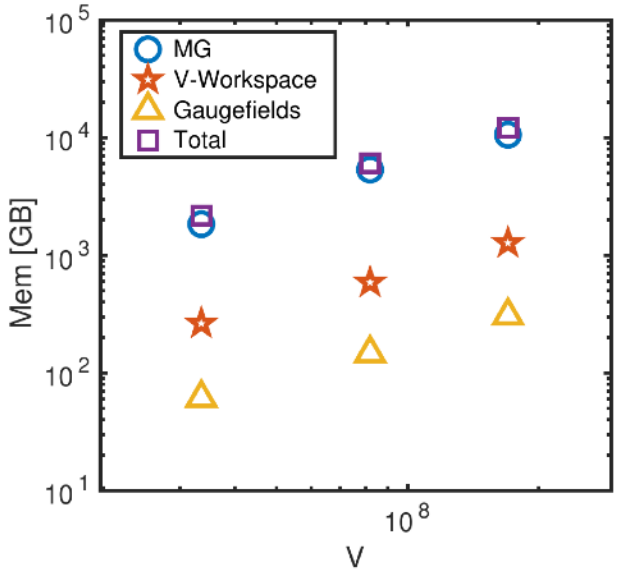
- Limited by the size of the coarsest grid
- on SuperMUC-NG window scales with

$$\propto V^{\frac{3}{4}}$$



Memory:

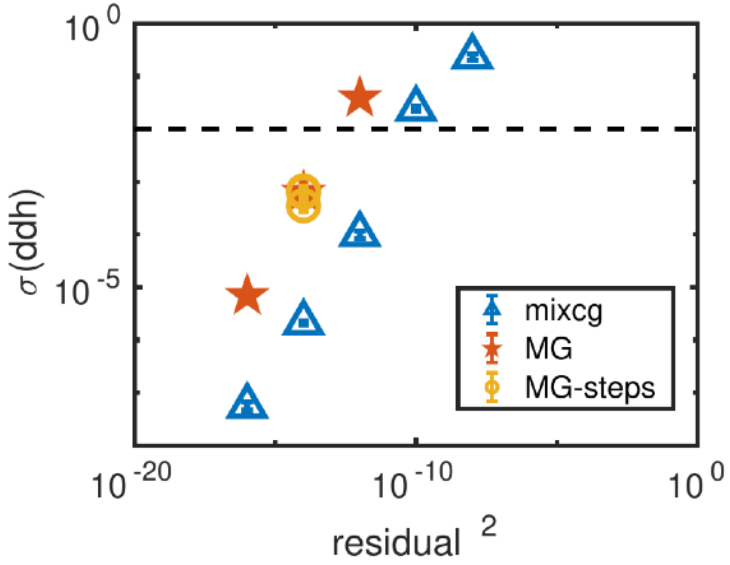
- multigrid increases memory requirements 5x
- hard limit for minimal parallelization :
  - L=64 ~ 16 Nodes
  - L=96 ~ 80 Nodes



Reversibility:

- reused of multigrid subspace can impact reversibility
- using a higher solver precision for force computation:

factor 10 higher precision for MG



ETMC Simulation effort on SuperMUC

Pioneering Multigrid solver
(work with S. Bacchio)

- Nf =2 physical point Simulations
  - two ensembles at larger box sizes

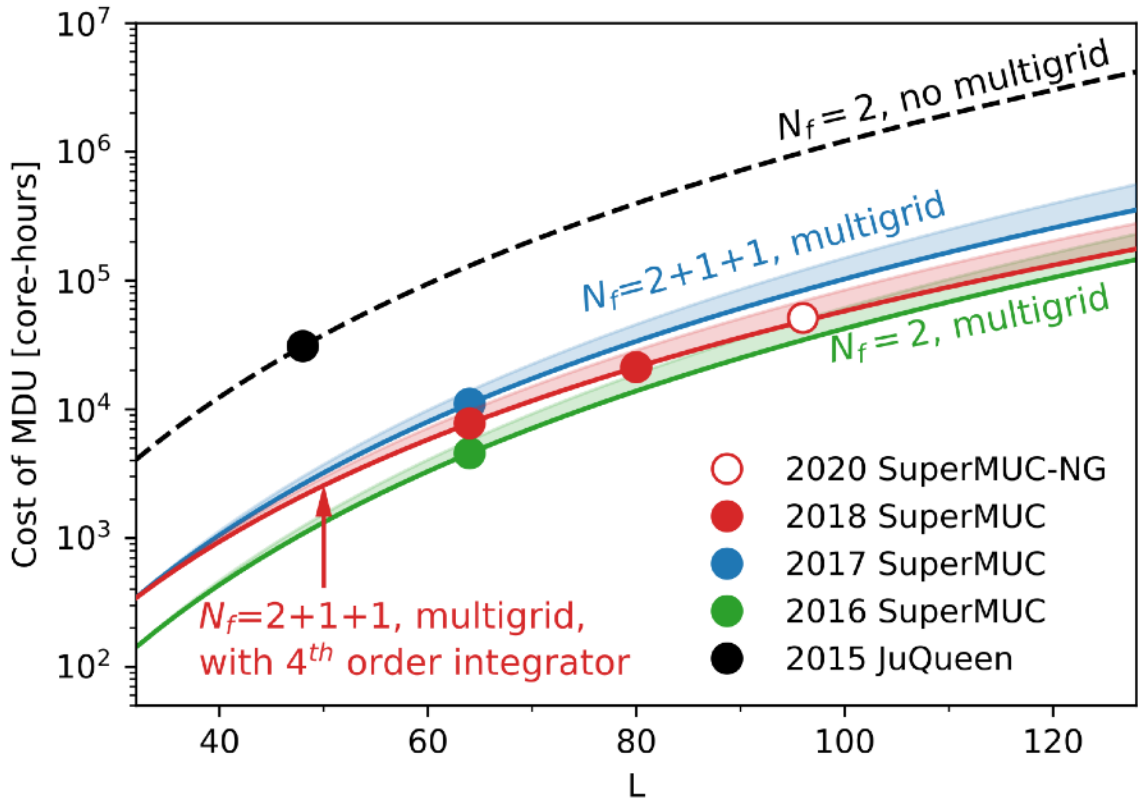
C. Alexandrou et al., PRD 94 (2016) 11, 114509

