A nested Krylov subspace method to compute the sign function of large complex matrices

Jacques C.R. Bloch, Simon Heybrock
Institute for Theoretical Physics, University of Regensburg, 93040 Regensburg, Germany

Abstract

We present an acceleration of the well-established Krylov-Ritz methods to compute the sign function of large complex matrices, as needed in lattice QCD simulations involving the overlap Dirac operator at both zero and nonzero baryon density. Krylov-Ritz methods approximate the sign function using a projection on a Krylov subspace. To achieve a high accuracy this subspace must be taken quite large, which makes the method too costly. The new idea is to make a further projection on an even smaller, nested Krylov subspace. If additionally an intermediate preconditioning step is applied, this projection can be performed without affecting the accuracy of the approximation, and a substantial gain in efficiency is achieved for both Hermitian and non-Hermitian matrices. The numerical efficiency of the method is demonstrated on lattice configurations of sizes ranging from $4^4$ to $10^4$, and the new results are compared with those obtained with rational approximation methods.

1. Introduction

In the study of quantum chromodynamics on the lattice it is essential to discretize the Dirac operator such that it respects chiral symmetry. This is most faithfully achieved using the overlap Dirac operator \[1, 2\], whose use is especially favored in studies relying on the chiral properties of the theory. To study QCD at nonzero baryon density the overlap formulation was recently extended to include a quark chemical potential \[3, 4\]. A major ingredient in the overlap operator, which makes its use very challenging, is the computation of the sign function of a complex matrix, which is Hermitian at zero baryon density, but becomes non-Hermitian when a quark chemical potential is introduced.

The search for efficient numerical methods to compute the sign function for the large sparse matrices encountered in this context is an ongoing field of research. Typically, Krylov subspace methods are employed to evaluate the operation of a matrix function on an arbitrary vector. We distinguish two main variants: the Krylov-Ritz approximation, which we discuss in this paper, is based on the construction of a Krylov basis and its accompanying Ritz matrix. Depending on the algorithm used to construct the basis we distinguish between the Lanczos approximation in the Hermitian case \[6\], and the Arnoldi approximation \[11\] or two-sided Lanczos

Supported by the DFG collaborative research center SFB/TR-55 “Hadron Physics from Lattice QCD”.

Preprint submitted to Elsevier

December 22, 2009
approximation \cite{12} in the non-Hermitian case. The latter clearly yields the more efficient function approximation for non-Hermitian matrices \cite{12}. In the Krylov-Ritz approximation the large complex matrix is projected on the Krylov subspace, and its sign function is approximated by lifting the sign function of its projected image (Ritz matrix) back to the original space. The latter sign function is computed to high accuracy using the spectral definition of a matrix function or using a matrix-iterative method. When a large Krylov subspace is needed to reach the desired accuracy, the computation of this matrix sign function becomes a bottle neck for the algorithm.

Herein we will introduce an enhancement of the Krylov-Ritz approximation method which substantially reduces the cost of this internal sign computation and boosts the efficiency of the overall method, such that it competes with, and even surpasses, the rational function approximation in both the Hermitian and non-Hermitian case. The dramatic reduction in computation time is achieved by projecting the Ritz matrix on an even smaller, nested Krylov subspace, after performing a suitable preconditioning step first. The desired sign function is then computed via the sign function of the inner Ritz matrix, which yields the same accuracy as the original Krylov-Ritz approximation.

The outline of the paper is as follows. In Sec. 2 we introduce the overlap operator and the matrix sign function. In Sec. 3 we show how the matrix function of large matrices is computed using Krylov-Ritz approximation methods. In Sec. 4 we introduce the nested Krylov subspace method, which substantially enhances the efficiency of the Krylov-Ritz approximation to the sign function. We study its convergence properties and present numerical results for various lattice sizes, including a comparison with rational approximation methods. Finally, our conclusions are given in Sec. 5. For completeness we have added some algorithms in Appendix.

2. Overlap operator and the matrix sign function

Our motivation to develop numerical algorithms to compute the matrix sign function of large, sparse, complex matrices comes from its application in lattice quantum chromodynamics (LQCD). The overlap formulation of the Dirac operator \cite{1} \cite{2}, which ensures that chiral symmetry is preserved in LQCD, is given in terms of the matrix sign function \cite{3}, and its definition in the presence of a quark chemical potential \( \mu \) is given by

\[
D_{ov}(\mu) = \mathbb{1} + \gamma_5 \text{sgn}(\gamma_5 D_w(\mu)),
\]

where \( \gamma_5 = \gamma_1\gamma_2\gamma_3\gamma_4 \) with \( \gamma_1, \ldots, \gamma_4 \) the Dirac gamma matrices in Euclidean space, sgn is the matrix sign function, and

\[
D_w(\mu) = \mathbb{1} - \kappa \sum_{i=1}^{3} (T_i^+ + T_i^-) - \kappa(e^{\mu T_4^+} + e^{-\mu T_4^-})
\]

is the Wilson Dirac operator at nonzero chemical potential \cite{14} with \( (T_i^\pm)_{\lambda\kappa} = (\mathbb{1} \pm \gamma_i)U_{\lambda\nu\kappa\sigma\eta}U_\lambda^\dagger \delta_\nu\sigma\eta, \kappa = 1/(8 + 2m_u) \), \( m_u \in (-2, 0) \) and \( U_{\lambda\nu\kappa\sigma\eta} \in \text{SU}(3) \), where \( U_{\lambda\nu\kappa\sigma\eta} = U_{\lambda\nu\kappa\sigma\eta}^\dagger \). The exponential factors \( e^{\mu T_4} \) implement the quark chemical potential on the lattice. For \( \mu = 0 \) the argument of the sign function is Hermitian, while for \( \mu \neq 0 \) it is non-Hermitian. To compute the overlap operator we need to define the matrix sign function for a general complex matrix \( A \) of dimension \( n \). A generic matrix function \( f(A) \) can be defined by

