Krylov subspace methods for shifted unitary matrices and eigenvalue deflation applied to the Neuberger Operator and the matrix sign function

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

am Fachbereich Mathematik der
Bergischen Universität Wuppertal
genehmigte

Dissertation

von

Dipl.-Math. Katrin Schäfer

Tag der mündlichen Prüfung:	29. 8. 2008
Referent:	Prof. Dr. A. Frommer
Korreferent:	Prof. Dr. M. Günther

Danke

Ich danke meinem Doktorvater Prof. Dr. Andreas Frommer sowie Prof. Dr. Bruno Lang, Prof. Dr. Michael Günther, Prof. Dr. Francesco Knechtli, Brigitte Schultz, Dr. Holger Arndt sowie der gesamten Gruppe FGAngInf, von denen mich jeder auf seine Weise unterstützt hat. Meiner Familie und meinen Freunden danke ich für ihre Geduld und ihr Verständnis.

Contents

List of Tables

List of Figures

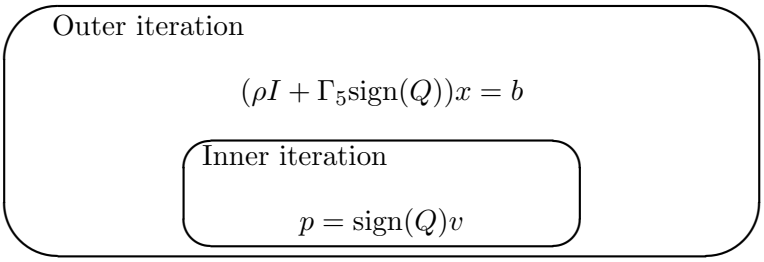
Preface

The solution of linear systems $Ax = b$ with a large and sparse matrix A plays an important role in numerical linear algebra. There is no strict definition of when a matrix is called sparse or large. With view on the methods used to solve the linear system one should call a matrix *sparse* when it has enough zero entries to exploit this fact and *large* when it is too large to be handled with direct methods. The combination of being large and sparse usually allows to use a matrix vector product with A, though.
Linear systems with large sparse matrices thus cannot be solved with direct methods. These matrices exist only as their action on a vector. The methods of choice are therefore iteration methods: Starting with an initial guess, an (improved) approximation is computed in each step. In this work we concentrate on Krylov subspace methods, projection methods choosing the approximation in each step from certain subspaces of $\mathbb{C}^n$, the Krylov subspaces. Krylov subspaces depend on the matrix A. Building a basis for a Krylov subspace requires matrix vector products with A in a way that in the respective iteration methods as well the matrix A is needed only as its action on a vector.

As motivation and example for the methods presented in this thesis we use an application from lattice QCD (quantum chromodynamics). QCD is the theory of the interaction of quarks, the physical particles that build protons and neutrons, for example. This interaction is called strong interaction. Lattice QCD gives a formulation of QCD on a 4-dimensional space-time lattice. On each lattice site there are 12 unknowns, so that for a lattice of size N in each direction we have $12 \cdot N^4$ unknowns.
An important property in QCD is the so-called chiral symmetry. On the lattice chiral symmetry can only be fulfilled in the continuum limit, i.e, for lattice spacing $a \to 0$. To achieve this, the Ginsparg-Wilson relation is required. Therefore we use the Neuberger overlap operator $\rho I + \Gamma_5 \text{sign}(Q)$ which fulfills the Ginsparg-Wilson relation and thus realizes chiral symmetry on the lattice.

Actually, the Neuberger overlap operator is not sparse. Even though the matrix Q is sparse, $\text{sign}(Q)$ is not sparse. The matrix vector product with

sign(Q) can thus not be computed directly. To use iteration methods we have to compute an approximate matrix vector product with sign(Q). This is done with iteration methods as well, now requiring matrix vector products with the sparse matrix Q. The result is a nested iteration: The inner iteration approximates the action of sign(Q) on a vector, the outer iteration uses the result of the inner iteration to approximate the solution of a linear system with the Neuberger overlap operator.



We have to distinguish the Neuberger overlap operator at zero chemical potential from the one at non-zero chemical potential. The chemical potential being zero or non-zero results in different properties of the Dirac matrix $D = \Gamma_5 Q$. Depending on the properties of the Dirac matrix the Neuberger operator shows properties we can exploit – or leaves us without any utilizable structure.

At zero chemical potential the Dirac matrix D is Γ_5-symmetric which leads to $Q = \Gamma_5 D$ being hermitian and $\Gamma_5 \text{sign}(Q)$ being unitary. This can be exploited in the outer iteration as well as in the inner iteration. For the outer iteration the shifted unitary structure of the Neuberger overlap operator allows iteration methods with short recurrences. Following the Faber-Manteuffel theorem this seemed to be impossible since (shifted) unitary matrices do not fulfil the requirement of this theorem, i.e., unitary matrices are in general not normal(s) for small s. But it was shown more recently that using a different reccurence scheme a bigger class of matrices allows for short recurrences.
The inner iteration profits from Q being hermitian since that leads to a small number of poles in a rational approximation and allows to use short recurrence multishift methods.
At non-zero chemical potential the Dirac matrix is no longer Γ_5-symmetric. Therefore, the Neuberger overlap operator shows no structure to allow for short recurrence iterations for the outer or inner iteration. In addition, the number of poles required for an accurate rational approximation of sign(Q)b for non-hermitian Q is significantly higher than in the case of Q hermitian.

This thesis is organized as follows.

In Chapter 1 the fundamental principles of iteration methods and Krylov subspaces are presented. Classical Krylov subspace methods are introduced. We present the extension of scalar functions to matrix functions and the matrix sign function. The Neuberger operator is introduced for zero and non-zero chemical potential and the fundamental symmetries of QCD, Γ_5-symmetry and chiral symmetry are explained.

In Chapter 2 we investigate iteration methods for shifted unitary matrices. Two versions of the Arnoldi method are presented. For unitary matrices these methods build Krylov subspace bases with short recurrences. In combination with the minimal residual or Galerkin condition we obtain four Krylov subspace methods to solve linear systems with shifted unitary matrices. Three of these methods (SUOM, SHUMR and SUMR) are already known [1, 7, 8, 9]. We complete the set of methods by combining the isometric Arnoldi method with a Galerkin condition, resulting in SUFOM (shifted unitary FOM). In addition, we give a more detailed theoretic foundation for SUOM and SHUMR than found in the literature up to now. Finally, we modify SHUMR and present a breakdown-free version.

In Chapter 3 we invesigate multishift methods to compute rational approximations to the matrix sign function. We concentrate on methods for non-hermitian matrices with regard to the Neuberger overlap operator at non-zero chemical potential. To accelerate convergence we use eigenvalue information. Two variants of eigenvalue deflation, Schur-deflation and LR-deflation are presented. The idea is to augment Krylov subspaces by some Schur vectors or eigenvectors to eigenvalues with small real part. These deflation methods are applied to multishift versions of FOM, GMRES, BiCG, and QMR to accelerate the rational approximation of $\mathrm{sign}(Q)v$. For the long recurrence methods FOM and GMRES we investigate restarts to limit storage requirements. In this case we combine thus restarts, multishifts and deflation. Finally we reduce the number of poles for the rational approximation of the matrix sign function using the information of the eigenvalue deflation. It turns out that this is only possible for LR-deflation.

In Chapter 4 we give numerical results for the methods presented in the previous two chapters. The shifted unitary methods are tested with the Neuberger overlap operator at zero chemical potential. The sample matrices are taken from matrix market[1] and refer to 4^4-lattices. The deflation methods are tested for the rational approximation of the matrix sign function for a non-hermitian matrix, the Dirac operator at non-zero chemical

[1] http://math.nist.gov/MatrixMarket/data/misc/qcd/

potential. The sample matrices refer to a 4^4-lattice and a 6^4-lattice, respectively.

Throughout the thesis the following notation is used:

$\langle \cdot, \cdot \rangle$	inner product (Euclidean if not indicated otherwise)
e_i	i-th unit vector
$\delta_{i,j}$	Kronecker delta function: $\delta_{i,j} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases}$
$\| \cdot \|_2$	the Euclidian vector norm or the induced matrix norm
$\| \cdot \|_F$	the Frobenius (matrix) norm
$\| \cdot \|_A$	norm induced by the inner product $\langle \cdot, \cdot \rangle_A = \langle A \cdot, \cdot \rangle_2$
I_n	$n \times n$ unit matrix; if n is suppressed, the dimension is obvious from the context
A^H, v^H	the Hermitian adjoint of a matrix or vector
A^T, v^T	the transpose of a matrix or vector
$A^\dagger$	the Moore-Penrose pseudo inverse
$[\cdot]$	matrix or vector composed of submatrices or -vectors
$(\cdot)$	matrix or vector given elementwise
$A \otimes B$	the Kronecker product of the matrices A and B
span(S), $S \subseteq \mathbb{C}^n$	the set of all linear combinations of vectors in S
rank(A)	the rank of the matrix A, i.e. the dimension of the space spanned by its rows/columns
spec(A)	the spectrum of the matrix A, i.e. the set of eigenvalues of the matrix A
range(A)	the range of the matrix A, i.e. the space spanned by the columns of the matrix A
null(A)	the nullspace of the matrix A, i.e. the set of vectors x with $Ax = 0$
Π_n	polynomials of degree at most n

Table 1: Notation

Part of this thesis was supported by DFG project Fr755 "Effiziente Löser für das Overlap-Modell der Fermion Diskretisierung in der QCD" (efficient solvers for the overlap model of the fermion discretization in QCD).

Chapter 1

Introduction

1.1 Krylov subspace methods

Let $A \in \mathbb{C}^{n\times n}$ be a non-singular complex matrix. We consider the linear system

$$Ax = b \tag{1.1}$$

with A being large, so that it is not possible to solve (1.1) with direct methods. Iteration methods require the computation of matrix vector products with A though. Therefore we assume A to be sparse or at least involve a structure that allows to compute the desired matrix vector product up to a given accuracy.

Iteration methods compute in step k an approximation x_k to the solution of (1.1). This can be done in several ways. In projection methods the approximation x_k is chosen from an affine k-dimensional subspace K_k of $\mathbb{C}^n$, the search space. The approximation is chosen such that the residual $r_k = b - Ax_k$ is orthogonal to a second k-dimensional subspace L_k of $\mathbb{C}^n$, the so-called test space. In every step of the iteration, the test and search space are enlarged so that the next approximation x_{k+1} is chosen from the $(k+1)$-dimensional search space $K_{k+1} \supseteq K_k$ such that r_{k+1} is orthogonal to the $(k+1)$-dimensional test space $L_{k+1} \supseteq L_k$. Both search and test space thus have to be built iteratively.

1.1.1 Krylov subspaces

Krylov subspaces are perfectly suited to be built iteratively using only one matrix vector multiplication per iteration step. In addition, it turns out that when using Krylov subspaces only one subspace has to be built and used as search and test space.

Definition 1.1. For a matrix $A \in \mathbb{C}^{n\times n}$ and a vector $r \in \mathbb{C}^n$ the m-th *Krylov subspace* generated by A and r is

$$K_m(A, r) = \text{span}\{r, Ar, A^2r, \ldots, A^{m-1}r\}.$$

Obviously, Krylov subspaces are shift invariant, i.e. for a shifted matrix $\rho I + A$ it holds

$$K_m(\rho I + A, r) = K_m(A, r).$$

The dimension of a Krylov subspace with regard to A and r is bounded by the degree of the minimal polynomial of r in the following way.

Definition 1.2. A *minimal polynomial* p_r of r with regard to A is a polynomial with

$$\deg(p_r) = \min_{p\in\Pi_n}\{\deg(p) : p \neq 0, p(A)r = 0\}.$$

The minimal polynomial is unique up to scaling, so its degree is unique. So we call $\deg(p_r)$ the *degree of* r (with regard to A).

Lemma 1.3. Let $A \in \mathbb{C}^{n\times n}$ and $r \in \mathbb{C}^n$, $r \neq 0$, and let m_0 be the degree of the minimal polynomial of r with regard to A. Then it holds

$$\dim(K_m(A, r)) = m \quad \text{for } m \leq m_0$$

and

$$K_m(A, r) = K_{m_0}(A, r) \quad \text{for } m \geq m_0.$$

Proof. See [41]. □

Thus, from m_0 on the Krylov subspaces are A-invariant. The solution of (1.1) is then contained in a shifted Krylov subspace.

Lemma 1.4. Let $A \in \mathbb{C}^{n\times n}$ be nonsingular and $b \in \mathbb{C}^n$. For every $x_0 \in \mathbb{C}^n$ it holds

$$A^{-1}b \in x_0 + K_{m_0}(A, r_0), \tag{1.2}$$

where $r_0 = b - Ax_0$ and m_0 the degree of r_0, and

$$A^{-1}b \notin x_0 + K_m(A, r_0), \quad \text{for } m < m_0. \tag{1.3}$$

Proof. Let p_{m_0} be a minimal polynomial of r_0, i.e. $p_{m_0}(A)r_0 = 0$ and $\deg(p_{m_0}) = m_0$. Without loss of generality we assume $p_{m_0}(0) = 1$ such that $p_{m_0}(t) = 1 - tq_{m_0-1}(t)$ with $\deg(q_{m_0-1}) = m_0 - 1$. The residual for $x_{m_0} = x_0 + q_{m_0-1}(A)r_0 \in K_{m_0}(A, r_0)$ is

$$\begin{aligned} b - Ax_{m_0} &= r_0 - Aq_{m_0-1}(A)r_0 \\ &= p_{m_0}(A)r_0 \\ &= 0. \end{aligned}$$

Therefore (1.2) holds. Since the degree of p_{m_0} is minimal, it follows (1.3). □

1.1.2 Iteration methods

Krylov subspace methods are projection methods for solving systems of the form (1.1) using affine Krylov subspaces as search spaces: In every iterative step an iterate x_m is chosen from an affine Krylov subspace

$$x_m \in x_0 + K_m(A, r_0),$$

where x_0 is the initial guess and $r_0 = b - Ax_0$. The initial guess is often chosen as $x_0 = 0$ which gives $r_0 = b$. Since we will need non-zero x_0 when investigating restarts in the following, we assume a general, not necesarrily zero, initial guess.

From the definition of Krylov subspaces it is clear that the iterates can be written as

$$x_m = x_0 + q_{m-1}(A)r_0,$$

with a polynomial q_{m-1} of degree less or equal to $m-1$. The corresponding residuals are

$$r_m = b - Ax_m = p_m(A)r_0, \tag{1.4}$$

with a polynomial $p_m(t) = 1 - tq_{m-1}(t)$ of degree less or equal m and $p_m(0) = 1$. From (1.4) it directly follows that $r_m \in K_{m+1}(A, r_0)$. For the error it holds

$$e_m = A^{-1}b - x_m = A^{-1}r_m$$

and using the same polynomials p_m as for the residuals it holds same as for the residuals

$$e_m = p_m(A)e_0.$$

The various Krylov subspace methods differ in the way the iterates are chosen from the Krylov subspace. For this purpose we can either demand for $x_m \in x_0 + K_m(A, r_0)$ a (Petrov-)Galerkin condition

$$r_m \perp L \tag{1.5}$$

with a subspace L or the *minimal residual condition*

$$\|r_m\|_2 = \min_{x \in x_0 + K_m(A,r_0)} \|b - Ax\|. \tag{1.6}$$

Definition 1.5. The condition (1.5) for choosing $x_m \in x_0 + K_m(A, r_0)$ is called

- *Galerkin* condition if $L = K_m(A, r_0)$,
- *Petrov-Galerkin* condition if $L \neq K_m(A, r_0)$.

Obviously, using a (Petrov-)Galerkin condition, the subspace L is the test space. In fact, choosing the iterate with the minimal residual condition (1.6) results in residuals orthogonal to $AK_m(A, r_0)$. The minimal residual condition can thus be seen as a Petrov-Galerkin condition with the test space $L = AK_m(A, r_0)$.

For hermitian matrices both Galerkin and minimal residual condition result in a minimization of the errors e_m, in different norms though. While for the Galerkin condition

$$\|e_m\|_A = \min_{x \in x_0 + K_m(A,r_0)} \|A^{-1}b - x\|_A,$$

see [41], for the minimal residual condition it holds

$$\|e_m\|_{A^H A} = \min_{x \in x_0 + K_m(A,r_0)} \|A^{-1}b - x\|_{A^H A}.$$

1.1.3 Krylov bases

Krylov subspace bases are naturally built iteratively by multiplying the last basis vector with A and orthogonalizing against the previous vectors.

Let A be non hermitian. Using the modified Gram-Schmidt orthogonalization to produce orthonormal vectors v_k, $k = 1, \ldots, m$ that span $K_m(A, r_0)$ leads to the so-called Arnoldi method [2].

Algorithm 1.6. Arnoldi

{**Input** $m \leq m_0$, r_0, A}

$\tilde{v}_1 = r_0$
$h_{1,0} = \|\tilde{v}_1\|_2$
for $k = 1, \ldots, m$ **do**
  $v_k = \tilde{v}_k / h_{k,k-1}$
  $\tilde{v}_{k+1} = A v_k$
  for $i = 1, \ldots, k$ **do**
    $h_{i,k} = \langle \tilde{v}_{k+1}, v_i \rangle$
    $\tilde{v}_{k+1} = \tilde{v}_{k+1} - h_{i,k} v_i$
  end for
  $h_{k+1,k} = \|\tilde{v}_{k+1}\|_2$
end for

The matrices $V_m = [v_1, \ldots, v_m]$ and $H_m = (h_{i,j}) \in \mathbb{C}^{m \times m}$ satisfy the Arnoldi relations

$$V_m^H A V_m = H_m \tag{1.7}$$

and

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T = V_{m+1} \widehat{H}_m, \tag{1.8}$$

with

$$\widehat{H}_m = \begin{bmatrix} H_m \\ h_{m+1,m} e_m^T \end{bmatrix} \in \mathbb{C}^{(m+1)\times m}.$$

The Arnoldi method has a breakdown when $k = m_0$. In this situation the vectors $v_1, \ldots, v_{m_0}$ already span an A-invariant subspace of $\mathbb{C}^n$ such that $Av_{m_0} \in K_{m_0}(A, r_0)$. Therefore it holds $\tilde{v}_{m_0+1} = 0$ and thus $h_{m_0+1,m_0} = 0$. According to Lemma 1.4 this is the natural situation to stop the iteration anyway and the solution of (1.1) is contained in the corresponding affine Krylov subspace.

The matrix $\widehat{H}_m$ has rank m for $m \leq m_0$. This is due to its $(m+1) \times m$ Hessenberg structure with non-zero subdiagonal. The matrix H_m can be singular, though. However, if A is positive real, H_m is positive real, too. This is because

$$\langle H_m x, x \rangle = \langle V_m^H A V_m x, x \rangle = \langle A V_m x, V_m x \rangle$$

and V_m has full rank.

A disadvantage of the Arnoldi method is its long recursion. For the orthogonalization of a new Arnoldi vector, all previous Arnoldi vectors have to be stored. To circumvent storage problems one can bound the number of vectors to be stored by restarting after $m < m_0$ steps which results in blocks of m orthogonal vectors. Alternatively, instead of demanding the basis $\{v_1, \ldots, v_m\}$ of $K_m(A, r_0)$ to be orthogonal we can demand $\{v_1, \ldots, v_m\}$ to be orthogonal to a basis $\{w_1, \ldots, w_m\}$ of $K_m(A^H, \tilde{r}_0)$, i.e. we build biorthogonal bases $\{v_1, \ldots, v_m\}$ and $\{w_1, \ldots, w_m\}$. This is done in the unsymmetric Lanczos method. Note that we have to invest two matrix vector products instead of one to gain the short recurrence.

Algorithm 1.7. Unsymmetric Lanczos

{**Input** $m \leq m_0$, r_0, A}

$\tilde{v}_1 = r_0$, $\tilde{w}_1 = \tilde{r}_0 \neq 0 \in \mathbb{C}^n$, $v_0 = w_0 = 0$
for $k = 1, \ldots, m$ **do**
  chose β_k, γ_k: $\beta_k \gamma_k = \langle \tilde{v}_k, \tilde{w}_k \rangle$
  $v_k = \tilde{v}_k / \gamma_k$
  $w_k = \tilde{w}_k / \bar{\beta}_k$
  $\alpha_k = \langle A v_k, w_k \rangle$
  $\tilde{v}_{k+1} = A v_k - \alpha_k v_k - \beta_k v_{k-1}$
  $\tilde{w}_{k+1} = A^H w_k - \bar{\alpha}_k w_k - \bar{\gamma}_k w_{k-1}$
end for

With $V_m = [v_1, \ldots, v_m]$, $W_m = [w_1, \ldots, w_m]$ the biorthogonality reads $V_m^H W_m = W_m^H V_m = 0$, and with

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & 0 \\ \gamma_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_m \\ 0 & & \gamma_m & \alpha_m \end{pmatrix}$$

the following relations hold:

$$\begin{aligned} AV_m &= V_m T_m + \gamma_{m+1} v_{m+1} e_m^T \\ A^H W_m &= W_m T_m^H + \bar{\beta}_{m+1} w_{m+1} e_m^T \\ W_m^H A V_m &= T_m \end{aligned} \tag{1.9}$$

The unsymmetric Lanczos method has a breakdown when $k = m_0$, since $v_1, \ldots, v_{m_0}$ span an A-invariant subspace and $Av_{m_0} \in K_{m_0}(A, r_0)$. Therefore it holds $\tilde{v}_{m_0+1} = 0$ and thus $\langle \tilde{v}_{m_0+1}, \tilde{w}_{m_0+1} \rangle = 0$. Again, this is the natural situation to stop the iteration, see Lemma 1.4. Actually, the unsymmetric Lanczos method can have a breakdown for $k < m_0$, namely when $\tilde{w}_{k+1} = 0$, i.e., when $K_{k+1}(A^H, \tilde{r}_0) = K_k(A^H, \tilde{r}_0)$ or even when $\langle \tilde{v}_{k+1}, \tilde{w}_{k+1} \rangle = 0$ while $\tilde{v}_{k+1} \neq 0$ and $\tilde{w}_{k+1} \neq 0$.

Breakdowns with $\tilde{v}_m = 0$ or $\tilde{w}_m = 0$ are thus called *lucky*, other breakdowns are called *serious*. Most of the serious breakdowns can be avoided using look-aheads, [14].

As long as no breakdowns occur, the matrix

$$\widehat{T}_m = \begin{bmatrix} T_m \\ \gamma_{m+1} e_m^T \end{bmatrix}$$

is non-singular since $\gamma_i \neq 0$ for $i = 1, \ldots, m+1$. The matrix T_m might be singular though.

In the case of a symmetric matrix $A = A^H$ both the Arnoli and the unsymmetric Lanczos method reduce to the same symmetric method when $\tilde{r}_0$, γ_k, and β_k are chosen right. The inner **for**-loop of the Arnoldi method $i = 1, \ldots, k$ reduces to $i = k-1, k$ such that H_m is tridiagonal. In the unsymmetric Lanczos method, choosing $\tilde{r}_0 = r_0$ and $\beta_k = \gamma_k$, we get $V_m = W_m$. Actually, it even holds $T_m = H_m$, T_m and H_m resulting from the unsymmetric Lanczos method and the Arnoldi method, respectively. The symmetric method is called Lanczos method.

For shifted matrices $A = \rho I + U$ the Krylov subspaces are built with regard to the matrix U since $K_m(A, r_0) = K_m(U, r_0)$. Therefore, the Arnoldi relations (1.7), (1.8) and (1.9), respectively, hold with regard to U.

We get Arnoldi relations for the shifted matrix A directly by shifting the respective relations for the unitary matrix U. This is summarized in the following lemma, the proof of which is trivial.

Lemma 1.8. Let $V_m = [v_1, \ldots, v_m]$ and $H_m^{(U)} = V_m^H U V_m$ be obtained from the Arnoldi method. Then

$$H_m^{(A)} = V_m^H A V_m = \rho I + H_m^{(U)}$$

and

$$A V_m = V_m(\rho I + H_m^{(U)}) + h_{m+1,m} v_{m+1} e_m^T.$$

Let $V_m = [v_1, \ldots, v_m]$, $W_m = [w_1, \ldots, w_m]$ and $T_m^{(U)} = W_m^H U V_m$ be obtained by the Lanczos biorthogonalization method. Then

$$\begin{aligned} A V_m &= V_m(\rho I + T_m^{(U)}) + \gamma_{m+1} v_{m+1} e_m^T, \\ A^H W_m &= W_m(\bar{\rho} I + {T_m^{(U)}}^H) + \bar{\beta}_{m+1} w_{m+1} e_m^T, \\ T_m^{(A)} &= \rho I + T_m^{(U)}. \end{aligned}$$

1.1.4 Classical Krylov subspace methods

Up to now a large number of Krylov subspace methods exist. Most of them are variants of a few basic methods. These basic methods differ in the underlying method to build the Krylov basis and the condition used to choose the iterate from the Krylov subspace.

Let A be non hermitian. There exist four basic Krylov subspace methods combining the Arnoldi method or the unsymmetric Lanczos method with a (Petrov-)Galerkin condition or a minimal residual condition.

Having a basis $\{v_1, \ldots, v_m\}$ for $K_m(A, r_0)$ given, computed with the Arnoldi method, the iterates read

$$x_m = x_0 + V_m y_m,$$

where $V_m = [v_1, \ldots, v_m]$ and $y_m \in \mathbb{C}^m$, and the corresponding residuals are

$$r_m = r_0 - A V_m y_m.$$

The Galerkin condition $r_m \perp K_m(A, r_0)$ translates to $V_m^H r_m = 0$ such that y_m is the solution of the $m \times m$ linear system

$$V_m^H A V_m y_m = H_m y_m = V_m^H r_0. \tag{1.10}$$

Computing the iterates this way leads to the so called full orthogonalization method (FOM) [40].

The FOM iterates exist when A is positive real since then H_m is positive real and thus non-singular. On the other hand, when A is indefinite, H_m can be singular and the FOM iterates do not necessarily exist.

The FOM residuals are always multiples of Arnoldi vectors.

Lemma 1.9. The FOM residuals are

$$r_m = -h_{m+1,m} e_m^T y_m v_{m+1}.$$

Proof. Using the Arnoldi relation (1.8) it holds

$$\begin{aligned} r_m &= r_0 - AV_m y_m \\ &= r_0 - V_m H_m y_m - h_{m+1,m} e_m^T y_m v_{m+1}. \end{aligned}$$

Since $r_0 = \|r_0\|_2 v_1$ and y_m is the solution of (1.10), it holds

$$r_0 - V_m H_m y_m = 0.$$

□

For the minimal residual condition we minimize in the Arnoldi case

$$\begin{aligned} \|r_m\|_2 = \|r_0 - AV_m y_m\|_2 &= \|r_0 - V_{m+1}\widehat{H}_m y_m\|_2 \\ &= \|V_{m+1}(\beta e_1 - \widehat{H}_m y_m)\|_2 \\ &= \|\beta e_1 - \widehat{H}_m y_m\|_2. \end{aligned}$$

The least squares problem

$$\|\beta e_1 - \widehat{H}_m y_m\|_2 = \min_{y \in \mathbb{C}^m} \|\beta e_1 - \widehat{H}_m y\|_2 \tag{1.11}$$

is usually solved using the QR-decomposition $\widehat{H}_m = Q_{m+1}\widehat{R}_m$, where

$$\widehat{R}_m = \begin{bmatrix} R_m \\ 0 \end{bmatrix}.$$

Since Q_{k+1} is unitary, (1.11) is equivalent to

$$\|\beta e_1 - \widehat{H}_m y_m\|_2 = \min_{y \in \mathbb{C}^m} \|\beta Q_{k+1}^H e_1 - \widehat{R}_m y\|_2. \tag{1.12}$$

The resulting method is called generalized minimum residual method (GMRES), see [42].

In contrast to the FOM iterates the GMRES iterates always exist if $m < m_0$. This is because $\text{rank}(\widehat{H}_m) = m$.

Obviously, for the norm of the GMRES residuals it holds

$$\|r_m\|_2 = |\beta e_{k+1}^T Q_{k+1}^H e_1|.$$

Equivalently, y_m can be expressed using the Moore-Penrose pseudo-inverse $\widehat{H}_m^\dagger$ of $\widehat{H}_m$:

Lemma 1.10. The least squares problem (1.11) is equivalent to

$$\widehat{H}_m^H \widehat{H}_m y_m = \widehat{H}_m^H \beta e_1, \tag{1.13}$$

i.e.

$$y_m = \widehat{H}_m^\dagger \beta e_1 = (\widehat{H}_m^H \widehat{H}_m)^{-1} \widehat{H}_m^H \beta e_1.$$

Proof. (1.13) is the normal equation for (1.11) and $\text{rank}(\widehat{H}_m) = m$.

□

Using the Lanczos biorthogonalization and a Petrov-Galerkin condition with L the span of the columns of W_m, the orthogonality condition translates thus to $W_m^H r_m = 0$, and y_m is the solution of the linear system of size $m \times m$

$$W_m^H A V_m y_m = T_m y_m = W_m^H r_0. \tag{1.14}$$

From the system (1.14) short recurrence updates for $x_m = V_m y_m$ can be obtained by exploiting the tridiagonal structure of T_m. The resulting method is called BiCG, see [20].

In addition to the breakdowns of the Lanczos biorthogonalization, the BiCG iterates do not necessarily exist since T_m might be singular.

Anologous to the FOM residuals being multiples of the Arnoldi vectors, the BiCG residuals are multiples of the Lanczos vectors for $K_m(A, r_0)$.

Lemma 1.11. The BiCG residuals are

$$r_m = -\gamma_{m+1} e_m^T y_m v_{m+1}.$$

Proof. Using relation (1.9) it holds

$$\begin{aligned} r_m &= r_0 - A V_m y_m \\ &= r_0 - V_m T_m y_m - \gamma_{m+1} e_m^T y_m v_{m+1} \\ &= r_0 - V_m W_m^H r_0 - \gamma_{m+1} e_m^T y_m v_{m+1}. \end{aligned}$$

Since $r_0 = \gamma_1 v_1$ and $W_m^H v_1 = e_1$ it holds

$$r_0 - V_m W_m^H r_0 = 0.$$

□

To derive a method analogous to GMRES based on the Lanczos biorthogonalization the minimal residual condition has to be weakend. This is because

$$\|r_m\|_2 = \|r_0 - AV_m y_m\|_2 = \|V_{m+1}(\gamma_1 e_1 - \widehat{T}_m y_m)\|_2 \neq \|\gamma_1 e_1 - \widehat{T}_m y_m\|_2 \quad (1.15)$$

since V_{m+1} is not orthogonal in this case. Still, we can demand

$$\|r_0 - AV_m y_m\|_2 = \min_{y \in \mathbb{C}^m} \|\gamma_1 e_1 - \widehat{T}_m y\|_2 \quad (1.16)$$

instead of the minimal residual condition (1.11). The condition (1.16) is called *quasi minimal residual condition*. As for GMRES, (1.16) is usually solved using the QR-decomposition $\widehat{T}_m = Q_{m+1}\widehat{R}_m$. Since $\widehat{T}_m$ is tridiagonal, the unitary matrix Q_{m+1} can be written as a product of m Givens matrices, each one chosen to zero out one subdiagonal element of $\widehat{T}_m$. In this way, exploiting the tridiagonal structure of $\widehat{T}_m$ leads to short recurrences for $x_m = V_m y_m$. The resulting method is called quasi minimum residual method (QMR), see [15].

As long as $m < m_0$ and no breakdown occurs, $\operatorname{rank}(\widehat{T}_m) = m$ and thus the QMR iterates exist.

Due to the inequality in (1.15), the QMR residuals are not as simply obtained as the GMRES residuals.

Lemma 1.12. Let Q_{m+1} be written as a product of Givens matrices

$$Q_{m+1} = G_1(c_1) \dots G_m(c_m)$$

with

$$G_i(c_i) = \begin{bmatrix} I_{i-1} & & & \\ & -c_i & s_i & \\ & s_i & \bar{c}_i & \\ & & & I_{m-i-1} \end{bmatrix}.$$

Then for the QMR residuals it holds

$$r_m = s_m \frac{\eta_m}{\eta_{m-1}} r_{m-1} + \eta_m \bar{c}_m v_{m+1}$$

with $\eta_m = \gamma_1 e_{m+1}^T Q_{m+1}^H e_1$.

Proof. The QMR residuals are

$$r_m = V_{m+1} Q_{m+1}(\gamma_1 Q_{m+1}^H e_1 - \widehat{R}_m y_m).$$

Since y_m is the solution of (1.16) it holds

$$\gamma_1 Q_{m+1}^H e_1 - \widehat{R}_m y_m = (\gamma_1 e_{m+1}^T Q_{m+1}^H e_1) e_{m+1} = \eta_m e_{m+1}$$

such that

$$\begin{aligned} r_m &= \eta_m V_{m+1} Q_{m+1} e_{m+1} \\ &= \eta_m \left[V_m v_{m+1}\right] \begin{bmatrix} Q_m & \\ & 1 \end{bmatrix} (0, \ldots, 0, s_m, \bar{c}_m)^T \\ &= \eta_m \left[s_m V_m Q_m e_m + \bar{c}_m v_{m+1}\right] \\ &= s_m \frac{\eta_m}{\eta_{m-1}} r_{m-1} + \eta_m \bar{c}_m v_{m+1}. \end{aligned}$$

□

For hermitian matrices FOM and BiCG reduce to the conjugate gradient method (CG), see [22], and GMRES and QMR reduce to the minimum residual method (MINRES), see [37]. Table 1.1 gives an overview over the presented six classical Krylov subspace methods.

method/condition	Galerkin	(quasi) minimal residual
Arnoldi	FOM	GMRES
Lanczos biorthogonalization	BiCG	QMR
Lanczos	CG	MINRES

Table 1.1: Classical Krylov subspace methods

1.1.5 Inexact methods

All Krylov subspace methods contain a matrix vector multiplication as the core computation in each step. When we apply the methods of the preceding sections, e.g. to the Neuberger operator $\rho I + \Gamma_5 \text{sign}(Q)$ we cannot compute this matrix vector product directly since $\text{sign}(Q)$ is large but not sparse even though Q is sparse. So we cannot directly compute the exact product $Uv = \Gamma_5 \text{sign}(Q)v$ for a vector v.

