PROCEEDINGS OF SCIENCE

Deflation and polynomial preconditioning in the application of the overlap operator at nonzero chemical potential

PoS(LATTICE2024)284

Gustavo Ramirez-Hidalgo[a,*] **and Jacob Finkenrath**[b]

[a] *Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH,*
Wilhelm-Johnen-Straße 52428 Jülich, Germany

[b] *Department of Theoretical Physics, European Organization for Nuclear Research, CERN,*
CH-1211 Geneva 23, Switzerland

E-mail: g.ramirez.hidalgo@fz-juelich.de, j.finkenrath@cern.ch

When solving linear systems with the overlap operator at nonzero chemical potential μ in lattice QCD one needs, at every iteration of the iterative solver, to apply the sign function evaluated on a non-Hermitian operator Q_μ times a vector, i.e., $\mathrm{sign}(Q_\mu)v$. In this work we describe how deflation and (the more recently proposed) polynomial preconditioning can be applied to this problem, in particular in the context of lattice QCD. Furthermore, we describe how both methods can be combined, we compare them in numerical experiments and explore whether there might be any synergy between both that can be exploited.

The 41st International Symposium on Lattice Field Theory (LATTICE2024)
28 July - 3 August 2024
Liverpool, UK

[*] Speaker

1. Introduction

The overlap discretization in lattice Quantum Chromodynamics (QCD) allows the simulation of QCD on the lattice with the attractive feature of having a lattice version of *chiral symmetry* [1]. Chiral symmetry is relevant for some physical observables such as hadron spectra in the presence of magnetic fields. The realization of chirality on the lattice is done via the Ginsparg-Wilson relation [2]; Neuberger has provided in [3] a construction of the *overlap* Dirac operator, that fulfills the Ginsparg-Wilson relation.

A nonzero chemical potential in lattice QCD can be employed to describe various physical objects in different areas of physics, e.g., in cosmology for descriptions of the early universe or in high energy physics for studying heavy-ion collisions [4]. Neuberger's overlap operator can be equipped with a nonzero chemical potential, leading to the necessity of evaluating the sign function on a non-Hermitian operator [5].

Our focus in this work is on stressing on what are, still, computationally expensive and problematic components in solving linear systems with the overlap operator, in particular at a nonzero chemical potential. In doing so, we describe a recently proposed method (polynomial preconditioning) for accelerating the evaluation of the sign function times a vector, its use in the context of lattice QCD, and we analyze how this compares to LR deflation, and moreover whether there is any synergy between polynomial preconditioning and deflation that can be exploited.

The contents of this work are distributed as follows. In sect. 2, we list some of the most relevant recent algorithmic advances in the solution of linear systems with the overlap operator D_N. In particular, in sect. 2.4 we go deeply into the recently proposed algorithm of polynomial preconditioning for the action of the inverse square root of a matrix on a vector. Then, in sect. 3 we describe how one can combine deflation and polynomial preconditioning for this sign function problem. Furthermore, we briefly explain, in sect. 4, how we have implemented and analyzed, via numerical experiments, the interplay of deflation and polynomial preconditioning. Finally, with all of the previous algorithmic pieces in place, we give some motivation on how a proper overlap solver should look like on modern heterogeneous supercomputing architectures and comment on some future work in sect. 5.

2. Some algorithms for solving linear systems with overlap fermions

The linear systems emerging in lattice QCD simulations with the overlap discretization take the form [1]:

$$D_{ov}(\mu)x = b, \tag{1}$$

where the overlap operator at nonzero chemical potential can be written as $D_{ov}(\mu) = \rho I + \Gamma_5 \text{sign}\left(Q_\mu\right)$, with $Q_\mu := \Gamma_5 D_W(m_W, \mu)$. Here, the matrix $D_W(m_W, \mu)$ is the usual Wilson-Dirac operator at nonzero mass m_W, and nonzero chemical potential μ. The value of m_W is tuned for improving the locality of the theory, with $m_W \approx 1.4$ set for optimality in general. The nonzero chemical potential is realized via the rescaling of the gauge links in the time direction with exponential factors e^μ and $e^{-\mu}$ on $+T$ and $-T$, respectively [5]. Having a nonzero chemical potential renders Q_μ non-Hermitian; we note that $Q_\mu^H = Q_{-\mu}$. Our interest here is on $\mu \neq 0$.