- Inclusion of heavy quark doublet
  - ongoing: four ensembles at different lattice spacings

C. Alexandrou et al., CPC. 236 (2019), 51-64

- Higher order integrators
  - fourth order integrator

D. Shcherbakov, CCP 21 (2017) 4, 1141-1153



However, for exa-scale era
- adaptation to GPU machines are (likely) necessary

Adaptation to GPUs

Work with

Bartosz Kostrzewa | Marco Garofalo | Simone Romiti | Simone Bacchio | Ferenc Pittler

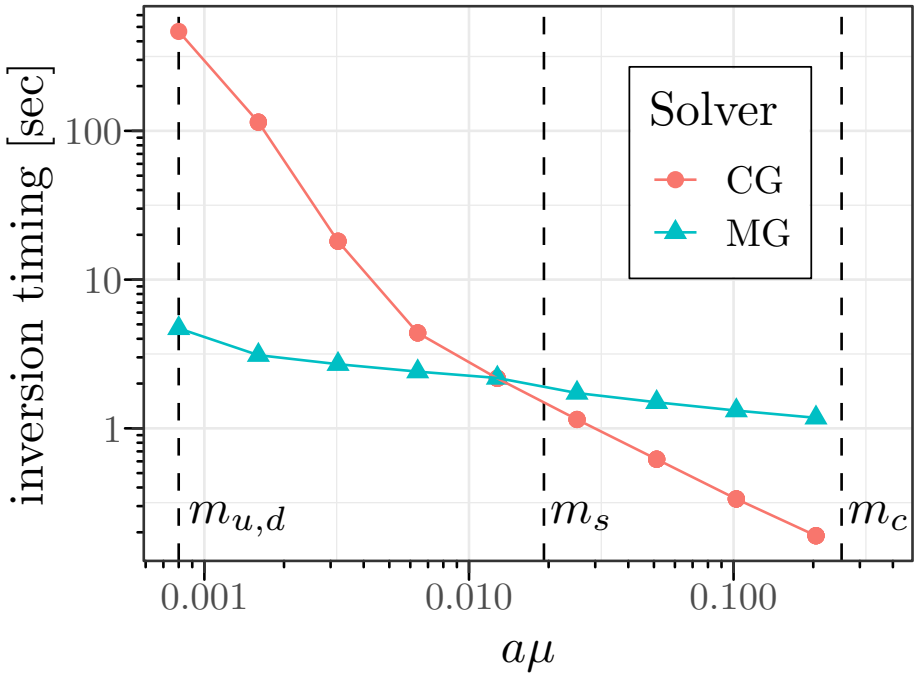
BERGISCHE UNIVERSITÄT WUPPERTAL

Idea: Use tmLQCDs linkage to QUDA

- QUDA comes with highly optimized solvers for GPUs (includes now also AMD and Intel kernels)

Similar to MG-adaptation

- outsource computation intensive parts to the GPU
  - Start with Inversions
    - enable multigrid solver
    - enable multi-mass-shift CG
  - (on-going) Extend to force calculation
    - Pure gauge part available
    - Currently fermion force is under work



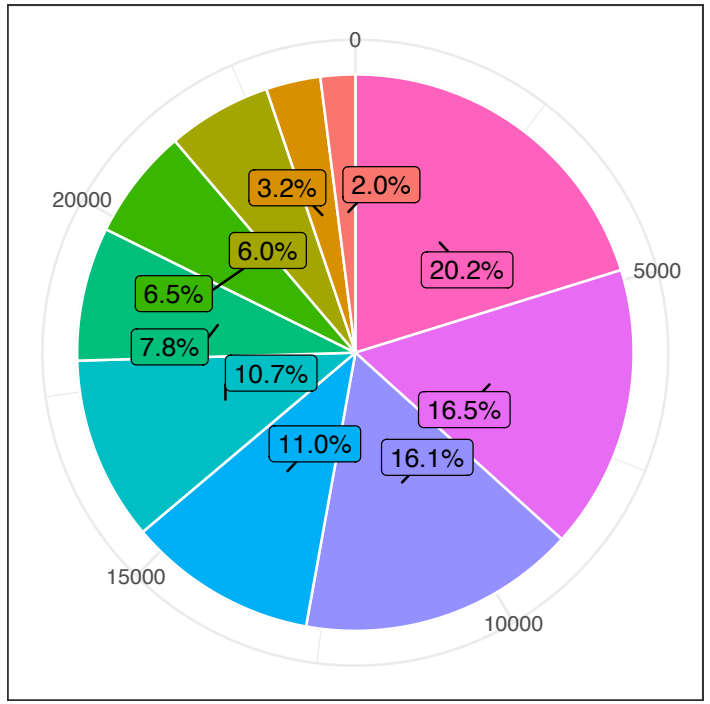
- MG on GPU is even further limited (due to memory and scalability)
- CG solver more efficient due to utilisation of mix-precision

FOR 5269

GPU utilisation

B. Kostrzewą, arXiv:2212.06635

Breakdown of time spend in monomials
Marconi 100 (16 Nodes with 4 V100)



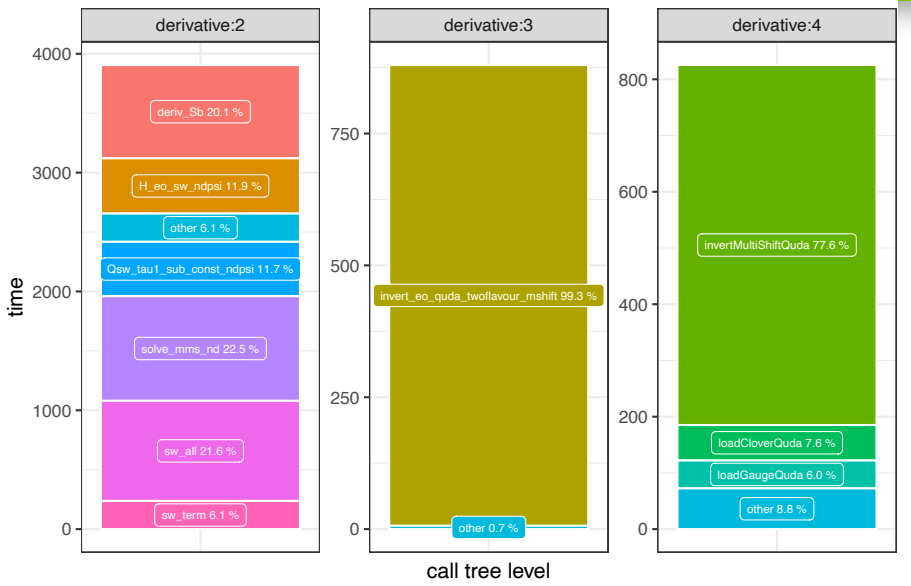
reorder(monomial, prop)