Replacing the exact product by an approximation, for example using a rational approximation to the sign function, leads to so-called inexact Krylov subspace methods [10, 44, 46]. When the approximation is obtained from an iterative method, we get a nested iteration with an inner iteration for the approximation of the matrix vector product run in every step of the outer iteration.

An inexact matrix vector product can be viewed as a pertubation of the exact product, i.e.

$$w = Av + g$$

with a perturbation g, $\|g\|_2 \leq \eta\|A\|_2\|v\|_2$. The question that arrises is: How does this perturbation (i.e. η) influence the convergence of the inexact method?

Running the Arnoldi process with an inexact matrix vector product changes the Arnoldi relation (1.8) to

$$AV_m + G_m = V_{m+1}\widehat{H}_m,$$

where

$$G_m = [g_1, \ldots, g_m], \quad \|g_i\|_2 \leq \eta_i,$$

since $\|v_i\|_2 = 1$. This is equivalent to running the Arnoldi method with the perturbed matrix $\tilde{A} = A + G_m V_m^H$:

$$(A + G_m V_m^H)V_m = V_{m+1}\widehat{H}_m.$$

The iterates, which are approximations to the matrix vector product in this case, can again be chosen by using a minimal residual or Galerkin condition. Independent from the chosen condition, the (true) residuals are

$$\begin{aligned} b - (\rho I + A)x_m &= r_0 - (\rho I + A)V_m y_m \\ &= r_0 - \rho V_m y_m - V_{m+1}\widehat{H}_m y_m + G_m y_m. \end{aligned} \tag{1.17}$$

Note that $r_m = r_0 - \rho V_m y_m - V_{m+1}\widehat{H}_m y_m$ are the residuals computed in the iteration. The vector $G_m y_m = (b - (\rho I + A)x_m) - r_m$ is called residual gap.

The computed residuals can be monitored during the iteration, while we are actually interested in the true residuals. From (1.17) we get the obvious bound

$$\|b - (\rho I + A)x_m\|_2 \leq \|r_m\|_2 + \|G_m y_m\|_2.$$

Therefore, to assure that for the true residual we have

$$\|b - (\rho I + A)x_m\|_2 = \mathcal{O}(\epsilon),$$

we should choose the η_i such that for the residual gap

$$\|G_m y_m\|_2 = \mathcal{O}(\epsilon). \tag{1.18}$$

This can be achieved by choosing $\eta_i = \epsilon$ throughout the iteration. Actually, it suffices to demand less. By choosing η_i as shown in Table 1.2 the inexact method starts with a matrix vector product approximated to high accuracy, but the accuracy may be decreased during the iteration, i.e. η_j grows with the residual norm getting smaller. Still (1.18) is achieved. For a detailed argumentation see [44].

Inexact methods with this property are called relaxed [46]. The obvious advantage of relaxation is, that computing the matrix vector product to lower accuracy causes less computational cost.

condition	tolerance η_j
Galerkin	$\epsilon \cdot \sqrt{\sum_{i=0}^{j} \|r_i\|_2^{-2}}$
minimal residual	$\epsilon / \|r_j\|_2$

Table 1.2: Precision of the matrix vector product

1.2 Matrix functions

There are several ways to extend functions to matrix functions, i.e. to extend a function $f : \mathbb{C} \to \mathbb{C}$ to a function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$. A compact overview is given in [18], for a thorough treatment see [23]. One definition uses Cauchy's integral formula:

Definition 1.13. Let f be analytic on and inside a closed contour C that encloses spec(A). Then the *matrix function* is defined as

$$f(A) = \frac{1}{2\pi i} \int_C f(\zeta)(\zeta I - A)^{-1} d\zeta.$$

An alternative definition of matrix functions uses the Jordan decomposition. The Definitions 1.14 and 1.13 are consistent, see [18, 23].

Definition 1.14. Let $A \in \mathbb{C}^{n\times n}$ and $A = X\text{diag}(J_1(\lambda_1), \ldots, J_t(\lambda_t))X^{-1}$ the Jordan decomposition of A with Jordan blocks $J_i \in \mathbb{C}^{m_i \times m_i}$. Assume that the function $f : \mathbb{C} \to \mathbb{C}$ is $m_i - 1$ times differentiable at λ_i, $i = 1, \ldots, t$. Then the *matrix function* is defined as

$$f(A) = X\text{diag}(f(J_1(\lambda_1)), \ldots, f(J_t(\lambda_t)))X^{-1}$$

where

$$f(J_i(\lambda_i)) = \begin{pmatrix} f(\lambda_i) & f^{(1)}(\lambda_i) & \ldots & \ldots & \frac{f^{(m_i-1)}(\lambda_i)}{(m_i-1)!} \\ 0 & f(\lambda_i) & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & f^{(1)}(\lambda_i) \\ 0 & \ldots & \ldots & 0 & f(\lambda_i) \end{pmatrix} \in \mathbb{C}^{m_i \times m_i}.$$

The matrix $J = \text{diag}(J_1(\lambda_1), \ldots, J_t(\lambda_t))$ is unique up to ordering of the $J_i(\lambda_i)$ along the diagonal while X is not unique [20, 23]. Anyway, the matrix function $f(A)$ is independent from the actual choice of the Jordan decomposition [25].

Obviously, when $m_i = 1$ for $i = 1, \ldots, t$, for example when A is normal, no differentiability assumption on f is needed.

In the following proposition we summarize some properties of matrix functions.

Proposition 1.15. Let $A \in \mathbb{C}^{n\times n}$ and let $f(A)$ be defined. Then $f(A)$ has the following properties:

a) If the matrix X commutes with A then X commutes with $f(A)$.

b) For non-singular X it holds $f(XAX^{-1}) = Xf(A)X^{-1}$.

c) The eigenvalues of $f(A)$ are $f(\lambda_i)$ where the λ_i are the eigenvalues of A.

Proof. See [23]. □

For some functions the extension to matrix functions is quite intuitive:

Proposition 1.16. Let $f(A)$ and $g(A)$ be defined.

a) If $f(x) = c \in \mathbb{C}$ then $f(A) = c \cdot I$.

b) If $f(x) = x$ then $f(A) = A$.

c) If $h(x) = f(x) + g(x)$ then $h(A) = f(A) + g(A)$.

d) If $h(x) = f(x) \cdot g(x)$ then $h(A) = f(A) \cdot g(A)$.

Proof. a) and b) follow directly from the Definition 1.14. For the proof of c) and d) see [23].

□

From Definition 1.14 it follows directly that for any two functions f and g it holds

$$f(A) = g(A) \Leftrightarrow f^{(j)}(\lambda_i) = g^{(j)}(\lambda_i) \tag{1.19}$$

for $i = 1, \ldots, t$ and $j = 0, \ldots, m_i - 1$. In particular, (1.19) shows the existence of a polynomial p with $\deg(p) \leq n - 1$ and $f(A) = p(A)$ since (1.19) holds for the polynomial which interpolates f at λ_i in the Hermite sense. The polynomial depends on A, of course, so that in general there will be no polynomial p with $f(A) = p(A)$ for any matrix A.

If g is an approximation to f then $g(A)$ is an approximation to $f(A)$. If $A = TJT^{-1}$ is diagonalizable and

$$|f(x) - g(x)| \leq \epsilon \text{ for } x \in \operatorname{spec}(A)$$

then for the matrix function it holds

$$\|f(A) - g(A)\|_2 \leq \epsilon \|T\|_2 \cdot \|T^{-1}\|_2.$$

1.2.1 Matrix sign function

One matrix function of special interest, e.g., for the Neuberger overlap operator, is the matrix sign function. The matrix sign function is the extension of the scalar sign function

$$\operatorname{sign}(z) = \begin{cases} +1 & \text{for } \operatorname{Re}(z) > 0 \\ -1 & \text{for } \operatorname{Re}(z) < 0. \end{cases}$$

Note that the (scalar) sign function is not defined for z with $\operatorname{Re}(z) = 0$.

Since outside the imaginary axis $\operatorname{sign}(z)$ is infinitely often differentiable with all derivatives equal to zero, the definition of the matrix sign function using the Jordan decomposition is quite simple.

Definition 1.17. Let $A \in \mathbb{C}^{n\times n}$ and let $A = XJX^{-1}$ be the Jordan decomposition of A with

$$J = \begin{bmatrix} J_+ & 0 \\ 0 & J_- \end{bmatrix},$$

where the eigenvalues of J_+ lie in the right half plane and the eigenvalues of J_- lie in the left half plane. Then the matrix sign function is

$$\operatorname{sign}(A) = X \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} X^{-1}.$$

Analogous to the scalar sign function not being defined for $\operatorname{Re}(z) = 0$, the matrix sign function is not defined if A has eigenvalues on the imaginary axis.

For the integral definition of the matrix sign function we therefore have to look at the positive and negative real part seperately, see [38].

Definition 1.18. Let A be a matrix without eigenvalues on the imaginary axis. Let C^+ be a closed contour enclosing the eigenvalues of A with positive real part and C^- a closed contour enclosing the eigenvalues of A with negative real part. Then

$$\operatorname{sign}(A) = \frac{1}{2\pi i}\left(\int_{C^+} (\zeta I - A)^{-1} d\zeta - \int_{C^-} (\zeta I - A)^{-1} d\zeta\right).$$

If the matrix sign function is defined, the following properties are obvious:

- $A \cdot \operatorname{sign}(A) = \operatorname{sign}(A) \cdot A$.
- $(\operatorname{sign}(A))^2 = I$.
- $(\operatorname{sign}(A))^{-1} = \operatorname{sign}(A)$.

- If $A = A^H$ then $\text{sign}(A) = \text{sign}(A)^H$.

The matrix sign function can be computed using the Jordan decomposition for example, even though its computation is numerically instable, since the size of the Jordan blocks is of no importance. Only the sign of the the eigenvalues is needed.

For an approximative computation, there exist several iteration schemes, see for example [3, 23, 28]. One simple iteration scheme is the Newton iteration:

$$X_{i+1} = \frac{1}{2}(X_i + X_i^{-1}), \quad X_0 = A. \tag{1.20}$$

The Newton iteration results from Newton's method applied to the equation $X^2 - I = 0$ and converges quadratically and globally. A iteration scheme with more matrix multiplications instead of the matrix inversion is given by the Newton-Schulz iteration:

$$X_{i+1} = \frac{1}{2}X_i(3I + {X_i}^2), \quad X_0 = A. \tag{1.21}$$

The Newton-Schulz (1.21) iteration converges quadratically same as the Newton iteration (1.20) but only locally.

In practice, i.e. if rounding errors are involved, the Newton method can be unstable. Since the matrix A is involved in the iteration only as starting matrix, the error depends on the condition number of the iterates X_i and accumulated rounding errors, see [23].

Actually, we are mostly more interested in the action of $\text{sign}(A)$ on a vector than in computing $\text{sign}(A)$ itself.

1.2.2 Rational approximation of the sign function

Having a rational approximation for the (scalar) sign function given in the form

$$\text{sign}(t) \approx r(t) = \sum_{i=1}^{s} \omega_i \frac{t}{t^2 - \sigma_i}, \tag{1.22}$$

the action of $\text{sign}(Q)$ on a vector v can be approximated as

$$\text{sign}(Q)v \approx Q \sum_{i=1}^{s} \omega_i (Q^2 - \sigma_i I)^{-1} v. \tag{1.23}$$

The right-hand side of the approximation (1.23) involves the solution of s linear systems $(Q^2 - \sigma_i I)x^{(i)} = v$. The systems only differ in the shift σ_i,

all s systems have the same right-hand side. This is a useful quality when the solutions to these systems are approximated in Krylov subspaces. Since Krylov subspaces are shift-invariant, i.e.

$$K_m(A, r_0) = K_m(A - \sigma I, r_0),$$

only one Krylov subspace has to be built for all s systems. For all classical Krylov subspace methods there exist multishift variants to solve s shifted systems simultaneously in the same Krylov subspace, see Section 3.1 and [13, 16, 17, 43].

Let $x_m^{(i)} \in x_0 + K_m(Q^2, r_0)$, $i = 1, \ldots, s$, be approximations to $x^{(i)}$, computed with a multishift method. Then the rational/multishift approximation to the action of the sign function is

$$\text{sign}(Q)v \approx Q \sum_{i=1}^{s} \omega_i (Q^2 - \sigma_i I)^{-1} v \approx Q \sum_{i=1}^{s} \omega_i x_m^{(i)}.$$

To achieve a required accuracy ϵ for the error, the two approximation levels have to be investigated. How accurate does the rational approximation have to be and when can the multishift iteration be stopped?

For the error of the rational/multishift approximation it holds

$$\begin{aligned} \|e_m^s\|_2 &= \|\text{sign}(Q)v - Q \sum_{i=1}^{s} \omega_i x_m^{(i)}\|_2 \\ &\leq \|\text{sign}(Q)v - r(Q)v\|_2 + \|r(Q)v - Q \sum_{i=1}^{s} \omega_i x_m^{(i)}\|_2. \end{aligned}$$

Therefore, to achieve the error bound

$$\|e_m^s\|_2 \leq \epsilon \|v\|_2,$$

we require for the rational approximation

$$\|\text{sign}(Q)v - r(Q)v\|_2 \leq \frac{\epsilon}{2} \|v\|_2 \tag{1.24}$$

and

$$\|r(Q)v - Q \sum_{i=1}^{s} \omega_i x_m^{(i)}\|_2 \leq \frac{\epsilon}{2} \|v\|_2 \tag{1.25}$$

for the multishift approximation.

For the rational approximation $r(t)$, Neuberger [35, 36] proposed

$$r_m(t) = \frac{(t+1)^{2m} - (t-1)^{2m}}{(t+1)^{2m} + (t-1)^{2m}}. \tag{1.26}$$

Actually, (1.26) just summarizes $\log_2(m) + 1$ steps of the iteration (1.20), more precisely $X_m = \frac{1}{r_{2^{m-1}}}$. The rational approximation (1.26) can be written in the form (1.22), see [35], with

$$\omega_i = \frac{1}{s}\cos^{-2}\left(\frac{\pi}{2s}(i-\frac{1}{2})\right), \qquad \sigma_i = \tan^2\left(\frac{\pi}{2s}(i-\frac{1}{2})\right).$$

Hermitian matrices can be approximated with less poles using the Zolotarev best rational approximation, [26, 44, 47]. Let $\mathcal{R}_{i,j}$ denote the space of rational functions $r = p/q$ with polynomials p, q and $\deg(p) \le i$, $\deg(q) \le j$. An approximation $r \in \mathcal{R}_{i,j}$ is called best approximation for a function f on the set $\mathcal{I}$ if it minimizes

$$\sup_{t\in\mathcal{I}} |f(t) - r(t)|.$$

The Zolotarev best approximation is a best rational approximation for $\mathrm{sign}(A)$ on the set $[-\lambda_{max}, -\lambda_{min}] \cup [\lambda_{min}, \lambda_{max}] \supset \mathrm{spec}(A)$.

Proposition 1.19. (Zolotarev [47]) Let $\tilde{r} \in \mathcal{R}_{m-1,m}$ be the best relative approximation to $t^{-1/2}$ on $[1, (\lambda_{max}/\lambda_{min})^2]$. Then the best approximation to the sign function from $\mathcal{R}_{2m-1,2m}$ on $[-\lambda_{max}/\lambda_{min}, -1] \cup [1, \lambda_{max}/\lambda_{min}]$ is

$$r(t) = t\tilde{r}(t^2),$$

and on $[-\lambda_{max}, -\lambda_{min}] \cup [\lambda_{min}, \lambda_{max}]$ the best approximation to the sign function is $r(t/\lambda_{min})$.

The rational approximation $\tilde{r}(t)$ is given in terms of the Jacobian elliptic function $\mathrm{sn}(w, \kappa)$ which is defined by the elliptic integral

$$w = \int_0^x \frac{1}{\sqrt{(1-t^2)(1-\kappa^2 t^2)}} dt.$$

Theorem 1.20. (Zolotarev [47]) The best relative approximation $\tilde{r}(t)$ from $\mathcal{R}_{m-1,m}$ for $t^{-1/2}$ on $[1, (\lambda_{max}/\lambda_{min})^2]$ is given by

$$\tilde{r}(t) = D\frac{\prod_{i=1}^{m-1}(t + c_{2i})}{\prod_{i=1}^{m}(t + c_{2i-1})},$$

where

$$c_i = \frac{\mathrm{sn}^2(iK/(2m); \kappa)}{1 - \mathrm{sn}^2(iK/(2m); \kappa)},$$

with $\kappa = \sqrt{1 - (\lambda_{min}/\lambda_{max})^2}$ and K is the complete elliptic integral. The constant D is uniquely determined by the condition

$$\max_{t\in[1,(\lambda_{max}/\lambda_{min})^2]}(1 - \sqrt{t}\cdot\tilde{r}(t)) = -\min_{t\in[1,(\lambda_{max}/\lambda_{min})^2]}(1 - \sqrt{t}\cdot\tilde{r}(t)).$$

$\lambda_{\min}/\lambda_{\max}$	number of poles (Neuberger)	number of poles (Zolotarev)
10^{-1}	15	8
10^{-2}	48	13
10^{-3}	152	17
10^{-4}	478	22
10^{-5}	1512	26

Table 1.3: Number of poles necessary to achieve an accuracy of 10^{-8}

The number of poles needed to achieve a required accuracy is significantly smaller using the Zolotarev rational approximation than using the Neuberger rational approximation, see Table 1.3.

1.3 Neuberger Overlap operator

The application we use to motivate and demonstate the methods presented in the following chapters arrises from lattice QCD (quantum chromodynamics). Here a core numerical task is the solution of a linear system with the Neuberger overlap operator [34]. The Neuberger overlap operator implements chiral symmetry on the lattice since it provides an exact solution of the Ginsparg-Wilson relation [19] $\gamma_5 D + D\gamma_5 = aD\gamma_5 D$ (a is the lattice spacing, γ_5 and D as defined below).

The Neuberger overlap operator reads

$$D = \rho I + \Gamma_5 \text{sign}(Q) \tag{1.27}$$

with a mass parameter $\rho \in \mathbb{R}$, $\rho > 1$, and $Q = \Gamma_5 D_W$, $D_W \in \mathbb{C}^{n\times n}$ being the Wilson-Dirac operator[1].

The Wilson-Dirac operator $D_W = I - \kappa M \in \mathbb{C}^{12N^4 \times 12N^4}$ represents a nearest neighbour coupling on a four-dimensional space-time lattice

$$\Omega = \{x = (x_i, x_j, x_k, x_l), \quad 1 \le i, j, k, l \le N\}.$$

The coupling parameter $\kappa \in \mathbb{R}$ corresponds to the quark mass m_q in the following way

$$\kappa = \frac{\kappa_c}{2m_q\kappa_c + 1},$$

with κ_c defined by the eigenvalues λ_i of D_W, $i = 1, \ldots, n$,

$$\kappa_c = \frac{1}{\max \text{Re}(\lambda_i)}.$$

[1]For the definition of Γ_5 see Appendix A.

Obviously, small quark masses correspond to κ values close to κ_c.

The definition of the hopping term M of the Wilson-Dirac operator involves the matrices $\gamma_1, \ldots, \gamma_4$[2] and the lattice gauge fields represented by matrices $U_i \in \mathrm{SU}(3)$, $i = 1, \ldots, 4$.

Definition 1.21. The special unitary Group $\mathrm{SU}(3) \subset \mathbb{C}^{3\times 3}$ is defined as

$$\mathrm{SU}(3) = \{U \in \mathbb{C}^{3\times 3} : U^H U = I_3, \det(U) = 1\}.$$

The hopping term $M = ((M)_{xy})_{x,y=1,\ldots,N^4}$ consists of the 12×12-blocks

$$(M)_{xy} = \sum_{i=1}^{4} (I - \gamma_i) \otimes U_i(x)\delta_{x,y-e_i} + (I + \gamma_i) \otimes U_i^H(x - e_i)\delta_{x,y+e_i}$$

where $x, y \in \Omega$ such that $x \pm e_i$ describes the neighbours of x with respect to the i-th dimension.

Additionaly, a chemical potential can be introduced. For non-zero chemical potential μ the Wilson-Dirac operator reads $D_W^\mu = I - \kappa M^\mu$ with

$$\begin{aligned}(M^\mu)_{xy} \; = &\sum_{i=1}^{3} (I - \gamma_i) \otimes U_i(x)\delta_{x,y-e_i} + (I + \gamma_i) \otimes U_i^H(x - e_i)\delta_{x,y+e_i} \\ &+ e^{\mu}(I - \gamma_4) \otimes U_4(x)\delta_{x,y-e_4} + e^{-\mu}(I + \gamma_4) \otimes U_4^H(x - e_4)\delta_{x,y+e_4}.\end{aligned}$$

The definitions of D_W and D_W^μ are consistent, i.e. $D_W^0 = D_W$.

Whether the chemical potential μ is zero or non-zero influences the basic properties of the Wilson-Dirac operator and thus of the Neuberger overlap operator D. One important property that is influenced is the Γ_5-symmetry:

Definition 1.22. A matrix D is called Γ_5-*symmetric* or Γ_5-*hermitian* if

$$\Gamma_5 D = D^H \Gamma_5.$$

At zero chemical potential, i.e. $\mu = 0$, we find

- the Wilson-Dirac operator D_W^0 is Γ_5-hermitian,
- $Q = \Gamma_5 D_W^0$ is hermitian,
- $\mathrm{sign}(Q)$ is hermitian,
- $\Gamma_5 \mathrm{sign}(Q)$ is unitary,

[2] For the definition of $\gamma_1, \ldots, \gamma_4$ see Appendix A.

- the Neuberger overlap operator D is shifted unitary.

Since at zero chemical potential the Neuberger operator is not hermitian, one could expect that Krylov subspace methods require long recurrences. But the structure as a shifted unitary matrix can be exploited to build effective Krylov subspace methods with short recurrences, see Chapter 2. These methods need a matrix vector product with sign(Q) in every step. Since the matrix Q is hermitian there exist effective approximation methods for the matrix vector product with sign(Q): The Zolotarev best rational approximation can be used resulting in a small number of shifts, and for the multishift approximation multishift CG works with short recurrences.

At non-zero chemical potential, i.e. $\mu \neq 0$, we find

- the Wilson-Dirac operator D_W^μ is not Γ_5-hermitian,
- neither Q nor sign(Q) are hermitian,
- $\Gamma_5$sign(Q) is not unitary.

Since at non-zero chemical potential the Neuberger overlap operator has no structure to exploit, classical Krylov subspace methods for non-hermitian matrices have to be used. Long recurrences cannot be avoided. Since Q is not hermitian either, the matix vector product with sign(Q) needed in every step of these methods cannot be approximated with the Zolotarev rational approximation. The number of poles is thus significantly higher. Furthermore, for the multishift approximation of the rational approximation, CG has to be replaced by FOM or GMRES resulting in long recurrences or by BiCG or QMR.
In Chapter 3 we investigate the two eigenvalue deflation methods to reduce the computational cost for approximating sign(Q)b. Applied to FOM and GMRES they are combined with restarts.

Chapter 2

Krylov subspace methods for shifted unitary matrices

Computing a basis for a Krylov subspace is a central task in Krylov subspace methods. For hermitian matrices we have with the Lanczos method and its three term recurrence a relatively cheap method compared to the Arnoldi method. While for general non-hermitian matrices there is no such method with short recurrence, there is one for a special class of non-hermitian matrices, namely unitary matrices.

Since Krylov subspaces are shift invariant, short recurrence methods to build a Krylov subspace basis therefore exist for matrices of the form

$$A = \rho I + U$$

with U a unitary matrix.

2.1 The Faber-Manteuffel theorem

The theorem of Faber and Manteuffel [12] gives a characterization for matrices that allow a short recurrence to build an orthogonal basis for $K_m(A, r_0)$. At first glance this famous theorem seems to doom all effort concerning unitary matrices to fail as it states A being normal(s) as a necessary condition for the existence of a short s-term recurrence. And unitary matrices are not normal(s) for s small, in general.

Definition 2.1.

1. A matrix A is *normal* if $A^H = p(A)$ for some polynomial p.
2. A matrix A is *normal(s)* if

$$s = \min\{d = \deg(p) : p \text{ polynomial}, A^H = p(A)\}.$$

Hermitian matrices – for which with the Lanczos iteration we already know a short recursion – are normal(1) as $A^H = A = p(A)$ with $p(t) = t$ and $\deg(p) = 1$ is minimal.

For unitary matrices we have $A^H = A^{-1}$, such that $A^H = p(A)$ with $\deg(p) \leq n - 1$. In this case, the degree of the polynomial p is related to the degree of the minimal polynomial of A:

Proposition 2.2. If m is the degree of the minimal polynomial of the non-singular matrix A, then $A^{-1} = p(A)$ with $\deg(p) = m - 1$.

Proof. Let $q_m(t) = \alpha_0 + \alpha_1 t + \cdots + \alpha_m t^m$ be the mimimal polynomial of A, thus

$$q_m(A) = \alpha_0 + \alpha_1 A + \cdots + \alpha_m A^m = 0.$$

Since A is non-singular, α_0 is non-zero and we can write

$$A(\alpha_1 + \cdots + \alpha_m A^{m-1}) = -\alpha_0 I.$$

Therefore

$$A^{-1} = \frac{-1}{\alpha_0}(\alpha_1 + \cdots + \alpha_m A^{m-1}) = p_{m-1}(A)$$

with $\deg(p_{m-1}) = m - 1$.

□

Thus, unitary matrices are normal$(m-1)$ for m the degree of the mimimal polynomial, but of course, m depends on the actual matrix such that unitary matrices are not normal(s) for small s in general.

The theorem of Faber and Manteuffel tells for which matrices, namely those which are normal(s), an $(s+2)$-term recurrence is possible.

Theorem 2.3 (Faber-Manteuffel)**.**
A matrix A allows a $(s+2)$-term recurrence of the form

$$v_{j+1} = Av_j - \sum_{i=j-s}^{j} h_{i,j} v_i \quad , \qquad h_{i,j} = \frac{v_i^H A v_j}{v_i^H v_i} \tag{2.1}$$

to construct an orthogonal basis $\{v_1, \ldots, v_m\}$ of $K_m(A, r_0)$ if and only if A is normal(s).

Proof. [11, 29] □

For hermitian matrices Theorem 2.3 guarantees a 3-term-recurrence ($s = 1$), and the Lanczos method gives a realization of such.

For unitary matrices a short recurrence of the form (2.1) is thus impossible. But it is in fact possible to construct an orthogonal basis of $K_m(A, r_0)$ for A unitary with a short recurrence, using a recurrence formula of a different form than (2.1). Gragg [21] presented a realization of such a short recurrence for unitary matrices and Barth and Manteuffel [4] gave a theoretical foundation using a generalized definition of being normal.

Definition 2.4. A matrix A is *normal(ℓ, m)* if A is normal and

$$A^H q_m(A) = p_\ell(A)$$

for polynomials p and q of degree ℓ and m, respectively.

Obviously normal(ℓ, 0) matrices are normal(ℓ). Hermitian matrices are thus normal(1, 0). Unitary matrices are normal(0, 1) as in this case we have $A^H q(A) = p(A)$ with $q(t) = t$ and $p(t) = 1$.

Barth and Manteuffel showed in [4] that being normal(ℓ, m) is a sufficient condition for a short recursion, though of a form different from that in (2.1).

Theorem 2.5. If a matrix A is normal(ℓ, m), it allows an $\ell + m + 2$-recurrence to construct an orthogonal basis $\{v_1, \ldots, v_m\}$ of $K_m(A, r_0)$ of the form

$$v_{j+1} = \sum_{i=j-m}^{j} \widehat{h}_{i,j} A v_i - \sum_{i=j-\ell}^{j} h_{i,j} v_i. \tag{2.2}$$

Proof. [4] □

For hermitian matrices (2.2) results in the well known form of its 3-term recurrence, while for unitary matrices we get a 3-term recurrence of the form

$$v_{j+1} = \sum_{i=j-1}^{j} \widehat{h}_{i,j} A v_i - h_{j,j} v_j. \tag{2.3}$$

The $\ell + m + 2$-recurrence (2.2) can be reformulated as a coupled recursion. For unitary matrices this leads to the formulation of Gragg [21, 27]:

$$\begin{aligned} \sigma_j v_{j+1} &= A v_j + \gamma_j \widehat{v}_j \\ \widehat{v}_{j+1} &= \sigma_j \widehat{v}_j + \bar{\gamma}_j v_{j+1} \end{aligned} \tag{2.4}$$

In [4], a possible breakdown of the single recursion (2.2) was identified that does not occur in the coupled formulation.

In the following section two realizations of such short recurrences to construct a basis for $K_m(A, r_0)$ with unitary A are presented. The first one (unitary Arnoldi) uses a single recursion of the form (2.3) while the second

one (isometric Arnoldi) uses a coupled recursion (2.4). Both are thus realizations of the recursion (2.2).

These methods can then be combined with the Galerkin or the minimal residual condition to gain tailor-made methods for shifted unitary matrices.

2.2 Short recurrence Arnoldi

2.2.1 Unitary Arnoldi

The unitary Arnoldi method was introduced by Borici in [7, 8, 9]. As a foundation, the following statement found in [39] was used.

> Almost every orthogonal matrix U has a decomposition $U = LR^{-1}$, where L is lower and R upper triangular matrix.

The idea is to apply this decomposition to the matrix $H_{m_0} = {V_{m_0}}^H U V_{m_0}$ that is computed in the last step of the Arnoldi method, i.e., when the columns of V_{m_0} span an A-invariant subspace. While for $k < m_0$ the matrices H_k are not unitary, H_{m_0} is unitary.

Lemma 2.6. Let U be a unitary matrix and V_m the matrix obtained by the Arnoldi method started with $\tilde{v}_1 = r_0$. Then the matrix $H_{m_0} = {V_{m_0}}^H U V_{m_0}$ is unitary for m_0 the degree of r_0 with respect to U.

Proof. While for $k < m$ the Arnoldi relation (1.8) holds, for $k = m_0$ it holds

$$UV_{m_0} = V_{m_0} H_{m_0}.$$

Therefore

$$H_{m_0}^H H_{m_0} = H_{m_0}^H V_{m_0}^H V_{m_0} H_{m_0} = V_{m_0}^H U^H U V_{m_0} = I.$$

□

What we need to know is, whether an LU-decomposition for H_{m_0} exists or not. It does, if H_{m_0} and its minors, i.e., the matrices H_k are regular for $k = 1, \ldots, m_0,$. The following lemma (see [8]) helps to answer this question.

Lemma 2.7. For a unitary matrix U, the matrix $\widehat{H}_k$ resulting from the Arnoldi method has orthogonal columns.

Proof. Multiplying both sides of the Arnoldi relation

$$UV_k = V_{k+1}\widehat{H}_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T$$

from the left by $(UV_k)^H$ results in

$$\begin{aligned} I &= V_k^H U^H V_k H_k + h_{k+1,k} V_k^H U^H v_{k+1} e_k^T \\ &= H_k^H H_k + h_{k+1,k} (v_{k+1}^H U V_k)^H e_k^T \\ &= H_k^H H_k + h_{k+1,k} h_{k+1,k} e_k e_k^T \\ &= \widehat{H}_k^H \widehat{H}_k. \end{aligned}$$

□

From Lemma 2.7 we can read under which circumstances the desired LU-decomposition exists.

Theorem 2.8. For a unitary matrix U the matrices $H_k = V_k^H U V_k$ resulting from the Arnoldi method are regular if and only if $v_i^H U v_k \neq 0$ for at least one index $i \leq k$.

Proof. From Lemma 2.7 we know that

$$H_k^H H_k = I - h_{k+1,k}^2 e_k e_k^T.$$

This shows that H_k is regular iff $h_{k+1,k} \neq 1$. By construction it is

$$h_{k+1,k} = \|U v_k - \sum_{i=1}^{k} v_i^H U v_k v_i\|_2.$$

Since $\|U v_k\|_2 = 1$ and $\sum_{i=1}^{k} v_i^H U v_k v_i$ is the orthogonal projection of $U v_k$ onto $K_k(U, r_0)$, this proves that $h_{k+1,k} = 1$ iff $v_i^H U v_k = 0$, $i = 1, \ldots, k$.

□

Theorem 2.8 tells that not all unitary matrices have an LU-decomposition. From now on we assume $h_{k+1,k} \neq 1$ for $k = 1, \ldots, m_0$.