There are many studies on new algorithmic improvements for solving the linear system in eq. 1, e.g., [6–8], with associated convergence analyzes that permit one to relax the accuracy up to which certain components of the computation need to be attained.

We describe now what we believe are the main algorithms needed for setting up a fast solver for linear systems with the overlap operator, i.e., eq. 1. We start with a brief description of preconditioning at the outermost level of the solver for eq. 1. Then we explain how deflation and polynomial preconditioning can be applied in the inner part of the linear solve, namely, in the application of the sign function on a vector $\text{sign}(Q_\mu)v$. In sect. 4 we analyze the interplay of deflation and polynomial preconditioning.

2.1 Preconditioning with multigrid

When solving the linear system in eq. 1, due to Q_μ being non-Hermitian, one can use for example the GMRES algorithm [9]. “Traditional" iterative methods, e.g., Gauss-Seidel, GMRES, etc. [10], they all suffer from *critical slowing down*, that is, the convergence rate worsens as the condition number increases (which usually happens when one drives one or more parameters in the physical system to their continuum values). To alleviate this, one can use *multigrid* methods [11]. In particular, aggregation based algebraic multigrid [12, 13] has been very successful in lattice QCD [14–21]. In [22], FGMRES (the flexible variant of GMRES [10]) is employed for solving eq. 1 (with $\mu = 0$), and the preconditioner is the application of $D_W^{-1}(m_{prec})$ up to a certain tolerance ϵ. By tuning m_{prec}, critical slowing down is then considerably reduced, but of course it is offloaded to the application of $D_W^{-1}(m_{prec})$. Then, by using multigrid in the evaluation of $D_W^{-1}(m_{prec})$ times a vector, the authors in [22] find that critical slowing down is dramatically reduced for this problem. Furthermore, they find that $\epsilon = 10^{-1}$ is an approximately optimal value from a total execution time point of view.

But once this problem in the outer solver is “fixed", one has to turn to a very expensive piece of the computation, namely the evaluation of $\text{sign}(Q_\mu)v$, and figure out ways of reducing the cost of that operation. We now discuss two algorithms for precisely that purpose, whether one of them is better than the other and their possible interplay.

2.2 The Arnoldi method for $f(A)v$

A simple yet robust method for computing the application of a function of a matrix times a vector, in the non-Hermitian case, is the Arnoldi method. As the algorithms described in the following sections all rely on the Arnoldi method, it is worth briefly explaining this technique before diving into the less simple deflation and polynomial preconditioning.

The Arnoldi method consists of the following steps. First, one runs k steps of the Arnoldi iteration [10] with $\frac{v}{\|v\|_2}$ as the first Arnoldi vector, obtaining with this the Arnoldi vectors $V_k \in \mathbb{C}^{n\times k}$ and the Hessenberg matrix $H_{k+1} \in \mathbb{C}^{(k+1)\times k}$ fulfilling $AV_k = V_{k+1}H_{k+1}$. The eigenvalues of H_k serve as approximations to part of the spectrum of A. A second and last step is to use V_k and H_k to build an approximation to $f(A)v$, $f(A)v \approx V_k f(H_k)V_k^H v$, which consists of a “lifting” of $f(H_k)$ from the small subspace up to the large subspace, i.e., that where A acts.