- a GAUGE
- a ndcloverratcor
- a other and unaccounted for
- a cloverdetratio1light
- a cloverdetlight
- a ndcloverrat2
- a ndcloverrat3
- a cloverdetratio3light
- a ndcloverrat1
- a cloverdetratio2light

~ 36% associated with MG solver
Other parts better scalable

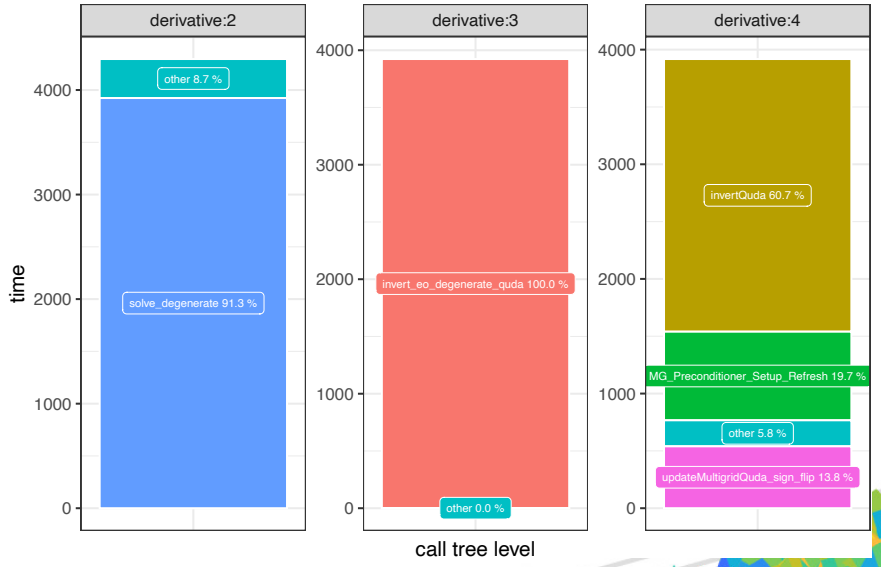
❖ Example of CPU dominant

ndcloverrat1 derivative



❖ Example of GPU dominant

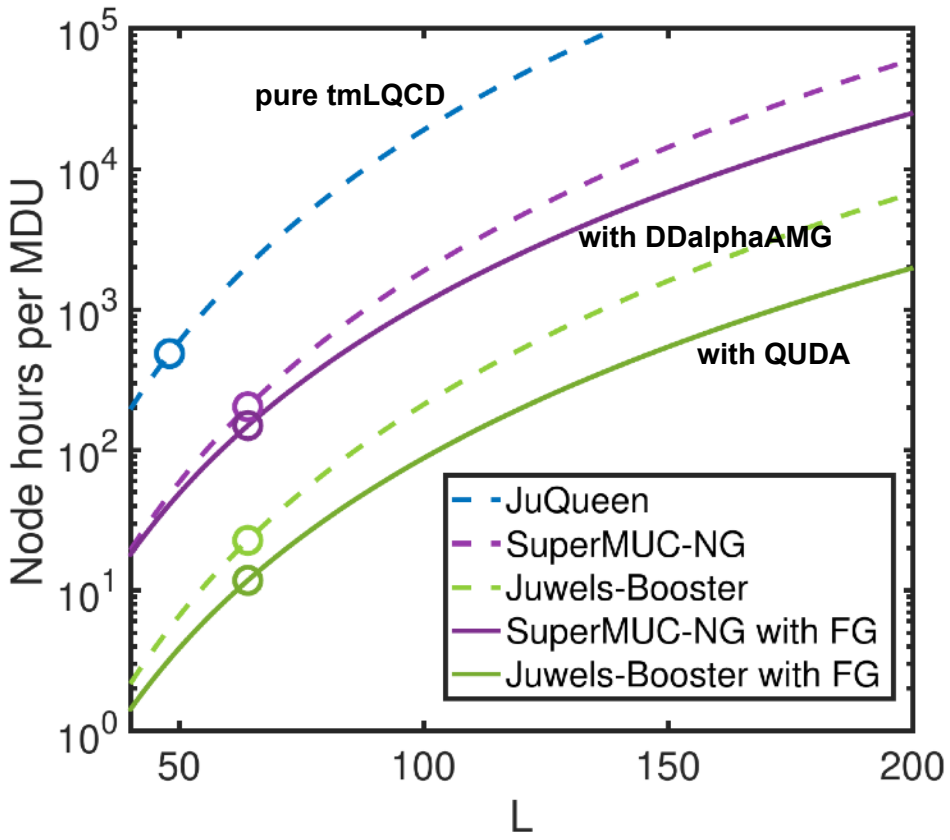
cloverdetratio2light derivative



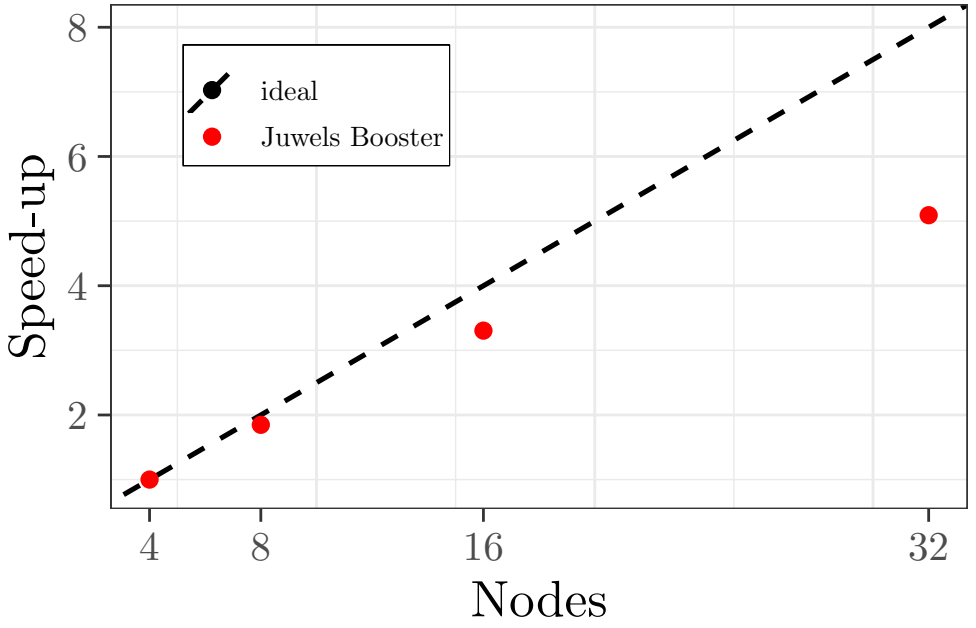
HMC on GPUs

Cost for HMC:
by 10x with MG, by 10x with GPUs

B. Kostrzewą et al., arXiv:2212.06635



$L = 64$



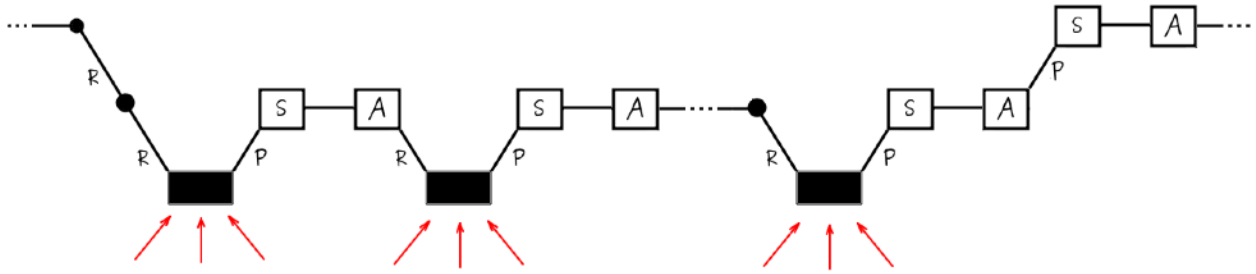
- HMC accelerated with GPU, enables MD times below 2 hours