Following Rutishauser – and assuming that the LU-decomposition for H_{m_0} exists – we write $H_{m_0} = L_{m_0} R_{m_0}^{-1}$ with a lower triangular matrix $L_{m_0} = (l_{i,j})$ and an upper triangular matrix $R_{m_0} = (r_{i,j})$. Furthermore, H_{m_0} is upper Hessenberg so that L_{m_0} is in fact bidiagonal, and since H_{m_0} is unitary, L_{m_0} and R_{m_0} satisfy

$$L_{m_0}^H L_{m_0} = R_{m_0}^H R_{m_0},$$

so that R_{m_0} is bidiagonal, too.

To find short recurrences for the Arnoldi vectors we use the LU-decomposition of H_{m_0} to write the Arnoldi relations as

$$V_{m_0}{}^H U V_{m_0} R = L \tag{2.5}$$

or

$$UV_{m_0}R = V_{m_0}L. \tag{2.6}$$

W.l.o.g. let R have entries 1 on the diagonal. With (2.5) we get

$$\begin{aligned} l_{1,1} &= v_1^H U v_1 \\ l_{2,1} &= v_2^H U v_1 \\ \tilde{v}_2 &= U v_1 - l_{11} v_1 \end{aligned}$$

and the following recursions for $r_{i,j}$, $l_{i,j}$ and $\tilde{v}_i$:

$$\begin{aligned} r_{k-1,k} &= -\frac{v_{k-1}^H U v_k}{v_{k-1}^H U v_{k-1}} \\ l_{k,k} &= r_{k-1,k} v_k^H U v_{k-1} + v_k^H U v_k \\ l_{k,k-1} &= r_{k-2,k-1} v_k^H U v_{k-2} + v_k^H U v_{k-1} \\ \tilde{v}_{k+1} &= r_{k-1,k} U v_{k-1} + U v_k - l_{k,k} v_k. \end{aligned}$$

The orthonormal vectors v_i result from normalizing $\tilde{v}_i$. The norm of $\tilde{v}_i$ directly gives $l_{i,i-1}$ as the following lemma shows.

Lemma 2.9. With $l_{k+1,k}$ and $\tilde{v}_k$ as above the following holds

$$l_{k+1,k} = \|\tilde{v}_{k+1}\|_2.$$

Proof. Obviously $l_{k+1,k} = v_{k+1}^H \tilde{v}_{k+1}$. As $v_{k+1} = \tilde{v}_{k+1}/\|\tilde{v}_{k+1}\|_2$, the proof is completed.

□

Algorithm 2.10. Unitary Arnoldi method

{**Input** $m \leq m_0$, r_0, unitary matrix U}

$v_1 = r_0$
$l_{1,1} = v_1^H U v_1$
$\tilde{v}_2 = U v_1 - l_{1,1} v_1$
$l_{2,1} = \|\tilde{v}_2\|_2$
for k = 2, 3, . . . , m **do**
  $v_k = \tilde{v}_k / l_{k,k-1}$
  $r_{k-1,k} = -\frac{v_{k-1}^H U v_k}{v_{k-1}^H U v_{k-1}}$
  $l_{k,k} = v_k^H U v_{k-1} r_{k-1,k} + v_k^H U v_k$
  $\tilde{v}_{k+1} = U v_{k-1} r_{k-1,k} + U v_k - l_{k,k} v_k$
  $l_{k+1,k} = \|\tilde{v}_{k+1}\|_2$
end for

2.2.2 Isometric Arnoldi

The isometric Arnoldi method was first introduced by Gragg [21]. The basic idea is to write the upper Hessenberg matrix

$$H_{m_0} = {V_{m_0}}^H U V_{m_0}$$

that we get in the last step of the ordinary Arnoldi method as a product of unitary matrices. To see that this is possible, let $H_{m_0} = Q_{m_0} R_{m_0}$ be the QR-decomposition of H_{m_0}. From Lemma 2.6 we know that H_{m_0} is unitary and thus

$$R_{m_0}^H R_{m_0} = R_{m_0}^H Q_{m_0}^H Q_{m_0} R_{m_0} = H_{m_0}^H H_{m_0} = I,$$

i.e., R_{m_0} is unitary (and therefore diagonal).

Realizing the QR-decompostion with Givens rotations it is actually

$$H_{m_0} = G_1(\gamma_1) G_2(\gamma_2) \cdots G_{m_0-1}(\gamma_{m_0-1}) \tilde{G}_{m_0}(\tilde{\gamma}_{m_0}),$$

where $G_i(\gamma_i)$ are the Givens matrices

$$G_i(\gamma_i) = \begin{bmatrix} I_{i-1} & & & \\ & -\gamma_i & \sigma_i & \\ & \sigma_i & \bar{\gamma}_i & \\ & & & I_{m_0-i-1} \end{bmatrix} \in \mathbb{C}^{m_0 \times m_0}$$

with $\gamma_i \in \mathbb{C}$, $\sigma_i \in \mathbb{R}^+$ and $|\gamma_i|^2 + \sigma_i^2 = 1$, and

$$\tilde{G}_{m_0}(\tilde{\gamma}_{m_0}) = \operatorname{diag}(1, \ldots, 1, -\tilde{\gamma}_{m_0})$$

with $\tilde{\gamma}_{m_0} \in \mathbb{C}$, $|\tilde{\gamma}_{m_0}| = 1$.

Thus in the last step it holds

$$U V_{m_0} = V_{m_0} H_{m_0} = V_{m_0} G_1(\gamma_1) G_2(\gamma_2) \ldots G_{m_0-1}(\gamma_{m_0-1}) \tilde{G}_{m_0}(\tilde{\gamma}_{m_0}).$$

By comparing columns we get for $k < m_0$

$$U V_k = V_{k+1} \widehat{H}_k, \tag{2.7}$$

where

$$\widehat{H}_k = G_1(\gamma_1) G_2(\gamma_2) \ldots G_{k-1}(\gamma_{k-1}) \widehat{G}_k(\gamma_k),$$

and

$$\widehat{G}_k(\gamma_k) = \begin{bmatrix} I_{k-1} & & \\ & -\gamma_k & \\ & \sigma_k & \end{bmatrix} \in \mathbb{C}^{(k+1)\times k}.$$

Note that now the Givens matrices are $G_i(\gamma_i) \in \mathbb{C}^{(k+1)\times(k+1)}$.

To determine the desired recurrences we take a look at (2.7). The first Givens matrix $G_1(\gamma_1)$ produces

$$V_k G_1(\gamma_1) = [-\gamma_1 v_1 + \sigma_1 v_2, \sigma_1 v_1 + \bar{\gamma}_1 v_2, v_3, \ldots, v_k].$$

Writing $\widehat{v}_2 = \sigma_1 v_1 + \bar{\gamma}_1 v_2$, the second Givens matrix $G_2(\gamma_2)$ produces

$$V_k G_1(\gamma_1) G_2(\gamma_2) = [-\gamma_1 v_1 + \sigma_1 v_2, -\gamma_2 \widehat{v}_2 + \sigma_2 v_3, \sigma_2 \widehat{v}_2 + \bar{\gamma}_2 v_3, v_4, \ldots, v_k],$$

Further investigating the effect of the Givens matrices on V_k, we get

$$\widehat{v}_{k+1} = \sigma_k \widehat{v}_k + \bar{\gamma}_k v_{k+1}$$

and

$$v_{k+1} = \sigma_k^{-1}(U v_k + \gamma_k \widehat{v}_k)$$

with $-\gamma_k = \widehat{v}_k^H U v_k$.

These coupled recurrences of v_k and $\widehat{v}_k$ allow short recurrences in this case. As mentioned before, the coupled recursion could be written as a single recursion, see [4].

Algorithm 2.11. Isometric Arnoldi method

{**Input** $m \leq m_0$, r_0, unitary matrix U}

$v_1 = \frac{r_0}{\|r_0\|_2}$; $\widehat{v}_1 = v_1$
for $k = 1, 2, \ldots, m-1$ **do**
  $u = U v_k$
  $\gamma_k = -\widehat{v}_k^H u$
  $\sigma_k = ((1 - |\gamma_k|)(1 + |\gamma_k|))^{1/2} = \|u + \gamma_k \widehat{v}_k\|_2$
  $v_{k+1} = \sigma_k^{-1}(u + \gamma_k \widehat{v}_k)$
  $\widehat{v}_{k+1} = \sigma_k \widehat{v}_k + \bar{\gamma}_k v_{k+1}$
end for
$\gamma_m = -\widehat{v}_m^H U v_m$; $\sigma_m = ((1 - |\gamma_m|)(1 + |\gamma_m|))^{1/2}$

The vectors $v_1, \ldots, v_m$ and the vectors $\widehat{v}_1, \ldots, \widehat{v}_m$ computed with the isometric Arnoldi method are normal as the following lemma shows. The vectors $v_1, \ldots, v_m$ are of course orthogonal.

Lemma 2.12. In the isometric Arnoldi method the following holds

$$v_j^H v_j = \widehat{v}_j^H \widehat{v}_j = 1 \quad \text{and} \quad v_{j+1}^H \widehat{v}_j = 0.$$

Proof. It holds:

$$\widehat{v}_1^H \widehat{v}_1 = v_1^H v_1 = r_0^H r_0 / \|r_0\|_2^2 = 1$$

$$v_2^H \widehat{v}_1 = \frac{1}{\sigma_1}(u + \gamma_1 \widehat{v}_1)^H \widehat{v}_1 = \frac{1}{\sigma_1}(u^H \widehat{v}_1 + \bar{\gamma}_1 \widehat{v}_1^H \widehat{v}_1) = \frac{1}{\sigma_1}(-\bar{\gamma}_1 + \bar{\gamma}_1) = 0$$

Via induction we get:

$$\begin{aligned} v_{j+1}^H v_{j+1} &= \frac{1}{\sigma_j \bar{\sigma}_j} (u + \gamma_j \widehat{v}_j)^H (u + \gamma_j \widehat{v}_j) \\ &= \frac{1}{\sigma_j \bar{\sigma}_j} \left(u^H u + \gamma_j u^H \widehat{v}_j + \bar{\gamma}_j \widehat{v}_j^H u + \bar{\gamma}_j \gamma_j \widehat{v}_j^H \widehat{v}_j \right) \\ &= \frac{1}{\sigma_j \bar{\sigma}_j} (1 + \gamma_j(-\bar{\gamma}_j) + \bar{\gamma}_j(-\gamma_j) + \bar{\gamma}_j \gamma_j) \\ &= \frac{1}{\sigma_j \bar{\sigma}_j} (1 - \gamma_j \bar{\gamma}_j) \\ &= 1 \end{aligned}$$

$$\begin{aligned} \widehat{v}_{j+1}^H \widehat{v}_{j+1} &= (\sigma_j \widehat{v}_j + \bar{\gamma}_j v_{j+1})^H (\sigma_j \widehat{v}_j + \bar{\gamma}_j v_{j+1}) \\ &= \bar{\sigma}_j \sigma_j \widehat{v}_j^H \widehat{v}_j + \bar{\sigma}_j \bar{\gamma}_j \widehat{v}_j^H v_{j+1} + \gamma_j \sigma_j v_{j+1}^H \widehat{v}_j + \gamma_j \bar{\gamma}_j v_{j+1}^H v_{j+1} \\ &= \bar{\sigma}_j \sigma_j + \gamma_j \bar{\gamma}_j \\ &= 1 \\ v_{j+2}^H \widehat{v}_{j+1} &= \frac{1}{\bar{\sigma}_{j+1}} (u + \gamma_{j+1} \widehat{v}_{j+1})^H \widehat{v}_{j+1} \\ &= \frac{1}{\bar{\sigma}_{j+1}} (u^H \widehat{v}_{j+1} + \bar{\gamma}_{j+1} \widehat{v}_{j+1}^H \widehat{v}_{j+1}) \\ &= \frac{1}{\bar{\sigma}_{j+1}} (-\bar{\gamma}_{j+1} + \bar{\gamma}_{j+1}) \\ &= 0 \end{aligned}$$

□

If implemented as in Algorithm 2.11 the isometric Arnoldi method suffers from numerical instabilities:
In each step we have $\sigma_j = \|u + \gamma_j \widehat{v}_j\|_2 = \|v_{j+1}\|_2$. Therefore v_{j+1} has norm one, and this is obtained by explicit normalization. The normalization of $\widehat{v}_{j+1}$ is only implicit and not done explicitly in the algorithm. It actually vanishes during the iteration due to rounding errors. At the same time we lose the orthogonality of the vectors v_j. The right plot of Figure 2.1 shows that even the orthogonality of the last four vectors gets lost soon when we rely on the implicit normalization of Algorithm 2.11.

To preserve orthogonality, $\widehat{v}_{j+1}$ has to be normalized explicitly in each step. Of course, we can not expect to prevent the general loss of orthogonality, i.e., the last vectors will not be orthogonal to the first ones after a certain number of steps, see left plot of Figure 2.1. But every vector will be orthogonal to a reasonable number of its predecessors, see right plot of Figure 2.1.

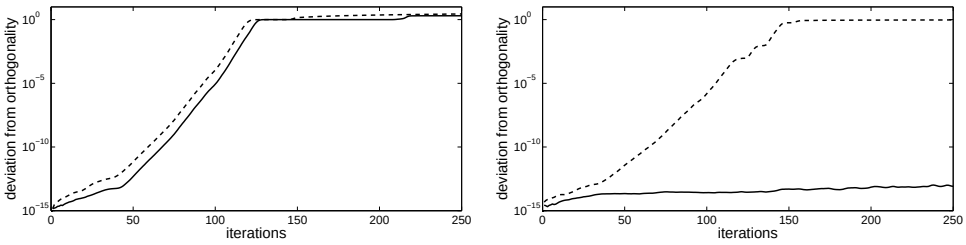


Figure 2.1: The isometric Arnoldi method with (solid line) and without (dashed line) explicit normalization. Left plot shows $\|V^H V - I\|_F$, right plot shows $\|[v_{i-3}, \ldots, v_i]^H [v_{i-3}, \ldots, v_i] - I\|_F$.

2.3 Shifted unitary methods

2.3.1 SUOM

In this section we combine the unitary Arnoldi method with a Galerkin condition. The resulting method is called SUOM (Shifted Unitary Orthogonal Method) [7, 8, 9]. We will see that not only the basis of the Krylov subspace can be calculated with a short recurrence by the unitary Arnoldi method but also the SUOM iterates can be calculated by a simple update. Therefore, only a constant number of basis vectors of the Krylov subspace has to be stored.

Let $A = \rho I + U$ with a unitary matrix U so that the unitary Arnoldi method can be applied, i.e., the decomposition ${V_{m_0}}^H U V_{m_0} = L_{m_0} R_{m_0}^{-1}$ exists.

To solve

$$ {V_k}^H (\rho I + U) V_k y_k = {V_k}^H b, $$

we set $y_k = R_k \zeta_k$. Instead of solving a system with the upper Hessenberg matrix ${V_k}^H (\rho I + U) V_k = \rho I + H_k$, the system to solve is then

$$ (\rho R_k + L_k) \zeta_k = {V_k}^H b $$

with $(\rho R_k + L_k)$ tridiagonal. This system can be solved using its LU-decomposition.

Lemma 2.13. If the shift parameter $\rho > 1$ and therefore A is positive real, the LU-decomposition $(\rho R_k + L_k) = \Lambda_k \Psi_k$ exists.

Proof. If the matrix A is positive real, so are the matrices $H_k^{(A)} = {V_k}^H A V_k$ for $k = 1, \ldots, m_0$. Thus, all minors of $H_{m_0}^{(A)}$ are non-singular and $H_{m_0}^{(A)}$ has an LU-decomposition $H_{m_0}^{(A)} = \Lambda_{m_0} \widehat{\Psi}_{m_0}$. Since $(\rho R_{m_0} + L_{m_0}) = H_{m_0}^{(A)} R_{m_0}$ it has an LU-decomposition $(\rho R_{m_0} + L_{m_0}) = \Lambda_{m_0} \Psi_{m_0}$ with $\Psi_{m_0} = \widehat{\Psi}_{m_0} R_{m_0}$. □

Since $(\rho R_k + L_k)$ is tridiagonal, Λ_k and Ψ_k are bidiagonal. W.l.o.g we can assume Λ_k having ones on the diagonal. The other entries $\lambda_{i,j}$ and $\psi_{i,j}$ can be calculated easily with the following recurrences:

$$\begin{aligned} \psi_{1,1} &= \rho + l_{1,1} \\ \psi_{k-1,k} &= \rho r_{k-1,k} \\ \lambda_{k,k-1} &= l_{k,k-1}/\psi_{k-1,k-1} \\ \psi_{k,k} &= \rho + l_{k,k} - \lambda_{k,k-1}\psi_{k-1,k} \end{aligned}$$

With these notations we can write the SUOM iterates as

$$\begin{aligned} x_k &= V_k R_k (\rho R_k + L_k)^{-1} {V_k}^H b \\ &= V_k R_k {\Psi_k}^{-1} {\Lambda_k}^{-1} {V_k}^H b. \end{aligned}$$

To see that x_{k+1} can be calculated by a simple update of x_k, we seperately investigate

$$\omega_k = {\Lambda_k}^{-1} {V_k}^H b$$

and

$$Z_k = V_k R_k {\Psi_k}^{-1}.$$

For $\omega_k = (w_1, \ldots, w_k)^T$ we directly get, since $V_k^H b = \beta e_1$,

$$\begin{aligned} w_1 &= \beta \\ w_i &= -\lambda_{i,i-1} w_{i-1}, \quad i = 2, \ldots, k. \end{aligned}$$

For Z_k we compare columns in $V_k R_k = Z_k \Psi_k$

$$[V_{k-1} R_{k-1}, r_{k-1,k} v_{k-1} + v_k] = [Z_{k-1}\Psi_{k-1}, \psi_{k-1,k} z_{k-1} + \psi_{k+1,k+1} z_k]$$

and get

$$\begin{aligned} z_1 &= \psi_{1,1}^{-1} v_1 \\ z_i &= \psi_{i,i}^{-1} (r_{i-1,i} v_{i-1} + v_i - \psi_{i-1,i} z_{i-1}), \quad i = 2, \ldots, k. \end{aligned}$$

The iterates are therefore updated as

$$x_{k+1} = [Z_k | z_{k+1}](\omega_k, w_{k+1})^T = x_k + w_{k+1} z_{k+1}.$$

The SUOM residuals are thus

$$r_k = -h_{k+1,k} e_k^T y_k v_{k+1} = -l_{k+1,k} \frac{1}{\psi_{k,k}} w_k v_{k+1}.$$

Algorithm 2.14. SUOM

{**Input** $m \leq m_0$, x_0 $r_0 = b - Ax_0$, ϵ}

$v_1 = r_0$
$\beta = \|v_1\|_2$
$l_{1,1} = v_1^H U v_1$
$\tilde{v}_2 = Uv_1 - l_{1,1}v_1$
$l_{2,1} = \|\tilde{v}_2\|_2$
$\psi_{1,1} = \rho + l_{1,1}$
$w_1 = \beta$
$z_1 = \psi_{1,1}^{-1} v_1$
$x_1 = x_0 + w_1 z_1$
for k = 2, 3, . . . , m **do**
  $v_k = \tilde{v}_k / l_{k,k-1}$
  $r_{k-1,k} = -\frac{v_{k-1}^H U v_k}{v_{k-1}^H U v_{k-1}}$
  $l_{k,k} = v_k^H U v_{k-1} r_{k-1,k} + v_k^H U v_k$
  $\psi_{k-1,k} = \rho r_{k-1,k}$
  $\lambda_{k,k-1} = l_{k,k-1} / \psi_{k-1,k-1}$
  $\psi_{k,k} = \rho + l_{k,k} - \lambda_{k,k-1}\psi_{k-1,k}$
  $w_k = -\lambda_{k,k-1} w_{k-1}$
  $z_k = \psi_{k,k}^{-1}(r_{k-1,k} v_{k-1} + v_k - \psi_{k-1,k} z_{k-1})$
  $x_k = x_{k-1} + w_k z_k$
  $\tilde{v}_{k+1} = Uv_{k-1} r_{k-1,k} + Uv_k - l_{k,k} v_k$
  $l_{k+1,k} = \|\tilde{v}_{k+1}\|_2$
end for

Note that for the derivation of the SUOM method, we assumed A to be positive real. If A is not positive real, the combination of the unitary Arnoldi method and a Galerkin condition can be realized using a QR-decomposition of $(\rho R_k + L_k)$ instead of using the LU-decomposition. Nevertheless, the resulting method can have breakdowns, since its iterates do not necessarily exist due to the possibility of H_k being singular.

2.3.2 SHUMR

In this section we combine the unitary Arnoldi method with the minimal residual condition instead of a Galerkin condition. As proposed by Borici [8], the resulting method is called SHUMR (SHifted Unitary Minimal Residual), although we present here a slightly different realization than the one proposed by Borici. The original SHUMR algorithm (see [8]) involves the SUOM iterates to obtain a simple update for its iterates. Since the SUOM iterates do not necessarily exist when A is not positive real, this introduces

possible breakdowns to SHUMR. As the derivation below shows, it is not necessary to use the SUOM iterates. The version we present here is thus breakdown free, even for A not positive real.

With the minimal residual condition we demand for the SHUMR iterates

$$\|r_k\|_2 = \|b - (\rho I + U)x_k\|_2 = \min_{x \in x_0 + K_k(U, r_0)} \|b - (\rho I + U)x\|_2, \tag{2.8}$$

and requiring $x_k = V_k y_k$ leads to

$$\begin{aligned} r_k &= b - (\rho I + U)V_k y_k \\ &= \beta V_{k+1} e_1 - V_{k+1}(\rho \widehat{I}_k + \widehat{H}_k) y_k, \end{aligned}$$

where $\widehat{I}_k \in \mathbb{R}^{(k+1)\times k}$ is the identity matrix expanded by an extra row of zeros.

Assuming that the unitary Arnoldi method can be applied, we know that $H_k = L_k R_k^{-1}$ and we get

$$\widehat{H}_k = \widehat{L}_k R_k^{-1}$$

with

$$\widehat{L}_k = \begin{bmatrix} L_k \\ 0 \ldots 0 \; l_{k+1,k} \end{bmatrix} \in \mathbb{C}^{(k+1)\times k}.$$

As in SUOM, we set $y_k = R_k z_k$ and get

$$\begin{aligned} r_k &= V_{k+1}(\beta e_1 - (\rho \widehat{I}_k + \widehat{L}_k R_k^{-1}) R_k z_k) \\ &= V_{k+1}(\beta e_1 - (\rho \widehat{R}_k + \widehat{L}_k) z_k) \end{aligned}$$

with $\widehat{R}_k = \widehat{I}_k R_k$.

To construct z_k we use the QR-decomposition

$$(\rho \widehat{R}_k + \widehat{L}_k) = Q_k \widehat{\Theta}_k, \; \widehat{\Theta}_k = \begin{bmatrix} \Theta_k \\ 0 \end{bmatrix}$$

with $Q_k \in \mathbb{C}^{(k+1)\times(k+1)}$ unitary and $\Theta_k \in \mathbb{C}^{k\times k}$ an upper tridiagonal matrix. The least squares problem (2.8) reads thus

$$\|\beta_k Q_k^H e_1 + \widehat{\Theta}_k z_k\|_2 = \min_{z \in \mathbb{C}^k} \|\beta_k Q_k^H e_1 + \widehat{\Theta}_k z\|_2. \tag{2.9}$$

We can recursively write Q_k^H as a product of Givens matrices

$$G_k(c_k) = \begin{bmatrix} I_{k-1} & & \\ & -c_k & s_k \\ & s_k & \bar{c}_k \end{bmatrix}, \; |c_k|^2 + s_k^2 = 1.$$

such that

$$\beta Q_k^H e_1 = \beta G_k(c_k) \begin{bmatrix} Q_{k-1}^H & \\ & 1 \end{bmatrix} e_1 = (\tau_1, \ldots, \tau_k, \widehat{\tau}_{k+1})^T,$$

with

$$\begin{aligned} \widehat{\tau}_1 &= \beta \\ \tau_k &= -c_k \widehat{\tau}_k \\ \widehat{\tau}_{k+1} &= s_k \widehat{\tau}_k. \end{aligned}$$

With the recursion for τ_k the least squares problem (2.9) is further simplified and we have

$$z_k = \Theta_k^{-1}(\tau_1, \ldots, \tau_k)^T.$$

Recall that $\rho\widehat{R}_k + \widehat{L}_k$ is tridiagonal such that column k is only affected by the Givens matrices G_k, G_{k-1}, and G_{k-2}. The k-th column of Θ_k therefore consists of the three non-zero entries

$$\begin{aligned} \theta_{k-2,k} &= s_{k-2}\rho r_{k-1,k} \\ \theta_{k-1,k} &= -c_{k-1}\bar{c}_{k-2}\rho r_{k-1,k} + s_{k-1}(\rho + l_{k,k}) \\ \theta_{k,k} &= -c_k \widehat{\theta}_k + s_k l_{k+1,k} \end{aligned}$$

with $\widehat{\theta}_k = s_{k-1}\bar{c}_{k-2}\rho r_{k-1,k} + \bar{c}_{k-1}(\rho + l_{k,k})$ and

$$\bar{c}_k = \frac{\widehat{\theta}_k}{(|\widehat{\theta}_k|^2 + l_{k+1,k}^2)^{1/2}}, \quad s_k = \frac{-l_{k+1,k}}{(|\widehat{\theta}_k|^2 + l_{k+1,k}^2)^{1/2}}.$$

After all, we have to compute $V_k R_k z_k$, therefore we need a short recurrence formula for $P_k = [p_1, \ldots, p_k] = V_k R_k \Theta_k^{-1}$. By comparing columns in

$$P_k \Theta_k = V_k R_k,$$

we get the following recurrence

$$\begin{aligned} p_1 &= v_1/\theta_{1,1} \\ p_2 &= (v_2 + r_{1,2}v_1 - \theta_{1,2}p_1)/\theta_{2,2} \\ p_k &= (v_k + r_{k-1,k}v_{k-1} - \theta_{k-1,k}p_{k-1} - \theta_{k-2,k}p_{k-2})/\theta_{k,k}. \end{aligned}$$

For our SHUMR iterate we get thus

$$x_k = x_{k-1} + \tau_k p_k.$$

Since SHUMR uses the minimal residual condition, we get the norm of the residual r_k as

$$\|r_k\|_2 = |\beta e_{k+1}^T Q_{k+1}^H e_1| = |\widehat{\tau}_{k+1}|.$$

Algorithm 2.15. SHUMR

{**Input** $m \leq m_0$, x_0, $r_0 = b - Ax_0$, ϵ}

$\theta_{0,2} = 0$, $c_0 = 1$, $p_0 = 0$
$v_1 = r_0$, $\beta = \|v_1\|_2$, $w_1 = \beta$, $\widehat{\tau}_1 = \beta$
$l_{1,1} = v_1^H U v_1$
$\tilde{v}_2 = Uv_1 - l_{1,1}v_1$
$l_{2,1} = \|\tilde{v}_2\|_2$
$s_1 = \frac{-l_{2,1}}{(l_{2,1}^2 + |\rho + l_{1,1}|^2)^{1/2}}$
$\bar{c}_1 = \frac{\rho + l_{1,1}}{(l_{2,1}^2 + |\rho + l_{1,1}|^2)^{1/2}}$
$\tau_1 = -c_1\widehat{\tau}_1$
$\widehat{\tau}_2 = s_1\widehat{\tau}_1$
$\theta_{1,1} = -c_1(\rho + l_{1,1}) + s_1 l_{2,1}$
$p_1 = v_1/\theta_{1,1}$
$x_1 = x_0 + \tau_1 p_1$
for k = 2, 3, . . . , m **do**
  $v_k = \tilde{v}_k / l_{k,k-1}$
  $r_{k-1,k} = -\frac{v_{k-1}^H U v_k}{v_{k-1}^H U v_{k-1}}$
  $l_{k,k} = v_k^H U v_{k-1} r_{k-1,k} + v_k^H U v_k$
  $\tilde{v}_{k+1} = Uv_{k-1}r_{k-1,k} + Uv_k - l_{k,k}v_k$
  $l_{k+1,k} = \|\tilde{v}_{k+1}\|_2$
  $\widehat{\theta}_k = s_{k-1}\bar{c}_{k-2}\rho r_{k-1,k} + \bar{c}_{k-1}(\rho + l_{k,k})$
  $s_k = \frac{-l_{k+1,k}}{(|\widehat{\theta}_k|^2 + l_{k+1,k}^2)^{1/2}}$
  $\bar{c}_k = \frac{\widehat{\theta}_k}{(|\widehat{\theta}_k|^2 + l_{k+1,k}^2)^{1/2}}$
  $\tau_k = -c_k\widehat{\tau}_k$
  $\widehat{\tau}_{k+1} = s_k\widehat{\tau}_k$
  $\theta_{k-2,k} = s_{k-2}\rho r_{k-1,k}$
  $\theta_{k-1,k} = -c_{k-1}\bar{c}_{k-2}\rho r_{k-1,k} + s_{k-1}(\rho + l_{k,k})$
  $\theta_{k,k} = -c_k\widehat{\theta}_k + s_k l_{k+1,k}$
  $p_k = (v_k + r_{k-1,k}v_{k-1} - \theta_{k-1,k}p_{k-1} - \theta_{k-2,k}p_{k-2})/\theta_{k,k}$
  $x_k = x_{k-1} + \tau_k p_k$
end for

Not using the SUOM iterates the SHUMR iterates always exist. Note that this holds of course under the assumption that the unitary Arnoldi method has no breakdown, i.e., the LU-decomposition needed for the unitary Arnoldi method exists. If the LU-decomposition in the unitary Arnoldi method does not exist, both SUOM and SHUMR have a breakdown.

2.3.3 SUFOM

In this section we combine the isometric Arnoldi method with a Galerkin condition. The resulting method is called SUFOM (Shifted Unitary FOM).

With the notations from the isometric Arnoldi method we can write the upper Hessenberg matrix H_k as

$$H_k = D_k^{-1} \begin{pmatrix} -\bar{\gamma}_0\gamma_1 & -\bar{\gamma}_0\gamma_2 & \cdots & -\bar{\gamma}_0\gamma_{k-1} & -\bar{\gamma}_0\gamma_k \\ \sigma_1^2 & -\bar{\gamma}_1\gamma_2 & \cdots & -\bar{\gamma}_1\gamma_{k-1} & -\bar{\gamma}_1\gamma_k \\ & \sigma_2^2 & \cdots & -\bar{\gamma}_2\gamma_{k-1} & -\bar{\gamma}_2\gamma_k \\ & & \ddots & \vdots & \vdots \\ & & & \sigma_{k-1}^2 & -\bar{\gamma}_{k-1}\gamma_k \end{pmatrix} D_k$$

where $\gamma_0 = 1$ and $D_k = \mathrm{diag}(\delta_0, \delta_1, \ldots, \delta_{k-1})$ with $\delta_0 = \beta$ and $\delta_i = \delta_{i-1}\sigma_i$ for $i \geq 1$, see [21, 27]. The quantities γ_i and σ_i are computed in the isometric Arnoldi method.

The matrix H_k can be written as

$$H_k = \begin{bmatrix} H_{k-1} & -\gamma_k\delta_{k-1}D_{k-1}^{-1}(\bar{\gamma}_0, \ldots, \bar{\gamma}_{k-2})^T \\ \sigma_{k-1}e_{k-1}^T & -\bar{\gamma}_{k-1}\gamma_k \end{bmatrix}.$$

For the computation of $y_k = (H_k + \rho I)^{-1}V_k^H b = \beta(H_k + \rho I)^{-1}e_1$ we use the QR-decomposition

$$Q_k^H(H_k + \rho I) = \widehat{R}_k,$$

where the unitary matrix Q_k^H can be written as a product of Givens matrices

$$Q_k^H = G_{k-1}(c_{k-1}) \cdots G_1(c_1)$$

with

$$G_i(c_i) = \begin{bmatrix} I_{i-1} & & & \\ & -c_i & s_i & \\ & s_i & \bar{c}_i & \\ & & & I_{k-i-1} \end{bmatrix}, |c_i|^2 + s_i^2 = 1.$$

In order to update $\widehat{R}_k$ we have to know how the Givens matrices affect the last column of $H_k + \rho I$ and the $(k-1, k-1)$ element.

