DDalphaAMG for Twisted Mass Fermions

Simone Bacchio[∗, a, b]**, Constantia Alexandrou**[a,c]**, Jacob Finkenrath**[c]**,
Andreas Frommer**[b]**, Karsten Kahl**[b]** and Matthias Rottmann**[b]

[a]*Department of Physics, University of Cyprus, PO Box 20537, 1678 Nicosia, Cyprus*
[b]*Fakultät für Mathematik und Naturwissenschaften, Bergische Universität Wuppertal*
[c]*Computation-based Science and Technology Research Center, The Cyprus Institute*

E-mail: s.bacchio@hpc-leap.eu

We present the Adaptive Aggregation-based Domain Decomposition Multigrid method extended to the twisted mass fermion discretization action. We show comparisons of results as a function of tuning the parameters that enter the twisted mass version of the DDalphaAMG library [3]. Moreover, we linked the DDalphaAMG library to the tmLQCD software package and give details on the performance of the multigrid solver during HMC simulations at the physical point.

∗Speaker.

1. Introduction

The Adaptive Aggregation-based Domain Decomposition Multigrid method, referred to as DD-αAMG, has been introduced in Ref. [1] as a solver for the Wilson clover operator, D_W. In DD-αAMG a flexible iterative Krylov solver is preconditioned at every iteration step by a multigrid approach given by the error propagation

$$\varepsilon \leftarrow (I - MD)^k \left(I - PD_c^{-1}P^\dagger D \right) (I - MD)^j \varepsilon, \tag{1.1}$$

where M is the smoother, j and k are the number of pre- and post-smoothing iterations respectively, P is the interpolation operator and $D_c = P^\dagger D_W P$ is the coarse grid operator. The multigrid preconditioner exploits domain decomposition strategies having for instance as a smoother the Schwarz Alternating Procedure (SAP) [5] and as a coarse grid correction an aggregation-based coarse grid operator. The method is designed to deal efficiently with both, infrared (IR)- and ultra-violet (UV)-modes of D_W. Indeed the smoother reduces the error components belonging to the UV-modes [1], while the coarse grid correction deals with the IR-modes. This is achieved by using a interpolation operator P, which approximately spans the eigenspace of the small eigenvalues. Thanks to the property of local coherence [6] the subspace can be approximated by aggregating over a small set of $N_v \simeq \mathscr{O}(20)$ test vectors v_i, which are computed in DD-αAMG via an adaptive setup phase [1]. We remark that the interpolation operator in DD-αAMG is Γ_5-compatible, i.e. $\Gamma_5 P = P\Gamma_{5,c}$. Thanks to this property the Γ_5-hermiticity of D_W is preserved on the coarse grid as well – i.e. $D_c^\dagger = \Gamma_{5,c}D_c\Gamma_{5,c}$.
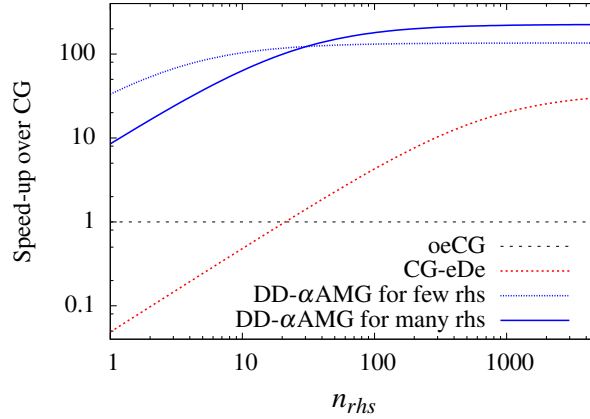

Figure 1: Speed-up of the DD-αAMG solver compared to odd-even preconditioned CG (oeCG) and CG with exact deflation (CG-eDe) by using 1600 eigenvectors. The timings for the DD-αAMG method and CG-eDe include the time for the build-up and the setup. The setup is required just once and then applied to several rhs on the same configuration. The optimized performance of DD-αAMG for few and many rhs are obtained by changing the setup procedure, i.e. three setup iterations in case of few rhs and five setup iterations for many rhs, see also Ref. [2].

Recently, we extended the DD-αAMG approach to the Twisted Mass (TM) fermions discretization with the TM operator $D(\pm\mu) = D_W \pm i\Gamma_5\mu$ [2]. When the PCAC (partial-conserved axial current) mass is tuned to zero, the TM discretization cancels the linear lattice discretization effects [7]. The squared operator $D^\dagger(\mu)D(\mu) = D_W^\dagger D_W + \mu^2$ is bounded from below by μ^2. Thus

a finite twisted mass term $i\Gamma_5\mu$ protects the TM operator $D(\mu)$ from being singular, unlike the Wilson clover operator D_W where this can happen for small quark masses.