2.3 Deflating the sign function

As introduced in [23], one can deflate LR (left-right) eigenmodes when applying $f(A)v$, for some function f (e.g., sign) and some matrix $A \in \mathbb{C}^{n\times n}$. The motivation behind this is the following. Let us assume that A is nonsingular, therefore it admits an eigendecomposition of the form $A = X\Lambda X^{-1}$. We call X the *right eigenvectors*, as we can write the previous expression in the form $AX = X\Lambda \rightarrow AR = R\Lambda$. Furthermore, by rewriting it as $X^{-1}A = \Lambda X^{-1}$, we then call X^{-1} the *left eigenvectors*, due to the notation $X^{-1}A = \Lambda X^{-1} \rightarrow L^H A = \Lambda L^H \rightarrow A^H L = L\Lambda^H$. Finally, one can then write the eigendecomposition of A as $A = R\Lambda L^H$.

One can then furthermore write this as a summation of rank-1 updates: $A = \sum_{i=1}^n r_i \lambda_i l_i^h$. Let us now deflate the LR eigenmode for $i = 1$. To do so, we have to apply the projector $\pi_1 = I - r_1 l_1^H$ from the right: $A\pi_i = r_1\lambda_1 l_1^h(I - r_1 l_1^H) + \sum_{i=2}^n r_i\lambda_i l_i^h(I - r_1 l_1^H)$. The first term vanishes thanks to $L^H R = X^{-1}X = I$, and in the second term $l_i^H r_1$ is zero due to the same connection between L^H and R. We are then left with $A\pi_i = \sum_{i=2}^n r_i\lambda_i l_i^h$. Therefore, in general:

$$\Pi_m = I - R_m L_m^H \Rightarrow A\Pi_m = \sum_{i=m+1}^{n} r_i\lambda_i l_i^h. \tag{2}$$

These relations are used in [23] to deflate LR eigenmodes in $f(A)v$:

$$f(A)v = f(A)R_m L_m^H v + f(A)\Pi_m v = f(A)R_m L_m^H v + f(A)v_\ominus = R_m f(\Lambda_m) L_m^H v + f(A)v_\ominus, \tag{3}$$

with: $AR_m = R_m\Lambda_m$, $L_m^H A = \Lambda_m L_m^H$, $v_\ominus = \Pi_m v = v - R_m L_m^H v$. One then uses the Arnoldi method presented in sect. 2.2 on the deflated term $f(A)v_\ominus$. The "direct" term in eq. 3 reads $R_m f(\Lambda_m) L_m^H v$.

Can we do something similar with the SVD instead of the eigendecomposition? Take the SVD of A to be $A = U\Sigma V^H = \sum_{i=1}^n u_i\sigma_i v_i^H = U_m\Sigma_m V_m^H + \sum_{i=m+1}^n u_i\sigma_i v_i^H$. We can multiply from the right with $\Pi_m = I - V_m V_m^H$ to remove the first m singular modes: $A\Pi_m = \sum_{i=m+1}^n u_i\sigma_i v_i^H$. This looks just like the deflated expression in eq. 2. Let us furthermore use this in $f(A)v$: $f(A)v = f(A)V_m V_m^H v + f(A)\Pi_m v$. The second term can be evaluated again via the Arnoldi method. The first term, though, does not seem to be able to be written in closed form as in the LR deflation case. To see this, write $AV_m V_m^H = U_m\Sigma_m V_m^H \Rightarrow A^2 V_m V_m^H = AU_m\Sigma_m V_m^H = U\Sigma V^H U_m\Sigma_m V_m^H$. The problem with not being able to do SVD deflation is rooted in the fact that the connection between V and U is via A and not direct, as it was in the case in the LR eigendecomposition where $RL^H = L^H R = I$. Therefore, we cannot write the "direct" term in the SVD case in a closed form, e.g., via an evaluation of $f(\Sigma_m)$.

By means of LR deflation with 64 deflation vectors, the authors in [23] accomplish a reduction of 4× in the size of the Krylov subspace needed for the evaluation of the sign function times a vector, for a lattice of size 4^4, to attain a relative error of 10^{-10}. In order to do this deflation, though, one has to first compute the m smallest LR eigenmodes which is, as we will explain later in this work, a very expensive task in the particular case of the sign function in lattice QCD.