Starting with $\widehat{\phi}_1 = \bar{\gamma}_0/\delta_0$ we see that

$$G_{k-2}(c_{k-2}) \cdots G_1(c_1)D_{k-1}^{-1}(\bar{\gamma}_0, \ldots, \bar{\gamma}_{k-2})^T = (\phi_1, \ldots, \phi_{k-2}, \widehat{\phi}_{k-1})^T$$

where

$$\begin{aligned} \phi_i &= -c_i\widehat{\phi}_i + s_i\bar{\gamma}_i/\delta_i, \\ \widehat{\phi}_{i+1} &= s_i\widehat{\phi}_i + \bar{c}_i\bar{\gamma}_i/\delta_i. \end{aligned}$$

This leads to

$$G_{k-2}(c_{k-2})\cdots G_1(c_1)(H_k+\rho I) = \left[\begin{array}{c|c} \widehat{R}_{k-1} & \begin{array}{c} -\gamma_k\delta_{k-1}(\phi_1,\ldots,\phi_{k-2})^T \\ -\gamma_k\delta_{k-1}\widehat{\phi}_{k-1} \end{array} \\ \hline \sigma_{k-1}e_{k-1}^T & -\bar{\gamma}_{k-1}\gamma_k+\rho \end{array}\right].$$

The $(k-1)$-st Givens matrix is chosen to zero out the $(k,k-1)$ element, i.e. σ_{k-1} and thus

$$\bar{c}_{k-1} = \frac{\widehat{r}_{k-1,k-1}}{(|\widehat{r}_{k-1,k-1}|^2+\sigma_{k-1}^2)^{1/2}}, \quad s_{k-1} = \frac{\sigma_{k-1}}{(|\widehat{r}_{k-1,k-1}|^2+\sigma_{k-1}^2)^{1/2}}.$$

The last Givens matrix $G_{k-1}(c_{k-1})$ affects $-\gamma_k\delta_{k-1}\widehat{\phi}_{k-1}$ and $-\bar{\gamma}_{k-1}\gamma_k+\rho$ in the following way

$$\begin{aligned} G_{k-1}\begin{pmatrix} -\gamma_k\delta_{k-1}\widehat{\phi}_{k-1} \\ -\bar{\gamma}_{k-1}\gamma_k+\rho \end{pmatrix} &= \begin{pmatrix} \gamma_k\delta_{k-1}c_{k-1}\widehat{\phi}_{k-1} - s_{k-1}\bar{\gamma}_{k-1}\gamma_k+\rho s_{k-1} \\ -\gamma_k\delta_{k-1}s_{k-1}\widehat{\phi}_{k-1} - \bar{c}_{k-1}\bar{\gamma}_{k-1}\gamma_k+\rho\bar{c}_{k-1} \end{pmatrix} \\ &= \begin{pmatrix} -\gamma_k\delta_{k-1}\phi_{k-1}+s_{k-1}\rho \\ -\gamma_k\delta_{k-1}\widehat{\phi}_k+\bar{c}_{k-1}\rho \end{pmatrix}, \end{aligned}$$

such that the following structure is produced

$$\widehat{R}_k = Q_k^H(H_k+\rho I) = \left[\begin{array}{c|c} R_{k-1} & \begin{array}{c} -\gamma_k\delta_{k-1}(\phi_1,\ldots,\phi_{k-2})^T \\ r_{k-1,k} \end{array} \\ \hline 0 & \widehat{r}_{k,k} \end{array}\right]$$

with

$$\begin{aligned} r_{k-1,k} &= -\gamma_k\delta_{k-1}\phi_{k-1}+s_{k-1}\rho, \\ \widehat{r}_{k,k} &= -\gamma_k\delta_{k-1}\widehat{\phi}_k+\bar{c}_{k-1}\rho, \\ r_{k-1,k-1} &= -c_{k-1}\widehat{r}_{k-1,k-1}+s_{k-1}\sigma_{k-1}. \end{aligned}$$

Algorithm 2.16 summarizes the computation of the QR decomposition of $(H_k+\rho I)$.

Algorithm 2.16. QR-decomposition for $(\rho I + H_k)$

{**Input** $m \leq m_0$, r_0, unitary matrix U, ρ}

$v_1 = \frac{r_0}{\|r_0\|_2}$; $\widehat{v}_1 = v_1$; $\delta_0 = \|r_0\|_2$; $\widehat{\phi}_1 = 1/\delta_0$; $\phi_0 = s_0 = 0$; $c_0 = 1$
$u = Uv_1$
$\gamma_1 = -\widehat{v}_1^H u$
$\widehat{r}_{1,1} = -\gamma_1 + \rho$
$\sigma_1 = \|u + \gamma_1 \widehat{v}_1\|_2$
$\delta_1 = \delta_0 \sigma_1$
$v_2 = \sigma_1^{-1}(u + \gamma_1 \widehat{v}_1)$
$\widehat{v}_2 = \sigma_1 \widehat{v}_1 + \bar{\gamma}_1 v_2$
$\widehat{v}_2 = \widehat{v}_2 / \|\widehat{v}_2\|_2$
for $k = 2, \ldots, m-1$ **do**
  $u = Uv_k$
  $\gamma_k = -\widehat{v}_k^H u$
  $\bar{c}_{k-1} = \widehat{r}_{k-1,k-1} / (|\widehat{r}_{k-1,k-1}|^2 + \sigma_{k-1}^2)^{1/2}$
  $s_{k-1} = -\sigma_{k-1} / (|\widehat{r}_{k-1,k-1}|^2 + \sigma_{k-1}^2)^{1/2}$
  $\phi_{k-1} = -c_{k-1}\widehat{\phi}_{k-1} + s_{k-1}\bar{\gamma}_{k-1}/\delta_{k-1}$
  $\widehat{\phi}_k = s_{k-1}\widehat{\phi}_{k-1} + \bar{c}_{k-1}\bar{\gamma}_{k-1}/\delta_{k-1}$
  $r_{k-1,k-1} = -c_{k-1}\widehat{r}_{k-1,k-1} + s_{k-1}\sigma_{k-1}$
  $r_{k-1,k} = -\gamma_k \delta_{k-1}\phi_{k-1} + s_{k-1}\rho$
  $\widehat{r}_{k,k} = -\gamma_k \delta_{k-1}\widehat{\phi}_k + \bar{c}_{k-1}\rho$
  $\sigma_k = ((1 - |\gamma_k|)(1 + |\gamma_k|))^{1/2} = \|u + \gamma_k \widehat{v}_k\|_2$
  $\delta_k = \delta_{k-1}\sigma_k$
  $\phi_k = -c_k\widehat{\phi}_k + s_k\bar{\gamma}_k/\delta_k$
  $\widehat{\phi}_{k+1} = s_k\widehat{\phi}_k + \bar{c}_k\bar{\gamma}_k/\delta_k$
  $v_{k+1} = \sigma_k^{-1}(u + \gamma_k \widehat{v}_k)$
  $\widehat{v}_{k+1} = \sigma_k \widehat{v}_k + \bar{\gamma}_k v_{k+1}$
  $\widehat{v}_{k+1} = \widehat{v}_{k+1} / \|\widehat{v}_{k+1}\|_2$
end for

For σ_k it holds $0 \leq \|u + \gamma_k \tilde{v}_k\| = \sigma_k = (1 - |\gamma_k|^2)^{1/2} \leq 1$. Therefore $\delta_k = \|r_0\| \prod_{i=1}^{k} \sigma_i$ might be very small, in which case $\widehat{\phi}_k = s_{k-1}\widehat{\phi}_{k-1} + \bar{c}_{k-1}\bar{\gamma}_{k-1}/\delta_{k-1}$ is very large while only the product $\delta_{k-1}\widehat{\phi}_k$ is needed. The same holds for ϕ_k. This could cause numerical instabilities.

For practical implementation we therefore suggest a (slight) modification of the Algorithm 2.16: Computing the quantities δ_k, $\widehat{\phi}_k$ and ϕ_k can be circumvented by computing

$$\begin{aligned}\tilde{\delta}_k = \delta_k\widehat{\phi}_k &= \delta_k\left(s_{k-1}\widehat{\phi}_{k-1} + \bar{c}_{k-1}\bar{\gamma}_{k-1}/\delta_{k-1}\right)\\ &= s_{k-1}\delta_k\widehat{\phi}_{k-1} + \sigma_k\bar{c}_{k-1}\bar{\gamma}_{k-1}\\ &= s_{k-1}\sigma_k\delta_{k-1}\widehat{\phi}_{k-1} + \sigma_k\bar{c}_{k-1}\bar{\gamma}_{k-1}\\ &= s_{k-1}\sigma_k\tilde{\delta}_{k-1} + \sigma_k\bar{c}_{k-1}\bar{\gamma}_{k-1}.\end{aligned}$$

Obviously $\tilde{\delta}_k$ can be obtained via the recursion

$$\begin{aligned}\tilde{\delta}_1 &= \sigma_1\\ \tilde{\delta}_k &= s_{k-1}\sigma_k\tilde{\delta}_{k-1} + \sigma_k\bar{c}_{k-1}\bar{\gamma}_{k-1}.\end{aligned}$$

This changes the computation of $r_{k-1,k}$ and $\widehat{r}_{k,k}$ to

$$\begin{aligned}r_{k-1,k} &= -\gamma_k\delta_{k-1}\varphi_{k-1} + s_{k-1}\rho\\ &= -\gamma_k\delta_{k-1}\left(-c_{k-1}\widehat{\phi}_{k-1} + s_{k-1}\bar{\gamma}_{k-1}/\delta_{k-1}\right) + s_{k-1}\rho\\ &= \gamma_k c_{k-1}\tilde{\delta}_{k-1} - \gamma_k s_{k-1}\bar{\gamma}_{k-1} + s_{k-1}\rho\\ \widehat{r}_{k,k} &= -\gamma_k\delta_{k-1}\widehat{\phi}_k + \bar{c}_{k-1}\rho\\ &= -\gamma_k\delta_{k-1}\left(s_{k-1}\widehat{\phi}_{k-1} + \bar{c}_{k-1}\bar{\gamma}_{k-1}/\delta_{k-1}\right) + \bar{c}_{k-1}\rho\\ &= -\gamma_k s_{k-1}\tilde{\delta}_{k-1} - \gamma_k\bar{c}_{k-1}\bar{\gamma}_{k-1} + \bar{c}_{k-1}\rho\end{aligned}$$

Computing $r_{k-1,k}$ and $\widehat{r}_{k,k}$ like this, it is thus not necessary anymore to calculate ϕ_i and $\widehat{\phi}_i$.

Finally, we want to obtain the SUFOM iterates

$$\widehat{x}_k = x_0 + V_k\widehat{y}_k,$$

with

$$\widehat{y}_k = (H_k + \rho I)^{-1}V_k^H b = \beta\widehat{R}_k^{-1}Q_k^H e_1.$$

Note that $\beta Q_k^H e_1 = (\tau_1, \ldots, \tau_{k-1}, \widehat{\tau}_k)^T = \widehat{t}_k$ can be easily updated by

$$\begin{aligned}\tau_k &= -c_k\widehat{\tau}_k\\ \widehat{\tau}_{k+1} &= s_k\widehat{\tau}_k\end{aligned}$$

with $\widehat{\tau}_1 = \beta$.

So we have to solve

$$\widehat{R}_k\widehat{y}_k = (\tau_1, \ldots, \tau_{k-1}, \widehat{\tau}_k)^T. \tag{2.10}$$

For the moment we ignore the changes we just made for stability reasons. It will be easy to apply them afterwards.

With $\widehat{y}_k = (\widehat{\eta}_1, \ldots, \widehat{\eta}_k)^T$ and $t_{k-1} = (\tau_1, \ldots, \tau_{k-1})^T$ equation (2.10) reads

$$\begin{aligned} R_{k-1}(\widehat{\eta}_1, \ldots, \widehat{\eta}_{k-1})^T + (-\gamma_k\delta_{k-1}\phi_1, \ldots, -\gamma_k\delta_{k-1}\phi_{k-2}, r_{k-1,k})^T\widehat{\eta}_k &= t_{k-1} \\ \widehat{r}_{k,k}\widehat{\eta}_k &= \widehat{\tau}_k, \end{aligned}$$

and we get

$$\begin{aligned} (\widehat{\eta}_1, \ldots, \widehat{\eta}_{k-1})^T &= R_{k-1}^{-1}t_{k-1} - R_{k-1}^{-1}(-\gamma_k\delta_{k-1}\phi_1, \ldots, -\gamma_k\delta_{k-1}\phi_{k-2}, r_{k-1,k})^T\widehat{\eta}_k \\ \widehat{\eta}_k &= \widehat{\tau}_k/\widehat{r}_{k,k}. \end{aligned}$$

In the same way we obtain for $y_k = (\eta_1, \ldots, \eta_k)^T = R_k^{-1}t_k$

$$\begin{aligned} (\eta_1, \ldots, \eta_{k-1})^T &= R_{k-1}^{-1}t_{k-1} - R_{k-1}^{-1}(-\gamma_k\delta_{k-1}\phi_1, \ldots, -\gamma_k\delta_{k-1}\phi_{k-2}, r_{k-1,k})^T\eta_k \\ \eta_k &= \tau_k/r_{k,k}. \end{aligned}$$

We need $g_{k-1} = R_{k-1}^{-1}(-\gamma_k\delta_{k-1}\phi_1, \ldots, -\gamma_k\delta_{k-1}\phi_{k-2}, r_{k-1,k})^T$ for both $\widehat{y}_k$ and y_k. With the definition of $r_{k-1,k}$ we can split

$$\begin{aligned} g_{k-1} &= R_{k-1}^{-1}(-\gamma_k\delta_{k-1}\phi_1, \ldots, -\gamma_k\delta_{k-1}\phi_{k-2}, -\gamma_k\delta_{k-1}\phi_{k-1} + s_{k-1}\rho)^T \\ &= -\gamma_k\delta_{k-1}R_{k-1}^{-1}(\phi_1, \ldots, \phi_{k-1})^T + s_{k-1}\rho R_{k-1}^{-1}e_{k-1}. \end{aligned}$$

In the same way as for $\widehat{y}_k$ we get recursions for $z_k = (\widehat{z}_{k-1}, \zeta_k)^T = R_k^{-1}e_k$ and $l_k = (\widehat{l}_{k-1}, \lambda_k)^T = R_k^{-1}(\phi_1, \ldots, \phi_k)^T$:

$$\begin{aligned} \lambda_k &= \phi_k/r_{k,k} \\ \widehat{l}_{k-1} &= l_{k-1} - g_{k-1}\lambda_k \end{aligned}$$

$$\begin{aligned} \zeta_k &= 1/r_{k,k} \\ \widehat{z}_{k-1} &= -1/r_{k,k}g_{k-1}. \end{aligned}$$

The iterates therefore read

$$\begin{aligned} \widehat{x}_k &= x_0 + V_k\widehat{y}_k \\ &= x_0 + V_{k-1}(\widehat{\eta}_1, \ldots, \widehat{\eta}_{k-1})^T + \widehat{\eta}_k v_k \\ &= x_0 + V_{k-1}(y_{k-1} - g_{k-1}\widehat{\eta}_k) + \widehat{\eta}_k v_k \\ &= x_{k-1} - V_{k-1}g_{k-1}\widehat{\eta}_k + \widehat{\eta}_k v_k \end{aligned}$$

with

$$\begin{aligned} x_k &= x_0 + V_k y_k \\ &= x_0 + V_{k-1}(\eta_1, \ldots, \eta_{k-1})^T + \eta_k v_k \\ &= x_0 + V_{k-1}(y_{k-1} - g_{k-1}\eta_k) + \eta_k v_k \\ &= x_{k-1} - V_{k-1} g_{k-1}\eta_k + \eta_k v_k. \end{aligned}$$

For $w_k = V_k g_k$ it holds

$$\begin{aligned} w_k &= V_k(-\gamma_{k+1}\delta_k l_k + s_k \rho z_k) \\ &= -\gamma_{k+1}\delta_k(V_{k-1}l_{k-1} - V_{k-1}g_{k-1}\lambda_k + \lambda_k v_k) + s_k\rho(-\frac{1}{r_{k,k}}w_{k-1} + \frac{1}{r_{k,k}}v_k) \\ &= -\gamma_{k+1}\delta_k V_{k-1}l_{k-1} + \frac{\gamma_{k+1}\delta_k\phi_k - s_k\rho}{r_{k,k}}(w_{k-1} - v_k) \\ &= -\gamma_{k+1}\delta_k V_{k-1}l_{k-1} - r_{k,k+1}/r_{k,k}(w_{k-1} - v_k). \end{aligned}$$

Introducing $p_k = V_k l_k$ we get

$$\begin{aligned} p_k &= V_{k-1}\widehat{l}_{k-1} + \lambda_k v_k \\ &= p_{k-1} - \lambda_k(w_{k-1} - v_k) \end{aligned}$$

which completes the short recurrence for the SUFOM iterates $\widehat{x}_k$.

Finally, taking into account the changes we suggested for the stability of the QR-decomposition, we see that the computation of w_k is the only one that involves the quantities δ_k and ϕ_k, the latter through p_k and λ_k.

Introducing $\widehat{p}_{k-1} = -\gamma_{k+1}\delta_k p_{k-1}$ with the recurrence

$$\begin{aligned} \widehat{p}_{k-1} &= -\gamma_{k+1}\delta_k(p_{k-2} - \lambda_{k-1}(w_{k-2} - v_{k-1})) \\ &= \frac{\gamma_{k+1}}{\gamma_k}\sigma_k\widehat{p}_{k-2} + \gamma_{k+1}\sigma_k\delta_{k-1}\phi_{k-1}/r_{k-1,k-1}(w_{k-2} - v_{k-1}) \\ &= \frac{\gamma_{k+1}}{\gamma_k}\sigma_k\widehat{p}_{k-2} + \gamma_{k+1}\sigma_k\frac{-c_{k-1}\tilde{\delta}_{k-1} + s_{k-1}\bar{\gamma}_{k-1}}{r_{k-1,k-1}}(w_{k-2} - v_{k-1}) \end{aligned}$$

eliminates both δ_k and ϕ_k.

Finally, the SUFOM residuals are

$$r_k = -h_{k+1,k}e_k^T\widehat{y}_k v_{k+1} = -\sigma_k\widehat{\eta}_k v_{k+1}.$$

Algorithm 2.17. SUFOM

{**Input** $m \leq m_0$, x_0, $r_0 = b - Ax_0$, ϵ}

$v_1 = r_0/\|r_0\|_2$, $\widehat{v}_1 = v_1$, $\delta_0 = \|r_0\|_2$, $\gamma_0 = 1$, $w_0 = 0$, $\widehat{\tau}_1 = \|r_0\|_2$, $s_0 = 0$,
$c_0 = 1$, $\widehat{p}_0 = 0$
$u = Uv_1$
$\gamma_1 = -\widehat{v}_1^H u$
$\widehat{r}_{1,1} = -\gamma_1 + \rho$
$\widehat{\eta}_1 = \widehat{\tau}_1/\widehat{r}_{1,1}$
$\widehat{x}_1 = x_0 + \widehat{\eta}_1 v_1$
$\sigma_1 = \|u + \gamma_1 \widehat{v}_1\|_2$
$\tilde{\delta}_1 = \sigma_1$
$v_2 = \sigma_1^{-1}(u + \gamma_1 \widehat{v}_1)$
$\widehat{v}_2 = \sigma_1 \widehat{v}_1 + \bar{\gamma}_1 v_2$
$\widehat{v}_2 = \widehat{v}_2/\|\widehat{v}_2\|_2$
for $k = 2, 3, \ldots, m$ **do**
  $u = Uv_k$
  $\gamma_k = -\widehat{v}_k^H u$
  if $k \geq 3$ **then**
    $\widehat{p}_{k-2} = \sigma_{k-1}\frac{\gamma_k}{\gamma_{k-1}}\widehat{p}_{k-3} + \gamma_k \sigma_{k-1}\frac{(-c_{k-2}\tilde{\delta}_{k-2} + s_{k-2}\bar{\gamma}_{k-2})}{r_{k-2,k-2}}(w_{k-3} - v_{k-2})$
  end if
  $\bar{c}_{k-1} = \widehat{r}_{k-1,k-1}/(|\widehat{r}_{k-1,k-1}|^2 + |\sigma_{k-1}|^2)^{1/2}$
  $s_{k-1} = -\sigma_{k-1}/(|\widehat{r}_{k-1,k-1}|^2 + |\sigma_{k-1}|^2)^{1/2}$;
  $r_{k-1,k-1} = -c_{k-1}\widehat{r}_{k-1,k-1} + s_{k-1}\sigma_{k-1}$
  $\tau_{k-1} = -c_{k-1}\widehat{\tau}_{k-1}$
  $\eta_{k-1} = \tau_{k-1}/r_{k-1,k-1}$
  $x_{k-1} = x_{k-2} - \eta_{k-1} w_{k-2} + \eta_{k-1} v_{k-1}$
  $r_{k-1,k} = \gamma_k c_{k-1}\tilde{\delta}_{k-1} - \gamma_k s_{k-1}\bar{\gamma}_{k-1} + s_{k-1}\rho$
  $\widehat{r}_{k,k} = -\gamma_k s_{k-1}\tilde{\delta}_{k-1} - \gamma_k \bar{c}_{k-1}\bar{\gamma}_{k-1} + \bar{c}_{k-1}\rho$
  $w_{k-1} = \widehat{p}_{k-2} - r_{k-1,k}/r_{k-1,k-1}(w_{k-2} - v_{k-1})$
  $\widehat{\tau}_k = s_{k-1}\widehat{\tau}_{k-1}$
  $\widehat{\eta}_k = \widehat{\tau}_k/\widehat{r}_{k,k}$
  $\widehat{x}_k = x_{k-1} - \widehat{\eta}_k w_{k-1} + \widehat{\eta}_k v_k$
  $\sigma_k = \|u + \gamma_k \widehat{v}_k\|_2$
  $\tilde{\delta}_k = s_{k-1}\sigma_k\tilde{\delta}_{k-1} + \sigma_k \bar{c}_{k-1}\bar{\gamma}_{k-1}$
  $v_{k+1} = \sigma_k^{-1}(u + \gamma_k \widehat{v}_k)$
  $\widehat{v}_{k+1} = \sigma_k \widehat{v}_k + \bar{\gamma}_k v_{k+1}$
  $\widehat{v}_{k+1} = \widehat{v}_{k+1}/\|\widehat{v}_{k+1}\|_2$
end for

2.3.4 SUMR

In this section we combine the isometric Arnoldi method with the minimal residual condition. The resulting method was first presented by Jagels and Reichel [27] and called SUMR (Shifted Unitary Minimal Residual) in [1].

The least squares problem resulting from the minimal residual condition is

$$\|\beta e_1 - (\rho I + \widehat{H}_k)y_k\|_2 = \min_{y\in\mathbb{C}^k} \|\beta e_1 - (\rho I + \widehat{H}_k)y\|_2. \tag{2.11}$$

With $D_k = \mathrm{diag}(\delta_0, \delta_1, \ldots, \delta_{k-1})$, where $\delta_0 = \beta$, $\delta_i = \delta_{i-1}\sigma_i$ for $i \geq 1$, and $\gamma_0 = 1$ we can write

$$\rho I + \widehat{H}_k = \begin{bmatrix} \rho I + H_{k-1} & -\gamma_k \delta_{k-1} D_{k-1}^{-1}(\bar{\gamma}_0, \ldots, \bar{\gamma}_{k-2})^T \\ \sigma_{k-1} e_{k-1}^T & -\bar{\gamma}_{k-1}\gamma_k + \rho \\ & \sigma_k \end{bmatrix}.$$

The quantities γ_i and σ_i are computed in the isometric Arnoldi method.

The least squares problem (2.11) is solved via the QR-decomposition

$$Q_{k+1}^H(\rho I + \widehat{H}_k) = \begin{bmatrix} R_k \\ 0 \end{bmatrix} = \widehat{R}_k,$$

where the unitary matrix Q_{k+1}^H can be written as a product of Givens matrices

$$Q_{k+1}^H = G_k(c_k)G_{k-1}(c_{k-1})\cdots G_1(c_1).$$

In order to update R_k we have to know how the Givens matrices affect the last column of $\rho I + \widehat{H}_k$. Starting with $\widehat{\phi}_1 = \bar{\gamma}_0/\delta_0 = 1/\beta$ we see that

$$G_{k-2}(c_{k-2})\cdots G_1(c_1)D_{k-1}^{-1}(\bar{\gamma}_0, \ldots, \bar{\gamma}_{k-2})^T = (\phi_1, \ldots, \phi_{k-2}, \widehat{\phi}_{k-1})^T,$$

where

$$\begin{aligned} \phi_i &= -c_i\widehat{\phi}_i + s_i\bar{\gamma}_i/\delta_i, \\ \widehat{\phi}_{i+1} &= s_i\widehat{\phi}_i + \bar{c}_i\bar{\gamma}_i/\delta_i. \end{aligned}$$

The next Givens matrix $G_{k-1}(c_{k-1})$ affects the entries $-\gamma_k\delta_{k-1}\widehat{\phi}_{k-1}$ and $-\bar{\gamma}_{k-1}\gamma_k + \rho$ in the following way

$$G_{k-1}\begin{pmatrix} -\gamma_k\delta_{k-1}\widehat{\phi}_{k-1} \\ -\bar{\gamma}_{k-1}\gamma_k + \rho \end{pmatrix} = \begin{pmatrix} -\gamma_k\delta_{k-1}\phi_{k-1} + s_{k-1}\rho \\ -\gamma_k\delta_{k-1}\widehat{\phi}_k + \bar{c}_{k-1}\rho \end{pmatrix}.$$

The first $k-1$ Givens matrices produce thus the following structure

$$Q_k^H(\rho I + \widehat{H}_k) = \left[\begin{array}{c|c} R_{k-1} & \begin{array}{c} -\gamma_k\delta_{k-1}(\phi_1, \ldots, \phi_{k-2})^T \\ r_{k-1,k} \end{array} \\ \hline \begin{array}{c} 0 \\ 0 \end{array} & \begin{array}{c} \widehat{r}_{k,k} \\ \sigma_k \end{array} \end{array}\right]$$

with $r_{k-1,k} = -\gamma_k \delta_{k-1} \phi_{k-1} + s_{k-1}\rho$ and $\widehat{r}_{k,k} = -\gamma_k \delta_{k-1} \widehat{\phi}_k + \bar{c}_{k-1}\rho$.

Finally, the last Givens matrix $G_k(c_k)$ has to zero out σ_k. Therefore it is calculated as

$$\bar{c}_k = \frac{\widehat{r}_{k,k}}{(|\widehat{r}_{k,k}|^2 + \sigma_k^2)^{1/2}}, \quad s_k = \frac{\sigma_k}{(|\widehat{r}_{k,k}|^2 + \sigma_k^2)^{1/2}}$$

to obtain the QR-decomposition

$$Q_{k+1}^H(\rho I + \widehat{H}_k) = \left[\begin{array}{c|c} R_{k-1} & -\gamma_k \delta_{k-1}(\phi_1, \ldots, \phi_{k-2})^T \\ & r_{k-1,k} \\ \hline 0 & r_{k,k} \\ 0 & 0 \end{array}\right]$$

with $r_{k,k} = -c_k \widehat{r}_{k,k} + s_k \sigma_k$.

Therefore, the least squares problem (2.11) reduces to

$$\|\beta Q_{k+1}^H e_1 - \widehat{R}_k y_k\|_2 = \min_{y \in \mathbb{C}^k} \|\beta Q_{k+1}^H e_1 - \widehat{R}_k y\|_2.$$

Note that $\beta Q_{k+1}^H e_1 = (\tau_1, \ldots, \tau_k, \widehat{\tau}_{k+1})^T$ can be easily updated by

$$\begin{aligned} \tau_k &= -c_k \widehat{\tau}_k \\ \widehat{\tau}_{k+1} &= s_k \widehat{\tau}_k \end{aligned}$$

with $\widehat{\tau}_1 = \beta$.

Algorithm 2.18. QR-decomposition for $\rho I + \widehat{H}_k$

{**Input** $m \le m_0$, r_0, unitary matrix U, ρ}

$v_1 = \frac{r_0}{\|r_0\|}$; $\widehat{v}_1 = v_1$; $\delta_0 = \|r_0\|$; $\widehat{\phi}_1 = 1/\delta_0$; $\phi_0 = s_0 = 0$; $c_0 = 1$
for $k = 1, 2, \ldots, m-1$ **do**
  $u = U v_k$
  $\gamma_k = -\widehat{v}_k^H u$
  $\sigma_k = ((1 - |\gamma_k|)(1 + |\gamma_k|))^{1/2} = \|u + \gamma_k \widehat{v}_k\|_2$
  $r_{k-1,k} = -\gamma_k \delta_{k-1} \phi_{k-1} + s_{k-1}\rho$
  $\widehat{r}_{k,k} = -\gamma_k \delta_{k-1} \widehat{\phi}_k + \bar{c}_{k-1}\rho$
  $\bar{c}_k = \widehat{r}_{k,k}/(|\widehat{r}_{k,k}|^2 + \sigma_k^2)^{1/2}, \quad s_k = -\sigma_k/(|\widehat{r}_{k,k}|^2 + \sigma_k^2)^{1/2}$
  $r_{k,k} = -c_k \widehat{r}_{k,k} + s_k \sigma_k$
  $\delta_k = \delta_{k-1} \sigma_k$
  $\phi_k = -c_k \widehat{\phi}_k + s_k \bar{\gamma}_k / \delta_k$
  $\widehat{\phi}_{k+1} = s_k \widehat{\phi}_k + \bar{c}_k \bar{\gamma}_k / \delta_k$
  $v_{k+1} = \sigma_k^{-1}(u + \gamma_k \widehat{v}_k)$, $\widehat{v}_{k+1} = \sigma_k \widehat{v}_k + \bar{\gamma}_k v_{k+1}$, $\widehat{v}_{k+1} = \widehat{v}_{k+1}/\|\widehat{v}_{k+1}\|_2$
end for

For the same reasons as in SUFOM, the QR-decompostion for $\rho I + H_k$ can be numerically instable if implemented like Algorithm 2.18.

We therefore suggest to avoid the explicit computation of the quantities δ_k, $\widehat{\phi}_k$ and ϕ_k and replace them by $\tilde{\delta}_k = \delta_k \widehat{\phi}_k$, which can be obtained via the recursion

$$\begin{aligned} \tilde{\delta}_1 &= \sigma_1 \\ \tilde{\delta}_k &= s_{k-1}\sigma_k \tilde{\delta}_{k-1} + \sigma_k \bar{c}_{k-1}\bar{\gamma}_{k-1}. \end{aligned}$$

This changes the computation of $r_{k-1,k}$ and $\widehat{r}_{k,k}$ to

$$\begin{aligned} r_{k-1,k} &= -\gamma_k \delta_{k-1}\varphi_{k-1} + s_{k-1}\rho \\ &= \gamma_k c_{k-1}\tilde{\delta}_{k-1} - \gamma_k s_{k-1}\bar{\gamma}_{k-1} + s_{k-1}\rho \\ \widehat{r}_{k,k} &= -\gamma_k \delta_{k-1}\widehat{\phi}_k + \bar{c}_{k-1}\rho \\ &= -\gamma_k s_{k-1}\tilde{\delta}_{k-1} - \gamma_k \bar{c}_{k-1}\bar{\gamma}_{k-1} + \bar{c}_{k-1}\rho. \end{aligned}$$

Finally, to obtain the iterate $x_k = x_0 + V_k y_k$, we have to solve

$$R_k y_k = t_k = (\tau_1, \ldots, \tau_k)^T. \tag{2.12}$$

For the derivation of recursions for y_k and x_k, we first ignore the modifications we just made for stability reasons. It will be easy to apply them afterwards.

With $y_k = (\widehat{y}_{k-1}, \eta_k)^T$ and $\widehat{y}_{k-1} \in \mathbb{C}^{k-1}$, (2.12) reads

$$\begin{aligned} R_{k-1}\widehat{y}_{k-1} + (-\gamma_k \delta_{k-1}\phi_1, \ldots, -\gamma_k \delta_{k-1}\phi_{k-2}, r_{k-1,k})^T \eta_k &= (\tau_1, \ldots, \tau_{k-1})^T \\ r_{k,k}\eta_k &= \tau_k, \end{aligned}$$

and we get

$$\begin{aligned} \widehat{y}_{k-1} &= y_{k-1} - R_{k-1}^{-1}(-\gamma_k \delta_{k-1}\phi_1, \ldots, -\gamma_k \delta_{k-1}\phi_{k-2}, r_{k-1,k})^T \eta_k \\ \eta_k &= \tau_k / r_{k,k}. \end{aligned}$$

For $g_{k-1} = R_{k-1}^{-1}(-\gamma_k \delta_{k-1}\phi_1, \ldots, -\gamma_k \delta_{k-1}\phi_{k-2}, r_{k-1,k})^T$ we find

$$\begin{aligned} g_{k-1} &= R_{k-1}^{-1}(-\gamma_k \delta_{k-1}\phi_1, \ldots, -\gamma_k \delta_{k-1}\phi_{k-2}, -\gamma_k \delta_{k-1}\phi_{k-1} + s_{k-1}\rho)^T \\ &= -\gamma_k \delta_{k-1} R_{k-1}^{-1}(\phi_1, \ldots, \phi_{k-1})^T + s_{k-1}\rho R_{k-1}^{-1} e_{k-1}. \end{aligned}$$

In the same way as for y_k we get recursions for $z_k = (\widehat{z}_{k-1}, \zeta_k)^T = R_k^{-1} e_k$ and $l_k = (\widehat{l}_{k-1}, \lambda_k)^T = R_k^{-1}(\phi_1, \ldots, \phi_k)^T$:

$$\begin{aligned} \lambda_k &= \phi_k / r_{k,k} \\ \widehat{l}_{k-1} &= l_{k-1} - g_{k-1}\lambda_k \end{aligned}$$

$$\begin{aligned} \zeta_k &= 1/r_{k,k} \\ \widehat{z}_{k-1} &= -1/r_{k,k} g_{k-1}. \end{aligned}$$

The iterates x_k are finally

$$\begin{aligned} x_k &= x_0 + V_k y_k = x_0 + V_{k-1}\widehat{y}_{k-1} + \eta_k v_k \\ &= x_{k-1} + V_{k-1}(\widehat{y}_{k-1} - y_{k-1}) + \eta_k v_k \\ &= x_{k-1} + (-V_{k-1} g_{k-1} + v_k)\eta_k. \end{aligned}$$

Introducing $w_k = V_k g_k$ and $p_k = V_k l_k$ which fulfil the recursions

$$\begin{aligned} w_k &= -\gamma_{k+1}\delta_k p_{k-1} - r_{k,k+1}/r_{k,k}(w_{k-1} - v_k) \\ p_k &= p_{k-1} - \lambda_k(w_{k-1} - v_k) \end{aligned}$$

results in a recurrence for x_k.