  - Autotuner for MG parameters turned out to be very useful
- Allows in principle larger lattices, e.g. with L=192
- Enables us to utilize top machines in EU and US
  - e.g. Lumi in Finnland

Strong scaling of MG

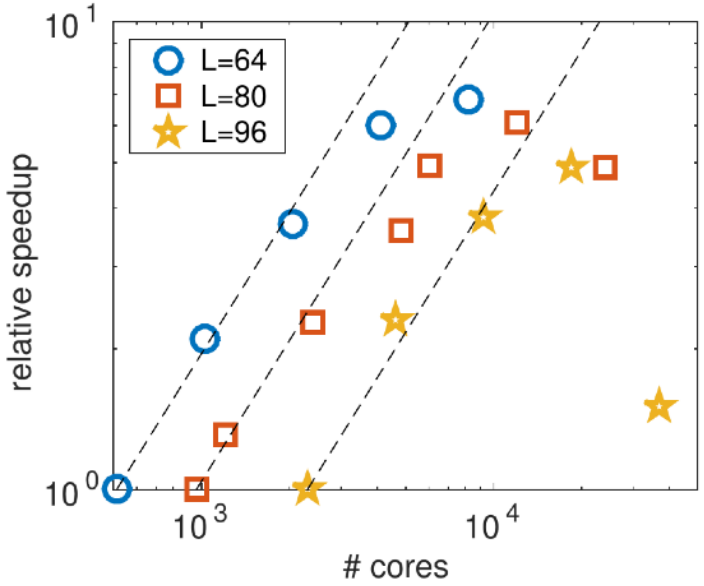
Overcoming network bounds

❖ Coarsest level improvements

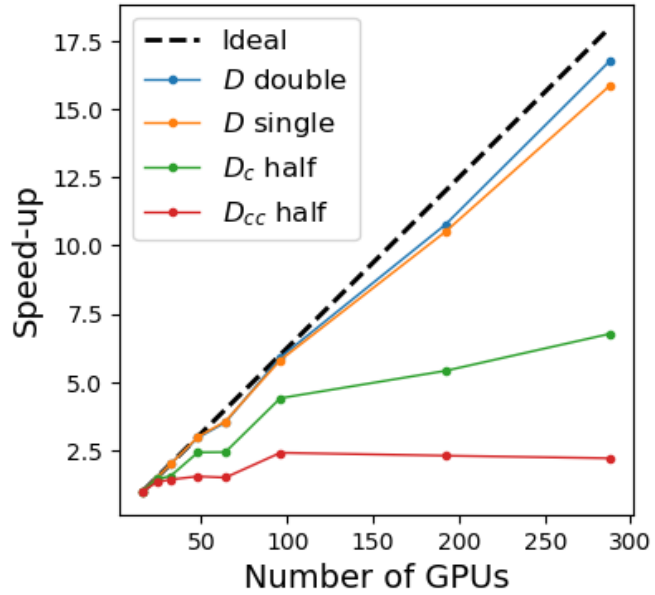
Three level method



SuperMUC-NG scaling



Juwels-Booster scaling



- ❖ Although the coarsest level is solved to 10^-1,
  - ❖ dominating in the strong scaling limit
- ❖ Coarse grid becomes too small, communication dominates

❖ Need for coarse grid improvements

❖ Coarsest level improvements

The Cyprus Institute

S. Yamamoto, PoS LATTICE2021 (2022) 536

Simone Bacchio | Shuhei Yamamoto | J.F.

