Deflation Based Krylov Subspace Methods for Sequences of Linear Systems

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

am Fachbereich Mathematik und Naturwissenschaften der
Bergischen Universität Wuppertal genehmigte

Dissertation

von

Dipl.-Math. Nemanja Božović

Gutachter: Prof. Dr. Andreas Frommer
Gutachter: Prof. Dr. Matthias Bolten

Prüfungskommission
Prof. Dr. Andreas Frommer
Prof. Dr. Matthias Bolten
Prof. Dr. Michael Günther
Assoc. Prof. Vladimir Kostić

Dissertation eingereicht am: 08.06.2017
Tag der Disputation: 21.07.2017

Die Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20170913-142254-9
[http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20170913-142254-9]

Acknowledgements

I would like to thank all people who helped me, in one way or another, to write down this thesis.

First, I would like to thank my advisor Prof. Andreas Frommer. Over the years he found time on numerous occasions to answer my questions, he provided me with many useful advice and we had countless fruitful discussions. In addition, I thank Prof. Matthias Bolten, who was in a role of my second advisor. Disregarding my (often) constant knocking on his door, he was always patient enough to help me out with any issues I had, as well as to suggest me different insights and ways to tackle problems. It was a pleasure working with them and in general in the Applied Computer Science group at Bergische Universität Wuppertal. Amongst all colleagues from the group, I would like to specially thank Sebastian Birk, Matthias Rottmann and Karsten Kahl who either helped me through advice and idea exchanges or by providing me codes and matrices for different applications. Prof. Francesco Knechtli and Björn Leder from the physics department also deserve to be mentioned, since without joint seminars and their lectures and talks, it would be a much harder task to understand basics of lattice QCD.

I had the opportunity to attend many conferences, summer schools and workshops, where I met many interesting people, and I am grateful for that. From all these people, I would like to thank Prof. Mike Peardon from Trinity College Dublin who helped me work out gauge fixing from the mathematical point of view, as well as Graham Moir (at that moment also from TCD) who was always open for any lattice QCD related question I had.

I would like to express my deepest gratitude to my former mentor and advisor, Prof. Ljiljana Cvetković, who was probably the most influential person throughout my academic career. Her inspiring and "out of the ordinary" lectures rose my interest in Applied Numerical Linear Algebra momentarily, and made my choice for the field of a future research quite straightforward. I am also thankful to Vladimir Kostić, who was responsible for the exercise course at that moment, for his "can do" attitude and help and support whenever we met.

I would also like to thank Sebastian Birk and his family for beeing there

for us as friends, as well as helping us with many matters due to the lack of knowledge of the German language. Moreover, I would specially like to thank Dmitry Shcherbakov for beeing a true friend over the years, and hopefully beyond Ph.D.

Next, I would like to thank my parents, Vesna and Milan, for their mental support, not only over the past years, but rather throughout my whole academic career.

Last but not least, I dedicate this thesis to my lovely wife Milana and my two sons, Dušan and Filip, who endured with me through thick and thin. Their unwavering support at moments of weakness helped me overcome all problems and finish my work.

Contents

1 Introduction

Many applications in computational science and engineering require the solution of a sequence of slowly changing linear systems

$$A^{(i)}x^{(i)} = b^{(i)}, \quad A^{(i)} \in \mathbb{C}^{n\times n}, \quad b^{(i)} \in \mathbb{C}^n, \quad i = 1, \dots, m. \tag{1.1}$$

In this context, slowly changing means that the matrices and right-hand sides change slightly from one system to the next, as it will be demonstrated for some particular applications in Chapter 5. In our work, we have focused on solving (1.1) by exploiting the idea of *recycling*, which basically means keeping a carefully chosen subspace between systems, with the goal of reducing costs of subsequent systems. We should mention that this is not a new concept, it has been introduced and utilized by many authors in the last two decades.

Advances in technology led to a substantial growth in the size of the linear systems, e.g., lattice QCD calculations tend to have hundreds of millions of unknowns. In this case direct methods fail due to the excessive storage or computational time requirements and iterative methods become the only feasible option. The most popular choice are Krylov subspace methods. Not taking into account the "closeness" of the matrices, there is a huge variety of methods to choose from for solving each system separately in (1.1). Some well-known Krylov solvers are CG for the Hermitian matrices and GMRES, GCR etc. in the non-Hermitian case. However, even these methods encounter problems due to their excessive storage requirements. A straightforward remedy is to use the restarted or truncated version of these methods. Nonetheless, this is not always the best option, since restarted and truncated methods often experience slow convergence and even stagnation.