By extending DD-αAMG to the TM discretization we observe a significant increase of the iteration count of the multigrid method at the physical value of the pion mass – i.e. at small values of μ. We find that the eigenvalue density of the squared operator is densely populated close to μ^2 [2]. This increases significantly the iteration count on the coarse grid and slows down the method. By increasing the TM parameter on the coarsest grid $\mu_{\text{coarse}} = \delta \cdot \mu$ with $\delta \geq 1$ the coarse grid iteration count can be reduced by an order of magnitude while simultaneously the fine grid iteration count only increases slightly. For $\delta \sim 5$ this improves the time to solution by a factor 4 for configurations of the $cA2.09.48$ [8] with lattice size $48^3 \times 96$ and pion mass ~ 0.131 GeV. We are thus able to achieve a similar speed up like it is found in the case of the Wilson clover operator at near physical pion masses. In Ref. [2] we have presented a thorough analysis of the solver parameters achieving speed-ups of more than a magnitude in time compared to the Conjugate Gradient (CG) algorithm, e.g. a speed-up of $\mathcal{O}(100)$ when the solutions of at least ten right hand sides (rhs) are needed. The result is depicted in Figure 1.

One advantage of the DD-αAMG approach, when applied to the TM operator, is the Γ_5-compatibility. The TM term is still diagonal on the coarse grid similarly to the fine grid operator and the coarse grid operator is given by $D_c(\mu) = D_c + i\mu\Gamma_{5,c}$. Moreover, the same setup can be used for inverting both, $D(+\mu)$ and $D(-\mu)$, without affecting the performance of the solver as shown in Fig. 2. Therefore, the method does not require an additional setup procedure when linear systems with the squared operator $D^\dagger(\mu)D(\mu)$ have to be solved.
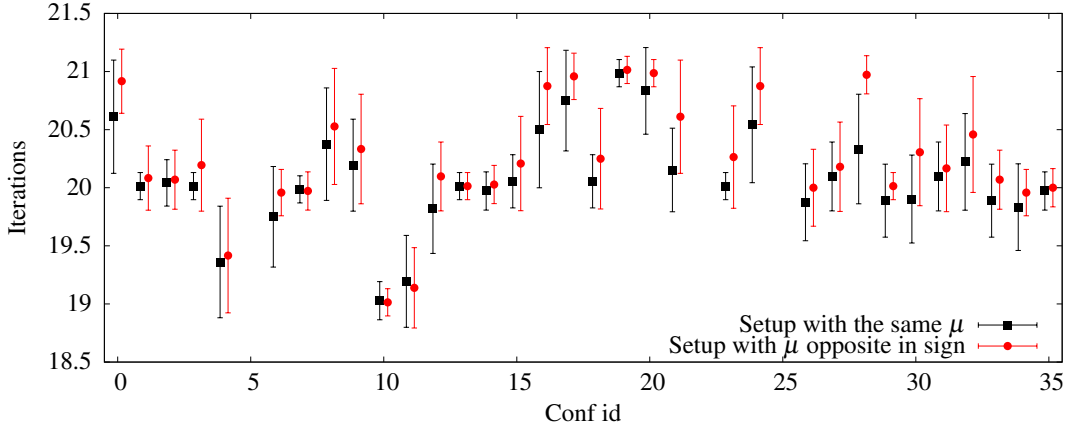

Figure 2: Average iteration count on computing quark propagators for several configurations of $cA2.09.48$.

A code package containing the DD-αAMG approach is publicly available in the DDalphaAMG library [9]. Our TM version of the code is also publicly available in the resource [3] and we provide an interface to the tmLQCD software package [12] available at the resource [4]. In the next sections we give some details about the library and in section 3 we described in detail how the solver can be called within tmLQCD.

2. Overview of the DDalphaAMG library

The DDalphaAMG solver library has been released under GNU General Public License at resource [9]. This software package includes an implementation of the DD-αAMG solver for the Wilson clover operator as described in Ref. [1]. The implementation is of production code quality, it includes a hybrid MPI/openMP parallelization, state-of-the-art mixed precision and odd-even preconditioning approaches and also SSE3 optimizations. Implementation details are available in Ref. [10].

Based on the DDalphaAMG code we have developed a version, which supports TM fermions, available at the resource [3]. At the moment the following features have been added to the library: TM operator with $N_f = 2$ and twisted boundary conditions are supported, a different TM parameter can be applied on the even and odd sites – required for the Hasenbusch mass preconditioning in the HMC simulations when odd-even preconditioning is used – and a new interface to the library is provided. All details about the interface can be found in the header of the library `DDalphaAMG.h` and a sample code, which links to the library is given in `tests/DDalphaAMG_sample.c`. Moreover the DDalphaAMG package can be used as an independent software package including features as reading `LIME` configurations and reading/writing `LIME` vectors. More information can be found in the package documentation in the `doc/` directory.