Enable multi-right-hand sides

- ❖ increase computational load compare to bandwidth
- ❖ Block-Krylov methods
- ❖ Simple vectorisations

Bergische Universitaet Wuppertal

J. Espinoza-Valverde et al., arXiv:2205.09104

Gustavo Ramírez | Jesus Espinoza-Valverde

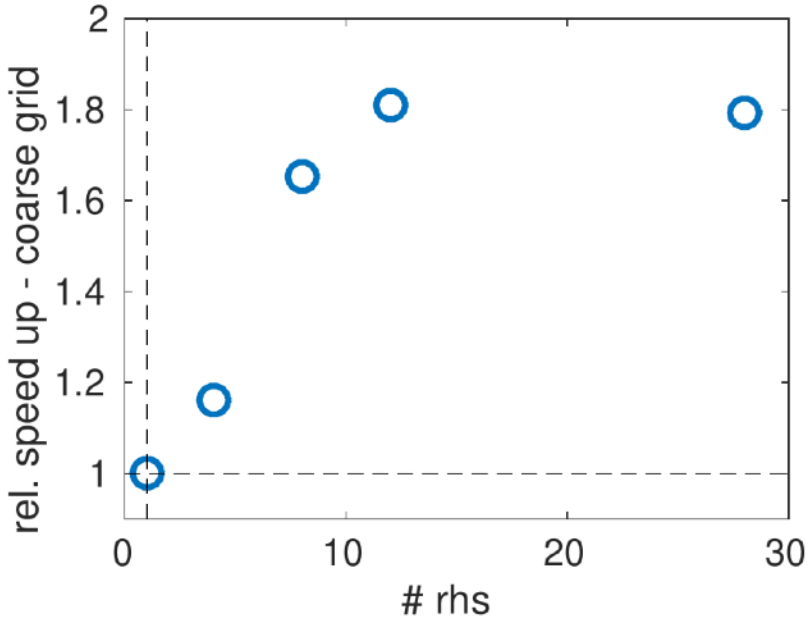
Introduced four new methods for coarsest-level

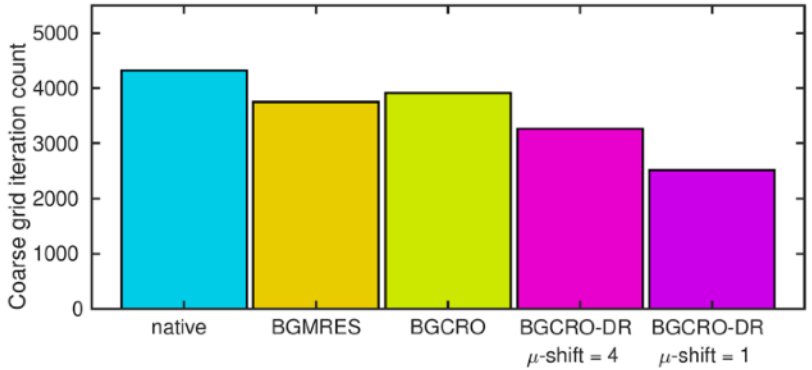
- ❖ Block-diagonal preconditioned: no communication involved
- ❖ polynomial preconditioned: exchanges dot products for matrix-vector multiplications, which improves scaling
- ❖ Deflation and recycling (via GCRO-DR): reduces the iteration count, which reduces the number of dot products, which in turns improves scalability
- ❖ pipelined (does not improve)

DDalphaAMG multiple RHS

Implementation:

- definition of a new data structure, index on vectors runs the fastest
  - re-written all low level kernels
- using auto-vectorisation to enhance portability
- On Skylake chips this shifts vectorisation from 128 bit to 256 bit
- Coarse grid computation speed up by a factor ~1.8





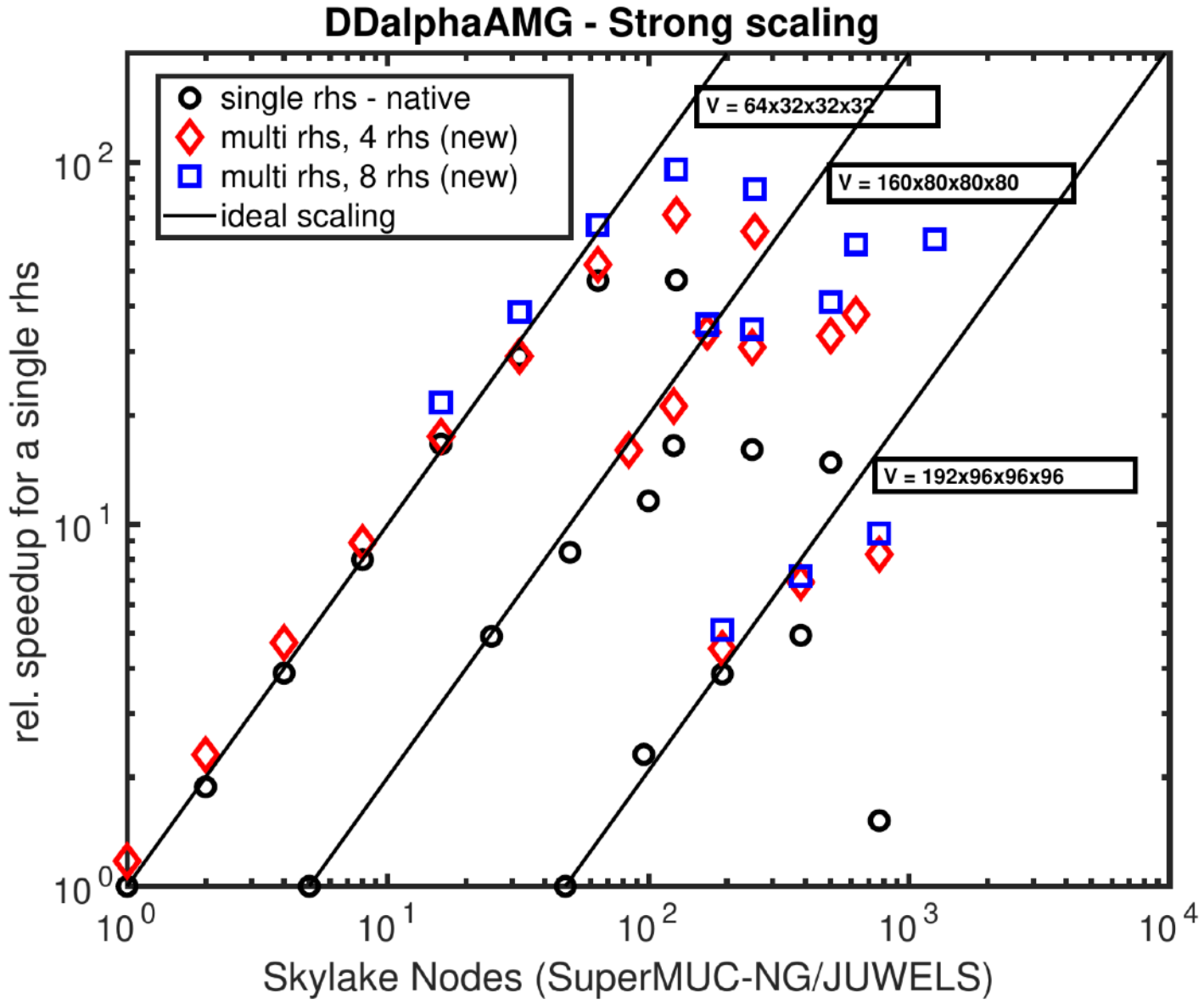
Combination with Block-Krylov solvers possible