Enhancing the robustness of restarted and truncated methods was, and still is, an everlasting task. Improving on restarted GMRES resulted in many advanced methods that are based on two concepts: *deflation* and *augmentation*, which play an important role throughout this thesis. The idea behind deflation is to remove the smallest eigenvalues from the spectrum, which should lead to a better convergence of the method, whereas augmentation simply means enlarging the current subspace with a carefully chosen subspace from the previous cycle. As it will be shown in Section 3.5 these two concepts come along together naturally. The way in which they are employed gives rise to different

methods. The GMRES-E and GMRES-DR methods are algebraically equivalent, since they use the same subspace, i.e. the eigenspace corresponding to the smallest eigenvalues for the augmentation, but are rather based on different strategies. Another example is the "loose" GMRES method where one aims to keep a few error approximations which in some sense represent the previously built subspace. On the other hand, truncation might lead to poor convergence, since we keep only a certain number of vectors that we orthogonalize against in the Arnoldi process. Improving on truncated methods leads to more efficient methods like GCRO, which is by construction a truncated method. However, an inner orthogonalization scheme which is of outmost importance throughout this thesis seems to be quite effective. Furthermore, de Sturler proposed the concept of *optimal truncation*, in which we choose the best possible subspace to keep, thus developing the GCROT method.

Even though some of these methods can be easily modified for solving (1.1), they all exhibit certain flaws. Nonetheless, combining some of these methods, i.e. combining ideas, concepts and frameworks, leads to elegant and efficient methods, that recycle a judiciously selected subspace between systems (cycles) and use it to reduce costs of subsequent systems in the ensemble. The GCRO-DR method uses the framework of the GCRO method, i.e. it has an inner/outer scheme. However, it performs deflation in the same way as in GMRES-DR. To clarify further, in the outer method, i.e. GCR, one computes the approximate eigenspace, which is later on used by the inner method, i.e. GMRES for building the augmented space, and deflating the smallest eigenvalues. An important detail is that augmentation and deflation correspond to performing the Arnoldi process with the operator $(I-CC^H)A$ within GMRES, where C is the approximate eigenspace and $I-CC^H$ is the orthogonal projector. In recent years, one more method named "loose" GCRO-DR (LGCRO-DR) for solving (1.1) was proposed, which was developed by straightforwardly incorporating the idea of recycling a few error approximations into GCRO-DR.

Our research is mainly based on the work by M. Gutknecht. In [Gut12] he compares two techniques, namely deflated GMRES and truly deflated GMRES, without giving any details about the subspaces that we aim to recycle. Choosing the right harmonic Ritz vectors as the recycle subspace, deflated GMRES basically corresponds to the GCRO-DR method. On the other hand, "true" deflation means deflating both, left and right, eigenspaces, and hence represents a theoretically better approach. In order to do that, one has to use the operator $(I - C\tilde{C}^H)A$ within Arnoldi instead of $(I - CC^H)A$, where the range of $\tilde{C}$ is spanned by left harmonic Ritz vectors, which have to be computed additionally. Therefore, on top of the Gutknecht's theory, we propose a cheap way of computing $\tilde{C}$, thus leading to a new method which represents

the truly deflated technique. We named the method Left-Right Deflated GMRES (LRDGMRES). The idea behind "loose" methods fits naturally into our method, and gives rise to the "loose" LRDGMRES method.

Deflated and augmented methods usually work better when used together with preconditioning. Often, the reliability of iterative methods for various applications depends more on the choice of the preconditioner, rather than on the acceleration technique employed. For this reason, preconditioning found its way into this thesis. Since the choice of the preconditioner depends on the application considered, we will discuss further this topic for the lattice QCD application, which was, due to our involvement in the projects *Marie Curie Initial Training Network STRONGnet* and *SFB/Transregio 55 Hadronenphysik mit Gitter-QCD*, of the most interest for us.

1.1 Outline

This thesis is organised as follows.

In Chapter 2 we gathered definitions and results that are scattered throughout the literature. The first three sections contain basic properties of projectors, projection methods and Krylov subspaces, so even if the reader is familiar with these terms, we suggest skimming through these sections for the sake of notation. However, we would like to point to Section 2.4, where we introduce the concept of *left* Ritz and harmonic Ritz vectors, which, to our knowledge, is not known from the literature, and prove some canonical properties they satisfy.

In Chapter 3 we introduce two mathematically equivalent Krylov subspace methods, GMRES and GCR, for the solution of the nonsymmetric linear systems and discuss further modifications of these methods. In practice, both methods are used in restarted or truncated form. In Section 3.5 we explain the concepts of augmentation and deflation and draw a connection between them. Furthermore, we present different techniques for using them, and we describe the resulting methods GMRES-E (Section 3.5.2), GMRES-DR (Section 3.5.3) and LGMRES (Section 3.5.4). Moreover, we describe the inner/outer scheme utilized by GMRESR (Section 3.3) and GCRO (Section 3.4), and later on by more advanced methods in Chapter 4. Since augmented and deflated methods often unfold their full potential when used with preconditioning, we briefly describe the right preconditioning in Section 3.6, which we further revisit in Chapter 5 for a particular application in lattice QCD.