2.4 Preconditioning the sign function with a polynomial

It has been recently proposed [24] to use polynomial preconditioning for the action of the square root and the inverse square root of a matrix times a vector. We focus on the inverse square root case here. For doing this preconditioning, one constructs first a polynomial $q(A)$ such that

$q(A) \approx A^{-1/2}$. To construct such a polynomial, one can first run an Arnoldi process[1] that builds $p(A) \approx A^{-1}$ [25], and then use the coefficients of $p(A)$ and via divided differences compute the coefficients for $q(A)$; this is one of the approaches followed in [24] to build $q(A)$.

From [24], polynomial preconditioning for the specific problem $f(A)v = A^{-1/2}v$ looks like: $A^{-1/2}v = \left((q(A))^2 A\right)^{-1/2} q(A)v$. Now, our interest here is on describing this method for lattice QCD, where we write $\text{sign}(Q_\mu)v = Q_\mu \left(Q_\mu^2\right)^{-1/2} v$. Then, preconditioning this problem:

$$\text{sign}(Q_\mu)v = Q_\mu \left(Q_\mu^2\right)^{-1/2} v = Q_\mu \left(\left(q\left(Q_\mu^2\right)\right)^2 Q_\mu^2\right)^{-1/2} q\left(Q_\mu^2\right) v. \tag{4}$$

Note how we need to build a polynomial in Q_μ^2 such that $q(Q_\mu^2) \approx \left(Q_\mu^2\right)^{-1/2}$. One then applies the Arnoldi method on the preconditioned problem. The insertion of the polynomial in eq. 4 corresponds to *left preconditioning*. One can also define right preconditioning. Whether one uses left or right preconditioning will depend on considerations such as computational effort and stopping criterium. We prefer left preconditioning here, as it allows one to compute convergence checks more cheaply and simply in general.

Polynomial preconditioning for the evaluation of the sign function on a matrix times a vector has been introduced in [24] and successfully used to drastically reduce the size of the Krylov subspace needed when using the Arnoldi method, with this reducing the dot products and memory footprint and furthermore the total execution time.

3. Combining deflation and preconditioning for the sign function

Now, let us further LR-deflate the preconditioned problem. But we have to be careful at this point: we want to deflate the inverse square root part only, and not the whole sign function application. So, let us first introduce: $\text{invsqrt}(Q_\mu)v := \left(\left(q\left(Q_\mu^2\right)\right)^2 Q_\mu^2\right)^{-1/2} q\left(Q_\mu^2\right) v \Rightarrow \text{sign}(Q_\mu)v = Q_\mu \text{invsqrt}(\mathrm{Q}_\mu)v$. Now we can focus on the inverse square root part: $c := q\left(Q_\mu^2\right) v$, $W := \left(q\left(Q_\mu^2\right)\right)^2 Q_\mu^2 \Rightarrow \text{invsqrt}(\mathrm{Q}_\mu)v = W^{-1/2}c$, $\text{sign}(Q_\mu) = Q_\mu W^{-1/2}c$. The procedure is then: we apply LR deflation to the problem $W^{-1/2}c$, then we multiply from the left by Q_μ, and that gives us the wanted result $\text{sign}(Q_\mu)v$, like this:

$$\text{sign}(Q_\mu)v = Q_\mu R_m \Lambda_m^{-1/2} L_m^H c + Q_\mu W^{-1/2} c_\ominus, \tag{5}$$

where: $c = q\left(Q_\mu^2\right) v$, $W R_m = R_m \Lambda_m$, $L_m^H W = \Lambda_m L_m^H$, $c_\ominus = c - R_m L_m^H c$. One then needs to run the Arnoldi method on the preconditioned and deflated term $W^{-1/2}c_\ominus$.