The library interface is designed to provide an easy integration of the solver to production codes, e.g. codes which are used in computations of fermionic observables or in HMC simulations. A minimal set of variables is required for the initialization of the solver, while a wide set of parameters is set by default but can be modified for further optimizations. The list of parameters is given in the interface header, and information about their analysis and performance improvement is given in Refs. [2, 1]. The code conventions are the following: the γ_5 matrix is given by

$$\gamma_5 = \begin{pmatrix} -1 & & & \\ & -1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}. \tag{2.1}$$

while the representation of the other γ_ν matrices can be freely chosen, and sets are provided in `Clifford.h`. The order of the lexicographical index is fixed to *TZYX*.

The standard setup of the DDαAMG approach is a three-level multigrid with aggregation block size of 4^4 between fine and first-coarse grid and of 2^4 between first- and second- (coarsest) coarse grid. While the lower aggregation block size is automatically set, the first aggregation block size can be further optimized by tuning at the same time the number of test vectors as it is described in Ref. [2].

The number of levels and block sizes limit the maximal number of MPI parallel processes. At least two lattice sites of the coarsest grid are required per process when odd-even preconditioning is used. Assuming the mentioned aggregation sizes the coarsest grid volume is given by $V/(4^4 \cdot 2^4)$ with V the fine grid volume. Thus the maximal number of MPI-processes is given by $V/(2 \cdot 4^4 \cdot 2^4)$. On recent machines, like Jureca, we observe an almost ideal strong scaling up to the maximal number of processes. Moreover, it is possible to parallelize further by using openMP. This increases the possible number of processes but without an ideal strong scaling. Additionally the library

interface provides a trivial parallelization by splitting up the MPI-communicator. This can be used if solutions to several rhs have to computed in parallel when large number of processes are required.

3. Employing the DD-αAMG solver within tmLQCD

The DDalphaAMG library has been integrated into the tmLQCD software package [12], which is commonly used by the European Twisted Mass (ETM) collaboration. The code with the interface branch is available at the resource [4]. The DD-αAMG solver can be used in all the applications of the software, which involve the inversion of the TM or the Wilson clover operator. The usage of the solver can be specified by `Solver = DDalphaAMG` in the tmLQCD-input-file while the parameters can be set by adding a similar parameter environment like it is shown in the Listing 1.

```
BeginDDalphaAMG
    MGBlockX = 4
    MGBlockY = 4
    MGBlockZ = 4
    MGBlockT = 4
    MGSetupIter = 5
    MGCoarseSetupIter = 3
    MGNumberOfVectors = 24
    MGNumberOfLevels = 3
    MGCoarseMuFactor = 5
EndDDalphaAMG
```

```
BeginOperator TMWILSON
    2kappaMu = 0.05
    kappa = 0.177
    Solver  = DDalphaAMG
    SolverPrecision  = 1e−14
EndOperator
```

Listing 1: DDalphaAMG parameters in a tmLQCD input file

A detailed description of the available parameters can be found in the package documentation in `doc/main.pdf`. The solver is tested within the applications `invert` and `hmc_tm`, while in any case an additional check of the residual is performed in tmLQCD. Note that even if the solution of the odd-even preconditioned operator is required, the DD-αAMG approach uses the full operator for the inversion. Since odd-even preconditioning reduces the sparsity of the operator, in DD-αAMG it is used only on the coarsest level and in the smoother. This is also the case if an inversion of the squared operator is required, where DDalphaAMG performs two inversions of the non-squared operator.

3.1 HMC simulation with DDalphaAMG

We perform an HMC simulation to generate a $64^3 \times 128$ ensemble at physical pion mass with an integration scheme equivalent to the one used in Ref. [8] for the ensemble $cA2z.09.48$. The DD-αAMG solver is employed in the force term computation, heat-bath and acceptance step. In the integration scheme the Hasenbusch mass preconditioning [11] is used. During the integration the squared operator $\hat{D}^\dagger(\mu)\hat{D}(\mu) + \rho_i^2$ has to be inverted, where $\hat{D}(\mu)$ is the odd-even reduced or odd-even preconditioned TM operator. In DDalphaAMG this is done by first inverting $\hat{D}(\mu) + i\rho_i\hat{\Gamma}_5$ and then $\hat{\Gamma}_5\hat{D}(-\mu)\hat{\Gamma}_5 - i\rho_i\hat{\Gamma}_5$, where $\hat{\Gamma}_5$ is Γ_5 restricted to the odd lattice sites. The same interpolation operator P can be used for all inversions of the operators involved in the HMC procedure. The setup is built once at the beginning of each trajectory, where three iterations on the fine grid and three on the first coarse grid are used. During the integration the setup is updated every time before the operator with the smallest Hasenbusch mass $\rho_0 = 0$ is inverted. The update is done by one fine