\[
f(A) = \frac{1}{2\pi i} \oint_{\Gamma} f(z)(zI - A)^{-1} \, dz,
\]

where \( \Gamma \) is a collection of contours in \( \mathbb{C} \) such that \( f \) is analytic inside and on \( \Gamma \) and such that \( \Gamma \) encloses the spectrum of \( A \). If \( A \) is diagonalizable, i.e., \( A = UAU^{-1} \), with diagonal eigenvalue matrix \( \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n) \) and \( U \in \text{Gl}(n, \mathbb{C}) \), then this general definition can be simplified to the well-known spectral form

\[
f(A) = U f(\Lambda) U^{-1},
\]

with

\[
f(\Lambda) = \text{diag}(f(\lambda_1), \ldots, f(\lambda_n)).
\]
Arnoldi algorithm, which suffers from long recurrences as each basis vector has to be orthogonalized with respect to the first column of the matrix. The method generates an orthonormal basis and a tridiagonal symmetric matrix approximation (10) we do not need to perform the matrix multiplications of Eq. (9) explicitly. First, one computes the vector in the Krylov subspace \( y = f(A)x \), where the \( v_i \) form an orthonormal basis of \( \mathcal{K}_k(A, x) \), which is defined by the matrix \( H_k = V_k^* A V_k \), which is often referred to as Ritz matrix. The components of \( H_k \) are the projection coefficients of \( A_k \) in the basis \( V_k \), as \( A_k \) and \( H_k \) are related by \( A_k = V_k H_k V_k^* \) (in analogy to the vector case).

The Krylov-Ritz approximation [16, 17] to \( f(A) \) consists in taking the function of the Ritz matrix \( H_k \) and lifting it back to the full \( n \)-dimensional space:

\[
    f(A) \approx V_k f(H_k) V_k^*.
\]

This approximation actually replaces the polynomial interpolating \( f \) at the eigenvalues of \( A_k \), also called Ritz values [17]. Substituting the approximation (9) in \( f(A) \) yields

\[
    y = V_k f(H_k) V_k^* x = |x| V_k f(H_k) e_1^{(k)},
\]

where we chose \( v_1 \) collinear with \( x \), i.e., \( v_1 = V_k e_1^{(k)} \equiv x/|x| \), with \( e_1^{(k)} \) the first unit vector of \( \mathbb{C}^k \). To evaluate the approximation (10) we do not need to perform the matrix multiplications of Eq. (9) explicitly. First, one computes the function \( f(H_k) \) of the \( k \)-dimensional Ritz matrix to high accuracy, using the spectral definition (4) or a matrix-iterative method. Then, the final approximation is simply a linear combination of the basis vectors \( v_i \), with coefficients given by the first column of \( f(H_k) \) multiplied with \( |x| \).

The Krylov-Ritz approximation described above uses an orthonormal basis of \( \mathcal{K}_k(A, x) \). For the Hermitian case such a basis can efficiently be constructed using the Lanczos algorithm, which we listed in Appendix A for completeness. It generates an orthonormal basis and a tridiagonal symmetric \( H_k \) using a three-term recurrence relation. The non-Hermitian case is more laborious as the construction of an orthonormal basis is typically performed using the Arnoldi algorithm, which suffers from long recurrences as each basis vector has to be orthogonalized with respect to the previous one.
to all the previous ones. The two-sided Lanczos algorithm is a suitable alternative \cite{12} which uses two three-term recurrences to construct bases $V_k = (v_1, \ldots, v_k)$ and $W_k = (w_1, \ldots, w_k)$ of the right, respectively left, Krylov subspaces $K_k(A, x)$ and $K_k(A^T, x)$, which are biorthonormal, i.e., $v_j w_i = \delta_{ij}$ (see Appendix B for a listing of the algorithm). The lack of orthogonality of the basis $V_k$ prevents the construction of the orthogonal projector needed for the Krylov-Ritz function approximation \cite{9}. Nevertheless, the biorthonormality between $V_k$ and $W_k$ can be used to construct an oblique projector $P = V_k W_k^\dagger$ on the right Krylov subspace. The oblique projection of $A$ is $A_k = PAP$ and its $k$-dimensional image is defined by $H_k = W_k^\dagger A V_k$, which we call two-sided Ritz matrix, such that $A_k = V_k H_k W_k^\dagger$. The matrix $H_k$ generated by the two-sided Lanczos algorithm is tridiagonal. The two-sided Krylov-Ritz approximation to $f(A)$ then consists in taking the matrix function of $H_k$ and lifting it back to the original space,

$$f(A) \approx V_k f(H_k) W_k^\dagger.$$ 

(11)

After applying this approximation of $f(A)$ to $x$ we find an expression which is similar to Eq. \eqref{10}.

$$y = V_k f(H_k) W_k^\dagger x = |x| V_k f(H_k) e_k^{(i)}.$$  

(12)

where the last step assumes that $v_1 = V e_k^{(i)} = x/|x|$. The price payed to achieve short recurrences in the non-Hermitian case is the loss of orthogonality of the projection on the Krylov subspace, which translates in a somewhat lower accuracy of the two-sided Lanczos approximation compared to the Arnoldi approximation, for equal Krylov subspace sizes. Nevertheless, the large gain in speed makes it by far the more efficient method \cite{12}.

In the case where $f$ is the sign function, the approximations \eqref{10} and \eqref{12} require the computation of $\text{sgn}(H_k)$. Although it could be computed directly with the spectral definition \eqref{4}, matrix-iterative methods are often cheaper for medium sized matrices. We choose to employ the Roberts-Higham iteration (RHi) \cite{13}: Set $S_0 = H_k$ and compute

$$S_{n+1} = \frac{1}{2}(S_n + S_n^{-1}).$$ \hspace{1cm} (13)

This iteration converges quadratically to $\text{sgn}(H_k)$, if the sign function for complex arguments is defined by Eq. \eqref{9}. The matrix inversion scales like $k^3$ and so will the RHi. For the QCD application considered here, typically 7 to 10 iterations are necessary to converge within machine precision \cite{11,12}.