Chapter 4 contains the main contribution of this thesis. In Section 4.1 we build on the GCRO framework by introducing the concept of *optimal trunca-*

tion. In Section 4.2 we give a detailed description of the GCRO-DR method, which was a starting point in our research, as well as two convergence results. Further, in Section 4.3 we show two ways of including the error approximations in the recycling process, which is the only difference between LGCRO-DR and GCRO-DR. Finally, in Section 4.4.1 we work out details for our method LRDGMRES. Moreover, we propose two results for cheaply obtaining the left harmonic Ritz vetors and discuss some disadvantages of our method when compared to GCRO-DR. In last Section 4.4.2 of this chapter, we briefly explain the LLRDGMRES method.

We compare the four methods described in Chapter 4 for various applications in Chapter 5. We consider in Section 5.1 an application which results in a sequence of symmetric matrices, only to demonstrate that GCRO-DR and LRDGMRES are equivalent in the Hermitian case. In Section 5.2 we construct a 3×3 example whose purpose is to show the power of the approach which utilizes the oblique projections. Further, we compare the methods for one nonsymmetric system from a fluid dynamics application (Section 5.3) and a sequence of nonsymmetric systems arising in the Korringa-Kohn-Rostoker method in solid-state physics. The application of most interest for us was definitely lattice QCD. We present results for 5 consecutive systems arising from the hybrid Monte Carlo integration in Section 5.5. Moreover, we apply the red-black multiplicative Schwarz method as a preconditioner, and show results in Section 5.5.1. Finally, Section 5.5.3 describes how diagonal similarity transformations of the matrices in a sequence can be exploited in lattice QCD to improve all four methods.

In Chapter 6 we give some final remarks and conclusions about the work done within this thesis and discuss some further plans.

2 Basic Concepts

In this chapter we gather some basic definitions and results that are scattered throughout the literature and that are useful for the remainder of the thesis. Projectors play an important role in numerical linear algebra, as well as in this thesis, and therefore, we give a brief overview of basic properties. Most of the iterative techniques covered in later chapters utilize a projection process. Thus, we describe a basic projection step in its general form and present some theory we found useful. Next, we also briefly recall the theory behind Krylov subspaces, as well as the Arnoldi method. While all of the results regarding projections and Krylov subspaces are known and widely used, the last section of this chapter is different, as its definitions and results are not widely used in the literature. The concept of Ritz and harmonic Ritz pairs can be found in many books and papers, but they usually refer to the *right* pairs. In Section 2.4 we define, in addition, the *left* pairs and prove some useful properties they satisfy.

2.1 Projectors

Definition 2.1. *A linear operator $P : \mathbb{C}^n \to \mathbb{C}^n$ is called a projector if $P^2 = P$.*

It follows immediately from the definition that if P is a projector, then so is $(I - P)$, and the following relations hold,

$$\mathrm{Ker}(P) = \mathrm{Ran}(I - P)$$
$$\mathrm{Ran}(P) = \mathrm{Ker}(I - P).$$

The next two lemmas show that each projector is uniquely characterized by two subspaces, its range and null space. For proofs we refer to [Saa03], e.g.

Lemma 2.2. *The space $\mathbb{C}^n$ can be decomposed as the direct sum*

$$\mathbb{C}^n = \mathit{Ker}(P) \oplus \mathit{Ran}(P).$$

Lemma 2.3. *Every pair of subspaces M and S which forms a direct sum of $\mathbb{C}^n$ defines a unique projector P such that $\mathit{Ran}(P) = M$ and $\mathit{Ker}(P) = S$. The associated projector P maps an element x of $\mathbb{C}^n$ into the M-component x_1 in the unique decomposition $x = x_1 + x_2$, $x_1 \in M, x_2 \in S$.*

It is said that the projector P projects onto the subspace M and along the subspace S. In the literature, projectors are usually defined through the orthogonal complement $L = S^{\perp}$ of the subspace S. The following equations define the projector P onto M and orthogonal to L

$$Px \in M \tag{2.1}$$
$$x - Px \perp L. \tag{2.2}$$

The following lemma gives us conditions under which it is possible to define such a projector. The proof follows immediately from Lemma 2.3 with $S = L^{\perp}$.

Lemma 2.4. *Given two subspaces M and L of the same dimension m, the following two conditions are mathematically equivalent.*

1. *No nonzero vector of M is orthogonal to L*
2. *For any $x \in \mathbb{C}^n$ there is a unique vector Px which satisfies* (2.1) *and* (2.2).

