B6: Multi-Level Iterative Solvers for Lattice Dirac Operators

Improving Algebraic Multigrid for LQCD via AMS & ML

Marc Alexander Schweitzer [1,2] Stefan Krieg [3]
Jaime Fabián Nieto Castellanos [3] (09/2024)
Pauline Schauerte [1,2] (04/2025)

[1]Institut für Numerische Simulation (INS)
Rheinische Friedrich-Wilhelms-Universität Bonn

[2]Fraunhofer-Institut für Algorithmen und Wissenschaftliches Rechnen (SCAI)

[3]Jülich Supercomputing Centre (JSC)

September 27, 2024

What is the problem?

- QCD theory is highly non-linear, cannot be solved directly.
- Must be approximated by numerical methods to make predictions.

Lattice QCD in a nutshell

- Discretize spacetime, i.e. four dimensional lattice of size $L_x \times L_y \times L_z \times L_t$.
- Finite spacetime implies periodic boundary conditions.
- Differential operators discretized by finite differences.

- Consumer of 10+% of public supercomputer cycles.
- Highly optimized on every single HPC platform for the past 30 years.

It's the linear solver, stupid!

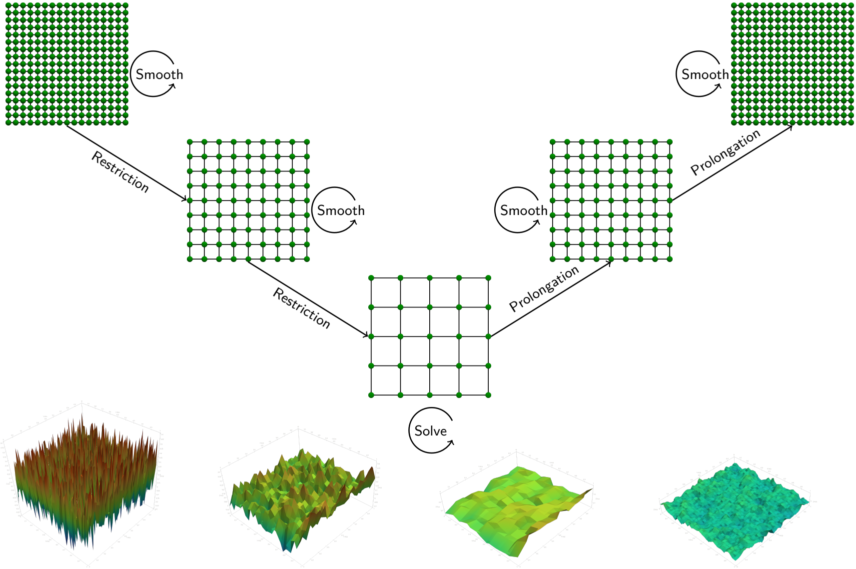
$$\mathsf{D}(\mathsf{U},\mathsf{m})\mathsf{z} = \mathsf{b}, \quad \text{gauge field } \mathsf{U}, \text{mass constant } \mathsf{m}$$

Computational Steps of LQCD

- Generate an ensemble of gluon field configurations, aka gauge generation
  - Hybrid Monte Carlo is the algorithm of choice
  - Produced in sequence, with hundreds needed per ensemble
  - Strong scaling required per task
  - $50-90\%$ of the runtime is in the linear solver
  - $O(1)$ solve per linear system
- Analyzing the configurations:
  - Task parallelism, can be farmed out
  - $80-99\%$ of the runtime is in the linear solver
  - Many solves per system, e.g., $O(10^6)$

Any substantial improvement in runtime can only be attained by better linear solvers!

Multigrid Methods - the optimal solver for discretized PDEs



marc.alexander.schweitzer@scai.fraunhofer.de

Different Flavors of Multigrid Methods

Geometric Multigrid

- Geometric coarsening of grids.
- Classical polynomial interpolation as prolongation.
- May employ complex smoothing schemes to improve robustness.
- Direct solution on coarsest level.