The hope is that the Krylov-Ritz approximations \eqref{10} and \eqref{12} are accurate for $k \ll n$. The method is known to work very well as long as no eigenvalues are close to a function discontinuity. However, for the sign function this method suffers from the sign discontinuity along the imaginary axis. If $A$ has eigenvalues close to this discontinuity the approximating polynomial must steeply change from $-1$ to $+1$ over a small interval to give an accurate approximation. This cannot be achieved with a low order polynomial, i.e., the Krylov subspace must be large, which makes the algorithm expensive. The common solution to this problem is to use deflation, where the contribution of the eigenvalues with smallest modulus to the sign function is computed exactly. The Krylov subspace approximation is then performed in a deflated space, i.e., the subspace where the directions along the critical eigenvectors have been removed. We refer to the literature for details \cite{11}.

The convergence of the Krylov-Ritz approximations to the matrix sign function is illustrated in Fig.\ref{fig1} the Lanczos approximation for the Hermitian case on the left, and the two-sided Lanczos approximation for the non-Hermitian case on the right. As expected, the accuracy improves with increasing Krylov subspace size $k$, and a larger deflation gap $\Delta$, corresponding to a higher number of deflated eigenvectors, leads to a faster convergence. For a given accuracy and equal deflation gap, the subspace size $k$ required for non-Hermitian $A$ is larger than for Hermitian $A$. The error in these and all following figures is given by $\varepsilon = |(\text{sgn}(A))^2 x - x|/2|x|$. This error determination proved to be consistent with the error obtained by comparing to the exact solution for $4^4$ and $6^4$ lattices, and will therefore be used for all lattice sizes.

To analyze the efficiency of the algorithm we briefly sketch the three major contributions to the total CPU time. For each matrix $A$ the deflation requires the computation of the critical eigenvalues and the corresponding eigenvectors. The time needed by the rest of the algorithm strongly depends on the eigenvalue gap, as the Krylov subspace size can be reduced if the deflation gap is increased. As mentioned at the beginning of this section, the product $f(A)x$ is usually needed for many source vectors $x$, e.g., as part of an iterative inversion. In this case the expensive deflation of $A$ only needs to be performed once in an initialization step, while the Krylov subspace part of the algorithm will be repeated
Figure 1: Accuracy of the Krylov subspace approximation for $y = \text{sgn}(A)x$, where $A$ is $\gamma_5 D_w(\mu)$ for a 64 lattice (for a lattice volume $V$ the matrix $\gamma_5 D_w$ has dimension 12V, such that dim($A$) = 15552 here). Left pane: Hermitian case ($\mu = 0$) using the Lanczos method, right pane: non-Hermitian case with chemical potential $\mu = 0.3$ using the two-sided Lanczos method. The relative error $\varepsilon$ is shown as a function of the Krylov subspace size $k$ for different deflation gaps $\Delta$ (given in parenthesis). The deflation gap is defined as the smallest distance between the origin and the spectrum of $A$.

Figure 2: CPU time $t$ (in seconds) versus accuracy for an 84 lattice configuration in the Hermitian case with deflation gap $\Delta = 0.055$ (left) and the non-Hermitian case with $\mu = 0.3$ and deflation gap $\Delta = 0.107$ (right). The full line shows the total time required to compute $\text{sgn}(A)x$, while the dashed line gives the time needed to construct the Krylov basis. The difference between both lines represents the time taken by the RHi to compute $\text{sgn}(H_k)$. The irregular convergence pattern for the non-Hermitian case is a well-known feature of the two-sided Lanczos algorithm.

for each new vector $x$. For this reason we assume from now on that an initial deflation has been performed and we will concentrate on the efficiency of the Krylov subspace part of the algorithm. We discern two main components in the Krylov-Ritz method: the construction of the Krylov basis using the Lanczos or two-sided Lanczos algorithms, where the computation time grows linearly with the subspace size $k$, and the RHi to compute $\text{sgn}(H_k)$, which scales as $k^3$. Figure 2 illustrates these last two contributions. For high accuracy the Krylov subspace becomes large such that the cost of the RHi dominates the total CPU time of the Krylov-Ritz approximation and the method becomes too costly. In the following, the implementation of the Krylov-Ritz approximation for which $\text{sgn}(H_k)$ is computed using Eq. (13) will be referred to as non-nested method. In the next section we will present a nested Krylov subspace method, which drastically reduces the cost to compute $\text{sgn}(H_k)k_1^{(k)}$ and vastly improves the overall efficiency of the Krylov-Ritz approximation.
4. Nested Krylov subspace method for the sign function

4.1. Nesting and preconditioning

We introduce a new method which speeds up the expensive computation of the vector \( \text{sgn}(H_\ell e_1^{(k)}) \) required in the Krylov-Ritz approximations (10) and (12) to \( \text{sgn}(A)x \). The idea is to approximate this matrix-vector product by a further Krylov-Ritz approximation, using a second, nested Krylov subspace (specified below) of size \( \ell \ll k \), i.e.,

\[
\text{sgn}(H_\ell e_1^{(k)}) = V_\ell \text{sgn}(H_\ell e_1^{(f)}),
\]

where \( V_\ell \) is the matrix containing the basis vectors of the inner Krylov subspace, constructed with the Lanczos or two-sided Lanczos method, and \( H_\ell \) is the inner Ritz or two-sided Ritz matrix. The \( \text{sgn}(H_\ell) \) is computed using the RHk on the inner Ritz matrix \( H_\ell \). After substituting this result in Eq. (10) and (12) we get the nested approximation

\[
y \approx |x| V_\ell V_\ell \text{sgn}(H_\ell e_1^{(f)})
\]
to \( \text{sgn}(A)x \). By introducing an additional Krylov subspace, the number of operations necessary to compute \( \text{sgn}(H_\ell e_1^{(k)}) \) is reduced from \( O(k^3) \) in the non-nested method to \( O(\ell^3) + O(k\ell) \). If \( \ell \ll k \) this will very much improve the efficiency of the Krylov-Ritz approximation.