Next, we consider matrix representations of projectors. Let us assume that the columns v_i and w_i of V and W form orthonormal bases for subspaces M and L, respectively. Since $Px \in M$, it can be written as

$$Px = Vy. \tag{2.3}$$

The constraint (2.2) is equivalent to the condition

$$\langle (x - Vy), w_j \rangle = 0, \quad j = 1, \dots, m.$$

This can be rewritten in the matrix form

$$W^H(x - Vy) = 0.$$

The previous equation yields the expression for computing y,

$$W^H x = W^H V y \quad \Leftrightarrow \quad y = (W^H V)^{-1} W^H x. \tag{2.4}$$

From (2.3) and (2.4) we get the matrix representation of the projector P,

$$P = V(W^H V)^{-1} W^H.$$

Under the assumptions of Lemma 2.4, the nonsingularity of the matrix $W^H V$ is guaranteed. In case that the two bases are biorthogonal, i.e. $W^H V = I$, we have, as a special case, the following representation of P,

$$P = V W^H.$$

We distinguish two different classes of projectors. In the case when the subspace L is equal to the subspace M it is said that P is the *orthogonal* projector onto M. A projector that is not orthogonal is called *oblique*. In order to provide the condition under which a projector is orthogonal, we have to define the adjoint P^H of the projector P and consider some of the properties of the adjoint.

Definition 2.5. *The mapping P^H is the adjoint of P if*

$$(P^H x, y) = (x, Py), \quad \forall x, \forall y.$$

It is easily shown that P^H is also a projector,

$$((P^H)^2 x, y) = (P^H x, Py) = (x, P^2 y) = (x, Py) = (P^H x, y).$$

The following relations

$$\begin{aligned} \mathrm{Ker}(P^H) &= \mathrm{Ran}(P)^\perp \\ \mathrm{Ker}(P) &= \mathrm{Ran}(P^H)^\perp \end{aligned}$$

hold as a consequence of Definition 2.5 and lead to this important result, see [Saa03].

Proposition 2.6. *A projector is orthogonal if and only if it is Hermitian.*

We conclude this section with a few basic properties of orthogonal projectors.

Lemma 2.7. *Let P be an orthogonal projector. Then the two vectors Px and $(I - P)x$ are orthogonal and the following holds*

$$\|x\|_2^2 = \|Px\|_2^2 + \|(I - P)x\|_2^2 . \tag{2.5}$$

This is just a consequence of Pythagoras' theorem. It follows directly from (2.5) that $\|Px\|_2^2 \leq \|x\|_2^2$, i.e. $\|Px\|_2^2 / \|x\|_2^2 \leq 1$. In addition, the value 1 is reached for any element in $\mathrm{Ran}(P)$. Thus, $\|P\|_2 = 1$, unless $P = 0$.

Remark 2.8. An orthogonal projector has only two eigenvalues, 0 or 1. Vectors in the range of P are eigenvectors associated with the eigenvalue 1, and vectors in the null space of P are the eigenvectors corresponding to the eigenvalue 0.

Geometrically, the orthogonal projection of a vector $x \in \mathbb{C}^n$ onto the subspace M is the shortest distance from that subspace, as formulated in the next theorem, see [Saa03], e.g.

Theorem 2.9. *Let P be the orthogonal projector onto a subspace M. Then for any given vector $x \in \mathbb{C}^n$, the following is true:*

$$\min_{y \in M} \|x - y\|_2 = \|x - Px\|_2 .$$

It is possible to reformulate this result in a form of necessary and sufficient conditions such that we can determine the best approximation to a given vector.

Corollary 2.10. *Let a subspace M and a vector $x \in \mathbb{C}^n$ be given and let $y^* = Px$. Then*

$$\min_{y \in M} \|x - y\|_2 = \|x - y^*\|_2 ,$$

if and only if the following two conditions are satisfied,

$$y^* \in M$$
$$x - y^* \perp M.$$

2.2 Projection Methods

The main subject of the thesis is solving sequences of linear systems. Let us first consider solving a single linear system

$$Ax = b, \quad A \in \mathbb{C}^{n \times n}, \quad x, b \in \mathbb{C}^n. \tag{2.6}$$

Most of the existing iterative methods for solving (2.6) utilize a projection process. The main idea of a projection method is to extract an approximate solution to the problem (2.6) from a subspace $\mathcal{K} \subseteq \mathbb{C}^n$, which is called the *search subspace*. Considering that we usually want to exploit the knowledge of an initial guess x_0, the solution is, therefore, sought in an affine space $x_0 + \mathcal{K}$. If the dimension of the subspace $\mathcal{K}$ is m, then, in general, m constraints must be imposed in order to uniquely extract such an approximation. Typically, the residual vector $r = b - Ax$ is constrained to be orthogonal to another subspace $\mathcal{L}$, which is called the *subspace of constraints*. This framework is well-known as the Petrov-Galerkin approach. In the special case when $\mathcal{L} = \mathcal{K}$, the Petrov-Galerkin approach is often called the Galerkin approach.