grid and three first coarse grid iterations. By using this approach the DD-αAMG solver showed very stable iteration counts for all ρ_i as it is depicted in Figure 3.
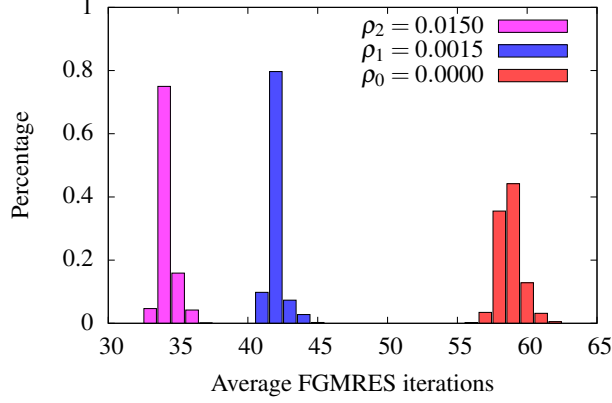

Figure 3: FGMRES iteration counts averaged over the trajectory when the solver is used for computing the force terms in the HMC simulation. The data are for the ensemble $cA2.09.64$ with statistics of 2000 trajectories. The parameters ρ_i set the mass for the Hasenbusch preconditioning. The iteration counts include two inversions of the non-squared TM operator.

The calculation of the force term requires a higher accuracy for the inversions when the DD-αAMG approach is used instead of the CG solver. This is required to maintain the reversibility of the HMC. Indeed, while the solution provided by CG only depends on the current configuration and the right hand side, the multigrid setup carries information from the previous configurations which thus influences the solution corresponding to the current configuration.

4. Conclusions and outlook

The DD-αAMG approach is extended to the case of $N_f = 2$ twisted mass fermions [2]. The code is publicly available in the twisted mass version of the DDalphaAMG library [3]. Moreover, we implement an interface in tmLQCD available under [4]. After tuning the parameters, the inversions are performed more than two orders of magnitude faster as compared to standard CG. Within the HMC simulations with $N_f = 2$, DD-αAMG achieves a speed-up of an order of magnitude compared to standard CG. Future steps will be the integration of the heavy quark twisted mass operator into the DDalphaAMG library [3]. Furthermore, we plan to update the vectorization to AVX instructions.

Acknowledgments

References

[1] A. Frommer, K. Kahl, S. Krieg, B. Leder, and M. Rottmann. Adaptive Aggregation Based Domain Decomposition Multigrid for the Lattice Wilson Dirac Operator. *SIAM J. Sci. Comput.*, 36:A1581–A1608, 2014.

[2] C. Alexandrou, S. Bacchio, J. Finkenrath, A. Frommer, K. Kahl and M. Rottmann. Adaptive Aggregation-based Domain Decomposition Multigrid for Twisted Mass Fermions. 2016. arXiv:1610.02370 [hep-lat].

[3] S. Bacchio and J. Finkenrath. DDalphaAMG, twisted mass version. Available at https://github.com/sbacchio/DDalphaAMG.

[4] S. Bacchio and J. Finkenrath. tmLQCD interfaced to DDalphaAMG. Available at https://github.com/finkerath/tmLQCD/tree/DDalphaAMG.

[5] M. Lüscher. Solution of the Dirac equation in lattice QCD using a domain decomposition method. *Comput. Phys. Commun.*, 156:209–220, 2004.

[6] M. Lüscher. Local coherence and deflation of the low quark modes in lattice QCD. *JHEP*, 07:081, 2007.

[7] R. Frezzotti and G. C. Rossi. Chirally improving Wilson fermions. 1. O(a) improvement. *JHEP*, 08:007, 2004.

[8] A. Abdel-Rehim et al. Simulating QCD at the Physical Point with $N_f = 2$ Wilson Twisted Mass Fermions at Maximal Twist. 2015.

[9] M. Rottmann, A. Strebel, S. Heybrock, S. Bacchio, and B. Leder. DDalphaAMG. Available at https://github.com/DDalphaAMG/DDalphaAMG.

[10] M. Rottmann. Adaptive Domain Decomposition Multigrid for Lattice QCD. *PhD thesis*, Bergische Universität Wuppertal, 2016.

[11] M. Hasenbusch and K. Jansen. Speeding up lattice QCD simulations with clover improved Wilson fermions. *Nucl. Phys.*, B659:299–320, 2003.

[12] K. Jansen and C. Urbach. tmLQCD: A Program suite to simulate Wilson Twisted mass Lattice QCD. *Comput. Phys. Commun.*, 180:2717–2738, 2009.