4. Numerical experiments

In our computations, we use a Dirac matrix for a grid with dimensions 64×32^3 coming from a physically relevant ensemble provided by the lattice QCD group at the University of Regensburg

[1] We do not refer here to the Arnoldi method presented in sect. 2.2, but rather to the underlying Arnoldi iteration used for obtaining V_k and H_{k+1}.

via the Collaborative Research Centre SFB-TRR55, with parameters $m_0 = -0.332159624413$ and $c_{sw} = 1.9192$ [26], and $m_\pi = 0.2946$ GeV. We have set $m_W = -1.4$ and $\mu = 0.3$, which are physically relevant values, for our nonzero chemical potential sign function computations.

All of our implementations have been made available on GitHub[2]. In there, we have implemented the method described in sect. 3, i.e., deflation on top of polynomial preconditioning. The computation of the LR eigenmodes is done via SLEPc [27]. Due to having to make use of a two-sided eigensolver, as we need LR eigenmodes, we are limited by SLEPc to the Krylov-Schur solver [28]. A perhaps better way of doing the LR extraction would be to call an eigensolver that allows for preconditioning, e.g., Jacobi Davidson or Generalized Davidson. As those preconditioned eigensolvers do not allow for two-sided solves in SLEPc, one then needs to pass the operator $W = \left(q\left(Q_\mu^2\right)\right)^2 Q_\mu^2$ to compute the right eigenvectors, and then do a second solve with W^H for the left eigenvectors. Furthermore, those preconditioned eigensolvers rely on a correction equation, with the preconditioner being applied during the approximate solution of this equation. The correction equation is of a shifted form, therefore one has to provide a preconditioner that allows for shifts; one way to alleviate this in the case of a polynomial preconditioner would be to make use of the shift-invariance property of the Arnoldi relation built when constructing a Krylov subspace via the Arnoldi iteration [10]. To simplify our implementation, we have opted for using the two-sided Krylov-Schur solver provided by SLEPc. In using that eigensolver, one has to provide W and W^H. To build a function that applies W^H on a vector, we have made use of the fact that $Q_\mu^H = Q_{-\mu}$, and then $W^H = \left(\bar{q}\left(Q_{-\mu}^2\right)\right)^2 Q_{-\mu}^2$, where $\bar{q}$ indicates that one has to also conjugate the coefficients of the polynomial.

We start then by making some observations regarding the computation of LR eigenmodes. To this purpose, we list in tab. 1 some values related to the extraction of the smallest 20 LR eigenmodes of W for various degrees of the polynomial, where *subsp. size* is the maximum size of the subspace, *restarts* is how many times the eigensolver needed to restart to reach convergence up to 10^{-10}, and *time* is the execution time taken by the eigensolver to converge. We have not presented there the results for the unpreconditioned case, i.e., $d = 0$, because the Krylov subspace needed is too large in our particular case and we did not manage to converge to the desired 20 LR eigenmodes in a reasonable amount of time. On the other hand, in tab. 2 we present the time taken to solve the problem at hand, i.e., $\mathrm{sign}(Q_\mu)v$, via polynomial preconditioning. Considering the small times displayed in tab. 2, we can already note how the non-deflated preconditioned method might not really benefit from deflation, mainly due to the costly eigensolving times. Perhaps if the addition of deflation considerably reduces the time taken to apply $\mathrm{sign}(Q_\mu)v$, and we have to do many solves of the form in eq. 1, then we would benefit from deflation. Or, if we can speedup the eigensolving time via, e.g., preconditioned eigensolvers, then we might see a benefit, as long as the Krylov subspace needed to obtain $\mathrm{sign}(Q_\mu)v$ is considerably reduced when using deflation.

We have run numerical experiments where we make use of deflation on top of polynomial preconditioning. Then, we see that deflation with 256 LR eigenmodes leads to no reduction in the size of the Krylov subspace needed by the preconditioned solver. To try and understand this, we have plotted the spectra until 0.02 of the matrices $W = \left(q\left(Q_\mu^2\right)\right)^2 Q_\mu^2$ for various values of

[2] See https://github.com/Gustavroot/sign_function_LQCD_with_polyprec.