A basic projection technique onto the subspace $\mathcal{K}$ and orthogonal to $\mathcal{L}$, as described above, can be defined as follows:

$$\text{Find } \tilde{x} \in x_0 + \mathcal{K} \quad \text{ such that } \quad b - A\tilde{x} \perp \mathcal{L}. \tag{2.7}$$

Writing $\tilde{x} = x_0 + \delta$, where $\delta \in \mathcal{K}$, the orthogonality condition can be rewritten as

$$\begin{aligned} & b - A\tilde{x} \perp \mathcal{L} \\ \Leftrightarrow\ & b - A(x_0 + \delta) \perp \mathcal{L} \\ \Leftrightarrow\ & r_0 - A\delta \perp \mathcal{L}, \end{aligned}$$

where $r_0 = b - Ax_0$ is the initial residual. This leads us to the basic projection step:

$$\tilde{x} = x_0 + \delta, \quad \delta \in \mathcal{K}, \tag{2.8}$$

$$\langle r_0 - A\delta, w \rangle = 0, \quad \forall w \in \mathcal{L}. \tag{2.9}$$

Most of the iterative methods use a succession of such projections, where, typically, in each step a new pair of subspaces $\mathcal{K}$ and $\mathcal{L}$ is used and the new initial guess x_0 is equal to the most recent approximation obtained from the previous projection step.

We distinguish two classes of projection techniques: *orthogonal* and *oblique*. In an orthogonal projection method, the subspace $\mathcal{L}$ is equal to the subspace $\mathcal{K}$, while in an oblique projection method they are different and may be completely unrelated.

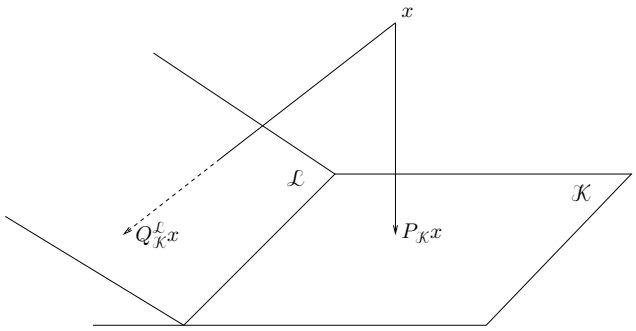


Figure 2.1: Orthogonal and oblique projectors

Figure 2.1 (reproduced from [Saa03]) illustrates $P_{\mathcal{K}}$, the orthogonal projector onto the subspace $\mathcal{K}$, while $Q_{\mathcal{K}}^{\mathcal{L}}$ illustrates the oblique projector onto $\mathcal{K}$, orthogonally to $\mathcal{L}$, i.e.

$$\begin{aligned} P_{\mathcal{K}}x \in \mathcal{K}, \quad & x - P_{\mathcal{K}}x \perp \mathcal{K} \\ Q_{\mathcal{K}}^{\mathcal{L}}x \in \mathcal{K}, \quad & x - Q_{\mathcal{K}}^{\mathcal{L}}x \perp \mathcal{L}. \end{aligned}$$

Choosing different subspaces $\mathcal{L}$ gives rise to many different algorithms. Throughout this thesis, we will focus on the case $\mathcal{L} = A\mathcal{K}$, where $\mathcal{K}$ is a Krylov subspace, which we define in the next section.

Geometrically, the orthogonality condition (2.9) for the case $\mathcal{L} = A\mathcal{K}$ means that the vector $A\delta$ is the orthogonal projection of the vector r_0 onto the subspace $A\mathcal{K}$. Hence, the following holds.

Proposition 2.11. *Let A be an arbitrary square matrix and assume that $\mathcal{L} = A\mathcal{K}$. Then, the following holds:*

1. *A vector $\tilde{x}$ is the result of an oblique projection method onto $\mathcal{K}$ orthogonally to $\mathcal{L}$ with the starting vector x_0 if and only if it minimizes the 2-norm of the residual vector $b - Ax$ over $x \in x_0 + \mathcal{K}$, i.e. iff*

$$R(\tilde{x}) = \min_{x \in x_0 + \mathcal{K}} R(x), \tag{2.10}$$

   where $R(x) \equiv \|b - Ax\|_2$.

2. *Let $\tilde{r} = b - A\tilde{x}$ be the residual associated with the approximate solution $\tilde{x}$. Then*

$$\tilde{r} = (I - P)r_0, \tag{2.11}$$

   where P denotes the orthogonal projector onto the subspace $A\mathcal{K}$. Consequently

$$\|\tilde{r}\| \leq \|r_0\|. \tag{2.12}$$

Proof. 1. It follows from Corollary 2.10 that for a vector $\tilde{x}$ to be the minimizer of $R(x)$ it is necessary and sufficient that $b - A\tilde{x}$ be orthogonal to all vectors of the form Ay, where y belongs to $\mathcal{K}$, i.e.

$$\langle b - A\tilde{x}, Ay \rangle = 0, \quad \forall y \in \mathcal{K},$$

   which is exactly the Petrov-Galerkin condition that defines the approximate solution $\tilde{x}$.