- Iteration count is indeed reduced
- Additional overhead to high so far
  - Re-ordering and orthogonalisation

Multi - rhs

Multiple rhs increase scalability

- comes with an additional memory overhead
- up to a factor 5 on Intel Skylake machines
- result so far only obtained on Intel and AMD CPUs



S. Yamamoto, PoS LATTICE2021 (2022) 536

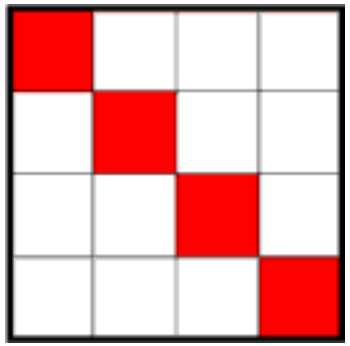
Work by Gustavo Ramírez

BERGISCHE UNIVERSITÄT WUPPERTAL

❖ Coarsest level improvements

J. Espinoza-Valverde et al.,CPC 292 (2023) 108869

Algorithm #1: block-diagonal preconditioned



- ❖ Lets write $$D_L = \begin{pmatrix} D_{ee} & D_{eo} \\ D_{oe} & D_{oo} \end{pmatrix}$$
- ❖ then the Schur complement is given by $$D_c = D_{ee} - D_{eo} D_{oo}^{-1} D_{oe}$$
- ❖ We can then focus on $$D_c x = b$$
- ❖ which can be preconditioned via: $$D_{ee}^{-1} D_c x = D_{ee}^{-1} b$$

m_0	without BDP	with BDP
-0.3515	21	15
-0.35371847789	35	26
-0.354	40	28
-0.3545	52	37

FOR 5269

❖ Coarsest level improvements

Algorithm #2: Polynomial preconditioned

Idea: use $q(D_c) \approx D_c^{-1}$

Then: $D_c q_d(D_c) y = b$ and $x = q_d(D_c) y$

And $q_{\mu-1}(t) = \sum_{i=1}^{\mu} \frac{1}{\theta_i} \prod_{j=1}^{i-1} \left(1 - \frac{t}{\theta_j}\right)$ with θ_i Ritz values

Algorithm #3: GCRO-DR

❖ Polynomial spreads the spectrum around zero, which is beneficial if we want to do deflation:

$$(I - C_k C_k^H) D_c$$

❖ by extracting k eigenvalues of the current Krylov cycle via $U = [u_1, \ldots u_k]$ and $C_k = D_c U_k$

❖ include the vectors $[c_1, \ldots, c_k]$ in the next Krylov cycle

Spectrum transformation:

$$D_c \rightarrow D_c q_{15}(D_c)$$

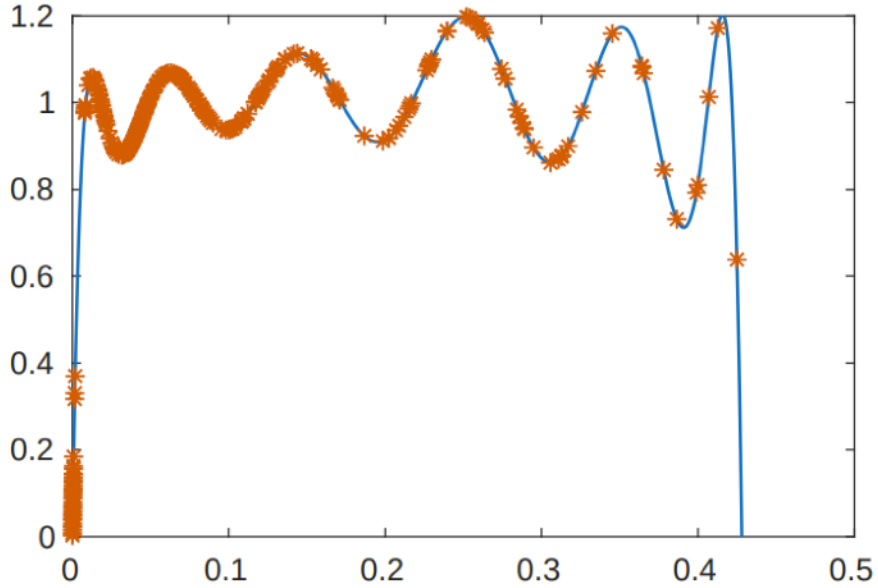
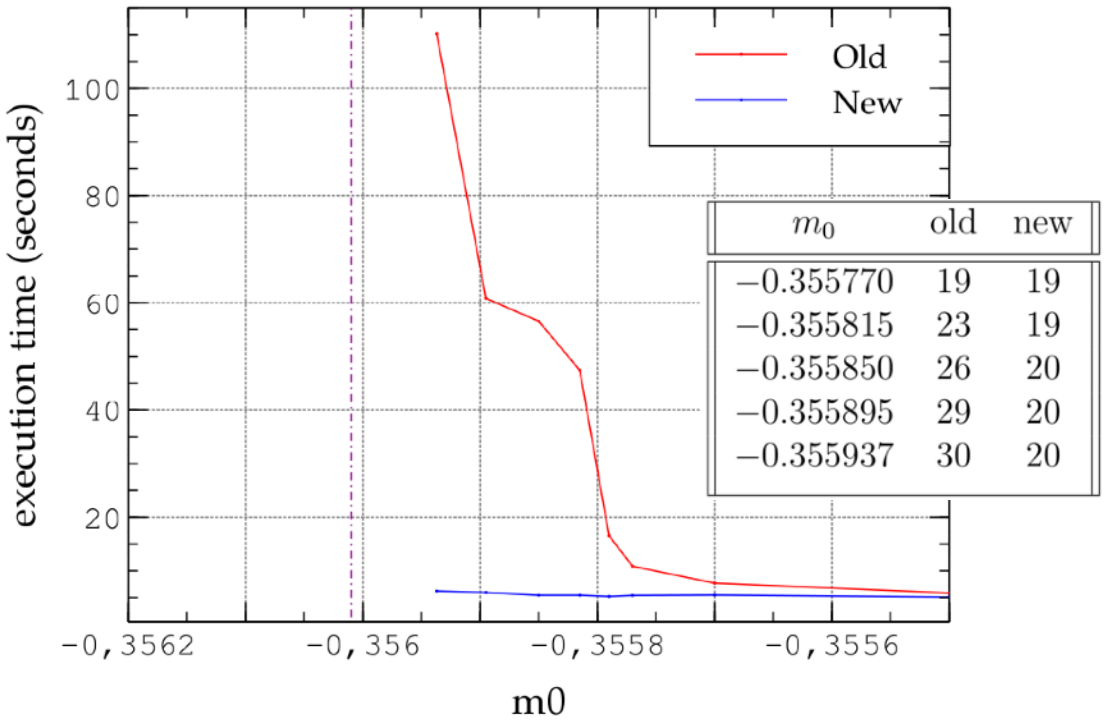


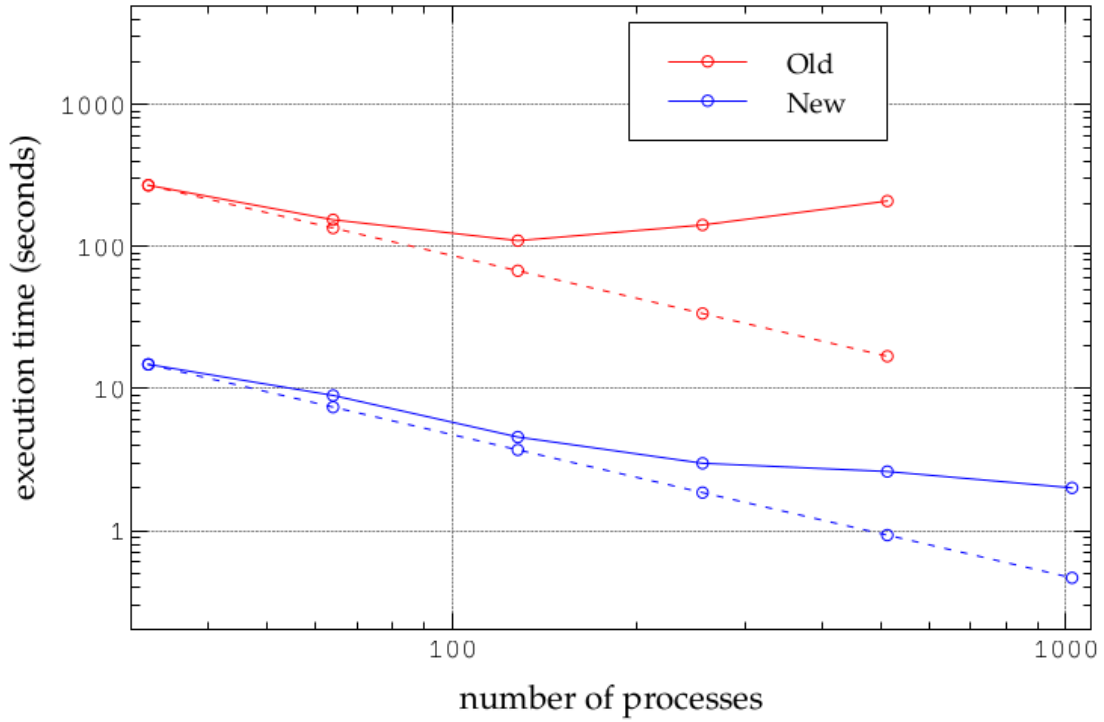
Figure 3.5.(a) in ``Toward efficient polynomial preconditioning for GMRES" by Loe and Morgan

❖ Coarsest level improvements

J. Espinoza-Valverde et al., arXiv:2205.09104



m_0	old	new
−0.355770	19	19
−0.355815	23	19
−0.355850	26	20
−0.355895	29	20
−0.355937	30	20



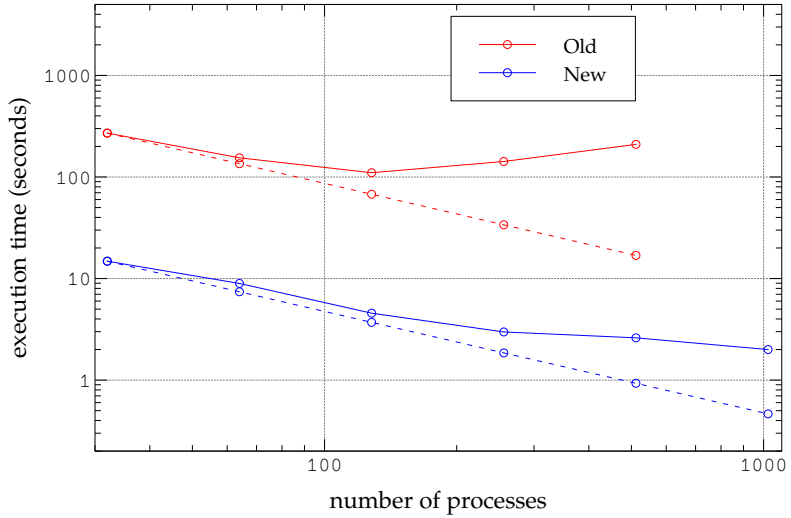
- ❖ On JUWELS cluster
- ❖ lattice 128x64x64x64
- ❖ Wilson-Dirac discretisation
- ❖ k = 25, d=4