Taking into account the changes we suggested for the stability of the QR-decomposition, we introduce – as in SUFOM – $\widehat{p}_{k-1} = -\gamma_{k+1}\delta_k p_{k-1}$ with the recurrence

$$\widehat{p}_{k-1} = \frac{\gamma_{k+1}}{\gamma_k}\sigma_k \widehat{p}_{k-2} + \gamma_{k+1}\sigma_k \frac{-c_{k-1}\tilde{\delta}_{k-1} + s_{k-1}\bar{\gamma}_{k-1}}{r_{k-1,k-1}}(w_{k-2} - v_{k-1})$$

to eliminate δ_k and ϕ_k and thus preserve stability.

The norm of the residuals r_k is

$$\|r_k\|_2 = |\beta e_{k+1}^T Q_{k+1}^H e_1| = |\widehat{\tau}_{k+1}|,$$

so its computation does not cause any additional cost.

Algorithm 2.19. SUMR

{**Input** $m \leq m_0$, x_0, $r_0 = b - Ax_0$, ϵ}

$\widehat{\tau}_1 = \|r_0\|_2$; $w_{-1} = v_0 = 0$; $s_0 = 0$
$r_{0,0} = \gamma_0 = \sigma_0 = c_0 = 1$; $v_1 = \tilde{v}_1 = r_0/\|r_0\|_2$
$\widehat{p}_1 = 0$; $\tilde{\delta}_0 = 0$
for $k = 1, 2, \ldots, m$ **do**
  $u = Uv_k$
  $\gamma_k = -\tilde{v}_k^H u$
  $\sigma_k = ((1 - |\gamma_k|)(1 + |\gamma_k|))^{1/2}$;
  $r_{k-1,k} = \gamma_k c_{k-1}\tilde{\delta}_{k-1} - \gamma_k s_{k-1}\bar{\gamma}_{k-1} + s_{k-1}\rho$
  $\widehat{r}_{k,k} = -\gamma_k s_{k-1}\tilde{\delta}_{k-1} - \gamma_k \bar{c}_{k-1}\bar{\gamma}_{k-1} + \bar{c}_{k-1}\rho$
  $\bar{c}_k = \widehat{r}_{k,k}/(|\widehat{r}_{k,k}|^2 + |\sigma_k|^2)^{1/2}$
  $s_k = -\sigma_k/(|\widehat{r}_{k,k}|^2 + |\sigma_k|^2)^{1/2}$
  $r_{k,k} = -c_k\widehat{r}_{k,k} + s_k\sigma_k$
  $\tau_k = -c_k\widehat{\tau}_k$
  $\widehat{\tau}_{k+1} = s_k\widehat{\tau}_k$
  $\eta_k = \tau_k/r_{k,k}$
  $\kappa_{k-1} = r_{k-1,k}/r_{k-1,k-1}$
  if $k > 1$ **then**
    $\widehat{p}_k = \frac{\gamma_k\sigma_{k-1}}{\gamma_{k-1}}\widehat{p}_{k-1} + \frac{(-c_{k-2}\tilde{\delta}_{k-2} + s_{k-2}\bar{\gamma}_{k-2})\gamma_k\sigma_{k-1}}{r_{k-2,k-2}}(w_{k-3} - v_{k-2})$
  end if
  $w_{k-1} = \widehat{p}_k - (w_{k-2} - v_{k-1})\kappa_{k-1}$
  $x_k = x_{k-1} - (w_{k-1} - v_k)\eta_k$
  $\tilde{\delta}_k = s_{k-1}\sigma_k\tilde{\delta}_{k-1} + \sigma_k\bar{c}_{k-1}\bar{\gamma}_{k-1}$
  $v_{k+1} = \sigma_k^{-1}(u + \gamma_k\tilde{v}_k)$
  $\tilde{v}_{k+1} = \sigma_k\tilde{v}_k + \bar{\gamma}_k v_{k+1}$
  $\tilde{v}_{k+1} = \tilde{v}_{k+1}/\|\tilde{v}_{k+1}\|_2$
end for

2.4 Discussion

For the unitary Arnoldi method we have to assume the existence of the LU-decomposition of the unitary matrix U. If for one index $k < m_0$ the product Uv_k is orthogonal to v_i, $i \leq k$, then the LU-decomposition does not exist, see Lemma 2.8. Although this is not very likely to happen, especially from the numerical point of view, we have to consider this as a weak point of the unitary Arnoldi method. The isometric Arnoldi method does not have this disadvantage.

Of course, problems of the underlying Arnoldi process communicate to the shifted unitary methods. Therefore, SUOM and SHUMR have to be used

always keeping in mind the possibility of a breakdown if the assumptions for the unitary Arnoldi method are not fulfilled.

Another cause for a breakdown can be the use of a Galerkin condition if A is not positive real, i.e., if $\rho \leq 1$. In this case there is no guarantee that $V_k^H A V_k$ is non-singular and therefore, the SUOM and SUFOM iterates do not necessarily exist.

From this point of view SUMR is the method of choice for A not positive real, since it is based on the isometric Arnoldi method and uses the minimal residual condition, such that its iterates always exist. For A positive real SUFOM is a safe alternative.

method/condition	Galerkin	minimal residual
unitary Arnoldi	SUOM	SHUMR
isometric Arnoldi	SUFOM	SUMR

Table 2.1: Shifted unitary methods

If no breakdowns occur, the numerical results of Chapter 4.2 show an equal performance of all four methods.

Chapter 3

Deflation for multishift methods

Although multishift methods play a role in other contexts too we will focus here on the rational approximation of $\mathrm{sign}(Q)b$ where they are of greatest importance.

Given a rational approximation $\mathrm{sign}(t) \approx \sum_{i=1}^{s} \frac{\omega_i t}{t^2 - \sigma_i}$ for all $t \in \mathrm{spec}(Q)$ we get the approximation

$$\mathrm{sign}(Q)b \approx Q \sum_{i=1}^{s} \omega_i x^{(i)}$$

where the $x^{(i)}$, $i = 1, \ldots, s$, are solutions of the s systems

$$(Q^2 - \sigma_i I)x^{(i)} = b.$$

For hermitian Q, e.g., the Wilson-Dirac operator at zero chemical potential, the Zolotarev rational approximation in combination with short recurrence multishift methods (CG, MINRES) have been investigated for example in [45].

At non-zero chemical potential the Wilson-Dirac operator is non-hermitian. In this case the short recurrences of CG and MINRES turn to long recurrences. In addition the Zolotarev approximation can not be used for non-hermitian matrices. We therefore have to face two problems.

- Replacing CG by FOM or MINRES by GMRES leads to long recurrences. Restarts bound the size of the Krylov subspaces and thus reduce storage problems. Still the computing efford is much higher than for a short recurrence. Alternatively we can replace Arnoldi by Lanczos biorthogonalization and use BiCG or QMR.

- Using the Neuberger rational approximation instead of Zolotarev increases the number s of poles significantly.

In the following sections we present a deflation approach which uses eigenvalue information to both accelerate convergence of the chosen Krylov subspace method and reduce the number of poles. We will add some eigenvectors to the Krylov subspaces and apply the restarted multishift methods (FOM and GMRES) and the short recurrence multishift methods (BiCG and QMR) to these augmented subspaces. In Section 3.3 we use Schur vectors to small eigenvalues to span the augmenting subspace while in Section 3.4 we use left and right eigenvectors to small eigenvalues instead. In Section 3.6 we show how to reduce the number of poles.

3.1 Shifts and restarts

Instead of solving one system we now have to solve s systems with the same right hand side:

$$(A - \sigma_i I)x^i = b,\ i = 1, \ldots, s$$

In the Krylov subspace context we get the approximations

$$x_m^i \in x_0^i + K_m(A - \sigma_i I, r_0^i),\ r_0^i = b - (A - \sigma_i I)x_0^i.$$

Of course, as in the non-shifted case, we can express both iterates and residuals in terms of polynomials in $A - \sigma_i I$:

$$x_m^i = x_0^i + q_{m-1}^i(A - \sigma_i I)r_0^i$$

with q_{m-1}^i a polynomial of degree less or equal $m - 1$ and

$$r_m^i = b - (A - \sigma_i I)x_m^i = p_m^i(A - \sigma_i I)r_0^i$$

with $p_m^i(t) = 1 - tq_{m-1}^i(t)$ and $p_m^i(0) = 1$. Obviously $r_m^i \in K_{m+1}(A - \sigma_i I, r_0^i)$.

Since Krylov subspaces are shift invariant we can use the same subspace for all s systems if the r_0^i are collinear.

In case of a zero initial guess $x_0^i = 0$ for all systems $i = 1, \ldots, s$, the starting residuals are $r_0^i = b - (A - \sigma_i I)x_0^i = b$. Considering restarts we do not have zero initial guesses, not even the same initial guesses for all systems. The initial guess of a restarted run is the last approximation of the preceding run, so the same holds for the residuals. For restarted methods, the crucial question thus is: Are the residuals collinear?

Whether they are or not depends on the way we choose the iterates from the Krylov subspace. Using the minimal residual condition, the residuals will in general not be collinear while using a (Petrov-)Galerkin condition, the residuals are automatically collinear as the following result from [16] shows.

Theorem 3.1. Let $W_1 \subseteq W_2 \subseteq \cdots \subseteq W_k \subseteq \mathbb{C}^n$ with $\dim(W_m) = m$ and $W_m \cap (K_{m+1}(A, r_0^i))^\perp = \{0\}$, $m = 1, \ldots, k$. Let the approximations $x_m^i \in x_0 + K_m(A, r_0^i)$ to the solution of

$$(A - \sigma_i I)x = b, \quad i = 1, \ldots, s$$

be chosen such that the residuals $r_m^i = b - (A - \sigma_i I)x_m^i = p_m^i(A - \sigma_i I)r_0^i$ fulfil the Petrov-Galerkin condition

$$r_m^i \perp W_m, \ m = 1, \ldots, k.$$

Then r_m^i and r_m^j are collinear if the starting residuals r_0^i and r_0^j are collinear.

Proof. See [16] □

Theorem 3.1 directly applies to FOM where $W_m = K_m(A, r_0^i)$. A formulation of restarted multishift FOM can be found in [43].

When the residuals are collinear, i.e., $r_m^{(i)} = \rho_m^{(i)} r_m^{(1)}$, then the collinearity factor $\rho_m^{(i)}$ can be expressed in terms of the polynomial p:

Lemma 3.2. Let the starting residuals be collinear, i.e., $r_0^{(i)} = \rho_0^{(i)} r_0^{(1)}$. If $r_m^{(i)} = \rho_m^{(i)} r_m^{(1)}$, then it holds

$$\rho_m^{(i)} = \frac{\rho_0^{(i)}}{p_m^{(1)}(\sigma_i - \sigma_1)}. \tag{3.1}$$

Proof. It holds

$$r_m^{(i)} = p_m^{(i)}(A - \sigma_i I)r_0^{(i)} = \rho_m^{(i)} r_m^{(1)} = \rho_m^{(i)} p_m^{(1)}(A - \sigma_1 I)r_0^{(1)}.$$

Since $r_0^{(1)}, Ar_0^{(1)}, A^2 r_0^{(1)}, \ldots$ are linearly independent it follows

$$\rho_0^{(i)} p_m^{(i)}(t - \sigma_i) = \rho_m^{(i)} p_m^{(1)}(t - \sigma_1),$$

and with $p_m^{(i)}(0) = 1$ we get (3.1). □

In case of methods like GMRES the residuals do not fulfil a Petrov-Galerkin condition. The minimal residual condition is used instead to choose the iterates from the Krylov subspace.
For such methods the residuals are in general not collinear. To use restarts, we have to either drop the aim of solving all s systems using one Krylov

subspace or demand the minimal residual condition for no more than one (the "worst") of the s systems.

The latter was proposed in [17]: drop the minimal residual condition for all except one system and force the residuals of the s systems to be collinear instead.

Let the starting residuals be collinear, such that $r_0^{(i)} = \rho_0^{(i)} r_0^{(1)}$, $i = 2, \ldots, s$. Let further be V_m and $\widehat{H}_m$ the matrices resulting from the Arnoldi method with starting vector $r_0^{(1)}$.
For the first system we demand the minimal residual condition and since $r_0^{(1)} - (A - \sigma_1 I)V_m y_m^{(1)} = \beta V_{m+1} e_1 - V_{m+1}(\widehat{H}_m - \sigma_1 I) y_m^{(1)}$ we therefore solve

$$\|\beta e_1 - (\widehat{H}_m - \sigma_1 I) y_m^{(1)}\|_2 = \min_{y \in \mathbb{C}^m} \|\beta e_1 - (\widehat{H}_m - \sigma_1 I) y\|_2.$$

This system is therefore solved by ordinary GMRES.

For the other systems $i = 2, \ldots, s$ we want to force the residuals to be collinear to the one of the first system, i.e.

$$r_m^{(i)} = \rho_0^{(i)} r_0^{(1)} - (A - \sigma_i I) V_m y_m^{(i)} = \rho_m^{(i)} r_m^{(1)},$$

which gives

$$V_{m+1}(\rho_0^{(i)} \beta e_1 - (\widehat{H}_m - \sigma_i I) y_m^{(i)}) = \rho_m^{(i)} r_m^{(1)}$$

and after multiplication with V_{m+1}^H

$$\rho_0^{(i)} \beta e_1 - (\widehat{H}_m - \sigma_i I) y_m^{(i)} = V_{m+1}^H \rho_m^{(i)} r_m^{(1)}. \tag{3.2}$$

Obviously, (3.2) is solved by the solution of

$$\left[\widehat{H}_m - \sigma_i I | V_{m+1}^H r_m^{(1)} \right] \begin{bmatrix} y_m^{(i)} \\ \rho_m^{(i)} \end{bmatrix} = \rho_0^{(i)} \beta e_1 \tag{3.3}$$

which provides both the subspace approximation $y_m^{(i)}$ and the scalar $\rho_m^{(i)}$. In [17] it is shown that if (3.3) has a solution, it is unique.

Although only for the system $i = 1$ a minimal residual property is demanded, all systems converge if $A - \sigma_1 I$ is positive real and the shifts are real with $\sigma_i < \sigma_1$, $i = 2, \ldots, s$.

Theorem 3.3. Let $A - \sigma_1 I$ be positive real and $\sigma_i < \sigma_1$, $i = 2, \ldots, s$. Then the restarted multishift GMRES converges for $i = 1, \ldots, s$ and

$$\|r^{(i)}\|_2 \leq |\rho_0^{(i)}| \cdot \|r^{(1)}\|_2$$

Proof. Theorem 3.3 in [17]. □

Algorithm 3.4. GMRES(m)-Sh

{**Input** A, $\{\sigma_1, \ldots, \sigma_s\}$, b}

$x^{(i)} = 0$, $i = 1, \ldots, s$
$r = b$
$\rho^{(i)} = 1$, $i = 1, \ldots, s$
$\beta = \|r\|$
while not all systems converged **do**
  $v_1 = r/\beta$
  compute V_m, $\widehat{H}_m$ by running Arnoldi
  compute GMRES approximation $y_m^{(1)}$
  for $i = 1, \ldots, s$ **do**
    compute $y_m^{(i)}$ and $\rho_m^{(i)}$ from (3.3)
  end for
  $x^{(i)} = x^{(i)} + V_m y_m^{(i)}$
  $r = r - V_{m+1}(\widehat{H}_m - \sigma_1 I) y_m^{(1)}$
  $\beta = \|r\|$
end while

Unfortunately, even though only one Krylov subspace has to be built for all shifts, for FOM and GMRES there is still a lot of additional cost for each system as there is no simple update of the iterates. So in each step a linear system or least squares problem of increasing size has to be solved. In addition to the computational cost, storage increases by $\mathcal{O}(s \cdot n) + \mathcal{O}(s \cdot m)$.

Finally there exist multishift versions for BiCG [16] and QMR [13]. Both methods are based on the Lanczos biorthogonalization which builds the matrices V_m, W_m and H_m or $\widehat{H}_m$, respectively, with H_m or $\widehat{H}_m$ being tridiagonal instead of full upper Hessenberg:

$$\widehat{H}_m = \begin{bmatrix} H_m \\ \gamma_m e_m^T \end{bmatrix} = \begin{pmatrix} \alpha_0 & \beta_1 & & & \\ \gamma_1 & \alpha_1 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{m-1} \\ & & & \ddots & \alpha_{m-1} \\ & & & & \gamma_m \end{pmatrix}$$

To obtain the multishift version of BiCG we use the residual recurrence

$$r_{k+1} = -\alpha_k A r_k + (1 - \frac{\alpha_k \beta_k}{\alpha_{k-1}}) r_k + \frac{\alpha_k \beta_k}{\alpha_{k-1}} r_{k-1}.$$

Since the iterates are chosen using a Petrov-Galerkin condition, we know from Theorem 3.1 that the residuals are collinear, such that $r_k^{(i)} = \rho_k^{(i)} r_k$, with r_k being the residual with respect to a non-shifted system. Note that for Algorithm 3.5 we used $\pi^{(i)} = 1/\rho^{(i)}$, i.e., $\pi_k^{(i)} r_k^{(i)} = r_k$.
For a detailed description how this information can be combined to get a simple update for the iterates $x_m^{(i)}$ such that all s systems can be solved for basically the cost of one see [16].

Algorithm 3.5. BiCG-Sh

{**Input** A, $\{\sigma_1, \ldots, \sigma_s\}$, b}

$x^{(i)} = 0, \quad i = 1, \ldots, s$
$r = b$, $\tilde{r} = r$
$u = \tilde{u} = 0$
$u^{(i)} = 0, \quad i = 1, \ldots, s$
$\rho_{old} = 1$, $\alpha_{old} = 1$
$\pi_{old}^{(i)} = 1, \quad i = 1, \ldots, s$
$\pi^{(i)} = 1, \quad i = 1, \ldots, s$
while not all systems converged **do**
  $\rho = \tilde{r}^H r$
  $\beta = -\rho/\rho_{old}$
  $u = r - \beta u$, $\tilde{u} = \tilde{r} - \bar{\beta}\tilde{u}$
  $q = Au$
  $\alpha = \rho/(\tilde{r}^H q)$
  for $i = 1, \ldots, s$ **do**
    $\pi_{new}^{(i)} = (1 - \alpha\sigma_i)\pi^{(i)} + (\alpha\beta)/\alpha_{old}(\pi_{old}^{(i)} - \pi^{(i)})$
    $\beta^{(i)} = (\pi_{old}^{(i)}/\pi^{(i)})^2 \beta$, $\alpha^{(i)} = \pi^{(i)}/\pi_{new}^{(i)} \alpha$
    $u^{(i)} = 1/\pi^{(i)} r - \beta^{(i)} u^{(i)}$
    $x^{(i)} = x^{(i)} + \alpha^{(i)} u^{(i)}$
  end for
  $r = r - \alpha q$, $\tilde{r} = \tilde{r} - \bar{\alpha} A^H \tilde{u}$
  $\alpha_{old} = \alpha$, $\rho_{old} = \rho$, $\pi_{old}^{(i)} = \pi^{(i)}$, $\pi^{(i)} = \pi_{new}^{(i)}$
end while

For multishift QMR, the least squares problems

$$\|\rho_0^{(i)} r_0^{(1)} - (\widehat{H}_m - \sigma_i I) y_m^{(i)}\|_2 = \min_{y \in \mathbb{C}^m} \|\rho_0^{(i)} r_0^{(1)} - (\widehat{H}_m - \sigma_i I) y\|_2$$

are solved using the QR-decompositions

$$(\widehat{H}_m - \sigma_i I) = {Q_m^{(i)}}^H \begin{bmatrix} R_m^{(i)} \\ 0 \end{bmatrix},$$

where

$$R_m^{(i)} = \begin{pmatrix} \delta_1^{(i)} & \epsilon_2^{(i)} & \theta_3^{(i)} & & 0 \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \theta_m^{(i)} \\ & & & \ddots & \epsilon_m^{(i)} \\ 0 & & & & \delta_m^{(i)} \end{pmatrix}$$

and $Q_m^{(i)}$ is the product of m Givens rotations. The QR-decomposition for the different shifts can be updated simultaneously, see [13] for details.

Algorithm 3.6. QMR-Sh

{**Input** A, $\{\sigma_1, \ldots, \sigma_s\}$, b}
$x^{(i)} = 0, \quad i = 1, \ldots, s$
$\tilde{v} = \tilde{w} = b$
$\gamma = \|\tilde{v}\|_2$
$\eta^{(i)} = \gamma, \quad i = 1, \ldots, s$
$c_{old}^{(i)} = c^{(i)} = 1$, $sn_{old}^{(i)} = sn^{(i)} = 0$, $p_{old}^{(i)} = p^{(i)} = 0 \quad i = 1, \ldots, s$
$w_{old} = v_{old} = 0$
while not all systems converged **do**
  $\beta = \langle \tilde{v}, \tilde{w} \rangle / \gamma$, $v = 1/\gamma \tilde{v}$, $w = 1/\bar{\beta} \tilde{w}$
  $q = Av$, $\alpha = \langle q, w \rangle$, $\tilde{q} = A^H w$
  $\tilde{v} = q - \alpha v - \beta v_{old}$
  $\tilde{w} = \tilde{q} - \bar{\alpha} w - \bar{\gamma} w_{old}$
  $\gamma = \|\tilde{v}\|_2$
  for $i = 1, \ldots, s$ **do**
    $\theta^{(i)} = sn_{old}^{(i)} \beta$, $\tilde{\epsilon}^{(i)} = c_{old}^{(i)} \beta$
    $\epsilon^{(i)} = c^{(i)} \tilde{\epsilon}^{(i)} + sn^{(i)} (\alpha - \sigma_i)$
    $\tilde{\delta}^{(i)} = -\bar{sn}^{(i)} \tilde{\epsilon}^{(i)} + c^{(i)} (\alpha - \sigma_i)$
    $\mu^{(i)} = (|\tilde{\delta}^{(i)}|^2 + |\gamma|^2)^{1/2}$
    $c_{old}^{(i)} = c^{(i)}$, $sn_{old}^{(i)} = sn^{(i)}$
    $c^{(i)} = |\tilde{\delta}^{(i)}| / \mu^{(i)}$
    $$\bar{sn}^{(i)} = \begin{cases} c^{(i)} \gamma / \tilde{\delta}^{(i)} & \text{if } \tilde{\delta}^{(i)} \neq 0 \\ 1 & \text{else} \end{cases}$$
    $\delta^{(i)} = c^{(i)} \tilde{\delta}^{(i)} + sn^{(i)} \gamma$
    $p_{new}^{(i)} = (v - \epsilon^{(i)} p^{(i)} - \theta^{(i)} p_{old}^{(i)}) / \delta^{(i)}$, $p_{old}^{(i)} = p^{(i)}$, $p^{(i)} = p_{new}^{(i)}$
    $\tilde{\eta}^{(i)} = c^{(i)} \eta^{(i)}$, $\eta^{(i)} = -\bar{sn}^{(i)} \eta^{(i)}$
    $x^{(i)} = x^{(i)} + \tilde{\eta}^{(i)} p^{(i)}$
  end for
  $v_{old} = v$, $w_{old} = w$
end while

As BiCG and QMR work with short recurrences, there is no need of restarting. Nevertheless, as for the methods above, BiCG preserves collinearity of residuals – Theorem 3.1 applies with $W_m = K_m(A^H, r_0^i)$ – while QMR residuals are not collinear.

3.2 Augmented subspaces

Eigenvalues with small real part cause problems in two ways. First they slow down convergence as the condition number of the matrix is most likely large, and second we need more poles in the rational approximation. If we have information about eigenvalues with small real part and their eigenvectors, we can use this information to deflate these eigenvalues in two ways:

1. Add the eigenvalue information to the Krylov subspace methods to accelerate their convergence.

2. Split the rational approximation and use less poles whereever possible.

In this section we investigate the first possibility. Based on the results of this section and Sections 3.3 and 3.4 we will address the second possibility in Section 3.6.

Let $\Omega \subseteq \mathbb{C}^n$ be the subspace we want to deflate, e.g., the subspace spanned by some eigenvectors of A. The basic idea is to search for an approximation x_m to the solution of $Ax = b$ in the augmented subspace

$$x_m \in x_0 + \Omega + K_m(A, r_0)$$

such that the iterate splits into $x_m = x_0 + x_m^{\Omega} + x_m^{\text{Krylov}}$ with $x_m^{\Omega} \in \Omega$, $x_m^{\text{Krylov}} \in K_m(A, r_0)$.

Of course, if $K_m(A, r_0) \cap \Omega \neq \{0\}$, computing x_m^{Krylov} would still involve the small eigenvalues and there would not be any advantage of doing so. To construct the Krylov subspace we therefore have to project out the parts lying in the subspace Ω. Depending on the projection P, this has to be done only once for the starting vector r_0 or in every single step of the Arnoldi method.

Let P be any projector on Ω. Then

$$x_0 + \Omega + K_m(A, r_0) = x_0 + \Omega + (I - P)K_m(A, r_0),$$

and $(I - P)K_m(A, r_0) \cap \Omega = \{0\}$.

The first question that arises is whether $(I - P)K_m(A, r_0)$ is still a Krylov subspace or not.

Lemma 3.7. If range$(P) = \Omega$ is A-invariant, then

$$(I - P)K_m(A, r_0) = K_m((I - P)A, (I - P)r_0),$$

i.e., $(I - P)K_m(A, r_0)$ is a Krylov subspace.

Proof. If range(P) is A-invariant, then for any $y \in \mathbb{C}^n$ there is a $\tilde{y} \in \mathbb{C}^n$ such that $APy = P\tilde{y}$ and thus

$$\begin{aligned}(I - P)Ay &= (I - P)A[(I - P)y + Py] \\ &= (I - P)A(I - P)y + (I - P)P\tilde{y} \\ &= (I - P)A(I - P)y.\end{aligned}$$

Therefore

$$\begin{aligned}& (I - P)K_m(A, r_0) \\ = \;& \text{span}\{(I - P)r_0, (I - P)Ar_0, \ldots, (I - P)A^{m-1}r_0\} \\ = \;& \text{span}\{(I - P)r_0, (I - P)A(I - P)r_0, \ldots, ((I - P)A)^{m-1}(I - P)r_0\} \\ = \;& K_m((I - P)A, (I - P)r_0).\end{aligned}$$

□

For the projections that we investigate in the following sections, range(P) is always A-invariant.
Lemma 3.7 tells that to build a basis $V_m = [v_1, \ldots, v_m]$ of $(I-P)K_m(A, r_0)$, we have to start with $v_1 = (I - P)r_0/\|(I - P)r_0\|_2$ and project out the Ω-parts after every multiplication with A.

Note that projecting out in every step of the Arnoldi method means running it with $(I-P)A$ instead of A, so the Arnoldi relations (1.7) and (1.8) change to

$$(I - P)AV_m = V_{m+1}\widehat{H}_m \tag{3.4}$$

and

$$V_m^H(I - P)AV_m = H_m \tag{3.5}$$

respectively, while $V_m^H(I - P)r_0 = \|(I - P)r_0\|_2 e_1$.

If P is an orthogonal projector, then $V_m^H P = 0$ because the columns of V_m span $K_m((I-P)A, (I-P)r_0) \subseteq$ range$(I-P)$ and range$(I-P) \perp$ range(P). In this case, the second Arnoldi relation (3.5) reads just the same as without projection as in this case $V_m^H P = 0$ and thus

$$V_m^H(I - P)AV_m = V_m^H AV_m = H_m.$$

The projections in every step are not necessary when range$(I - P)$ is A-invariant:

Lemma 3.8. If range$(I-P)$ is A-invariant, then

$$(I-P)K_m(A,r_0) = K_m(A,(I-P)r_0),$$

i.e., $(I-P)K_m(A,r_0)$ is a Krylov subspace.

Proof. Obviously, if range$(I-P)$ is A-invariant, then $(I-P)Ay = Ay$ for all $y \in \text{range}(I-P)$.

□

In this case the Arnoldi relations (1.7) and (1.8) do not change but of course $V_m^H(I-P)r_0 = \|(I-P)r_0\|_2 e_1$ still holds.

Having a basis W for Ω and a basis V_m for $(I-P)K_m(A,r_0)$, the approximations in the augmented space read

$$x_m = x_0 + Wy_1 + V_m y_2.$$

The residuals are from an augmented subspace of the same kind as well.

Lemma 3.9. When the iterates are chosen from an augmented Krylov subspace

$$x_m \in x_0 + \Omega + K_m((I-P)A,(I-P)r_0)$$

and Ω is A-invariant, then for the residuals $r_m = b - Ax_m$ it holds

$$r_m \in \Omega + K_{m+1}((I-P)A,(I-P)r_0).$$

Proof. Using Lemma 3.7 we get

$$\begin{aligned} r_m &= b - A(x_0 + Wy_1 + V_m y_2) \\ &= r_0 - AWy_1 - AV_m y_2 \\ &= Pr_0 - AWy_1 - PAVy_2 + (I-P)r_0 - (I-P)AVy_2 \\ &\in \Omega + K_{m+1}((I-P)A,(I-P)r_0). \end{aligned}$$

□

For methods using a (Petrov-)Galerkin condition we can apply Theorem 3.1 to the augmented case to see that the residuals stay collinear:

Lemma 3.10. Let $\dim(\Omega) = k$ and let $W_1 \subseteq W_2 \subseteq \cdots \subseteq W_j \subseteq \mathbb{C}^n$ with $\dim(W_m) = k+m$ and $W_m \cap (\Omega + K_{m+1}(A,r_0^i))^\perp = \{0\}$, $m = 1,\ldots,j$. Let the approximations $x_m^i \in x_0 + \Omega + K_m(A,r_0^i)$ to the solution of

$$(A-\sigma_i I)x = b$$

be chosen such that the residuals $r_m^i = b - (A-\sigma_i I)x_m^i$ fulfil the Petrov-Galerkin condition

$$r_m^i \perp W_m.$$

Then r_m^i and r_m^j are collinear if the starting residuals r_0^i and r_0^j are collinear.

Proof. If r_0^i and r_0^j are collinear, then $K_{m+1}(A, r_0^i) = K_{m+1}(A, r_0^j)$, so r_m^i and r_m^j are both from the same space

$$r_m^i, r_m^j \in \big(\Omega + K_{m+1}(A, r_0^i)\big) \cap (W_m)^\perp.$$

It holds $W_m^\perp + \Omega + K_{m+1}(A, r_0^i) = \mathbb{C}^n$ since $W_m \cap (\Omega + K_{m+1}(A, r_0^i))^\perp = \{0\}$. Since $\dim(K_{m+1}(A, r_0^i)) = m + 1$ and $\dim((W_m)^\perp) = n - m - k$, we get

$$\dim((\Omega + K_{m+1}(A, r_0^i)) \cap (W_m)^\perp) = 1.$$

□

The way we choose our approximations in the following sections either forces r_m to be orthogonal to Ω or zero out its Ω-part by minimization, so that actually $r_m \in K_{m+1}((I - P)A, (I - P)r_0)$.

We will investigate two variants of this deflation, namely Schur- and LR-deflation, and show how they apply to FOM, GMRES, BiCG and QMR.

3.3 Schur-Deflation

Let $S_k = [s_1, \ldots, s_k]$ be the matrix the columns s_i of which are the Schur vectors of the k smallest eigenvalues of the matrix A. So we have $S_k^H S_k = I_k$ and

$$AS_k = S_k T_k \tag{3.6}$$

where T_k is an upper triangular matrix with the k smallest eigenvalues of A on the diagonal.

In the case of a shifted matrix $A = M - \sigma I$ and S_k, T_k computed with respect to M we have

$$AS_k = MS_k - \sigma S_k = S_k(T_k - \sigma I_k).$$

As projector onto the subspace $\Omega_S = \text{span}\{s_1, \ldots, s_k\}$ we use the orthogonal projection $P = S_k S_k^H$.

Theorem 3.11. For the orthogonal projector $P = S_k S_k^H$ onto the subspace $\Omega_S = \text{span}\{s_1, \ldots, s_k\}$ it holds

1. $\text{range}(P) = \text{span}\{s_1, \ldots, s_k\}$ is A-invariant,
2. $\text{range}(I - P)$ is not A-invariant in general.

Proof.

1. Follows directly from (3.6).

2. Let $S_n = [s_1, \ldots, s_n]$ be the matrix of all Schur vectors, i.e.

$$AS_n = S_n T_n$$

with T_n upper triangular with the eigenvalues of A on the diagonal. Every $y \in \text{range}(I - P)$ can be written as

$$\sum_{i=k+1}^{n} \alpha_i s_i$$

with some α_i. Since $As_i = \sum_{j=1}^{i} T_{j,i} s_j$, multiplication with A will introduce Schur vectors s_i with index $i \leq k$.

□

Thus, to build a basis for $K_m((I - P)A, (I - P)r_0)$ projection will be necessary in every step.

For the Lanczos biorthogonalization we have to build a basis for the Krylov subspace $K_m(((I - P)A)^H, r) = K_m(A^H(I - P), r)$ for a starting vector r. Even though A^H and P do not commute, $K_m(A^H(I - P), r)$ is still orthogonal to Ω_S if $r \in \text{range}(I - P)$, i.e., if we choose $r = (I - P)r_0$, for example.

Theorem 3.12. For the orthogonal projector $P = S_k S_k^H$ onto the subspace $\Omega_S = \text{span}\{s_1, \ldots, s_k\}$ it holds

1. $\text{range}(P) = \text{span}\{s_1, \ldots, s_k\}$ is not A^H-invariant in general,
2. $\text{range}(I - P)$ is A^H-invariant.