The obvious choice for the inner Krylov subspace is \( \mathcal{K}_k(H_k, e_1^{(k)}) \). However, it is easy to see that approximations in this Krylov subspace will not improve the efficiency of the method. The Ritz matrix \( H_\ell \) of the Krylov subspace \( \mathcal{K}_k(H_k, e_1^{(k)}) \) will only contain information coming from the \( \ell \times \ell \) upper left corner of \( H_k \), because of the tridiagonal nature of \( H_k \) and the sparseness of the source vector \( e_1^{(k)} \). This will effectively cut down the size of the outer Krylov subspace from \( k \) to \( \ell \), which will substantially worsen the accuracy of the approximation if \( \ell \) is chosen much smaller than \( k \). Nonetheless, the nested Krylov subspace method can be made to work efficiently if we perform an initial preconditioning step on the tridiagonal Ritz matrix, replacing \( H_k \) by

\[
H_k \rightarrow H'_k = \frac{1}{2} \left[ p H_k + (pH_k)^{-1} \right],
\]

with \( p \) a positive real number, and construct the approximation to \( \text{sgn}(H_k e_1^{(k)}) \) in the Krylov subspace \( \mathcal{K}_k(H'_k, e_1^{(k)}) \). This alternate Krylov subspace can be used to compute \( \text{sgn}(H_k e_1^{(k)}) \) because the preconditioning step leaves the sign unchanged. Moreover, as it greatly improves the condition number of the matrix, we expect a large gain in efficiency for the inner Krylov-Ritz approximation. To show that both matrices have the same sign we note that they have identical eigenvectors, as a matrix and its inverse share the same eigenvectors, and that the sign of their eigenvalues satisfies

\[
\text{sgn} \left( \frac{1}{2} \left( \frac{1}{p} + \frac{1}{p^*} \right) \right) = \text{sgn} \text{Re} \left( \frac{1}{p} + \frac{1}{p^*} \right) = \text{sgn} \text{Re} \left( \frac{1}{p} + \frac{1}{|p|^2} \right) = \left( 1 + \frac{1}{|p|^2} \right) \text{sgn} \text{Re}(p) = \text{sgn}(z),
\]

where we used the definition (6), and hence \( \text{sgn}(H'_k) = \text{sgn}(H_k) \) according to Eq. (4).

The transformation induced by the preconditioning step is illustrated in Fig. 3 for real positive eigenvalues (for negative values the graph would be projected with respect to the origin). The factor \( p \) is chosen to optimize the effect of the transformation on the condition number. We examine the condition number for a real valued spectrum, assuming that the spectral support of \( H_k \) is similar to that of the original matrix \( A \), after deflation. As can be been from Fig. 3, after transformation the smallest eigenvalue (in absolute value) is \( z'_\min = 1 \), while the largest will be given by the transform of either the smallest or largest eigenvalues of \( H_k \). The smallest condition number will be achieved when both values are identical, i.e., for \( p \) satisfying\footnote{The factor 1/2 is chosen for convenience. For \( p = 1 \) the transformation actually mimics the first step of the RHk (13).}

\[
\frac{1}{2} \left( \frac{1}{p_{\min}} + \frac{1}{p_{\max}} \right) = \frac{1}{2} \left( \frac{p_{\max}}{p_{\min}} + \frac{1}{p_{\max}} \right) \Rightarrow \quad p_{\text{opt}} = \sqrt{\frac{1}{z_{\min} z_{\max}},}
\]

After substituting this result in Eq. (10) and (12) we get the nested approximation

\[
y \approx |x| V_\ell V_\ell \text{sgn}(H_\ell e_1^{(f)})
\]
to \( \text{sgn}(A)x \). By introducing an additional Krylov subspace, the number of operations necessary to compute \( \text{sgn}(H_\ell e_1^{(k)}) \) is reduced from \( O(k^3) \) in the non-nested method to \( O(\ell^3) + O(k\ell) \). If \( \ell \ll k \) this will very much improve the efficiency of the Krylov-Ritz approximation.

The obvious choice for the inner Krylov subspace is \( \mathcal{K}_k(H_k, e_1^{(k)}) \). However, it is easy to see that approximations in this Krylov subspace will not improve the efficiency of the method. The Ritz matrix \( H_\ell \) of the Krylov subspace \( \mathcal{K}_k(H_k, e_1^{(k)}) \) will only contain information coming from the \( \ell \times \ell \) upper left corner of \( H_k \), because of the tridiagonal nature of \( H_k \) and the sparseness of the source vector \( e_1^{(k)} \). This will effectively cut down the size of the outer Krylov subspace from \( k \) to \( \ell \), which will substantially worsen the accuracy of the approximation if \( \ell \) is chosen much smaller than \( k \). Nonetheless, the nested Krylov subspace method can be made to work efficiently if we perform an initial preconditioning step on the tridiagonal Ritz matrix, replacing \( H_k \) by

\[
H_k \rightarrow H'_k = \frac{1}{2} \left[ p H_k + (pH_k)^{-1} \right],
\]

with \( p \) a positive real number, and construct the approximation to \( \text{sgn}(H_k e_1^{(k)}) \) in the Krylov subspace \( \mathcal{K}_k(H'_k, e_1^{(k)}) \). This alternate Krylov subspace can be used to compute \( \text{sgn}(H_k e_1^{(k)}) \) because the preconditioning step leaves the sign unchanged. Moreover, as it greatly improves the condition number of the matrix, we expect a large gain in efficiency for the inner Krylov-Ritz approximation. To show that both matrices have the same sign we note that they have identical eigenvectors, as a matrix and its inverse share the same eigenvectors, and that the sign of their eigenvalues satisfies

\[
\text{sgn} \left( \frac{1}{2} \left( \frac{1}{p} + \frac{1}{p^*} \right) \right) = \text{sgn} \text{Re} \left( \frac{1}{p} + \frac{1}{p^*} \right) = \text{sgn} \text{Re} \left( \frac{1}{p} + \frac{1}{|p|^2} \right) = \left( 1 + \frac{1}{|p|^2} \right) \text{sgn} \text{Re}(p) = \text{sgn}(z),
\]

where we used the definition (6), and hence \( \text{sgn}(H'_k) = \text{sgn}(H_k) \) according to Eq. (4).