Table 1: cost of computing the 20 smallest LR eigenmodes of the operator $\left(q\left(Q_\mu^2\right)\right)^2 Q_\mu^2$ for various values of the degree d of the polynomial $q\left(Q_\mu^2\right)$. Running on 64 nodes of the JUWELS Cluster, with 48 MPI processes per node. The time is in seconds, and *subsp. size* and *restarts* indicate the maximum size of the subspace in the Krylov-Schur method and the number of restarts until convergence, respectively.

d	subsp. size	restarts	time (s)
128	50	4	59.69
64	50	10	68.26
32	50	21	73.47
16	100	16	86.01
8	100	40	142.00
4	200	41	393.27

Table 2: cost of computing $\text{sign}(Q_\mu)v$ via polynomial preconditioning, without deflation. Running on 64 nodes of the JUWELS Cluster, with 48 MPI processes per node. The times are in seconds, and d the degree of the polynomial. The column *polyn. time* is the time needed to construct the polynomial and *solve time* the time taken to compute $\text{sign}(Q_\mu)v$ up to a relative error of 10^{-9}.

d	polyn. time	solve time	total time
32	0.11	24.97	25.08
64	0.30	14.56	14.86
128	0.95	12.42	13.37

the polynomial degree d, see fig. 1. For small d, there are still many small eigenvalues, hence one might need to deflate many LR small eigenmodes to see any benefit. On the other hand, for larger d the spectra might have already been scattered enough such that the wrapping Arnoldi method manages to interpolate the spectrum of W well enough. The previous explanation might be partially incorrect, though, in the sense that the part of the spectrum of the preconditioned system W that needs to be deflated is not that corresponding to the smallest eigenvalues, but the spectrum might get reshaped due to preconditioning such that the Arnoldi method has a harder time interpolating a different part of the spectrum. This will require further exploration in future work. For now, we conclude that non-deflated polynomial preconditioning is a simpler and cheaper method for computing $\text{sign}(Q_\mu)v$, compared to either deflation alone or preconditioning plus deflation. Moreover, polynomial preconditioning with a large d will exhibit almost perfect strong scaling, assuming that the Dirac operator itself does as well, and also thanks to the drastic reduction in the size of the Krylov subspace, which also keeps the evaluation of $H_k^{-1/2}$ at a very low cost[3].

5. Outlook: a solver for overlap fermions on modern architectures

The linear system in eq. 1 can be solved via *iterative refinement* [29], where one can for example use GMRES in half precision as a preconditioner of FGMRES in double precision. In

[3] The cost of computing $H_k^{-1/2}$ grows as $O(k^3)$, hence it is good to keep the size of the Krylov subspace small to avoid the computation of this kernel becoming a significant part of the total execution time.

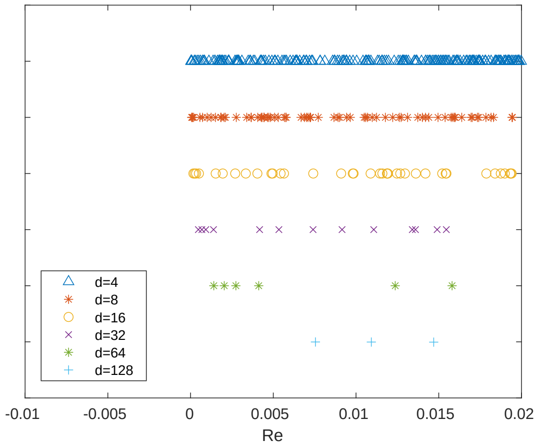

Figure 1: smallest part of the spectrum of the matrix $W = \left(q\left(Q_\mu^2\right)\right)^2 Q_\mu^2$ for various degrees d of the polynomial. The polynomial has collapsed the imaginary part of the eigenvalues of Q_μ^2, rendering the smallest eigenvalues of W virtually real.