Proof.

1. Analogous to the proof of Theorem 3.11 with $A^H S_n = S_n T_n^H$ and T_n^H a lower triangular matrix.
2. For $w = (I - P)y \in \text{range}(I - P)$ it holds

$$S_k^H A^H (I - S_k S_k^H) y = T_k^H S_k^H (I - S_k S_k^H) y = 0,$$

thus $A^H w \in \text{null}(P) = \text{range}(I - P)$.

□

Theorem 3.12 shows that $K_m(A^H(I - P), (I - P)r_0) = K_m(A^H, (I - P)r_0)$ and no projection, except for the starting vector, is needed to build a basis for this subspace.

In the following subsections we will investigate Schur-deflation for FOM, GMRES, BiCG and QMR. It will turn out that in all methods the Ω_S-part of the approximation depends on the Krylov-part. This is because every multiplication with A or M respectively introduces an additional Ω_S-part. The Ω_S-part does not have to be recomputed again and again in every step of the Krylov iteration, though. It can be computed once when the Krylov-part converged to the desired accuracy. The convergence of the Krylov iteration can be monitored cheaply using the residuals, which are actually the residuals with respect to the complete approximation, including the Ω_S-part.

3.3.1 FOM-Schur

Let now S_k and T_k be calculated with respect to M and let the columns of V_m span $K_m((I-P)A,(I-P)r_0)$. To choose the Schur deflated FOM approximation $x_m \in x_0 + \Omega_S + (I-P)K_m(A,r_0)$ to the solution of $Ax=b$, $A = M - \sigma I$, we use the Galerkin condition

$$r_m \perp \Omega_S + K_m((I-P)A,(I-P)r_0). \tag{3.7}$$

The projection P is orthogonal and thus $S_k^H V_m = 0$; the columns of V_m span $K_m((I-P)A,(I-P)r_0)$. Therefore, from (3.7) and Theorem 3.9 it directly follows that $r_m \in K_{m+1}((I-P)A,(I-P)r_0)$.

The chosen iterates are $x_m = x_0 + S_k y_m^1 + V_m y_m^2$ with $y_m^1 \in \mathbb{C}^k$ and $y_m^2 \in \mathbb{C}^m$ resulting from

$$[S_k\ V_m]^H A [S_k\ V_m] \begin{bmatrix} y_m^1 \\ y_m^2 \end{bmatrix} = \begin{bmatrix} S_k^H r_0 \\ V_m^H r_0 \end{bmatrix}.$$

Due to $V_m^H S_k = 0$, the representation of the projection and restriction of A onto the augmented subspace simplifies to block triangular structure

$$[S_k\ V_m]^H A [S_k\ V_m] = \begin{bmatrix} T_k - \sigma I_k & S^H A V_m \\ 0 & H_m - \sigma I_m \end{bmatrix} \in \mathbb{C}^{(k+m)\times(k+m)}.$$

For an arbitrary r_0 which might not lie in the orthogonal complement of Ω_S one therefore has

$$y_m^2 = (H_m - \sigma I)^{-1}(V_m^H P r_0 + V_m^H (I-P) r_0) = (H_m - \sigma I_m)^{-1} \|(I-P)r_0\|_2 e_1$$

and

$$y_m^1 = (T_k - \sigma I_k)^{-1}(S_k^H r_0 - S_k^H A V_m y_m^2).$$

As a part of y_m^1, namely $(T_k - \sigma I_k)^{-1} S_k^H r_0$, can be calculated beforehand, we can choose

$$\hat{x}_0 = x_0 + S_k (T_k - \sigma I_k)^{-1} S_k^H r_0$$

as starting vector. The starting residual

$$\hat{r}_0 = r_0 - AS_k(T_k - \sigma I_k)^{-1}S_k^H r_0 = (I - P)r_0$$

then lies in the orthogonal complement of Ω_S and therefore y_m^1 simplifies to

$$y_m^1 = -(T_k - \sigma I_k)^{-1}S_k^H AV_m y_m^2.$$

The iterate then reads

$$x_m = \hat{x}_0 - S_k(T_k - \sigma I_k)^{-1}S_k^H AV_m y_m^2 + V_m(H_m - \sigma I_m)^{-1}\|\hat{r}_0\|_2 e_1.$$

From now on we assume w.l.o.g that r_0 lies in the orthogonal complement of Ω_S, i.e., we assume that the $\hat{x}_0$ is taken as the starting vector. With this assumption we do not have to distinguish between the overall starting residual and the starting residuals after restarts.

Having multiple shifts σ_i, $i = 1, \ldots, s$, we start with collinear but not necessarily equal residuals $r_0^{(i)} = \rho_0^{(i)} r_0^{(1)}$, $i = 1, \ldots, s$. Obviously $\rho_0^{(1)} = 1$. We choose $r_0^{(1)}$ as starting vector for the Krylov subspace such that ${y_m^1}^{(i)}$ and ${y_m^2}^{(i)}$ are

$$\begin{aligned} {y_m^2}^{(i)} &= \rho_0^{(i)}(H_m - \sigma_i I_m)^{-1}\|r_0^{(1)}\|_2 e_1 \ , i = 1, \ldots, s \ , \\ {y_m^1}^{(i)} &= -(T_k - \sigma_i I_k)^{-1}S_k^H(M - \sigma_i I)V_m {y_m^2}^{(i)} \ , i = 1, \ldots, s \ . \end{aligned}$$

The residuals $r_m^{(i)}$ are collinear as well. This can be seen by directly applying Lemma 3.10 with $r_m^{(i)} \perp W_m$ where $W_m = \Omega_S + (I - P)K_m(A, r_0^{(1)})$ and $W_m \cap (\Omega_S + (I - P)K_{m+1}(A, r_0^{(1)}))^\perp = \{0\}$.

Besides the theoretical knowledge about the residuals staying collinear it will be necessary to compute the collinearity factors.

Proposition 3.13. For each $i = 1, \ldots, s$ let $x_m^{(i)} = x_0^{(i)} + S_k {y_m^1}^{(i)} + V_m {y_m^2}^{(i)}$ be the Schur deflated FOM approximation to $(M - \sigma_i I)x = r_0^{(i)} = \rho_0^{(i)} r_0^{(1)}$. Then

$$r_m^{(i)} = -h_{m+1,m}({y_m^2}^{(i)})_m v_{m+1}. \tag{3.8}$$

Proof. In the case of starting residuals without a part in Ω_S, i.e $r_0^{(i)} = \hat{r}_0^{(i)}$, we have for $i = 1, \ldots, s$

$$\begin{aligned} r_m^{(i)} &= r_0^{(i)} - (M - \sigma_i)x_m^{(i)} \\ &= r_0^{(i)} - (M - \sigma_i)(S_k {y_m^1}^{(i)} + V_m {y_m^2}^{(i)}) \\ &= r_0^{(i)} + S_k S_k^H(M - \sigma_i I)V_m {y_m^2}^{(i)} - (M - \sigma_i)V_m {y_m^2}^{(i)} \\ &= r_0^{(i)} - (I - S_k S_k^H)(M - \sigma_i I)V_m {y_m^2}^{(i)} \\ &= r_0^{(i)} - V_m(H_m - \sigma_i I_m){y_m^2}^{(i)} - h_{m+1,m}({y_m^2}^{(i)})_m v_{m+1} \\ &= -h_{m+1,m}({y_m^2}^{(i)})_m v_{m+1}. \end{aligned}$$

Setting $\rho_m^{(i)} = -h_{m+1,m}(y_m^{2\,(i)})_m$, $i = 1, \ldots, s$, we have $r_m^{(i)} = \rho_m^{(i)} v_{m+1}$.

The collinearity factor is given by (3.8) if $S_k S_k^H r_0^{(i)} \neq 0$, too. In this case we have for $i = 1, \ldots, s$

$$\begin{aligned} r_m^{(i)} &= r_0^{(i)} - (M - \sigma_i) x_m^{(i)} \\ &= -h_{m+1,m}(y_m^{2\,(i)})_m v_{m+1} + S_k S_k^H r_0^{(i)} - (M - \sigma_i) S_k (T - \sigma I_k)^{-1} S_k^H r_0^{(i)} \\ &= -h_{m+1,m}(y_m^{2\,(i)})_m v_{m+1} + S_k S_k^H r_0^{(i)} - S_k S_k^H r_0^{(i)} \\ &= -h_{m+1,m}(y_m^{2\,(i)})_m v_{m+1}. \end{aligned}$$

□

The residuals are thus not only in $K_{m+1}((I-P)A, (I-P)r_0)$, they are even multiples of the computed basis vectors.

Algorithm 3.14. FOM-Schur(k,m)
{**Input** $A = M - \sigma_i I$, $\{\sigma_1, \ldots, \sigma_s\}$, b, $S = S_k$, $T = T_k$}

$b^S = SS^H b$
$x^{(i)} = S(T - \sigma_i I)^{-1} S^H b^S$, $i = 1, \ldots, s$
$r = b - b^S$
$\rho^{(i)} = 1$, $i = 1, \ldots, s$
$\beta = \|r\|_2$
while not all systems converged **do**
  $v_1 = r/\beta$
  compute V_m, H_m by running Arnoldi with $(I - SS^H)M$
  for $i = 1, \ldots, s$ **do**
    $y_m^{2\,(i)} = \beta \rho^{(i)} (H_m - \sigma_i I_m)^{-1} e_1$
    $y_m^{1\,(i)} = -(T - \sigma_i I)^{-1} S^H (M - \sigma_i I) V_m y_m^{2\,(i)}$
    $x^{(i)} = x^{(i)} + S y_m^{1\,(i)} + V_m y_m^{2\,(i)}$
  end for
  $r = v_{m+1}$
  $\rho^{(i)} = -h_{m+1,m}(y_m^{2\,(i)})_m$, $i = 1, \ldots, s$
  $\beta = h_{m+1,m}$
end while

3.3.2 GMRES-Schur

Again, let S_k and T_k be calculated with respect to M. To choose the Schur deflated GMRES approximation $x_m \in x_0 + \Omega_S + (I - P)K_m(A, r_0)$ to the solution of $Ax = b$, $A = M - \sigma I$, we now use the minimal residual condition.

The chosen iterates are $x_m = x_0 + S_k y_m^1 + V_m y_m^2$ with $y_m^1 \in \mathbb{C}^k$ and $y_m^2 \in \mathbb{C}^m$ and the residual that has to be minimized is

$$r_m = b - (M - \sigma I)(x_0 + S_k y_m^1 + V_m y_m^2).$$

From the Arnoldi relation (3.4) we get

$$(M - \sigma I)V_m y_m^2 = V_{m+1}(\widehat{H}_m - \sigma I)y_m^2 + P(M - \sigma I)V_m y_m^2$$

such that we can separate the parts of r_m lying in Ω_S from those lying in $K_{m+1}((I-P)A, (I-P)r_0)$:

$$r_m = Pr_0 - S_k(T_k - \sigma I)y_m^1 - P(M - \sigma I)V_m y_m^2 + (I-P)r_0 - V_{m+1}(\widehat{H}_m - \sigma I)y_m^2.$$

Thus $r_m = S_k u + V_{m+1} w$ for some $u \in \mathbb{C}^k$, $w \in \mathbb{C}^{m+1}$ and S and V_{m+1} are orthogonal. Although the Krylov part influences the Ω_S part, the Ω_S part can be made zero for any choice of y_m^2 by solving

$$(T_k - \sigma I)y_m^1 = S_k^H r_0 - S_k^H (M - \sigma I)V_m y_m^2. \tag{3.9}$$

To minimize $\|r_m\|_2$ we therefore first choose the Krylov part y_m^2 such that it minimizes $\|(I-P)r_0 - V_{m+1}(\widehat{H}_m - \sigma I)y_m^2\|_2$. Afterwards y_m^1 can be computed by solving (3.9).

The residual is then actually just $r_m = (I-P)r_0 - V_{m+1}(\widehat{H}_m - \sigma I)y_m^2$, so as in the FOM case, the residuals are from $K_{m+1}((I-P)A, (I-P)r_0)$ and orthogonal to Ω_S.

In the same manner as in the FOM case, a part of y_m^1 can be calculated beforehand. Starting with $\hat{x}_0 = x_0 + S_k(T_k - \sigma I)^{-1} S_k^H r_0$ leads to a corresponding starting residual $\hat{r}_0 = (I-P)r_0$ and y_m^1 simplifies to

$$y_m^1 = -(T_k - \sigma I)^{-1} S_k^H (M - \sigma I)V_m y_m^2.$$

From now on we assume w.l.o.g that r_0 lies in the orthogonal complement of Ω_S.

Concerning multiple shifts we assume to have collinear starting residuals, i.e. $r_0^{(i)} = \rho_0^{(i)} r_0^{(1)}$, $i = 1, \ldots, s$. As starting vector for the Krylov subspace we choose $r_0^{(1)}$.
The latter computations to obtain ${y_m^1}^{(i)}$ and ${y_m^2}^{(i)}$ are done for the system $i = 1$ only, while for the systems $i = 2, \ldots, s$ we demand, similarly to the non-deflated version,

$$r_m^{(i)} = \rho_0^{(i)} r_0^{(1)} - (M - \sigma_i I)(S_k {y_m^1}^{(i)} + V_m {y_m^2}^{(i)}) = \rho_m^{(i)} r_m^{(1)}. \tag{3.10}$$

As $r_m^{(1)}$ is orthogonal to Ω_S, so are the $r_m^{(i)}$, $i = 2, \ldots, s$. So we keep for ${y_m^1}^{(i)}$

$$(T_k - \sigma_i I){y_m^1}^{(i)} = -S_k^H (M - \sigma_i I) V_m {y_m^2}^{(i)},$$

such that equation (3.10) reads

$$\rho_0^{(i)} r_0^{(1)} - V_{m+1}(\widehat{H}_m - \sigma_i I){y_m^2}^{(i)} = \rho_m^{(i)} r_m^{(1)}.$$

Therefore, ${y_m^2}^{(i)}$ and $\rho_m^{(i)}$ are obtained from the system

$$\left[\widehat{H}_m - \sigma_i I | V_{m+1}^H r_m^{(1)}\right] \begin{bmatrix} {y_m^2}^{(i)} \\ \rho_m^{(i)} \end{bmatrix} = \rho_0^{(i)} \|r_0^{(1)}\|_2 e_1. \tag{3.11}$$

Algorithm 3.15. GMRES-Schur(k,m)

{**Input** $A = M - \sigma_i I$, $\{\sigma_1, \ldots, \sigma_s\}$, b, $S = S_k$, $T = T_k$}

$b^S = SS^H b$
$x^{(i)} = S(T - \sigma_i I)^{-1} S^H b^S$, $i = 1, \ldots, s$
$r = b - b^S$
$\rho^{(i)} = 1$, $i = 1, \ldots, s$
$\beta = \|r\|_2$
while not all systems converged **do**
  $v_1 = r/\beta$
  compute V_m, H_m by running Arnoldi with $(I - SS^H)M$
  compute ${y_m^2}^{(1)}$ by minimizing $\|r - V_{m+1}(\widehat{H}_m - \sigma_1 I){y_m^2}^{(1)}\|_2$
  for $i = 2, \ldots, s$ **do**
    compute ${y_m^2}^{(i)}$ and $\rho_m^{(i)}$ from (3.11)
  end for
  for $i = 1, \ldots, s$ **do**
    compute ${y_m^1}^{(i)} = -(T - \sigma_i I)^{-1} S^H (M - \sigma_i I) V_m {y_m^2}^{(i)}$
  end for
  $x^{(i)} = x^{(i)} + S{y_m^1}^{(i)} + V_m {y_m^2}^{(i)}$
  $r = r - V_{m+1}(\widehat{H}_m - \sigma_1 I){y_m^2}^{(1)}$
  $\beta = \|r\|_2$
end while

Same as without deflation, see Theorem 3.3, in the Schur deflated version the first system (i.e. $M - \sigma_1 I$) rules the convergence of all systems. To show that, we follow the proof of Theorem 3.3, see [17].

In ordinary GMRES the iterates x_m are chosen from an affine Krylov subspace and can be represented as

$$x_m = x_0 + q_{m-1}(A) r_0,$$

with q_{m-1} a polynomial of degree at most $m-1$.

In Schur deflated GMRES this is no longer the case, as the iterates involve an additional part belonging to the deflated eigenvalues:

$$x_m = x_0 + Sy_m^1 + V_m y_m^2 = x_0 + Sy_m^1 + q_{m-1}((I-P)A)r_0,$$

with $P = SS^H$.

Nevertheless, for the corresponding residuals it holds, just as for ordinary GMRES,

$$\begin{aligned} r_m &= b - Ax_0 - (I-P)AV_m y_m^2 \\ &= r_0 - (I-P)AV_m y_m^2 \\ &= r_0 - (I-P)Aq_{m-1}((I-P)A)r_0 \\ &= p_m((I-P)A)r_0, \end{aligned}$$

where $p_m(t) = 1 - tq_{m-1}(t)$ is a polynomial of degree at most m with $p_m(0) = 1$.

For the shifted systems this reads

$$r_m^{(i)} = p_m^{(i)}((I-P)(M-\sigma_i I))r_0^{(i)},$$

and as we required $r_m^{(i)} = \rho_m^{(i)} r_m^{(1)}$ and $r_0^{(i)} = \rho_0^{(i)} r_0^{(1)}$, we get

$$\rho_0^{(i)} p_m^{(i)}((I-P)(M-\sigma_i I))r_0^{(1)} = \rho_m^{(i)} p_m^{(1)}((I-P)(M-\sigma_1 I))r_0^{(1)}. \quad (3.12)$$

Fortunately, since P is a projection and $(I-P)r_0 = r_0$, the following holds:

$$\begin{aligned} p_m^{(1)}((I-P)(M-\sigma_1 I))r_0^{(1)} &= p_m^{(1)}((I-P)(M-\sigma_i I + \sigma_i I - \sigma_1 I))r_0^{(1)} \\ &= p_m^{(1)}((I-P)(M-\sigma_i I) + (\sigma_i - \sigma_1)I)r_0^{(1)}. \end{aligned}$$

Therefore, (3.12) is equivalent to

$$\rho_0^{(i)} p_m^{(i)}(t) = \rho_m^{(i)} p_m^{(1)}(t + (\sigma_i - \sigma_1)).$$

Obviously $p_m^{(i)}(0) = 1$ is satisfied if and only if $\rho_0^{(i)} = \rho_m^{(i)} p_m^{(1)}(\sigma_i - \sigma_1)$ and this is fulfilled if and only if $p_m^{(1)}(\sigma_i - \sigma_1) \neq 0$ and $\rho_m^{(i)} = \rho_0^{(i)} / p_m^{(1)}(\sigma_i - \sigma_1)$.

The following lemma and theorem are the Schur deflated versions of the ones found in [17]. For the deflated versions the proofs had to be modified.

Lemma 3.16. Let $r_m = p_m((I-P)A)r_0$ be the Schur deflated GMRES residual and let A be positive real. Then all zeros ζ of p_m satisfy

$$\frac{1}{\zeta} \in \mathrm{F}(A^{-H}) \tag{3.13}$$

with $\mathrm{F}(A) = \{\langle Ax, x\rangle | x \in \mathbb{C}^n, \|x\|_2 = 1\}$ being the field of values and

$$\mathrm{Re}(\zeta) > 0. \tag{3.14}$$

Proof. Since $p_m(0) = 1$, we know that $\zeta \neq 0$ for any zero ζ of p_m, so we can write

$$p_m(t) = (1 - \frac{t}{\zeta})\widehat{p}_{m-1}(t)$$

with $\widehat{p}_{m-1}(0) = 1$. Writing $u = \widehat{p}_{m-1}((I-P)A)r_0$ and $w = (I-P)Au$ gives

$$\|r_m\|_2 = \|p_m((I-P)A)r_0\|_2 = \|u - \frac{1}{\zeta}w\|_2,$$

which is minimized for

$$\frac{1}{\zeta} = \frac{\langle w, u\rangle}{\langle w, w\rangle}.$$

It holds $(I-P)A^{-1}w = (I-P)A^{-1}(I-P)Au$ and since range(P) is A^{-1}-invariant, it follows $(I-P)A^{-1}w = (I-P)u$. Therefore, we get

$$\begin{aligned} \langle w, u\rangle &= \langle Au, (I-P)u\rangle \\ &= \langle Au, (I-P)A^{-1}w\rangle \\ &= \langle (I-P)Au, A^{-1}w\rangle \\ &= \langle A^{-H}w, w\rangle, \end{aligned}$$

and thus

$$\frac{1}{\zeta} = \frac{\langle w, u\rangle}{\langle w, w\rangle} = \frac{\langle A^{-H}w, w\rangle}{\langle w, w\rangle} \in \mathrm{F}(A^{-H})$$

which proves (3.13).

For $y = Ax$ it holds

$$\langle A^{-H}y, y\rangle = \langle Ax, x\rangle,$$

so if $\mathrm{F}(A)$ is contained in the right half-plane, so is $\mathrm{F}(A^{-H})$. Therefore, $\mathrm{Re}(\frac{1}{\zeta}) > 0$ and thus $\mathrm{Re}(\zeta) > 0$, which proves (3.14).

□

Theorem 3.17. Let $(M - \sigma_1 I)$ be positive real and $0 > \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_s$. Then the restarted shifted Schur deflated GMRES method converges for all shifted systems and for every restart value. The iterates for all shifted systems always exist, and we have

$$\|r_m^{(i)}\|_2 \leq |\rho_0^{(i)}| \cdot \|r_m^{(1)}\|_2. \tag{3.15}$$

Proof. Although the matrix $(I-P)(M-\sigma_1 I)$ is singular, the matrices $V_m^H(I-P)(M-\sigma_1 I)V_m = V_m^H(M-\sigma_1 I)V_m = H_m - \sigma_1 I$ are positive real because $\operatorname{rank}(V_m) = m$. Therefore, the convergence of the first system is a well-known result on restarted shifted GMRES, see [42, 41].

For (3.15) we have to show that $|\rho_m^{(i)}| \leq |\rho_0^{(i)}|$. We know that

$$\rho_m^{(i)} = \rho_0^{(i)} / p_m^{(1)}(\sigma_i - \sigma_1),$$

so we have to show $|p_m^{(1)}(\sigma_i - \sigma_1)| \geq 1$. For $\sigma_i = \sigma_1$ there is nothing to show since $p_m^{(1)}(0) = 1$. Therefore we assume $\sigma_i \neq \sigma_1$.

From Lemma 3.16 we know that

$$p_m^{(1)}(t) = \prod_{i=1}^{m}(1 - t/\zeta_i) \text{ with } \operatorname{Re}(\zeta_i) > 0, i = 1, \ldots, m.$$

For a $\tau < 0$ obviously $|1 - \tau/\zeta_i| > 1$ holds so that

$$|p_m^{(1)}(\sigma_i - \sigma_1)| = \prod_{i=1}^{m} |1 - (\sigma_i - \sigma_1)/\zeta_i| > 1$$

since $\sigma_i - \sigma_1 < 0$.

□

Theorem 3.17 tells us that the system with eigenvalues closest to the imaginary axis is the system with the slowest convergence. Monitoring the residual of this system is thus a safe way of controlling the overall convergence.

3.3.3 BiCG-Schur

Let S_k and T_k be calculated with respect to M. Let the columns of V_m span $K_m((I-P)A, (I-P)r_0)$ and the columns of W_m span $K_m(A^H, (I-P)r_0)$ with $V_m^H W_m = I$. To choose the Schur deflated BiCG approximation $x_m \in x_0 + \Omega_S + K_m((I-P)A, (I-P)r_0)$ to the solution of $Ax = b$, $A = M - \sigma I$, we use the Petrov-Galerkin condition

$$r_m \perp \Omega_S + K_m(A^H, (I-P)r_0). \tag{3.16}$$

As in the FOM case, the residual lies in $K_{m+1}((I-P)A, (I-P)r_0)$.

The chosen iterates are $x_m = x_0 + S_k y_m^1 + V_m y_m^2$ with $y_m^1 \in \mathbb{C}^k$ and $y_m^2 \in \mathbb{C}^m$ resulting from

$$[S_k\ W_m]^H A [S_k\ V_m] \begin{bmatrix} y_m^1 \\ y_m^2 \end{bmatrix} = \begin{bmatrix} S_k^H r_0 \\ W_m^H r_0 \end{bmatrix}.$$

Due to $W_m^H S_k = 0$, the representation of the projection and restriction of A onto the augmented subspace simplifies to block triangular structure in the same way as in the FOM case

$$[S_k\ W_m]^H A[S_k\ V_m] = \begin{bmatrix} T_k - \sigma I_k & S_k^H A V_m \\ 0 & H_m - \sigma I_m \end{bmatrix} \in \mathbb{C}^{(k+m)\times(k+m)}$$

such that for arbritrary r_0

$$\begin{aligned} y_m^1 &= (T_k - \sigma I_k)^{-1}(S_k^H r_0 - S_k^H A V_m y_m^2), \\ y_m^2 &= (H_m - \sigma I_m)^{-1} W_m^H r_0. \end{aligned}$$

Starting with

$$\hat{x}_0 = x_0 + S_k(T_k - \sigma I_k)^{-1} S_k^H r_0,$$

the starting residual

$$\hat{r}_0 = r_0 - A S_k (T_k - \sigma I_k)^{-1} S_k^H r_0 = (I - P) r_0$$

lies in the orthogonal complement of Ω_S. In this case $r_0 = \gamma V_m e_1$ and $W_m^H r_0 = \gamma e_1$, thus

$$y_m^2 = \gamma (H_m - \sigma I_m)^{-1} e_1.$$

Algorithm 3.18. BiCG-Schur(k,m)
{**Input** $A = M - \sigma_i I$, $\{\sigma_1, \ldots, \sigma_s\}$, b, $S = S_k$, $T = T_k$}

$b^S = S S^H b$
$x^{(i)} = S(T - \sigma_i I)^{-1} S^H b^S$, $i = 1, \ldots, s$
$r = b - b^S$
$\beta = \|r\|_2$
while not all systems converged **do**
  $\tilde{v}_1 = \tilde{w}_1 = r$
  for $i = 1, \ldots, s$ **do**
    compute $V_m y_m^{2\,(i)}$ by BiCG-Sh using $(I - P)M$ for building V_m
    compute $y_m^{1\,(i)} = -(T - \sigma_i I)^{-1} S^H (M - \sigma_i I) V_m y_m^{2\,(i)}$
    $x^{(i)} = x^{(i)} + S y_m^{1\,(i)} + V_m y_m^{2\,(i)}$
  end for
end while

3.3.4 QMR-Schur

Let S_k and T_k be calculated with respect to M. Let the columns of V_m span $K_m((I - P)A, (I - P)r_0)$ and the columns of W_m span $K_m(A^H, (I - P)r_0)$ with $V_m^H W_m = I$. To choose the Schur deflated QMR approximation $x_m \in$

$x_0 + \Omega_S + (I-P)K_m(A, r_0)$ to the solution of $Ax = b$, $A = M - \sigma I$, we split the residual

$$r_m = r_0 - S_k T_k y_m^1 - AV_m y_m^2$$

as we did for the GMRES case

$$r_m = Pr_0 - S_k T_k y_m^1 - PAV_m y_m^2 + (I-P)r_0 - V_{m+1}\widehat{H}_m y_m^2.$$

Minimizing the Ω_S-part of the residual by solving

$$T_k y_m^1 = S_k^H r_0 - S_k^H AV_m y_m^2$$

leaves

$$r_m = (I-P)r_0 - V_{m+1}\widehat{H}_m y_m^2 = \|(I-P)r_0\|_2 V_{m+1} e_1 - V_{m+1}\widehat{H}_m y_m^2.$$

As V_{m+1} is not orthogonal, we demand the quasi minimal residual property instead of the minimal residual property

$$\|\|(I-P)r_0\|_2 e_1 - \widehat{H}_m y_m\|_2 = \min_{y \in \mathbb{C}^m} \|\|(I-P)r_0\|_2 e_1 - \widehat{H}_m y\|_2. \qquad (3.17)$$

Algorithm 3.19. QMR-Schur(k,m)
{**Input** $A = M - \sigma_i I$, $\{\sigma_1, \ldots, \sigma_s\}$, b, $S = S_k$, $T = T_k$}

$b^S = SS^H b$
$x^{(i)} = S(T - \sigma_i I)^{-1} S^H b^S$, $i = 1, \ldots, s$
$r = b - b^S$
$\beta = \|r\|_2$
while not all systems converged **do**
  $\tilde{v}_1 = \tilde{w}_1 = r$
  for $i = 1, \ldots, s$ **do**
    compute $V_m y_m^{2\,(i)}$ by QMR-Sh using $(I-P)M$ for building V_m
    compute $y_m^{1\,(i)} = -(T - \sigma_i I)^{-1} S^H (M - \sigma_i I) V_m y_m^{2\,(i)}$
    $x^{(i)} = x^{(i)} + S y_m^{1\,(i)} + V_m y_m^{2\,(i)}$
    $r^{(i)} = r - V_{m+1}(\widehat{H}_m - \sigma_i I) y_m^{2\,(i)}$
  end for
end while

3.4 LR-deflation

Using both left and right eigenvalues leads to an alternative deflation scheme where additional projection is no longer necessary during the Arnoldi iteration.

Let $R_k = [r_1, \ldots, r_k]$ be the matrix containing the right eigenvectors corresponding to the k smallest eigenvalues of the matrix A and $L_k^H = [l_1, \ldots, l_k]^H$ the matrix containing the left eigenvectors. With Λ_k the diagonal eigenvalue matrix, the left and right eigenvectors satisfy

$$AR_k = R_k\Lambda_k$$

and

$$L_k{}^H A = \Lambda_k L_k{}^H.$$

If $A = M - \sigma I$ is a shifted matrix and R_k, L_k and Λ_k are calculated with respect to M, then

$$AR_k = R_k(\Lambda_k - \sigma I)$$

and

$$L_k{}^H A = (\Lambda_k - \sigma I)L_k{}^H.$$

Left and right eigenvectors corresponding to different eigenvalues are orthogonal [24]. Let the eigenvectors be furthermore normalized such that

$$L_k{}^H R_k = I_k.$$

As projector onto $\Omega_R = \text{span}\{r_1, \ldots, r_k\}$ we use $P = R_k L_k^H$. Obviously P is an oblique projector. More precisely, P is projector onto Ω_R along a subspace containing the remaining eigenvectors since

$$\text{range}(P) = \text{span}\{r_1, \ldots, r_k\}$$

and $\text{null}(P) = \text{span}\{r_{k+1}, \ldots, r_n\} = \text{null}(L_k{}^H)$.

For our purpose, the following properties of P are important.

Lemma 3.20. The projector $P = R_k L_k{}^H$ and A commute.

Proof. Obviously, the following holds

$$AR_k L_k{}^H = R_k\Lambda_k L_k{}^H = R_k L_k{}^H A$$

□

Lemma 3.21. For the oblique projector $P = R_k L_k{}^H$ both $\text{range}(P)$ and $\text{range}(I - P)$ are A-invariant.

Proof. Given $y = Pv \in \text{range}(P)$ we get

$$Ay = APv = PAv \in \text{range}(P).$$

The proof for $(I - P)$ is just the same. □

Therefore, when taking $P = R_k L_k^H$, there is no need of additional projection to build a basis for $(I-P)K_m(A, r_0) = K_m(A, (I-P)r_0)$.

Again, for the Lanczos biorthogonalization it will be necessary to build a basis for the Krylov subspace $K_m(((I-P)A)^H, r)$ for a starting vector r. In this case, as P and A commute, $K_m(((I-P)A)^H, r) = K_m((I-P)^H A^H, r)$.

Theorem 3.22. For the oblique projector $P = R_k {L_k}^H$ both range(P^H) and range$((I-P)^H)$ are A^H-invariant.

Proof. As A and P commute, so do A^H and P^H. □

Theorem 3.22 shows that starting with $r = (I-P)^H r_0$ no further projection is necessary to build a basis of $(I-P)K_m(A^H, r_0) = K_m(A^H, (I-P)^H r_0)$.

In the following subsections we will investigate LR-deflation for FOM, GMRES, BiCG and QMR. It will turn out that the Ω_R-part can be computed before starting the Krylov iteration. This is no surprise as neither by multiplication with A nor by multiplication with A^H, additional parts lying in Ω_R are produced.

3.4.1 FOM-LR

Let R_k, L_k and Λ_k be calculated with respect to M and let the columns of V_m span $K_m(A, (I-P)r_0)$. To choose the LR deflated FOM approximation $x_m \in x_0 + \Omega_R + K_m(A, (I-P)r_0)$ to the solution of $Ax = b$, $A = M - \sigma I$, we use the subspaces $\Omega_R = \text{span}\{r_1, \ldots, r_k\}$ and $\Omega_L = \text{span}\{l_1, \ldots, l_k\}$ and the Petrov-Galerkin condition

$$r_m \perp \Omega_L + K_m(A, (I-P)r_0). \tag{3.18}$$

Theorem 3.23. The LR-deflated FOM approximation is

$$x_m = x_0 + R_k y_m^1 + V_m y_m^2,$$

where $y_m^1 \in \mathbb{C}^k$ and $y_m^2 \in \mathbb{C}^m$ are the solutions of the independent linear systems

$${L_k}^H A R_k y_m^1 = (\Lambda_k - \sigma I) y_m^1 = {L_k}^H r_0 \tag{3.19}$$

and

$$V_m^H A V_m y_m^2 = V_m^H (I-P) r_0 = \beta e_1. \tag{3.20}$$

Proof. (3.18) leads to

$$\begin{bmatrix} \Lambda_k - \sigma I & 0 \\ V_m^H A R_k & V_m^H A V_m \end{bmatrix} \begin{bmatrix} y_m^1 \\ y_m^2 \end{bmatrix} = \begin{bmatrix} {L_k}^H r_0 \\ V_m^H r_0 \end{bmatrix}.$$

For the system for y_m^1 there is nothing more to show. For y_m^2 we get

$$V_m^H A V_m y_m^2 = V_m^H r_0 - V_m^H R_k(\Lambda_k - \sigma I) y_m^1 = V_m^H (I - P) r_0.$$

□

The system (3.19) can be solved before starting the Arnoldi iteration and does not have any influence on y_m^2. To compute y_m^2, a (non-deflated) FOM is run with starting vector $(I - P)r_0$ instead of r_0.