The transformation induced by the preconditioning step is illustrated in Fig. 3 for real positive eigenvalues (for negative values the graph would be projected with respect to the origin). The factor \( p \) is chosen to optimize the effect of the transformation on the condition number. We examine the condition number for a real valued spectrum, assuming that the spectral support of \( H_k \) is similar to that of the original matrix \( A \), after deflation. As can be seen from Fig. 3, after transformation the smallest eigenvalue (in absolute value) is \( z'_\min = 1 \), while the largest will be given by the transform of either the smallest or largest eigenvalues of \( H_k \). The smallest condition number will be achieved when both values are identical, i.e., for \( p \) satisfying\footnote{In practice \( p_{\text{opt}} \) is only known approximately, as it is computed from spectral information of \( A \) instead of \( H_k \). However, this has no significant impact on the performance of the nested method.}

\[
\frac{1}{2} \left( \frac{1}{p_{\min}} + \frac{1}{p_{\max}} \right) = \frac{1}{2} \left( \frac{p_{\max}}{p_{\min}} + \frac{1}{p_{\max}} \right) \Rightarrow \quad p_{\text{opt}} = \sqrt{\frac{1}{z_{\min} z_{\max}},}
\]
The transformation (16) therefore reduces the condition number $C$ by a factor

$$F = \frac{C}{C'} = \frac{z_{\text{max}}}{z_{\text{min}}} \left( \frac{1}{2} \sqrt{\frac{z_{\text{max}}}{z_{\text{min}}} + \frac{z_{\text{min}}}{z_{\text{max}}}} \right) \approx 2 \sqrt{\frac{z_{\text{max}}}{z_{\text{min}}}}. \quad (20)$$

At nonzero baryon density the spectrum of $\gamma_5 D_w$, after deflation, is contained in ellipses $E(-m, a, b) \cup E(+m, a, b)$, with center $m \in \mathbb{R}^+$ and major and minor axes $a$ and $b$ along the real and imaginary axes, respectively. One can show that the optimal $p$ is still given by Eq. (18) for such elliptic spectra, even though the improvement in condition number becomes less pronounced as the ellipse grows in the complex plane. Typically the ellipses in LQCD are quite narrow and the gain in condition number will be very close to that of the Hermitian case.

The effect of the preconditioning of the Ritz matrix is illustrated in Fig. 4 for the Hermitian case. The top and bottom graphs depict the spectra of $H_k$ and $H'_k$, respectively. The spectrum of the original Ritz matrix has only a small gap at zero, while the gap for the transformed matrix is large. In this example, the condition number is almost improved by a factor 20. In general, the value of $z_{\text{max}}$ for $\gamma_5 D_w$ varies only slightly with the choice of the simulation parameters and $F$ will mainly depend on the deflation gap. The picture remains essentially the same for the non-Hermitian matrices occurring at nonzero density.

As we will see below the reduction of the condition number significantly speeds up the Krylov-Ritz approximation.

4.2. Convergence

In this section we investigate the convergence properties of the nested method. The method was implemented to compute the sign function of $\gamma_5 D_w(\mu)$ needed by the overlap operator (1), for both the Hermitian and the non-Hermitian case. Whenever the matrix has eigenvalues close to the imaginary axis, these critical eigenvalues are first deflated to open up a deflation gap, necessary to keep the Krylov subspace within a reasonable size (see Sec. (3)). Our implementation uses Chroma [19] to compute the Wilson operator. The linear algebra is performed with BLAS and LAPACK routines. To ensure the efficiency of the nested method a judicious implementation of the preconditioning step (16), used to construct the inner Krylov subspace, is needed. Explicitly inverting the tridiagonal matrix $H_k$ to form the full matrix $H'_k$, then constructing the basis of the inner Krylov subspace by successive full matrix-vector multiplications would make a rather inefficient algorithm. To construct the inner Krylov subspace we do not need to

\[ \text{This is due to } z'_{\text{min}} \text{ moving into the complex plane and becoming smaller than one.} \]
Figure 4: Upper pane: Density of eigenvalues of $H_k$ in the Hermitian case for an $8^4$ lattice with $k = 1536$. The spectrum has a narrow deflation gap $\Delta = 0.055$. The optimal $p$-factor (19) for the transformation (16) is $p_{\text{opt}} = 1.86$ (using $z_{\text{min}} = \Delta$ and $z_{\text{max}} = 5.26$). The lower pane shows the corresponding eigenvalue density of the transformed matrix $H'_k$, where the condition number is improved by a factor $\mathcal{F} = 19.3$ (see Eq. (20)).

construct the full matrix $H'_k$ explicitly, but only have to apply $H'_k$ to $\ell - 1$ vectors of $C^k$ (in the non-Hermitian case $H'_k^\dagger$ is also needed). Those products are best computed using the $LU$ decomposition of $H_k$, which is especially efficient for tridiagonal matrices. A detailed listing of the algorithm is given in Appendix C.