2. The inequality (2.12) is an immediate consequence of the equation (2.10), while the equation (2.11) follows from the fact that the vector $A\delta$ in (2.9) is the orthogonal projection of the vector r_0 onto the subspace $A\mathcal{K}$.

□

Methods relying on Proposition 2.11 are known as *residual projection methods.*

We complete this section with the matrix representation of the expression for the approximate solution of (2.6). Let the columns of matrices V and W form bases of $\mathcal{K}$ and $\mathcal{L}$, respectively. The approximate solution can be written as

$$x = x_0 + Vy.$$

Then, the orthogonality condition (2.9) can be rewritten as

$$W^T AVy = W^T r_0.$$

Under the assumption that the matrix $W^T AV$ is nonsingular, we obtain the expression for the approximate solution $\tilde{x}$ of (2.6)

$$\tilde{x} = x_0 + V(W^T AV)^{-1} W^T r_0.$$

The matrix $W^T AV$ does not have to be nonsingular, i.e. the assertions of Lemma 2.4 are not necessarily fulfilled, even when the matrix A is nonsingular. Conditions under which the nonsingularity of the matrix $W^T AV$ is guaranteed are discussed next, see [Saa03].

Proposition 2.12. *Let A, $\mathcal{K}$ and $\mathcal{L}$ satisfy either one of the following conditions:*

1. *A is positive definite and $\mathcal{L} = \mathcal{K}$, or*
2. *A is nonsingular and $\mathcal{L} = A\mathcal{K}$.*

Then the matrix $W^T AV$ is nonsingular for any bases V and W of $\mathcal{K}$ and $\mathcal{L}$, respectively.

2.3 Krylov Subspaces

Since the 1950s, splitting methods were widely used to solve (2.6). The basic idea of splitting methods is to decompose the matrix $A = M - N$, where M is nonsingular. The solution is then obtained iteratively via the recurrence

$$x_{m+1} = M^{-1} N x_m + M^{-1} b, \quad m = 0, 1, ..., \tag{2.13}$$

where x_0 is an arbitrary vector. Convergence is not guaranteed for arbitrary splittings. The following theorem provides the necessary and sufficient condition for convergence.

Theorem 2.13. *The iteration* (2.13) *converges to the solution of the system $x^* = A^{-1} b$ for any starting vector x_0 and any right-hand side b if and only if $\rho(M^{-1} N) < 1$.*

In practice, one does not want to compute the spectral radius of a matrix, since this can be very expensive. Instead, one uses the following result, which utilizes the inequality $\rho(M^{-1} N) \leq \|M^{-1} N\|$.

Corollary 2.14. *Let $\|M^{-1}N\| < 1$ for some operator norm. Then the iteration* (2.13) *converges to the solution of the system $x^* = A^{-1}b$ for any starting vector x_0 and any right-hand side b.*

Today, splitting methods are used mostly as preconditioners or as smoothers in multigrid methods and more advanced iterative techniques are used to tackle (2.6). The most common of these advanced techniques are Krylov subspace methods, in which the subspace $\mathcal{K}$ in (2.7) is a Krylov subspace.

Definition 2.15. *Let $A \in \mathbb{C}^{n\times n}$ and $r \in \mathbb{C}^n$. A Krylov subspace of dimension m is defined as*

$$\begin{aligned}\mathcal{K}_m(A,r) &= \operatorname{span}\{r, Ar, \dots, A^{m-1}r\} \\ &= \{x \in \mathbb{C}^n \,:\, x = p_{m-1}(A)r, p_{m-1} \in \Pi_{m-1}\}\,. \end{aligned} \tag{2.14}$$

It follows from the definition that the sequence $\mathcal{K}_1(A,r), \dots, \mathcal{K}_m(A,r)$ of Krylov subspaces is nested. Another important property is that the dimension m cannot grow arbitrarily. To investigate this further we must recall the definition of the minimal polynomial and the grade.

Definition 2.16. *The minimal polynomial of a vector r with respect to the matrix $A \in \mathbb{C}^{n\times n}$ is the nonzero monic polynomial χ of lowest degree such that $\chi(A)r = 0$. The degree, γ, of the minimal polynomial is called the grade of r (with respect to A).*

The following proposition determines the dimension of $\mathcal{K}_m$ in general. For the proof consult [Saa03].