- ❖ Block diagonal preconditioned
- ❖ Polynomial preconditioned
- ❖ GCRO-DR

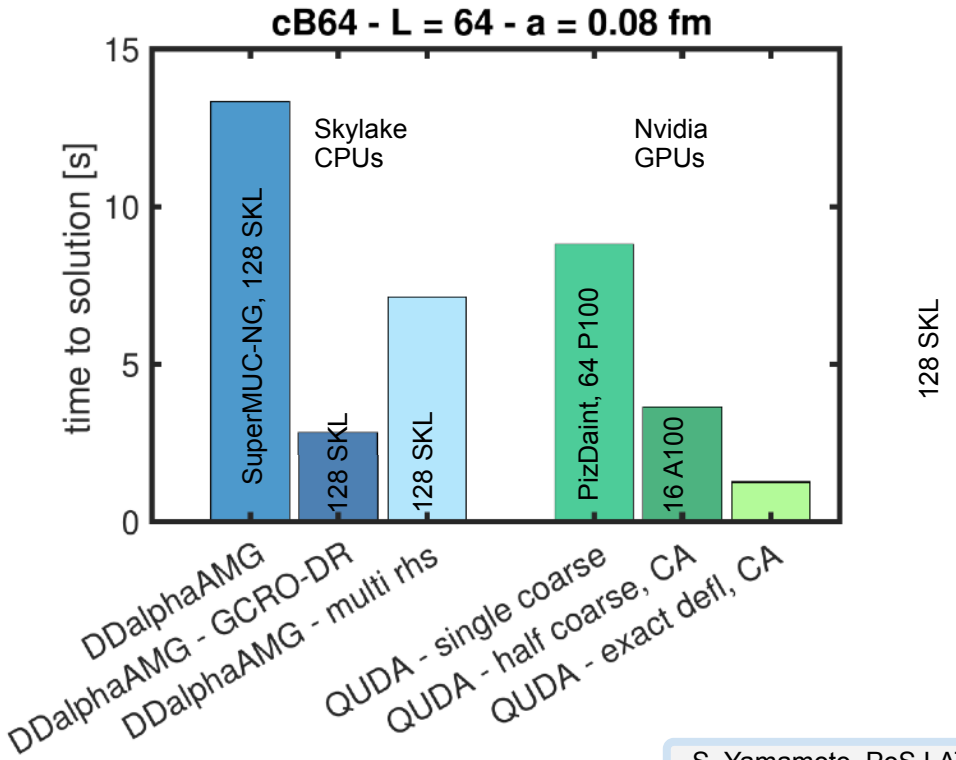
- ❖ m0 = -0.355937 (most ill-conditioned)

FOR 5269

Overview: Multigrid improvements



J. Espinoza-Valverde et al.,CPC 292 (2023) 108869



S. Yamamoto, PoS LATTICE2021 (2022) 536

We are working on: A. Frommer et al. in preparation

DDalphaAMG in C++

- ❖ testing bed for novel algorithms
- ❖ adding GPU-accelerated smoother
- ❖ openBC fully supported

Kernel improvements:

- ❖ Multiple rhs (block Krylov combination)

coarse grid improvements

- ❖ communication avoiding
- ❖ coarse grid deflation

Currently working on a comparison of available MG-methods

BERGISCHE UNIVERSITÄT WUPPERTAL

Conclusion

Multigrid is the method of choice for Wilson like fermions
- ❖ for physical light quarks
- ❖ accelerate HMC simulations

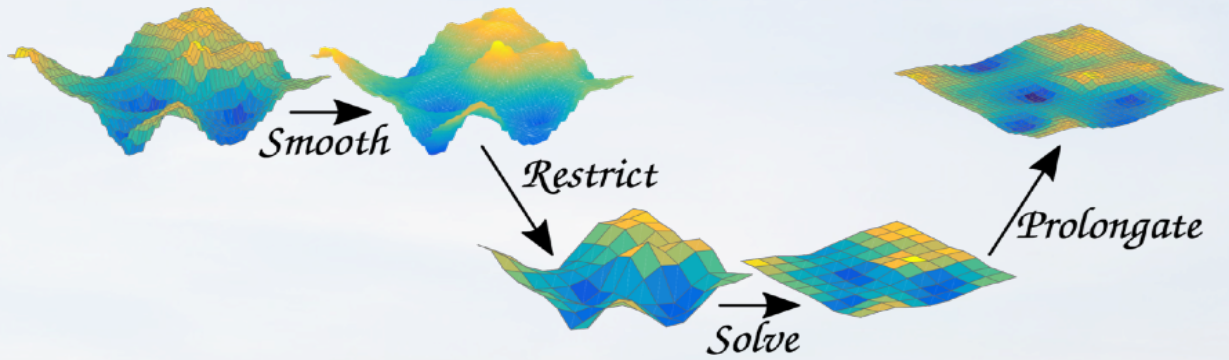
Coarse grid improvements needed to extend strong scaling

Multigrid method can be also utilised in different cases:
- ❖Trace estimation (Multi-Level Monte Carlo)
  - ❖Use coarser grid to split up trace into IR and UV dominated pieces

A. Frommer et al.,*SIAM J.Sci.Comput.* 44 (2022) 4, A2536-A2556

Outlook:
New DDalphaAMG lib written in C++,
- ❖use to test additional methods
  - ❖ Like LU-decomposition of the coarse grid

Next step: combination with multi-level algorithms …



Thank you

ご清聴（ごせいちょう）

ありがとうございました。