The overall accuracy of the nested approximation (15) depends on the parameters $k$ and $\ell$, defining the sizes of the outer and inner Krylov subspaces, respectively. Clearly the total error cannot drop below the accuracy $\varepsilon_k$ of the non-nested method with Krylov subspace size $k$, since the solution of the nested method is constructed in that same space and converges to the solution of the former when $\ell \to k$. To investigate the accuracy of the nested algorithm, our strategy is to fix the outer Krylov subspace size $k$, corresponding to a certain desired accuracy, and vary the inner Krylov subspace size $\ell$. We show the convergence results for an $8^4$ lattice configuration in Fig. 5, for both the Hermitian and non-Hermitian case. As expected the nested method reaches the accuracy of the non-nested method when its size is large enough. Surprisingly however, this happens for $\ell \ll k$, as the convergence of the inner Ritz approximation seems to be extremely fast. The smallest value of $\ell$ for which optimal convergence is reached will be called $\ell_{\text{opt}}$. The fast convergence is closely related to the large improvement in condition number discussed in the previous section. We also showed in Eq. (20) how the improvement of the condition number, due to the preconditioning of the Ritz matrix $H_k$, depends on the deflation gap. A smaller gap will yield a larger improvement, and vice-versa. This in turn will influence the convergence rate of the nested method. Figure 6 verifies that the result $\ell_{\text{opt}} \ll k$ remains valid for different deflation gaps. The figure also illustrates that the somewhat larger reduction in condition number achieved for a smaller gap yields an accordingly smaller ratio $\ell_{\text{opt}}/k$ (approximately proportional to the ratio of the respective improvement factors $\mathcal{F}$). This is an additional advantage as the size reduction is largest when the outer subspace is large. In all cases, the inner Krylov subspace can be taken much smaller than the outer subspace, such that the efficiency of the Krylov-Ritz method is substantially boosted, as will be shown in the benchmarks below.

We also verified that the convergence curves are fairly insensitive to the choice of the source vector and lattice configuration. The fast convergence property of the nested method is generic, regardless of the simulation details, for both the Hermitian and non-Hermitian case, even though the precise value of $\ell_{\text{opt}}$ depends on the lattice size, the simulation parameters, the deflation gap and the desired overall accuracy (determined by $k$).

4.3. Benchmarks

With the fast convergence ($\ell_{\text{opt}} \ll k$) discussed in the previous section, we can expect a substantial gain in computation time when using the nested method. The total CPU time consumed by the nested method is illustrated in Fig. 7 for the Hermitian case (left) and the non-Hermitian case (right). The size of the outer Krylov subspace is kept
CPU time needed for the computation of \( \text{sgn}(H_f) \) is completely dominated by the cost for building the basis in that subspace. The nested method is able to quench the error in this range is the minimal error achievable with the given size of the outer Krylov subspace, while the run time is no need to make a compromise between run time and accuracy, as both can be optimized simultaneously. The error with the RHi is negligible compared to the time required to construct the outer Krylov subspace. There is therefore no instance with \( \ell \) for which optimal convergence is reached is denoted by \( \ell_{opt} \). Note that we always restrict ourselves to even Krylov subspace sizes, as odd values systematically give a somewhat worse accuracy because of spurious near-zero eigenvalues occurring in the Ritz matrix.

Fixed, such that its construction gives a constant contribution to the run time, depicted by the horizontal dashed line. The contribution to the CPU time which varies with \( \ell \) mainly comes from the computation of \( \text{sgn}(H_f) \) with the RHi and is proportional to \( \ell^3 \). For \( \ell \approx k \) the total run time of the nested method is about equal to that of the non-nested method. However, as illustrated by the \( \epsilon(\ell) \) curve (red line) and discussed in Sec. 4.2, \( \ell \) can be chosen much smaller while preserving the accuracy of the non-nested method. The central result, illustrated by the vertical band in Fig. 7, is that there exists an interval in \( \ell \) for which the accuracy is still optimal, but the CPU time needed to compute \( \text{sgn}(H_f) \) with the RHi is negligible compared to the time required to construct the outer Krylov subspace. There is therefore no need to make a compromise between run time and accuracy, as both can be optimized simultaneously. The error in this range is the minimal error achievable with the given size of the outer Krylov subspace, while the run time is completely dominated by the cost for building the basis in that subspace. The nested method is able to quench the CPU time needed for the computation of \( \text{sgn}(H_f)\epsilon_1^{(k)} \) without affecting the accuracy of the Krylov-Ritz approximation.
4.4. Note on the memory usage

In the numerical tests we observed that, for a fixed deflation gap, the Krylov subspace size needed to achieve a certain accuracy is almost independent of the lattice volume in the Lanczos approximation and only grows slowly with the volume in the two-sided Lanczos approximation. Therefore, the memory consumed by the Krylov basis $V_k$ is roughly proportional to the lattice volume. For large lattice sizes this storage requirement might become too large to run the Krylov-Ritz approximation on a single node.

One solution, which only requires little storage, is to implement a double-pass version of the algorithm, which is possible due to the use of short recurrences. In double-pass mode only the two most recently generated basis vectors are stored during the construction of the outer Krylov subspace basis. In the first pass the matrix $H_k$ is built and the product $\text{sgn}(H_k)e_0(t)$ is computed with Eq. (14). In the second pass the basis vectors of the outer Krylov subspace are generated again and immediately added in a linear combination, whose coefficients were computed in the first pass. The drawback of this variant is that the Krylov basis is constructed twice, such that the corresponding CPU time will be doubled.

The more efficient solution is to parallelize the single-pass version of the algorithm, such that the memory requirement gets distributed over the available nodes. Benchmarks on larger volumes, using such a parallel implementation, are currently being performed.