As LR-deflation does not change the FOM iteration, applying multishifts and restarts is straightforward.

Algorithm 3.24. FOM-LR(k,m)
{**Input** $A = M - \sigma_i I$, $\{\sigma_1, \ldots, \sigma_s\}$, b, $L = L_k$, $R = R_k$, $\Lambda = \Lambda_k$}

$x^{(i)} = R(\Lambda - \sigma_i I)^{-1} L^H b$, $i = 1, \ldots, s$
$r = b - RL^H b$
$\rho^{(i)} = 1$, $i = 1, \ldots, s$
$\beta = \|r\|_2$
while not all systems converged **do**
  $v_1 = r/\beta$
  compute V_m, H_m by running Arnoldi with M
  for $i = 1, \ldots, s$ **do**
    ${y_m}^{(i)} = \beta\rho^{(i)}(H_m - \sigma_i I_m)^{-1} e_1$
    $x^{(i)} = x^{(i)} + V_m {y_m}^{(i)}$
  end for
  $r = v_{m+1}$
  $\rho^{(i)} = -h_{m+1,m}({y_m}^{(i)})_m$, $i = 1, \ldots, s$
  $\beta = h_{m+1,m}$
end while

3.4.2 GMRES-LR

Let again R_k, L_k and Λ_k be calculated with respect to M and let the columns of V_m span $K_m(A, (I - P)r_0)$. To choose the LR deflated GMRES approximation $x_m \in x_0 + \Omega_R + K_m(A, (I - P)r_0)$ to the solution of $Ax = b$, $A = M - \sigma I$, the intention is to use the minimal residual condition. Since the LR-projection is not orthogonal, it turns out that only a quasi minimal residual condition can be demanded.

Let $x_m = x_0 + [R_k V_m] y_m$ with $y_m = [y_m^1, y_m^2]^T$ and $y_m^1 \in \mathbb{C}^k$, $y_m^2 \in \mathbb{C}^m$. The

corresponding residual is

$$\begin{aligned} r_m &= r_0 - A[R_k V_m] y_m \\ &= r_0 - [R_k V_{m+1}] \begin{bmatrix} \Lambda_k - \sigma I & 0 \\ 0 & \widehat{H}_m - \sigma I \end{bmatrix}. \end{aligned}$$

Since $\beta v_1 = (I - P) r_0$ it holds

$$r_0 = P r_0 + (I - P) r_0 = [R_k V_{m+1}] \begin{bmatrix} L_k^H r_0 \\ \beta e_1 \end{bmatrix}$$

and thus

$$r_m = [R_k V_{m+1}] \left(\begin{bmatrix} L_k^H r_0 \\ \beta e_1 \end{bmatrix} - \begin{bmatrix} \Lambda_k - \sigma I & 0 \\ 0 & \widehat{H}_m - \sigma I \end{bmatrix} y_m \right).$$

Since $[R_k V_{m+1}]$ is not orthogonal, it holds

$$\|r_m\|_2 \neq \| \begin{bmatrix} L_k^H r_0 \\ \beta e_1 \end{bmatrix} - \begin{bmatrix} \Lambda_k - \sigma I & 0 \\ 0 & \widehat{H}_m - \sigma I \end{bmatrix} y_m \|_2.$$

Demanding the quasi minimal residual condition and thus minimizing

$$\| \begin{bmatrix} L_k^H r_0 \\ \beta e_1 \end{bmatrix} - \begin{bmatrix} \Lambda_k - \sigma I & 0 \\ 0 & \widehat{H}_m - \sigma I \end{bmatrix} y_m \|_2$$

splits the least squares problem into a $k \times k$ linear system

$$(\Lambda_k - \sigma I) y_m^1 = L_k^H r_0$$

and the smaller least squares problem

$$\|\beta e_1 - (\widehat{H}_m - \sigma I) y_m^2\|_2 = \min_{y \in \mathbb{C}^m} \|\beta e_1 - (\widehat{H}_m - \sigma I) y_m^2\|_2.$$

Actually, since V_{m+1} is orthogonal, y_m^2 is the non-deflated GMRES approximation with starting vector $(I - P) r_0$ instead of r_0.

As for FOM, LR-deflation does not change GMRES. Therefore, applying multishifts and restarts is straightforward here as well.

Algorithm 3.25. GMRES-LR(k,m)
{**Input** $A = M - \sigma_i I$, $\{\sigma_1, \ldots, \sigma_s\}$, b, $L = L_k$, $R = R_k$, $\Lambda = \Lambda_k$}

$x^{(i)} = R(\Lambda - \sigma_i I)^{-1} L^H b$, $i = 1, \ldots, s$
$r = b - RL^H b$
$\rho^{(i)} = 1$, $i = 1, \ldots, s$
$\beta = \|r\|_2$
while not all systems converged **do**
  $v_1 = r/\beta$
  compute V_m, $\widehat{H}_m$ by running Arnoldi
  compute GMRES approximation $y_m^{2\,(1)}$
  for $i = 2, \ldots, s$ **do**
    compute $y_m^{2\,(i)}$ and $\rho_m^{(i)}$ from (3.3)
  end for
  $x^{(i)} = x^{(i)} + V_m y_m^{2\,(i)}$
  $r = r - V_{m+1}(\widehat{H}_m - \sigma_1 I) y_m^{2\,(1)}$
  $\beta = \|r\|_2$
end while

3.4.3 BiCG-LR

Let R_k, L_k and Λ_k be calculated with respect to M and let the columns of V_m span $K_m(A, (I-P)r_0)$ while the columns of W_m span $K_m(A^H, (I-P)^H r_0)$ with $V_m^H W_m = I$. To choose the LR deflated BiCG approximation $x_m \in x_0 + \Omega_R + K_m(A, (I-P)r_0)$ to the solution of $Ax = b$, $A = M - \sigma I$, we use the Petrov-Galerkin condition

$$r_m \perp \Omega_L + K_m(A^H, (I-P)^H r_0).$$

The chosen iterates are $x_m = x_0 + R_k y_m^1 + V_m y_m^2$ with $y_m^1 \in \mathbb{C}^k$ and $y_m^2 \in \mathbb{C}^m$ resulting from

$$[L_k\ W_m]^H A [R_k\ V_m] \begin{bmatrix} y_m^1 \\ y_m^2 \end{bmatrix} = \begin{bmatrix} L_k^H r_0 \\ W_m^H r_0 \end{bmatrix}.$$

As $W_m^H R_k = 0$ and $L_k^H V_m = 0$, the representation of the projection and restriction of A reduces to block diagonal structure

$$[L_k\ W_m]^H A [R_k\ V_m] = \begin{bmatrix} \Lambda_k - \sigma I_k & 0 \\ 0 & H_m - \sigma I_m \end{bmatrix} \in \mathbb{C}^{(k+m)\times(k+m)}$$

such that for arbritrary r_0

$$\begin{aligned} y_m^1 &= (\Lambda_k - \sigma I_k)^{-1} L_k^H r_0 \\ y_m^2 &= (H_m - \sigma I_m)^{-1} W_m^H r_0. \end{aligned}$$

Here the Ω_R-part and the Krylov-part of the approximation are completely independent. The Ω_R-part can be calculated beforehand and included in x_0

$$\hat{x}_0 = x_0 + R_k(\Lambda_k - \sigma I_k)^{-1} L_k^H r_0$$

such that the starting residual is

$$\hat{r}_0 = r_0 - AR_k(\Lambda_k - \sigma I_k)^{-1} L_k^H r_0 = (I - P)r_0.$$

We can therefore assume to start with a residual $r_0 \in \text{null}(P)$. In this case, $r_0 = \gamma V_m e_1$ and $W_m^H r_0 = \gamma e_1$, yielding

$$y_m^2 = \gamma(H_m - \sigma I_m)^{-1} e_1.$$

Algorithm 3.26. BiCG-LR(k,m)
{**Input** $A = M - \sigma_i I$, $\{\sigma_1, \ldots, \sigma_s\}$, b, $L = L_k$, $R = R_k$, $\Lambda = \Lambda_k$}

$b^S = RL^H b$
$x^{(i)} = R(\Lambda - \sigma_i I)^{-1} L^H b$, $i = 1, \ldots, s$
$r = b - b^S$
$\beta = \|r\|_2$
while not all systems converged **do**
  $\tilde{v}_1 = (I - RL^H)b$, $\tilde{w}_1 = (I - LR^H)b$
  for $i = 1, \ldots, s$ **do**
    compute $V_m y_m^{2\,(i)}$ by BiCG-Sh
    $x^{(i)} = x^{(i)} + V_m y_m^{2\,(i)}$
    $r^{(i)} = r - V_{m+1}(\widehat{H}_m - \sigma_i I) y_m^{2\,(i)}$
  end for
end while

3.4.4 QMR-LR

Let R_k, L_k and Λ_k be calculated with respect to M and let the columns of V_m span $K_m(A, (I-P)r_0)$ while the columns of W_m span $K_m(A^H, (I-P)^H r_0)$ with $V_m^H W_m = I$. To choose the LR deflated QMR approximation $x_m \in x_0 + \Omega_R + K_m(A, (I-P)r_0)$ to the solution of $Ax = b$, $A = M - \sigma I$, we use the quasi minimal residual condition.

Let $x_m = x_0 + [R_k V_m] y_m$, with $y_m = [y_m^1, y_m^2]^T$ and $y_m^1 \in \mathbb{C}^k$, $y_m^2 \in \mathbb{C}^m$. The

corresponding residual is

$$\begin{aligned} r_m &= r_0 - A[R_k V_m] y_m \\ &= r_0 - [R_k V_{m+1}] \begin{bmatrix} \Lambda_k - \sigma I & 0 \\ 0 & \widehat{H}_m - \sigma I \end{bmatrix} \\ &= [R_k V_{m+1}] \left(\begin{bmatrix} L_k^H r_0 \\ \beta e_1 \end{bmatrix} - \begin{bmatrix} \Lambda_k - \sigma I & 0 \\ 0 & \widehat{H}_m - \sigma I \end{bmatrix} y_m \right). \end{aligned}$$

Since $[R_k V_{m+1}]$ is not orthogonal, we demand the quasi minimal residual condition and minimize

$$\| \begin{bmatrix} L_k^H r_0 \\ \beta e_1 \end{bmatrix} - \begin{bmatrix} \Lambda_k - \sigma I & 0 \\ 0 & \widehat{H}_m - \sigma I \end{bmatrix} y_m \|_2. \tag{3.21}$$

The least squares problem indicated by (3.21) splits thus into a $k \times k$ linear system

$$(\Lambda_k - \sigma I) y_m^1 = L_k^H r_0$$

and the smaller least squares problem

$$\|\beta e_1 - (\widehat{H}_m - \sigma I) y_m^2\|_2 = \min_{y \in \mathbb{C}^m} \|\beta e_1 - (\widehat{H}_m - \sigma I) y_m^2\|_2.$$

Different to GMRES-LR, y_m^2 is now the QMR approximation with starting vector $(I - P)r_0$ instead of r_0, since V_{m+1} is not orthogonal.

Algorithm 3.27. QMR-LR(k,m)

{**Input** $A = M - \sigma_i I$, $\{\sigma_1, \ldots, \sigma_s\}$, b, $L = L_k$, $R = R_k$, $\Lambda = \Lambda_k$}

$b^S = RL^H b$
$x^{(i)} = R(\Lambda - \sigma_i I)^{-1} L^H b$, $i = 1, \ldots, s$
$r = b - b^S$
$\beta = \|r\|_2$
while not all systems converged **do**
  $\tilde{v}_1 = (I - RL^H)b$, $\tilde{w}_1 = (I - LR^H)b$
  for $i = 1, \ldots, s$ **do**
    compute $V_m y_m^{2\,(i)}$ by QMR-Sh
    $x^{(i)} = x^{(i)} + V_m y_m^{2\,(i)}$
    $r^{(i)} = r - V_{m+1}(\widehat{H}_m - \sigma_i I) y_m^{2\,(i)}$
  end for
end while

3.5 Eliminating converged systems

With the (deflated) multishift methods presented in Sections 3.3 and 3.4 we simultaneously solve s systems

$$A - \sigma_i I = b, \quad i = 1, \ldots, s.$$

Although we use the same Krylov subspace for all s systems and therefore have no additional cost to build the subspace for additional systems, the cost still increases with the number s of systems to solve.
While for BiCG and QMR additional systems cause only little and above all constant additional cost, for FOM and GMRES the cost increases not only with s but with the iteration as well. The reason is that for each system a small but with the iteration number increasing system or least squares problem has to be solved. In the unsymmetric case, i.e., BiCG or QMR, a simple update of the iterates exists such that the (small) linear system or least squares problem, respectively, can be updated with little additional cost. In the symmetric case, i.e., FOM or GMRES, the inverse or least squares solution has to be computed seperately for each system as the matrix is non-hermitian and there is no simple update for the iterates. FOM and GMRES will therefore profit from any reduction of the number of systems to be solved.

Some of the systems will converge faster than others. Systems that have already reached a desired accuracy can be eliminated and no further computations have to be done for those systems.

To decide whether a system can be eliminated or not we use the residual norm. When combined with a rational approximation of the sign function, for example, the systems with small residual norm contribute only little to the error since

$$Q \sum_{i=1}^{s} \omega_i (Q^2 - \sigma_i I)^{-1} b - Q \sum_{i=1}^{s} \omega_i x_m^{(i)} = Q \sum_{i=1}^{s} \omega_i (Q^2 - \sigma_i I)^{-1} r_m^{(i)}.$$

For all methods we get $\|r_m^{(i)}\|_2$, or at least an upper bound, with basically no extra effort. No matter whether Schur- or LR-deflation is used, the residual norms are computed as listed in Table 3.1.

To get an idea of the benefits of elimination, Figure 3.1 shows how the number of systems to solve decreases in the course of the iteration. We used the matrix MAT3[1] for this example and ran FOM with LR-deflation. The shifts σ_i were chosen to approximate $\operatorname{sign}(Q)$ to an accuracy of 10^{-10}. The

[1] See Chapter 4.1 for the definition of MAT3.

method	residual norm $\|r_m^{(i)}\|_2$
FOM	$\|h_{m+1,m}(y_m^{2\ (i)})_m\|$
GMRES	$\|r_0^{(i)} - V_{m+1}(\widehat{H}_m - \sigma_1)y_m^{2\ (1)}\|_2$, $i = 1$ $\|\rho_m^{(i)}\| \cdot \|r_m^{(1)}\|_2$, $i \neq 1$
BiCG	$\|h_{m+1,m}(y_m^{2\ (i)})_m\|$
QMR	$(\leq)\|r_0\|_2\sqrt{m+1}\|sn_1 \cdot \cdots \cdot sn_m\|$

Table 3.1: Computation of residual norms

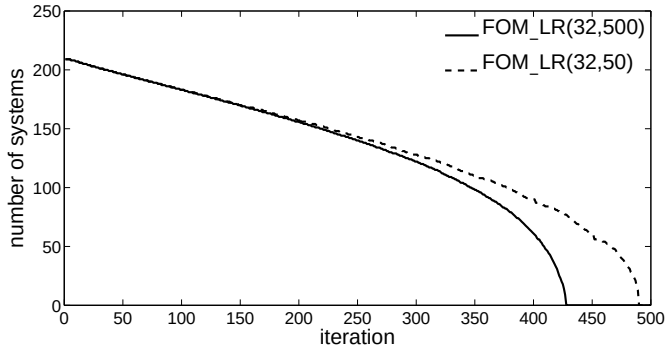


Figure 3.1: Decrease of the number of systems to solve

effect is the same no matter whether FOM or GMRES is used with Schur- or LR-deflation.

Of course the effect on the actual computational cost depends on the Krylov subspace size, i.e., the size of the systems to solve. But even when the size is bounded by using restarts, the cost is reduced significantly as Figure 3.2 shows, for the same sample matrix and shifts as chosen for Figure 3.1. For these plots we estimate the cost by taking into account only the linear system of size i in step i. As it is of Hessenberg form, the cost for m iteration steps sums up to

$$\text{cost}(m) = \sum_{i=1}^{m} \text{systems}(i) \cdot i^2,$$

where systems(i) denotes the number of systems to solve in step i.

3.6 Deflation of the rational approximation

To approximate the matrix sign function sign(Q) of a non-hermitian matrix Q, we have to use a rational approximation with significantly more poles

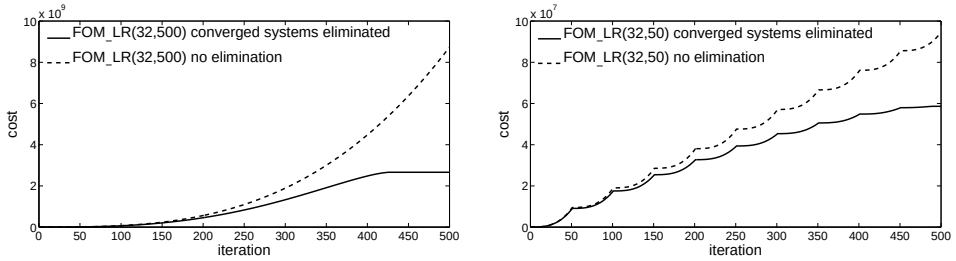


Figure 3.2: Effect on the cost without restart (left) and with restart (right)

than the Zolotarev approximation. Although only one Krylov subspace is built for all shifted systems, solving more systems still increases the computational cost significantly for FOM and GMRES, as discussed in the previous section. The aim in this section is therefore to reduce the number of poles by using the deflation applied to the multishift solver. Of course this would apply to the Zolotarev rational approximation as well.

The number of poles gets higher the closer some eigenvalues of Q lie to the imaginary axis. As some of those small eigenvalues are deflated by Schur- or LR-deflation, the idea is to split the rational approximation and use less poles for those parts that do not involve the deflated eigenvalues. It turns out that only LR-deflation is suitable for this pole reduction.

To approximate

$$\operatorname{sign}(Q)b \approx Q \sum_{i=1}^{s} \omega_i (Q^2 - \sigma_i I)^{-1} b,$$

we choose $r(t) = \sum_{i=1}^{s} \omega_i \frac{t}{t^2 - \sigma_i}$ such that

$$|\operatorname{sign}(t) - r(t)| \leq \epsilon/2 \qquad \forall t \in \operatorname{spec}(Q).$$

When approximating $(Q^2 - \sigma_i I)^{-1} b$ in a Krylov subspace for example by (ordinary) FOM as $(Q^2 - \sigma_i I)^{-1} b \approx V_m (H_m - \sigma_i I)^{-1} V_m^H b$, we compute

$$\operatorname{sign}(Q)b \approx Q V_m \sum_{i=1}^{s} \omega_i (H_m - \sigma_i I)^{-1} V_m^H b,$$

so r could as well be chosen with respect to $\operatorname{spec}(H_m)$.

Usually that does not make a big difference and does not help to reduce the number of poles, but in the augmented context the spectrum of H_m does not include the k smallest eigenvalues of Q due to the projection. Therefore, the rational approximation with respect to H_m will be as accurate as before with less poles. Of course, when reducing the number of poles, the shifts σ_i

size k of the augmented subspace	$\lambda_{\min}$	number s of poles
0	$4.2313 \cdot 10^{-3}$	168
8	0.0570	46
16	0.0950	36
32	0.1887	25
64	0.3195	19

Table 3.2: Number of poles needed for the matrix MAT3

and weights ω_i will change as well.

So, splitting in the LR case

$$\operatorname{sign}(Q)b \approx QR_k \sum_{i=1}^{s} \omega_i y_m^{1\,(i)} + QV_m \sum_{i=1}^{s} \omega_i y_m^{2\,(i)},$$

we can use an approximation with less poles for the second sum. The crucial property of LR-deflation in this context is that the Ω_R-part and Krylov part of the approximation are independent.

In the Schur case, the Krylov part influences the Ω_S-part. Here as well, we can split

$$\operatorname{sign}(Q)b \approx QS_k \sum_{i=1}^{s} \omega_i y_m^{1\,(i)} + QV_m \sum_{i=1}^{s} \omega_i y_m^{2\,(i)}, \tag{3.22}$$

and we could use an approximation with less poles for the second sum. But (3.22) actually reads

$$\operatorname{sign}(Q)b \approx QS_k \sum_{i=1}^{s} \omega_i (T_k - \sigma_i I)^{-1} S_k^H (b - (M - \sigma_i I) V_m y_m^{2\,(i)}) + QV_m \sum_{i=1}^{s} \omega_i y_m^{2\,(i)},$$

so that if we reduced the number of poles in the second sum we had to compute $y_m^{2\,(i)}$ for the big number of poles again for the first sum.

Therefore, reducing the number poles is possible only for LR-deflation. Instead of eigenvalue information of H_m we use the available eigenvalue information of A: Since the k smallest eigenvalues are deflated, we take $\lambda_{\min}$ the $(k+1)$-smallest eigenvalue for the calculation of the number of poles, the shifts, and the weights. Tables 3.2 and 3.3 show how many poles are actually needed to achieve an accuracy of 10^{-8} for the the matrix sign function of the matrices[2] MAT3 and MAT4.

[2]See Chapter 4.1 for the definition of MAT3 and MAT4.

size k of the augmented subspace	$\lambda_{\min}$	number s of poles
0	$3.0838 \cdot 10^{-4}$	582
16	0.0237	67
32	0.0431	50
64	0.0802	36
128	0.1483	27

Table 3.3: Number of poles needed for the matrix MAT4

3.7 Discussion

Looking at the numerical results, see Chapter 4.3, Schur- and LR-deflation show an equal performance with respect to iteration numbers. With respect to computation time, LR-deflation works significantly faster.

Of course, for LR-deflation both left and right eigenvectors have to be computed while for Schur-deflation only one set of Schur vectors is needed – we did not include this precalculation in the time measurements. In addition, the computation of Schur vectors is numerically stable in contrast to the computation of eigenvectors.

On the other hand, for Schur-deflation the number of poles cannot be reduced. That does not matter (much) when BiCG-Schur or QMR-Schur are used, but for FOM-Schur and GMRES-Schur it means a great increase of computational cost.
Regarding computation time, Schur-deflation has the second disadvantage that in each iteration step the Ω_S-part has to be projected out. The numerical results show clearly that this alone slows down the computation extremely.

	advantages	disadvantages
Schur-deflation	one set of (Schur-)vectors stable Schur vectors	projection in each step big number of poles
LR-deflation	no extra projection reduced number of poles	two sets of eigenvectors instable eigenvectors

Table 3.4: Advantages and disadvantages of Schur- and LR-deflation

Using the deflated methods for a rational approximation of the matrix sign function, the shifted matrices $M - \sigma I = Q^2 - \sigma I$ are positive real. The (deflated) FOM iterates therefore exist and even for (deflated) multishift

GMRES we have a guarantee for convergence. Of course, deflated BiCG and QMR still contain the possibility of breakdowns inherited from the underlying Lanczos process.

3.8 Other deflation techniques

Closest related to the approach presented in this work is the one given in [5, 6] where the matrix sign function is approximated using both Schur- and LR-deflation. Instead of using a rational approximation to the sign function they approximate

$$\text{sign}(Q)b \approx V_m \text{sign}(H_m) V_m^H b$$

which gives for Schur-deflation

$$\text{sign}(Q)b \approx [S_k V_m] \begin{bmatrix} \text{sign}(T_k) & Y \\ 0 & \text{sign}(H_m) \end{bmatrix} [S_k V_m]^H b,$$

where Y is the solution of the Sylvester equation

$$T_k Y - Y H_m = \text{sign}(T_k) S_k^H A V_m - S_k^H A V_m \text{sign}(H_m).$$

For LR-deflation they yield

$$\text{sign}(Q)b \approx R_k \text{sign}(\Lambda_k) L_k^H b + \beta V_m \text{sign}(H_m) e_1.$$

The sign function of the smaller submatrices is computed by Roberts' iterative method [38].

Morgan [30, 31, 32, 33] uses Ritz and harmonic Ritz vectors for the augmenting subspace and implicit restarts to include them. With Ω being spaned by (harmonic) Ritz vectors the augmented subspace $\Omega + K_m(A, r_0)$ is a Krylov subspace as well.

Chapter 4

Numerical results

4.1 Matrices

For the numerical experiments we use four matrices:

- MAT1, MAT2: zero chemical potential
  The shifted unitary methods of Chapter 2 are tested with Γ_5-hermitian Wilson-Dirac operators on a 4^4-lattice, i.e., a lattice with 4^4 lattice sites. The hopping matrices are taken from Matrix Market

  `http://math.nist.gov/MatrixMarket/data/misc/qcd/`.

  Note that for these matrices the γ-matrices were taken in Dirac representation, see Appendix A.

- MAT3, MAT4: non-zero chemical potential
  The eigenvalue deflation methods of Chapter 3 are tested with a non-Γ_5-hermitian Wilson-Dirac operator on a 4^4-lattice and a 6^4-lattice, respectively. Note that for these matrices the γ-matrices were taken in Weyl representation, see Appendix A.
  The matrices MAT3 and MAT4 were provided by Jacques Bloch from the Institute for Theoretical Physics at the University of Regensburg.

MAT1 uses the Matrix Market configuration `conf6.0-00l4x4-2000.mtx`
$\kappa_c = 0.15968$
$\kappa = 4/3\kappa_c = 0.2129$
$\min_{\lambda \in \mathrm{spec}(Q)}(|\lambda|) = 0.0767$
$\max_{\lambda \in \mathrm{spec}(Q)}(|\lambda|) = 2.4855$

The eigenvalues of the hopping term $M = I - \kappa D_W$ are plotted in Figure 4.1.

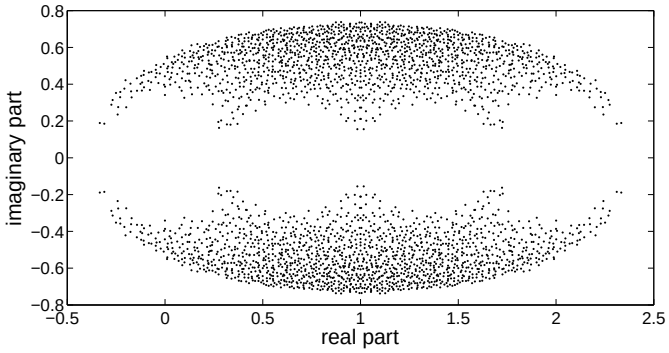


Figure 4.1: MAT1: eigenvalues of the matrix M

The eigenvalues of the Neuberger operator $D = \rho I + \Gamma_5 \text{sign}(Q)$ for $\rho = 1.01$ are plotted in Figure 4.2.

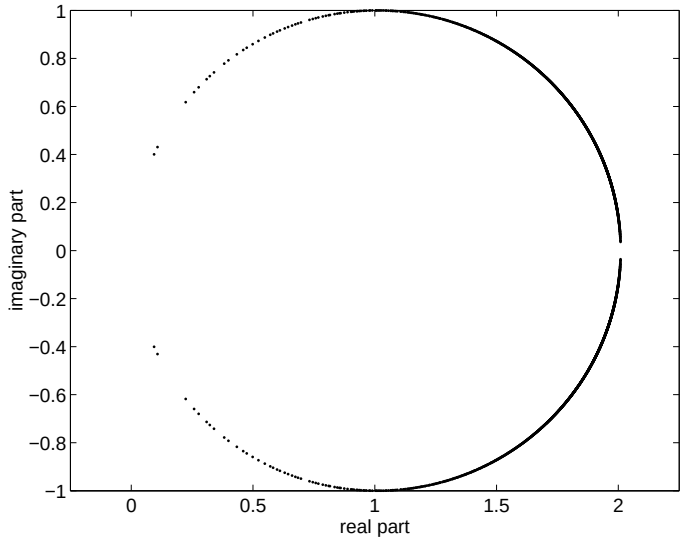


Figure 4.2: MAT1: eigenvalues of the matrix $\rho I + \Gamma_5 \text{sign}(Q)$, $\rho = 1.01$

MAT2 uses the Matrix Market configuration `conf5.0-00l4x4-2600.mtx`
$\kappa_c = 0.2107$
$\kappa = 4/3\kappa_c = 0.2809$
$\min_{\lambda \in \text{spec}(Q)}(|\lambda|) = 8.8923 \cdot 10^{-4}$
$\max_{\lambda \in \text{spec}(Q)}(|\lambda|) = 2.7868$

The eigenvalues of the hopping term $M = I - \kappa D_W$ are plotted in Figure 4.3.

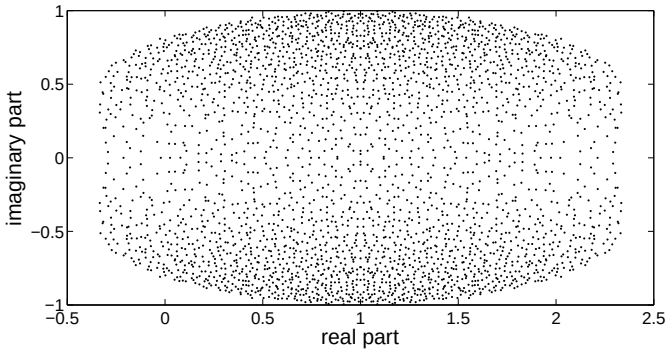


Figure 4.3: MAT2: eigenvalues of the matrix M

The eigenvalues of the Neuberger operator $D = \rho I + \Gamma_5 \text{sign}(Q)$ for $\rho = 1.01$ are plotted in Figure 4.2.

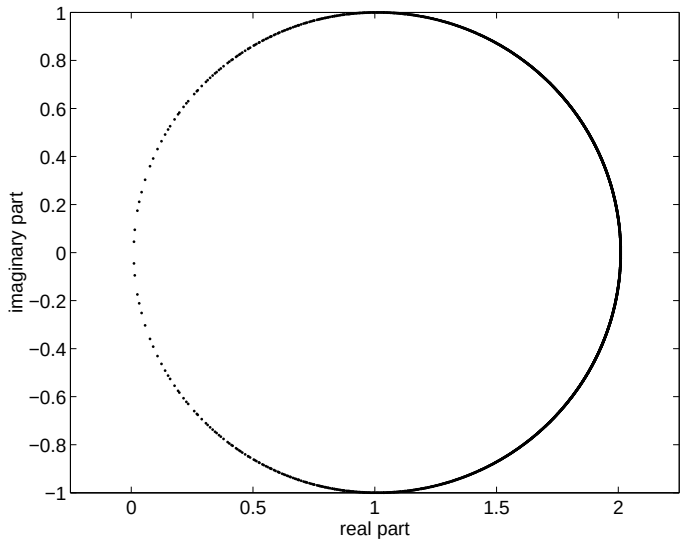


Figure 4.4: MAT1: eigenvalues of the matrix $\rho I + \Gamma_5 \text{sign}(Q)$, $\rho = 1.01$

MAT3

lattice size: 4^4
$\min_{\lambda \in \text{spec}(Q)}(|\lambda|) = 4.2313 \cdot 10^{-3}$
$\max_{\lambda \in \text{spec}(Q)}(|\lambda|) = 5.2161$
The eigenvalues of $Q = \Gamma_5 M$ are plotted in Figure 4.5.

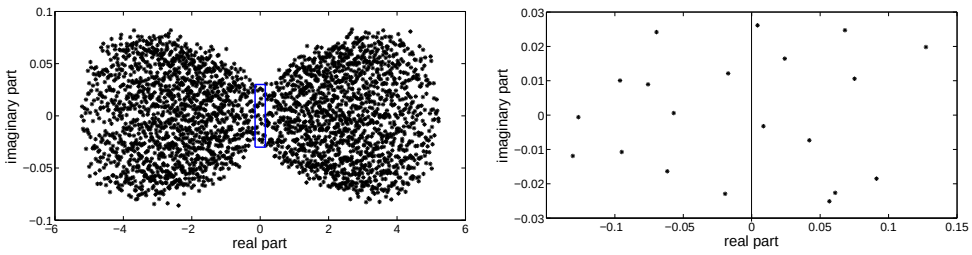


Figure 4.5: MAT3: eigenvalues of the matrix Q

MAT4

lattice size: 6^4
$\min_{\lambda \in \mathrm{spec}(Q)}(|\lambda|) = 3.0838 \cdot 10^{-4}$
$\max_{\lambda \in \mathrm{spec}(Q)}(|\lambda|) = 4.5599$
The eigenvalues of $Q = \Gamma_5 M$ are plotted in Figure 4.6.