Algebraic Multigrid

- Interpret system matrix as representation of graph.
- Compress and coarsen graph e.g. via agglomeration.
- Employs simple smoothing schemes.
- Construct prolongation from entries of system matrix and/or near kernel information.
- Direct solution on coarsest level.

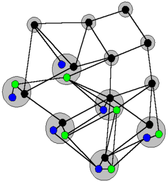
Typically used as preconditioner in a Krylov method to improve robustness and efficiency.

marc.alexander.schweitzer@scai.fraunhofer.de

Different Phases of Multigrid Methods

Setup Phase

- Coarsening of fine grid Ω_f to coarse grid Ω_c.
- Construction of P, $R = P^H$ & $D_c := RDP$.
- Apply recursively.



Solution Phase

- Outer (Flexible) Krylov Method, e.g. (F)GMRES, BiCGstab.
- Cycling schemes V, W or K-cycle.
- Smoother, e.g. block-Gauss-Seidel iteration or GMRES.
- Approximate coarse grid solver, e.g. GMRES, Gauss.

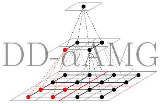
- Huge universe to select the various components in both phases.
- Computational effort in setup phase can differ substantially between different flavors of multigrid.
- Solution phase is essentially universal. Efficiency dominated by matrix-vector-products and coarse grid solution.

Current State of the Art

DD-α-AMG solver
[Rottmann; Frommer, Kahl, Krieg, Leder, Rottmann]

- Simple geometric agglomeration-based AMG.
- Prolongation based on local coherence assumption.
- Utilize block structure (12 degrees of freedom per lattice site/grid point on the fine grid, coarser grids $2 \times \mathrm{N}_\nu$ per agglomerate).
- Inexact multi-coloured block-Gauss-Seidel smoother (denoted as SAP).
- K-cycling in solution phase.
- Even-odd preconditioning, i.e. iteration on Schur complement.

- Widely used approach in LQCD with essentially these fixed components.
- No proof that selected components are optimal & robust.
- Implementation needs to be adapted to current architecture for maximal performance.



Goals of the Project

Development of efficient & scalable linear solver for lattice Dirac operators.

Use ML to improve AMG for LQCD

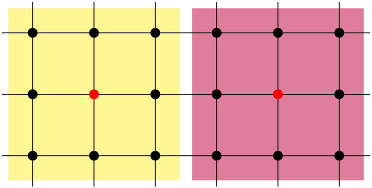
- Determine P by optimizing the spectral radius of the iteration matrix of the two-grid method, $\rho(\mathrm{M})$, as a function of P and A for a fixed smoother. A. Katrutsa *et al*, (2017) arXiv:1711.03825. D. Greenfeld *et al*, (2019) arXiv:1902.10248. I. Luz *et al*, (2020) arXiv:2003.05744
- Use ML to define the C/F splitting (coarsening scheme). A. Taghibakhshi *et al*, (2021) arXiv:2106.01854
- Construct smoothers that efficiently pair with standard coarsening. R. Huang *et al*, (2021) arXiv:2102.12071.
- Create a neural network to predict optimal free parameters (for instance ϵ). P. F. Antonietti, *et al*, (2021) arXiv:2111.01629. H. Zou *et al*, (2023) arXiv:2307.09879. M. Caldana, *et al*, (2024) arXiv:2304.10832

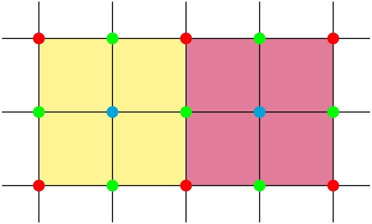
Bring modern multiscale techniques to LQCD

- AMG is designed for much more complex situations that LQCD
- Utilization of structured grid / lattice and upscaling techniques
- Extend algebraic multiscale approach to spacetime lattice and Dirac operator