Figure 7: Error and CPU usage of the nested method for lattice size $8^3$. Hermitian case with deflation gap $\Delta = 0.055$ and fixed $k = 1536$ (left), and non-Hermitian case with $\mu = 0.3$, $\Delta = 0.107$ and $k = 1580$ (right). $\varepsilon(\ell)$ shows the accuracy versus inner Krylov subspace size $\ell$ (red line). $\eta(\ell)$ shows the total CPU time in seconds (solid blue line), while the horizontal dashed line measures the time needed to construct the basis in the outer Krylov subspace. The difference between both lines corresponds to the time taken by the RHi to compute $\text{sgn}(H_\ell)$. The vertical band highlights the operational window of the nested method, i.e., the region in $\ell$ where the accuracy is optimal, but the CPU-time used to compute $\text{sgn}(H_\ell)$ is negligible.
Figure 8: Comparison of the nested Krylov subspace method (filled circles) with rational approximation methods (full lines) for lattices of sizes $4^4$, $6^4$, $8^4$ and $10^4$ for two different deflation gaps (given in parenthesis). Left: Hermitian case comparing the nested Lanczos approximation (Lanc) with the Zolotarev approximation (Zolo), evaluated using the Chroma QCD library. Right: non-Hermitian case with \( \mu = 0.3 \) comparing the nested two-sided Lanczos method (2sL) with the Neuberger approximation evaluated with a restarted FOM algorithm (rFOM). The timings were measured on a single 2.4 GHz Intel Core 2 core with 8 GB of memory.
4.5. Multi-level nesting

In principle, if the inner Krylov subspace in Eq. (15) is still too large for an efficient application of the RHi on the inner Ritz matrix, the nested method could be applied recursively. In this case we rename $k$ to $k_0$, $\ell$ to $k_1$, and add more recursively nested levels $k_i$ as necessary. Except for the deepest level, the matrix-vector product $\text{sgn}(H_k) e^{(k)}$ required at level $i$ will be computed with a Krylov-Ritz approximation (14) in the nested Krylov subspace $\mathcal{K}_{k_i}(H'_{k_i}, e^{(k_i)})$, where $H'_{k_i}$ is defined by Eq. (16) on $H_{k_i}$ and typically $k_{i+1} \ll k_i$. At the deepest level the sign function of the Ritz matrix will be evaluated with the RHi. This multi-level nesting is illustrated in Fig. 10 where we show the convergence curves for 1, 2, 3, 4 and 5 nested levels as a function of the size of the innermost Krylov subspace, with the sizes of all outer levels kept fixed to some value inside their convergence region (the convergence curves do not depend on the precise choice of the outer $k_i$’s). As before, the convergence criterion is set by the size $k_0$ of the outer Krylov subspace. Each additional level lowers the size of the Krylov subspace. In the case depicted in Fig. 10 the optimal Krylov subspace sizes, i.e. where convergence is reached, for the successive levels decreases from $1536 \to 90 \to 20 \to 8 \to 4 \to 2$. The improvement is most dramatic for the first nested level, but fast convergence is exhibited at all levels\(^4\). This can be related to the quadratically convergent RHi, as the preconditioning step at each level mimics a step of the RHi and compresses the spectrum more and more towards $\pm1$. Moreover, the judicious choice of $p$ at each level improves the convergence even more. It is intriguing to note that, in the example of Fig. 10 the sign of a matrix of dimension $n = 49152$ can be evaluated to an accuracy of $10^{-9}$ by computing the sign of a $2 \times 2$ matrix, which is then lifted back to the original $n$-dimensional space through linear combinations of Krylov vectors. This emphasizes again the power of Krylov subspace methods.

5. Conclusions

In this paper we have presented a nested Krylov subspace method which boosts the Krylov-Ritz approximations used to compute the sign function of both Hermitian and non-Hermitian matrices. The Krylov-Ritz approximation projects the matrix on a Krylov subspace in which it computes the sign function exactly, before lifting it back to the original space. Its standard implementation suffers from the CPU intensive computation of the sign of the Ritz matrix, which goes like the cube of the Krylov subspace size. By making an additional projection on a much smaller Krylov subspace, the nested method significantly reduces the total computation time of the Krylov-Ritz approximation, without affecting its accuracy. Numerical tests showed that the nested method works equally well for Hermitian and non-Hermitian matrices and is more efficient than state-of-the-art rational approximation methods. Moreover, it

\(^4\)For each level the $p$-factor for Eq. (16) is computed using Eq. (18), using appropriately approximated boundaries for the spectrum of the Ritz matrix of the previous level. Note that the factor $p$ converges to 1 as more levels are introduced, and the preconditioning step converges to the RHi.
exhibits a good, close to linear, volume scaling. We are currently investigating the efficiency of the nested method for larger lattice volumes using a parallel implementation of the algorithm.

To end, we comment on the relation between the nested method and the extended Krylov subspace methods introduced in Ref. [20]. An extended Krylov space is defined as

$$\mathcal{K}_k(A, A^{-1}, x) = \text{span}(x, Ax, A^{-1}x, A^2x, A^{-2}x, \ldots, A^{k-1}x, A^{-k+1}x),$$

and an approximation in that subspace approximates $f(A)$ by the sum $Q(A) = \sum_{i=-k+1}^{k-1} c_i A^i$. In the nested method we construct the $\ell$-dimensional Krylov subspace $\mathcal{K}_\ell(H_k, e_1^{(k)})$, which forms an $\ell$-dimensional subspace of the $(2\ell - 1)$-dimensional extended Krylov subspace $\mathcal{K}_\ell(pH_k, (pH_k)^{-1}, e_1^{(k)})$. The nested method implicitly fixes the coefficients of the positive and negative powers of $Q(H_k)$ to be equal, $c_{-i} = c_i$, which follows from the use of the property $\text{sgn}(H_k) = \text{sgn}(H_k + H_k^{-1})$. Hence, the nested method implicitly truncates the size of the extended Krylov subspace.