doing so, one then has to apply $\mathrm{sign}(Q_\mu)v$ very few times in double precision (thanks to the multigrid preconditioning briefly outlined in sect. 2.1), and most of its applications would then be in the lower precision, e.g., half precision. Then, if using polynomial preconditioning for the evaluation of $\mathrm{sign}(Q_\mu)v$, we would construct the polynomial in double precision once and then cast it down to the lower precision. With this algorithmic arrangement, one would then spend most of the time applying the polynomial $q(Q_\mu^2)$ in, e.g., half precision, as well as applying the multigrid preconditioner also in a reduced precision. Such solver is very well suited for running on modern GPUs, and our aim in future work is to implement such a mixed precision solver within the QUDA [30] library.

Furthermore, an important next step is to build a computationally efficient implementation that compares the recently proposed approach, for evaluating $\mathrm{sign}(Q_\mu)v$ via polynomial preconditinoing, to the one that makes use of a rational approximation (see, e.g., [6]).

Acknowledgments

This work is partially supported by the German Research Foundation (DFG) research unit FOR5269 "Future methods for studying confined gluons in QCD". J.F. has received financial support from the Inno4scale project, which received funding from the European High-Performance Computing Joint Undertaking (JU) under Grant Agreement No. 101118139. The JU receives support from the European Union's Horizon Europe Programme. J.F. acknowledges financial support by the Next Generation Triggers project (https://nextgentriggers.web.cern.ch). G.R-H. acknowledges financial support from the EoCoE-III project, which has received funding from the European High Performance Computing Joint Undertaking under grant agreement No. 101144014. The authors gratefully acknowledge the Gauss Centre for Supercomputing e. V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC), under the project with id MUL-TRA.

References

[1] C. Gattringer and C. Lang, *Quantum chromodynamics on the lattice: an introductory presentation*, **788** (2009) Springer Science & Business Media.

[2] P. H. Ginsparg and K. G. Wilson, *A remnant of chiral symmetry on the lattice*, *Phys. Rev. D* **25** (1982) 2649.

[3] H. Neuberger, *Exactly massless quarks on the lattice*, *Phys. Lett. B* **417** (1998) 141[hep-lat/9707022].

[4] M. A. Stephanov, *QCD phase diagram: An Overview*, *PoS* **LAT2006** (2006) 024 [hep-lat/0701002].

[5] J. Bloch and T. Wettig, *Overlap Dirac operator at nonzero chemical potential and random matrix theory*, *Phys. Rev. Lett.* **97** (2006) 012003 [hep-lat/0604020].

[6] J. van den Eshof, A. Frommer, T. Lippert, K. Schilling, and H. A. van der Vorst, *Numerical methods for the QCD overlap operator. I. Sign-function and error bounds*, *Comput. Phys. Commun.* **146** (2002) 203 [hep-lat/0202025].

[7] G. Arnold, N. Cundy, J. van den Eshof, A. Frommer, S. Krieg, T. Lippert, and K. Schäfer, *Numerical Methods for the QCD Overlap Operator: II. Optimal Krylov Subspace Methods*, in *QCD and Numerical Analysis III: Proceedings of the Third International Workshop on Numerical Analysis and Lattice QCD, Edinburgh June-July 2003*, pp. 153–167, 2005, Springer [hep-lat/0311025].

[8] N. Cundy, J. van den Eshof, A. Frommer, S. Krieg, T. Lippert, and K. Schäfer, *Numerical methods for the QCD overlap operator: III. Nested iterations*, *Comput. Phys. Commun.* **165** (2005) 221 [hep-lat/0405003].

[9] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, *SIAM J. Sci. Stat. Comput.* **7** (1986) 856.

[10] Y. Saad, *Iterative methods for sparse linear systems*, SIAM, 2003.

[11] U. Trottenberg, C. W. Oosterlee, and A. Schuller, *Multigrid*, Elsevier, 2000.

[12] F. Chatelin and W. L. Miranker, *Acceleration by aggregation of successive approximation methods*, *Linear Algebra and its Applications* **43** (1982) 17.