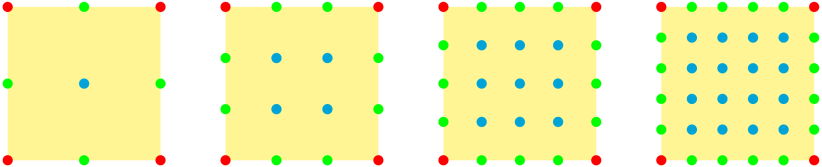
marc.alexander.schweitzer@scai.fraunhofer.de

Aggregation vs. Algebraic Multiscale Coarsening

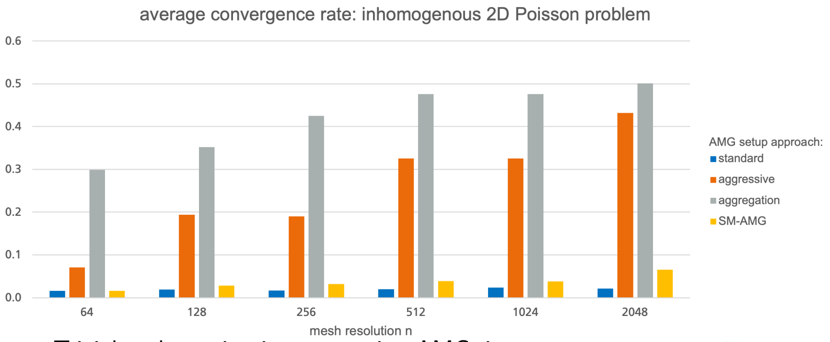




- Aggregates (disjoint) merge all points into single coarse point.
- AMS maintains notion of vertex, edge, ... degrees of freedom (connected).



Aggregation vs. Algebraic Multiscale Prolongation



- Trivial prolongation in aggregation AMG, i.e. piecewise constant
- Robustness and scalability: Unsmoothed vs. smoothed aggregation AMG.
- Algebraic AMS interpolation high quality via parallel construction.

Construction Algebraic Multiscale Prolongation

- Consider local linear system for single aggregate

$$\begin{pmatrix} A_{II} & A_{IE} & A_{IV} \\ A_{EI} & A_{EE} & A_{EV} \\ A_{VI} & A_{VE} & A_{VV} \end{pmatrix} \begin{pmatrix} x_I \\ x_E \\ x_V \end{pmatrix} = \begin{pmatrix} f_I \\ f_E \\ f_V \end{pmatrix}$$

- Direct interpolation of coarse points / vertices allows for

$$\begin{pmatrix} A_{II} & A_{IE} & A_{IV} \\ A_{EI} & A_{EE} & A_{EV} \\ 0 & 0 & \mathbb{I}_{VV} \end{pmatrix} \begin{pmatrix} u_I \\ u_E \\ u_V \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ u_V^c \end{pmatrix}$$

- Yielding the prolongation operator

$$P = O \cdot N \cdot \begin{pmatrix} -\hat{A}_{II}^{-1}(A_{IV} - A_{IE}A_{EE}^{-1}A_{EV}) \\ -\hat{A}_{EE}^{-1}(A_{EV} - A_{EI}A_{II}^{-1}A_{IV}) \\ \mathbb{I}_{VV} \end{pmatrix}$$

with $\hat{A}_{II} := A_{II} - A_{IE}A_{EE}^{-1}A_{EI}$ and $\hat{A}_{EE} := A_{EE} - A_{EI}A_{II}^{-1}A_{IV}$

Status & Outlook

Status

Positions recently filled / accepted.

Next Steps

- Evaluate if AMS can be easily realized in DD-α-AMG implementation.
- Evaluate if Dirac operator can be easily realized in our AMG implementations.
- Evaluate AMG components simplified to structured grid input.
- Evaluate AMG components for matrix-free realizations.
- Evaluate performance impact of coarse grid solution.

Long-term Outlook

- Implementation with B02 framework.
- ML optimization of parameters on MSA.