Proposition 2.17. *Let $A \in \mathbb{C}^{n\times n}$, $r \in \mathbb{C}^n$ and let γ be the grade of r. Then:*

a) $\mathcal{K}_\gamma$ is invariant under A and $\mathcal{K}_m = \mathcal{K}_\gamma$ for all $m \geq \gamma$.

b) The Krylov subspace $\mathcal{K}_m$ is of dimension m if and only if $\gamma \geq m$. Therefore,

$$dim(\mathcal{K}_m) = min\{m, \gamma\}.$$

The reason for Krylov subspaces being such an important concept in numerical linear algebra lies in the fact that $A^{-1}r$, the action of the inverse of a non-singular matrix $A \in \mathbb{C}^{n\times n}$ on a vector r, can be expressed as a polynomial p in A of degree $\gamma - 1$, with γ the grade of r. It follows from Definition 2.16 that

$$\chi_\gamma(A)r_0 = 0,$$

which further implies

$$A^{-1}r_0 = q_{\gamma-1}(A)r_0,$$

where

$$q_{\gamma-1}(A) = \frac{\chi_\gamma(A) - \chi_\gamma(0)}{\chi_\gamma(0)},$$

and $\chi_\gamma(0) \neq 0$, since A is a non-singular matrix. Thus, the solution of the system (2.6) can be computed as

$$x^* = x_0 + A^{-1}r_0 = x_0 + q_{\gamma-1}(A)r_0.$$

On the other hand, from (2.14), we see that the iterates extracted from the Krylov subspace are exactly of this form, i.e.

$$A^{-1}b \approx x_m = x_0 + p_{m-1}(A)r_0,$$

for some polynomial p_{m-1} of dimension $m-1$.

Due to the properties of the power iteration, the vectors $r, Ar, \ldots, A^{m-1}r$ become almost linearly dependent and are not a good choice for the basis of the Krylov subspace $\mathcal{K}_m$. Therefore, methods based on Krylov subspaces usually involve some orthogonalization scheme. Considering that we will be dealing with nonsymmetric systems, we will describe the Arnoldi process.

2.3.1 The Arnoldi Process

The Arnoldi process [Arn51] is an orthogonalization procedure which builds up matrices of Hessenberg form. It turns out that the eigenvalues of the Hessenberg matrix are good approximations to some eigenvalues of the original matrix, which leads to an efficient algorithm for approximating the eigenvalues of large sparse matrices. The Arnoldi method is based on the Gram-Schmidt algorithm. We will describe the version that uses modified Gram-Schmidt, a more stable algorithm than classical Gram-Schmidt. There are some other versions like Householder Arnoldi [Wal88] etc., but we will not discuss them here.

Most of the Krylov subspace methods described in the following chapters will be based on the Arnoldi method. Therefore, we give here some important basic properties of the process. The Arnoldi process is described as Algorithm 2.1. It computes a sequence of orthonormal vectors $v_1, v_2, \ldots$ such that $v_1, \ldots, v_m$ is an orthonormal basis of $\mathcal{K}_m(A, v_1)$.

Proposition 2.18. *Assume that Algorithm 2.1 does not stop before the mth step, i.e. $h_{k+1,k} \neq 0$. Then the vectors $v_1, v_2, \ldots, v_m$ form an orthonormal basis of the Krylov subspace $\mathcal{K}_m(A, v_1)$.*

Algorithm 2.1: ARNOLDI PROCESS

Input : $A \in \mathbb{C}^{n\times n}$ system matrix
$v_1 \in \mathbb{C}^n$ starting vector
m number of basis vectors to build

Output: $\{v_1, v_2, \dots, v_m\}$ orthonormal basis of $\mathcal{K}_m(A, v_1)$

1 $\beta = \|v_1\|_2$
2 $v_1 = v_1/\beta$
3 **for** $k = 1, 2, \dots, m$ **do**
4 $\quad w = Av_k$
5 $\quad$ **for** $i = 1, 2, \dots, k$ **do** // modified Gram-Schmidt
6 $\quad\quad h_{i,k} = \langle w, v_i \rangle$
7 $\quad\quad w = w - h_{i,k} v_i$
8 $\quad h_{k+1,k} = \|w\|_2$
9 $\quad$ **if** $h_{k+1,k} = 0$ **then**
10 $\quad\quad$ set $m = k$ and **Stop**
11 $\quad v_{k+1} = w/h_{k+1,k}$

Proposition 2.19. *Let V_m be the $n \times m$ matrix with column vectors $v_1, \dots, v_m$, $\bar{H}_m$ the $(m+1) \times m$ Hessenberg matrix whose nonzero entries h_{ij} are defined by Algorithm 2.1 and H_m the matrix obtained from $\bar{H}_m$ by deleting its last row. Then the following relations hold:*

$$\begin{aligned} AV_m &= V_m H_m + h_{m+1,m} v_{m+1} e_m^T \\ &= V_{m+1} \bar{H}_m \\ V_m^T A V_m &= H_m. \end{aligned} \tag{2.15}$$

The relation (2.15) *is known as the Arnoldi relation.*

Algorithm 2.1 may break down if $h_{k+1,k} = 0$ in line 8. In this case v_{k+1} cannot be computed and the algorithm stops. The following proposition gives the conditions under which this occurs.