Approximations for the sign function in extended Krylov subspaces have been briefly considered recently [21], however not in combination with the nesting of Krylov subspaces, i.e. the extended subspace is constructed for the original matrix $A$, not for the Ritz matrix $H_k$. Evidently this is not feasible in the application to lattice QCD as the inversion of the $\gamma_5$-Wilson Dirac operator is too expensive in order to construct extended Krylov subspaces.

Acknowledgements

We would like to thank Andreas Frommer and Tilo Wettig for useful discussions.

Appendix A. Lanczos algorithm

\[ \begin{align*}
  v_1 &\leftarrow \frac{x}{\|x\|} \\
  r &\leftarrow A v_1 \\
  \text{for } j = 1 \text{ to } k \text{ do} \\
  &H(j, j) \leftarrow v_j^T r \\
  &r \leftarrow r - H(j, j) v_j \\
  &\text{if } j = k \text{ then} \\
  &\text{stop} \\
  &\text{end if} \\
  &\beta \leftarrow \sqrt{r^T r} \\
  &H(j, j + 1) \leftarrow \beta
\end{align*} \]
\[ H(j + 1, j) \leftarrow \beta \vspace{1mm} \]
\[ v_{j+1} \leftarrow \frac{1}{\beta} r \vspace{1mm} \]
\[ r \leftarrow Av_{j+1} \vspace{1mm} \]
\[ r \leftarrow r - \beta v_j \vspace{1mm} \]
end for

All \( H(i, j) \) not assigned above are zero. Consequently \( H \) is tridiagonal and symmetric. The \( v_j \) are the column vectors of the matrix \( V_k \).

**Appendix B. Two-sided Lanczos algorithm**

\[ \begin{align*}
  v_1 & \leftarrow \frac{r}{\|r\|} \\
  w_1 & \leftarrow v_1 \\
  r & \leftarrow Av_1 \\
  l & \leftarrow A^\dagger w_1 \\
  \text{for } j = 1 \text{ to } k \text{ do} & \\
  & H(j, j) \leftarrow w_j^r \\
  & r \leftarrow r - H(j, j)w_j \\
  & l \leftarrow l - (H(j, j))^*w_j \\
  \text{if } j = k \text{ then} & \\
  & \text{stop} \\
  \text{end if} & \\
  \delta & \leftarrow r^vl \\
  \text{if } \delta \neq 0 \text{ then} & \\
  & \text{serious breakdown, stop} \\
  \text{end if} & \\
  \beta & \leftarrow \sqrt{\delta} \\
  H(j + 1, j) & \leftarrow \beta \\
  \gamma & \leftarrow \frac{\delta}{\beta} \\
  H(j, j + 1) & \leftarrow \gamma \\
  v_{j+1} & \leftarrow \frac{1}{\beta} r \\
  w_{j+1} & \leftarrow \frac{1}{\gamma} l \\
  r & \leftarrow Av_{j+1} \\
  l & \leftarrow A^\dagger w_{j+1} \\
  r & \leftarrow r - \gamma v_j \\
  l & \leftarrow l - \beta^* w_j \\
  \text{end for} \\
\end{align*} \]

The \( v_j \) and \( w_j \) are the column vectors of the matrices \( V_k \) and \( W_k \), respectively. All \( H(i, j) \) not assigned above are zero. Consequently \( H \) is tridiagonal, but not symmetric as in the Hermitian case. The coefficients \( \beta \) and \( \gamma \) are, non-uniquely, chosen to satisfy the biorthonormality condition

\[ w_j^T v_i = \delta_{ij}. \quad (B.1) \]

There are potential problems in the two-sided Lanczos process, namely serious breakdowns and near breakdowns, where \( \delta \approx 0 \), respectively \( \approx 0 \), however, these were not encountered in our numerical tests.

**Appendix C. Nested algorithm**

Given a (non-)Hermitian matrix \( A \), a source vector \( x \) and the critical eigenvectors \( r_i \) (left and right eigenvectors \( l_i \) and \( r_i \), with eigenvalues \( \lambda_i \), \( i = 1, \ldots, m \), do:
1. Apply Left-Right deflation (see Ref. [11]) to construct $x_0$, where the components of the source vector $x$ along the eigenvectors $r_i$ have been removed:

$$x_0 = x - \sum_{i=1}^{m} (l_i, x)r_i,$$

where $l_i = r_i$ for Hermitian $A$.

2. Run the (two-sided) Lanczos algorithm from [Appendix A] [Appendix B] with $A$ and $x_0$ to obtain $V_k$ and $H_k$.

3. Perform an LU decomposition of $pH_k$, e.g., with the LAPACK routine dgtrfs (zgtrfs). This yields a lower triangular matrix $L$ with unit diagonal and one sub-diagonal, and an upper triangular matrix $U$ with one diagonal and two super-diagonals. All other entries of $L$ and $U$ are zero.

4. Run the (two-sided) Lanczos algorithm with $H'_k = (pH_k + (pH_k)^{-1})/2$ and source vector $e^{(h)}_i$ to construct the Krylov basis $V'_k$ and the Ritz matrix $H'_k$. To do so apply $H'_k$ to each Krylov vector $v$:
   (a) Compute $(pH'_k)^{-1} v$ using a sparse LU back substitution, e.g., with the LAPACK routine dgtrfs (zgtrfs).
   (b) Compute $(pH'_k)v$ and add to the result of (a). This tridiagonal multiply and add can be done efficiently using the BLAS band-matrix-vector multiplication routine dsbmv (zgbmv).

5. Run the RHI (or any other suitable method to compute the sign function) on $H'_k$ to obtain $\text{sgn}(H'_k)$.

6. The final approximation is then given by

$$\text{sgn}(A)x \approx \sum_{i=1}^{m} \text{sgn}(4\lambda_i)(l_i, x)r_i + |x_0|V_k V'_k \text{sgn}(H'_k)e^{(h)}_i.$$

Note that steps (3-5) are done in real arithmetic in the Hermitian case.

References