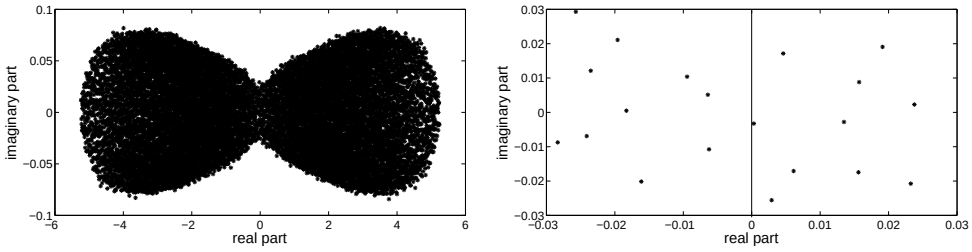


Figure 4.6: MAT4: eigenvalues of the matrix Q

4.2 Shifted unitary methods

For the shifted unitary methods we use the matrices MAT1 and MAT2, the right hand side is a complex random vector of unit length. For the inner iteration we require an accuracy of $\epsilon = 10^{-8}$. The matrix vector product is approximated by PFE/CG with the Zolotarev best approximation with 10 poles for MAT1 and 19 poles for MAT2, respectively. Note that the plots show matrix vector products instead of iteration numbers.

Figures 4.7 and 4.8 show the true and the computed residual. The norm of the true residual (solid) stagnates around the required accuracy while the computed residual (dashed) decreases further. To obtain the true residual, we computed the exact sign function of the hermitian matrix Q via a full singular value decomposition.

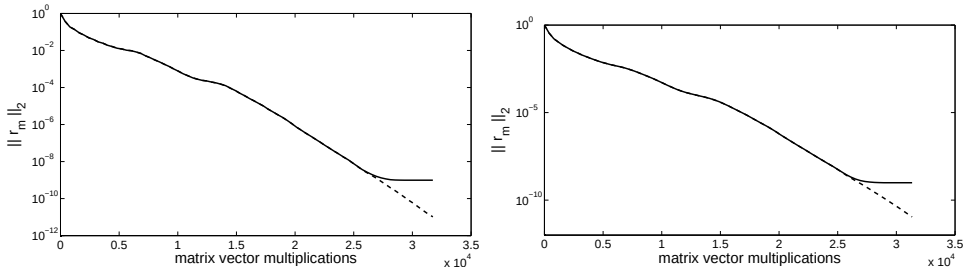


Figure 4.7: SUOM (left) and SHUMR (right) for MAT1

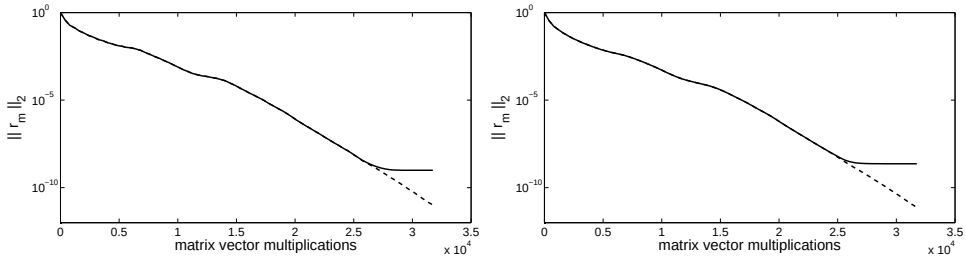


Figure 4.8: SUFOM (left) and SUMR (right) for MAT1

For relaxation we use the tolerances η_j as shown in Table 4.1. Note that these are the relaxation strategies proposed for FOM and GMRES. Since in exact arithmetic the shifted unitary methods are equivalent to FOM and GMRES, this is a reasonable choice. Nevertheless, better stategies might be possible taking into account the underlying short recurrence Arnoldi versions.

method	tolerance η_j
SUOM	$\epsilon \cdot \sqrt{\sum_{i=0}^{j} \|r_i\|_2^{-2}}$
SHUMR	$\epsilon / \|r_j\|_2$
SUFOM	$\epsilon \cdot \sqrt{\sum_{i=0}^{j} \|r_i\|_2^{-2}}$
SUMR	$\epsilon / \|r_j\|_2$

Table 4.1: Precision of the matrix vector product

For demonstration reasons we plot the true residuals computed as described above. For problems of a realistic size, the true residual is not known in general. In this case the computed residuals can be used for a stopping criteria but of course one should not iterate further once the accuracy of the inner iteration is reached.

The following plots (Figures 4.9 and 4.10) show the convergence of the relaxed and unrelaxed versions of the four shifted unitary methods using the matrix MAT1 and $\rho = 1.01$. We compare the unrelaxed version (solid) with the relaxed verion (dashed).

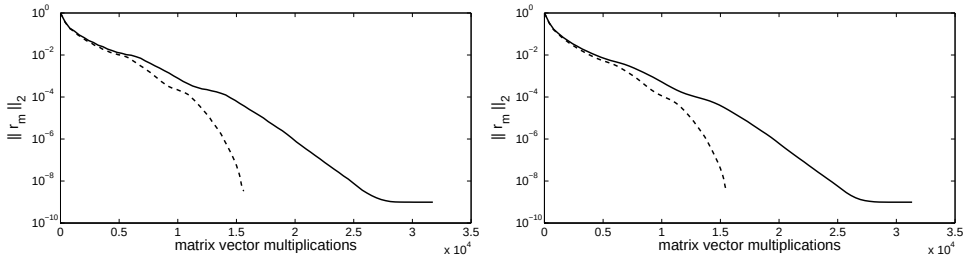


Figure 4.9: SUOM (left) and SHUMR (right) for MAT1

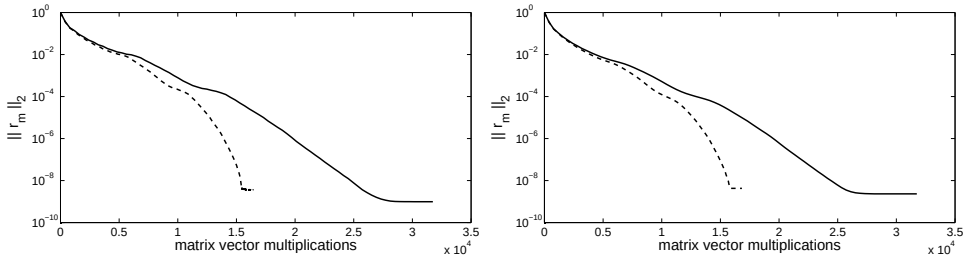


Figure 4.10: SUFOM (left) and SUMR (right) for MAT1

For MAT1 and $\rho = 1.01$ relaxation saves about 40% matrix vector multiplications while the loss in accuracy is negligible. No difference can be observed between the four methods. Although Figure 4.9 does not show it, SUOM and SHUMR stagnate at about the inner accuracy same as SUFOM and SUMR do.

The following plots (Figures 4.11 and 4.12) show the convergence of the four shifted unitary methods using the matrix MAT2 and $\rho = 1.01$. Again, we compare the unrelaxed version (solid) with the relaxed verion (dashed).

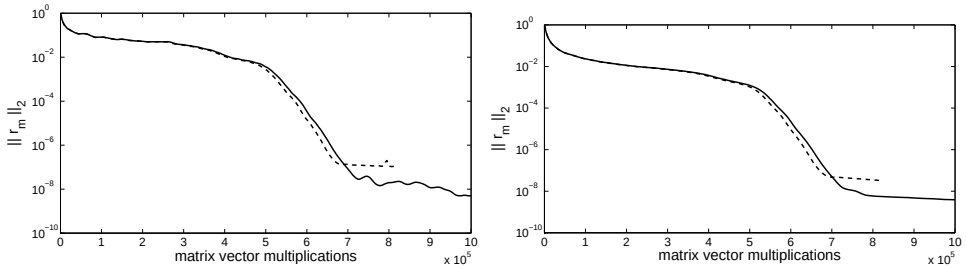


Figure 4.11: SUOM (left) and SHUMR (right) for MAT2

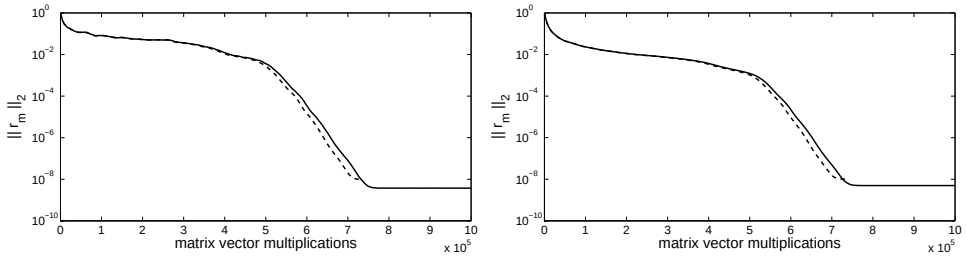


Figure 4.12: SUFOM (left) and SUMR (right) for MAT2

For MAT2 and $\rho = 1.01$ relaxation does not save much. The final accuracy reached with SUFOM and SUMR is slightly better than for SUOM and SHUMR.

The convergence of the shifted unitary methods does not only rely on the matrix Q but also on the shift parameter ρ. Figure 4.13 shows the convergence of SUMR for MAT1 and MAT2 with $\rho = 1.1$. The difference to $\rho = 1.01$ is little, but the convergence is faster – not surprisingly since the eigenvalues of $\Gamma_5 \text{sign}(Q)$ are shifted further away from the imaginary axis.

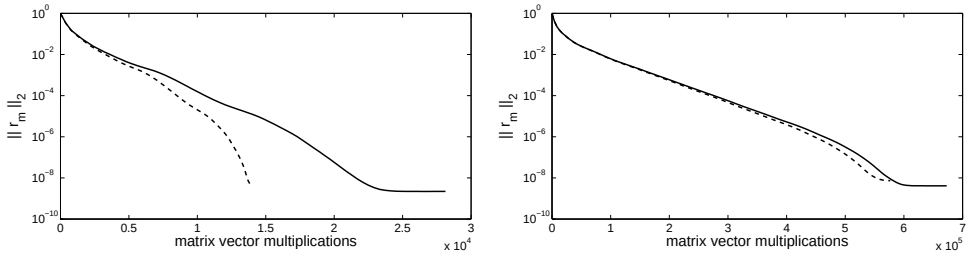


Figure 4.13: SUMR with $\rho = 1.1$ for MAT1 (left) and MAT2 (right)

4.3 Deflation

As sample matrices for the deflation methods we use the non-hermitian matrices MAT3 and MAT4. We compute the multishift approximations $\tilde{y}$

to the rational approximation of $y = \text{sign}(Q)b$ with $b = (1, \ldots, 1)^T$ to an accuracy of 10^{-8}. The error plotted in the convergence plots of Figures 4.14 - 4.22 is the relative error

$$\frac{\|\tilde{y} - y\|_2}{\|y\|_2}.$$

Since the matrices are non-hermitian we cannot use the Zolotarev best approximation. We use the Neuberger rational approximation instead, which works for hermitian matrices as well since it summarises a certain number of steps of the Newton iteration. To achieve an accuracy of η we use

$$s = \left\lceil \frac{1}{2} \cdot \log\left(\frac{\eta}{2-\eta}\right) / \log\left(\frac{\sqrt{\lambda_{\max}/\lambda_{\min}} - 1}{\sqrt{\lambda_{\max}/\lambda_{\min}} + 1}\right) \right\rceil$$

poles, see [45]. The number of poles needed to achieve the required accuracy is shown in Tables 3.2 and 3.3. The number of poles can be quite large. We therefore use pole reduction for all plots.

Figures 4.14 to 4.19 show the convergence plots for the Schur and LR deflated methods. In Figures 4.14 to 4.17 we compare the non-restarted and restarted versions of FOM and GMRES. The results for Schur and LR deflated BiCG and QMR are shown in Figures 4.18 and 4.19.

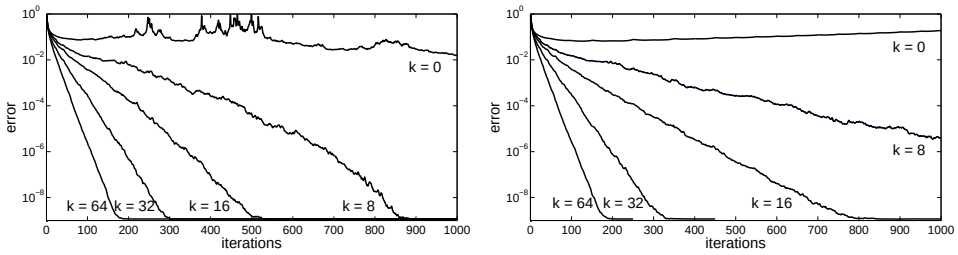


Figure 4.14: Schur deflated FOM for MAT3 – without restart (left) and with restart (right)

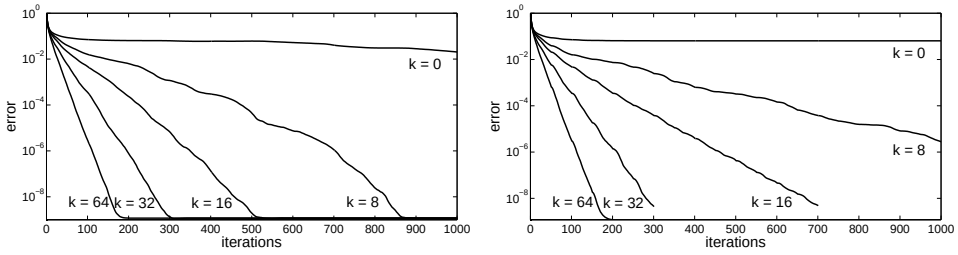


Figure 4.15: Schur deflated GMRES for MAT3 – without restart (left) and with restart (right)

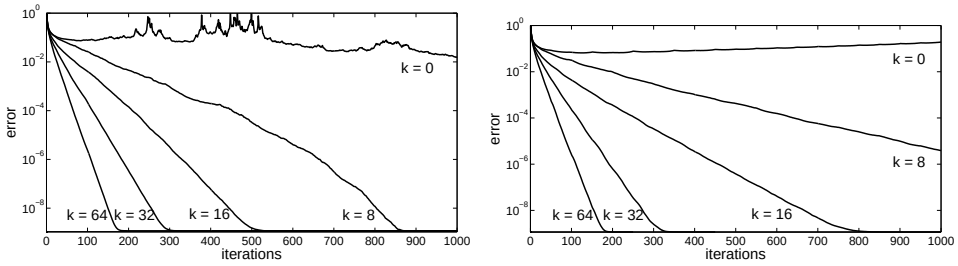


Figure 4.16: LR deflated FOM for MAT3 – without restart (left) and with restart (right)

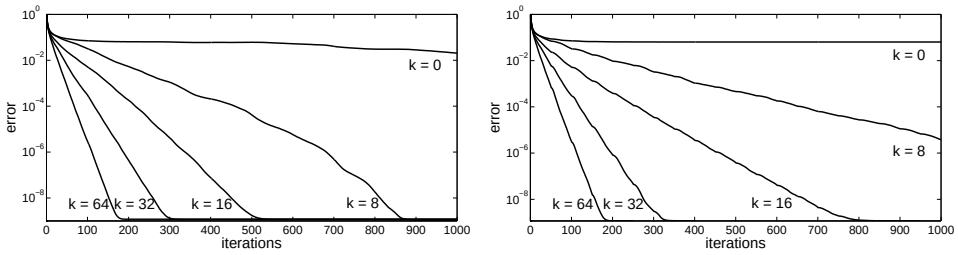


Figure 4.17: LR deflated GMRES for MAT3 – without restart (left) and with restart (right)

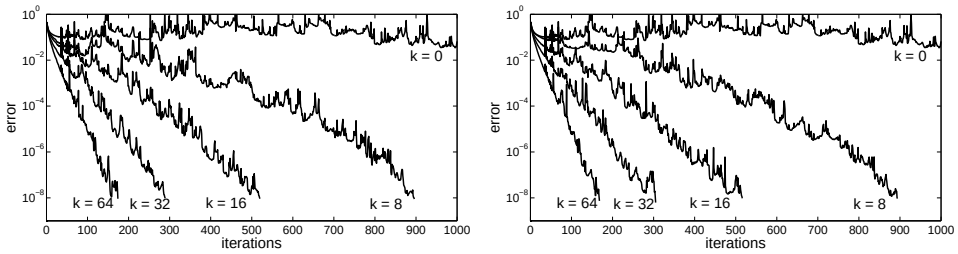


Figure 4.18: BICG for MAT3 – Schur deflated (left) and LR deflated (right)

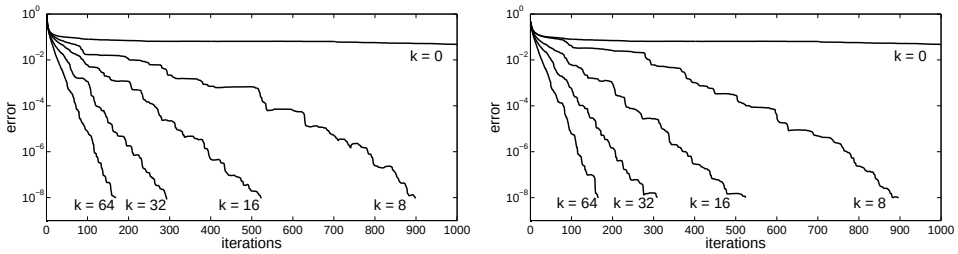


Figure 4.19: QMR for MAT3 – Schur deflated (left) and LR deflated (right)

Schur- and LR-deflation produce the same convergence behaviour. No significant difference in iteration numbers can be detected in the Figures 4.14 -

4.19. Still, there are diffenences between the Schur and LR deflated methods that can not be conveyed in those plots:

- For LR-deflation we have to compute left and right eigenvectors while for Schur-deflation we only need Schur vectors. This is an advantage of Schur-deflation. We do not investigate this difference further.
- In the Schur deflated methods, every new Arnoldi vector has to be orthogonalized against the augmenting Schur vectors, i.e. its Ω_S-part has to be projected out.
- LR-deflation allows a reduction of the number of poles in the rational approximation. As can be seen in Tables 3.2 and 3.3, the required number of poles is thus significantly higher using Schur-deflation.

Therefore even if they need the same number of iterations to achieve a given accuracy, Schur deflated methods and LR deflated methods differ in the computing time.

To demonstrate the effect of pole reduction and orthogonalization against the augmenting subspace on the computing time, we compare Schur deflated and LR deflated GMRES for the matrix MAT3. Table 4.2 shows the times in seconds to achieve an accuracy of 10^{-8} without restarts, Table 4.3 shows the resulting times for restarts after 50 iterations. The time messurements were done on an Intel Pentium 4, 2.8 GHz, using the `tic` and `toc` functions of MATLAB 7.5(2007b).

Schur deflated GMRES has to run with 168 poles no matter how many eigenvalues are deflated. LR deflated GMRES needs only 64/36/25/19 poles depending on the number of deflated eigenvalues (8/16/32/64). The last column of Tables 4.2 and 4.3 show the results for this reduced number of poles while for the second last column we ran GMRES-LR without pole reduction.

deflated eigenvalues	GMRES-Schur	GMRES-LR	GMRES-LR (reduced poles)
8	10881.3	3282.4	1013.7
16	4701.2	977.9	226.8
32	1798.6	258.4	52.0
64	855.6	71.2	15.7

Table 4.2: Time (in seconds) needed for GMRES (without restarts) with Schur- and LR-deflation

deflated eigenvalues	GMRES-Schur	GMRES-LR	GMRES-LR (reduced poles)
8	1320.7	220.4	100.3
16	880.6	89.9	36.4
32	455.1	38.1	13.4
64	396.2	22.2	6.9

Table 4.3: Time (in seconds) needed for GMRES (with restart after 50 iterations) with Schur- and LR-deflation

Though there are no differences in iteration numbers, the computing time differs significantly. Even without pole reduction, LR deflated GMRES is by a factor of 3-12 faster than Schur deflated GMRES, and even 6-18 times faster in the restarted version. With pole reduction the factor is 10-54, 13-57 in the restarted version. The more eigenvalues are deflated the bigger the difference between Schur-deflation and LR-deflation.

Table 4.4 shows that the same holds for QMR: QMR-LR is by a factor of 19-25 faster than QMR-Schur, by a factor of up to 124 when the number of poles is reduced.

deflated eigenvalues	QMR-Schur	QMR-LR	QMR-LR (reduced poles)
8	1953.7	102.3	15.7
16	1302.3	60.5	14.1
32	703.1	36.4	7.9
64	515.7	20.7	4.2

Table 4.4: Time (in seconds) needed for QMR with Schur- and LR-deflation

Finally, we compare the four LR deflated methods, see Table 4.5. Again, we use the matrix MAT3 and an accuracy of 10^{-8}. FOM-LR and GMRES-LR are restarted after 50 iterations, all methods are run with pole reduction. Even tough for the same number of iterations BiCG and QMR need twice as many matrix vector multiplications than FOM and GMRES, the short recurrence methods BiCG and QMR are significantly faster.

deflated eigenvalues	FOM-LR	GMRES-LR	BiCG-LR	QMR-LR
8	72.5	100.3	13.1	15.7
16	27.0	36.4	12.6	14.1
32	10.5	13.4	7.3	7.9
64	6.3	6.9	4.3	4.2

Table 4.5: Time (in seconds) needed for the LR deflated methods (FOM and GMRES with restart after 50 iterations)

For the 6^4-lattice MAT4 we only show the convergence behaviour of the LR deflated methods since the results for MAT3 show that Schur-deflation cannot compete with LR-deflation. In Figures 4.20 to 4.22 we compare the LR deflated methods for the matrix of the 6^4-lattice MAT4 with a required accuracy of 10^{-8}.

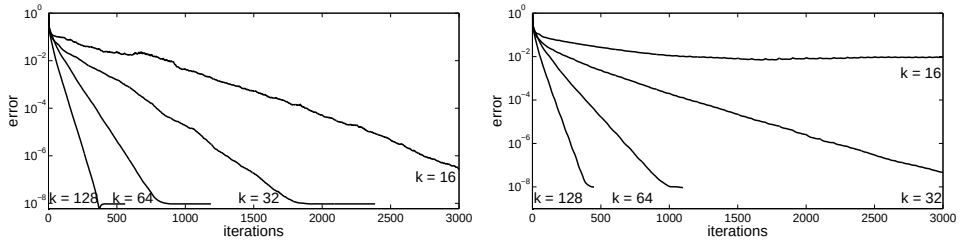


Figure 4.20: LR deflated FOM for MAT4 – without restart (left) and with restart (right)

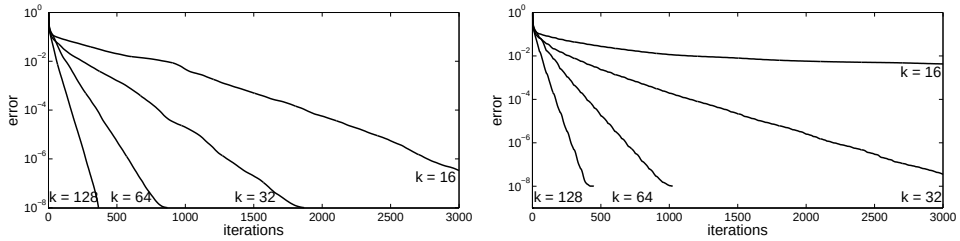


Figure 4.21: LR deflated GMRES for MAT4 – without restart (left) and with restart (right)

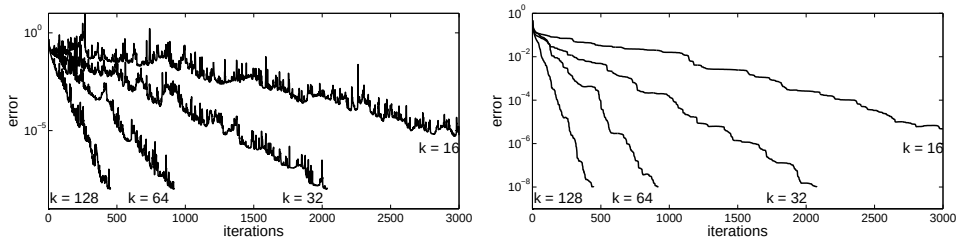


Figure 4.22: LR deflated BiCG (left) and QMR (right) for MAT4

Again, no difference in the convergence behaviour can be observed between the different methods. Using restarts obviously bears the risk of losing convergence as can be seen in Figures 4.20 and 4.21 for $k = 16$. Unfortunately we do not know beforehand how many eigenvalues we have to deflate to retain convergence for the restarted versions. As indicated by the time measurements of MAT3, without restarts FOM and GMRES are by far too slow to take them into consideration. BiCG and QMR on the other hand bear the risk of breakdowns. This is compensated by the fact that BiCG and QMR do not need restarts and are significantly faster.

The LR deflated methods, especially with pole reduction, are significantly faster than the Schur deflated methods. Note that for the presented results we ignored the precalculation required to obtain the Schur- or eigenvectors. Computing left and right eigenvalues will take longer than computing only one set of Schur vectors. If the left and right eigenvalues are not available and Schur-deflation has to be used, for the same reasons as in LR-deflation the methods of choice are BiCG and QMR.

Appendix A

Gamma matrices

The γ-matrices are not unique, there are several representations. All definitions use the Pauli matrices σ_1, σ_2, σ_3:

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

In the Dirac or Dirac-Pauli representation the Euclidean γ-matrices are defined as

$$\gamma_k = \begin{pmatrix} 0 & -i\sigma_k \\ i\sigma_k & 0 \end{pmatrix} \quad \text{for } k = 1, 2, 3$$

and

$$\gamma_4 = \begin{pmatrix} -1 & & & \\ & -1 & & \\ & & +1 & \\ & & & +1 \end{pmatrix}.$$

A fifth matrix γ_5 is defined as

$$\gamma_5 = \gamma_1\gamma_2\gamma_3\gamma_4 = \begin{pmatrix} & & +1 & \\ & & & +1 \\ +1 & & & \\ & +1 & & \end{pmatrix}.$$

In the Weyl or chiral representation the matrix γ_4 is replaced by

$$\gamma_4 = \begin{pmatrix} & & +1 & \\ & & & +1 \\ +1 & & & \\ & +1 & & \end{pmatrix}$$

such that

$$\gamma_5 = \gamma_1\gamma_2\gamma_3\gamma_4 = \begin{pmatrix} +1 & & & \\ & +1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix}.$$

The matrix Γ_5 is defined as

$$\Gamma_5 = I_{n/12} \otimes (\gamma_5 \otimes I_3).$$

Note that the matrices MAT1 and MAT2 use the Dirac representation while MAT3 and MAT4 use the Weyl representation.

Bibliography

[1] G. Arnold, N. Cundy, J. van den Eshof, A. Frommer, S. Krieg, Th. Lippert, and K. Schäfer. Numerical Methods for the QCD Overlap Operator: II. Optimal Krylov Subspace Methods. In *QCD and Numerical Analysis III*, volume 47 of *Lecture Notes in Computational Science and Engineering*, pages 153–167. Springer Berlin Heidelberg, 2005.

[2] W. E. Arnoldi. The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem. *Quarterly of Applied Mathematics*, 9:17–29, 1951.

[3] Zh. Bai and J. W. Demmel. Design of a parallel nonsymmetric eigenroutine toolbox, Part I. In R. F. Sincovec, D. E. Keyes, M. R. Leuze, L. R. Petzold, and D. A. Reed, editors, *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing, Volume I*, pages 391–398. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1993.

[4] T. Barth and Th. Manteuffel. Multiple recursion conjugate gradient algorithms part I: sufficient conditions. *SIAM J. Matrix Anal. Appl.*, 21:768–796, 2000.

[5] J. Bloch, A. Frommer, B. Lang, and T. Wettig. An iterative method to compute the overlap Dirac operator at nonzero chemical potential. *Proceedings of Science*, 2007. arXiv:hep-lat/0710.0341v1.

[6] 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. *Computer Physics Communications*, 177:933–943, 2007.

[7] A. Borici. The two-grid algorithm confronts a shifted unitary orthogonal method. *Nuclear Physics B Supplement*, 140:850–852, 2005.

[8] A. Borici and A. Allkoci. A fast minimal residual solver for overlap fermions. *arXiv:hep-lat/0602015v1*, 2006.

[9] A. Borici and A. Allkoci. Shifted unitary orthogonal methods for the overlap inversion. *arXiv:hep-lat/0601031v1*, 2006.

[10] N. Cundy, J. van den Eshof, A. Frommer, S. Krieg, Th. Lippert, and K. Schäfer. Numerical methods for the QCD overlap operator: III. Nested iterations. *Computer Physics Communications*, 165:221–242, 2005.

[11] V. Faber, J. Liesen, and P. Tichy. The Faber-Manteuffel theorem for linear operators. *SIAM J. on Numerical Analysis*, 46(3):1323–1337, 2008.

[12] V. Faber and Th. Manteuffel. Necessary and sufficient conditions for the existence of a conjugate gradient method. *SIAM J. Numer. Anal.*, 21:352–362, 1984.

[13] R. W. Freund. Solution of Shifted Linear Systems by Quasi-Minimal Residual Iterations. In L. Reichel, A. Ruttan, and R. S. Varga, editors, *Numerical Linear Algebra*, pages 101–121. W. de Gruyter, 1993.

[14] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. *SIAM Journal on Scientific Computing*, 14(1):137–158, 1993.

[15] R. W. Freund and N. M. Nachtigal. QMR: a Quasi-Minimal Residual Method for Non-Hermitian Linear Systems. *Numer. Math.*, 60:315–339, 1991.

[16] A. Frommer. BiCGStab(l) for families of shifted linear systems. *Computing*, 70(2):87–109, 2003.

[17] A. Frommer and U. Glässner. Restarted GMRES for shifted linear systems. *SIAM J. Sci. Comput.*, 19:15–26, 1998.

[18] A. Frommer and V. Simoncini. Matrix functions. To appear in "Model Order Reduction: Theory, Research Aspects and Applications", Mathematics in Industry, Schilders, Wil H. A. and van der Vorst, Henk A. eds, Springer, Heidelberg.

[19] P. H. Ginsparg and K. G. Wilson. A remnant of chiral symmetry on the lattice. *Phys. Rev. D25*, pages 2649–2657, 1982.

[20] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[21] W. Gragg. Positive definite Toeplitz matrices, the Arnoldi process for isometric operators, and Gaussian quadrature on the unit circle. *J. of Comput. and Appl. Mathematics*, 46:183–198, 1993.

[22] M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *J. Res. Nat. Bur. Stand.*, 49:409–436, 1952.

[23] N. J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, 2008.

[24] R. Horn and Ch. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[25] R. Horn and Ch. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[26] D. Ingerman, V. Druskin, and L. Knizhnerman. Optimal finite difference grids and rational approximations of the square root. I. Elliptic problems. *Comm. Pure Appl. Math.*, 53(8):1039–1066, 2000.

[27] C. Jagels and L. Reichel. A fast minimal residual algorithm for shifted unitary matrices. *Numerical Linear Algebra with Applications*, 1(6):555–570, 1994.

[28] Ch. Kenney and A. Laub. The Matrix Sign Function. *IEEE Transactions on Automatic Control*, 40(8):1330–1348, 1995.

[29] J. Liesen and Z. Strakos. On optimal short recurrences for generating orthogonal Krylov subspace bases. accepted for publication in SIAM Review.

[30] R. Morgan. A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Anal. Appl.*, 16:1154–1176, 1995.

[31] R. Morgan. Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations. *SIAM J. Matrix Anal. Appl.*, 21(4):1112–1135, 2000.

[32] R. Morgan. GMRES with deflated restarting. *SIAM J. Sci. Comp.*, 24, 2002.

[33] R. Morgan and W. Wilcox. Deflated iterative methods for linear equations with multiple right hand sides. *arXiv:0707.0505*, 2007.

[34] R. Narayanan and H. Neuberger. An alternative to domain wall fermions. *Phys. Rev.*, 2000. arXiv:hep-lat/0005004v2.

[35] H. Neuberger. A Practical Implementation of the Overlap Dirac Operator. *Phys. Rev. Lett.*, 81(19):4060 – 4062, 1998.

[36] H. Neuberger. Overlap Dirac operator. In A. Frommer, Th. Lippert, B. Medeke, and K. Schilling, editors, *Numerical challenges in Lattice Quantum Chromodynamics*. Springer Berlin, 2000.

[37] C. C. Paige and M. A. Saunders. Solution of Sparse Indefinite Systems of Linear Equations. *SIAM J. Num. Anal.*, 12:617–629, 1975.

[38] J. D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *Internat. J. Control*, 32:677–687, 1980.

[39] H. Rutishauser. Bestimmung der Eigenwerte orthogonaler Matrizen. *Numerische Mathematik*, 9:104–108, 1966.

[40] Y. Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Math. Comp.*, 37:105–126, 1981.

[41] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing, 1996.

[42] Y. Saad and M. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Statist. Comp.*, 7:856–869, 1986.

[43] V. Simoncini. Restarted full orthogonalization method for shifted linear systems. *BIT Numerical Mathematics*, 43:459–466, 2003.

[44] G. Sleijpen and J. van den Eshof. Inexact Krylov subspace methods for linear systems. *SIAM J. on Matrix Analysis and Applications*, 26:125–153, 2005.

[45] J. van den Eshof, A. Frommer, Th. Lippert, K. Schilling, and H. van der Vorst. Numerical methods for the QCD overlap operator: I. sign-function and error bounds. *Comput. Phys. Commun.*, 146:203–224, 2002.

[46] J. van den Eshof, G. Sleijpen, and M. van Gijzen. Relaxation strategies for nested Krylov methods. *SIAM J. of Computational and Applied Mathematics*, 177:347–365, 2005.

[47] E. I. Zolotarev. Application of elliptic functions to the question of functions deviating least and most from zero. *Zap. Imp. Akad. Nauk. St. Petersburg*, 30(5), 1877.