[13] M. Brezina et al, *Adaptive smoothed aggregation (α SA)*, *SIAM J. Sci. Stat. Comput.* **25** (2004) 1896.

[14] M. Lüscher, *Local coherence and deflation of the low quark modes in lattice QCD*, *JHEP* 07 (2007) 081 [0706.2298].

[15] R. Babich et al, *Adaptive multigrid algorithm for the lattice Wilson-Dirac operator*, *Phys. Rev. Lett.* **105** (2010) 201602 [1005.3043].

[16] A. Frommer et al, *Adaptive aggregation-based domain decomposition multigrid for the lattice Wilson–Dirac operator*, *SIAM J. Sci. Stat. Comput.* **36** (2014) A1581 [1303.1377].

[17] J. Brannick et al, *Adaptive multigrid algorithm for lattice QCD*, *Phys. Rev. Lett.* **100** (2008) 041601 [0707.4018].

[18] R. C. Brower et al, *Multigrid for chiral lattice fermions: Domain wall*, *Phys. Rev. D* **102** (2020) 094517 [2004.07732].

[19] J. C. Osborn, R. Babich, J. Brannick, R. C. Brower, M. A. Clark, S. D. Cohen, and C. Rebbi, *Multigrid solver for clover fermions*, *PoS* **LATTICE2010** (2010) 037 [1011.2775].

[20] S. Bacchio, *DDalphaAMG (twisted mass): A parallel implementation of the adaptive multigrid algorithm for domain decomposition with twisted mass fermions*, GitHub repository, 2023, https://github.com/sbacchio/DDalphaAMG.

[21] S. Bacchio, *Simulating maximally twisted mass fermions at the physical point with multigrid methods*, PhD thesis, University of Wuppertal, 2019.

[22] J. Brannick, A. Frommer, K. Kahl, B. Leder, M. Rottmann, and A. Strebel, *Multigrid preconditioning for the overlap operator in lattice QCD*, *Numerische Mathematik* **132** (2016) 463 [1410.7170].

[23] J. Bloch, A. Frommer, B. Lang, and T. Wettig, *An iterative method to compute the sign function of a non-Hermitian matrix and its application to the overlap Dirac operator at nonzero chemical potential*, *Comput. Phys. Commun.* **177** (2007) 933 [0704.3486].

[24] A. Frommer, G. Ramirez-Hidalgo, M. Schweitzer, and M. Tsolakis, *Polynomial preconditioning for the action of the matrix square root and inverse square root*, *Electronic Transactions on Numerical Analysis* **60** (2024) 381 [2401.06684].

[25] J. A. Loe and R. B. Morgan, *Toward efficient polynomial preconditioning for GMRES*, *Numerical Linear Algebra with Applications* **29** (2022) e2427 [1911.07065].

[26] G. S. Bali, S. Collins, B. Gläßle, M. Göckeler, J. Najjar, R. H. Rödl, A. Schäfer, R. W. Schiel, A. Sternbeck, and W. Söldner, *The Moment $\langle x \rangle_{u-d}$ of the Nucleon from $N_f = 2$ lattice QCD down to Nearly Physical Quark Masses*, *Phys. Rev. D* **90** (2014) 074510 [1408.6850].

[27] V. Hernandez, J. E. Roman, and V. Vidal, *SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems*, *ACM Transactions on Mathematical Software (TOMS)* **31** (2005) 351.

[28] Y. Saad, *Numerical Methods for Large Eigenvalue Problems: Revised Edition*, SIAM, 2011.

[29] E. Carson and N. J. Higham, *Accelerating the solution of linear systems by iterative refinement in three precisions*, *SIAM J. Sci. Stat. Comput.* **40** (2018) A817.

[30] M. A. Clark, R. Babich, K. Barros, R. C. Brower, and C. Rebbi, *Solving Lattice QCD systems of equations using mixed precision solvers on GPUs*, *Comput. Phys. Commun.* **181** (2010) 1517 [0911.3191].