Proposition 2.20. *The Arnoldi process breaks down at step k, i.e. $h_{k+1,k} = 0$ in line 8 of Algorithm 2.1 if and only if the grade of v_1 is k. Moreover, in this case the subspace $\mathcal{K}_k$ is invariant under A.*

The eigenvalues $\lambda_i, i = 1, \dots, m$ of the resulting Hessenberg matrix H_m are good approximations to some of the eigenvalues of the matrix A. They are known as the Ritz values of the matrix A w.r.t. $\mathcal{K}_m$. The corresponding Ritz vectors are $V_m y_i, i = 1, \dots, m$, where $y_i, i = 1, \dots, m$ are the eigenvectors of

the matrix H_m belonging to eigenvalues $\lambda_i, i = 1, \dots, m$. In practice, it is usually expensive to compute the eigenvalues of A. Thus, we rather compute approximations to the wanted eigenvalues, i.e. the Ritz values or some variants. Hence, we dedicate the next section to some general definitions and properties of Ritz and harmonic Ritz pairs.

2.4 Ritz and Harmonic Ritz Pairs

Definition 2.21. *Let $A \in \mathbb{C}^{n\times n}$ and $\mathcal{K} \subseteq \mathbb{C}^n$ be any subspace. Then $y \in \mathcal{K}$ is a* **right** *Ritz vector of A with respect to $\mathcal{K}$ with a Ritz value λ if*

$$Ay - \lambda y \perp \mathcal{K}. \tag{2.16}$$

The concept of a **left** Ritz pair is not widely used in the literature. Note that a left eigenpair (μ, z) satisfies $z^H A - \mu z^H = 0 \Leftrightarrow A^H z - \bar{\mu} z = 0$. Thus, the following definition for a left Ritz pair comes naturally.

Definition 2.22. *Let $A \in \mathbb{C}^{n\times n}$ and $\mathcal{K} \subseteq \mathbb{C}^n$ be any subspace. Then $z \in \mathcal{K}$ is a* **left** *Ritz vector of A with respect to $\mathcal{K}$ with a Ritz value μ if*

$$A^H z - \bar{\mu} z \perp \mathcal{K}. \tag{2.17}$$

The following Lemma shows that with Definition 2.22 certain useful properties are satisfied. Let us first denote with $\Lambda = \{\lambda_1, \dots, \lambda_j\}$ the set of all right Ritz values and with $M = \{\mu_1, \dots, \mu_j\}$ the set of all left Ritz values of the matrix A w.r.t. to the same subspace $\mathcal{K}$, where j is the dimension of the subspace $\mathcal{K}$.

Lemma 2.23. *Let (λ, y) be a right and (μ, z) be a left Ritz pair of A w.r.t. the same subspace $\mathcal{K}$. Then, if $\lambda \neq \mu$, we have $\langle y, z\rangle = 0$. In addition, $\Lambda = M$.*

Proof. We first prove that $\Lambda = M$. Let us assume that the columns of $V \in \mathbb{C}^{n\times m}$ form an orthonormal basis of $\mathcal{K}$, and let $y = Vu$. Then (2.16) is equivalent to

$$V^H A V u = \lambda V^H V u = \lambda u.$$

This shows that the right Ritz vectors y are of the form $y = Vu$ with u an eigenvector of $V^H A V$, and λ the corresponding eigenvalue. Similarly, with $z = Vw$, (2.17) is equivalent to

$$V^H A^H V w = \bar{\mu} V^H V w = \bar{\mu} w,$$

which shows that the left Ritz vectors z are of the form $z = Vw$ with w an eigenvector of $V^H A^H V$ and $\bar{\mu}$ the corresponding eigenvalue. We have

$$\overline{M} = spec(V^H A^H V) = \overline{spec}((V^H A^H V)^H) = \overline{spec}(V^H A V) = \overline{\Lambda},$$

which proves $M = \Lambda$.
Now let $\lambda \neq \mu$ with (λ, y) a right and (μ, z) a left Ritz pair. Then, by (2.16) we have

$$\langle Ay, z\rangle = \langle \lambda y + s, z\rangle \quad with \quad s \in \mathcal{K}^{\perp},$$

and thus

$$\langle Ay, z\rangle = \langle \lambda y, z\rangle = \lambda \langle y, z\rangle. \tag{2.18}$$

Similarly, using (2.17) we get

$$\langle y, A^H z\rangle = \langle y, \bar{\mu} z + t\rangle \quad with \quad t \in \mathcal{K}^{\perp},$$

and thus

$$\langle y, A^H z\rangle = \langle y, \bar{\mu} z\rangle = \mu \langle y, z\rangle. \tag{2.19}$$

From (2.18) and (2.19) we get

$$\lambda \langle y, z\rangle = \langle Ay, z\rangle = \langle y, A^H z\rangle = \mu \langle y, z\rangle,$$

and thus $\langle y, z\rangle = 0$ if $\lambda \neq \mu$. □

Ritz vectors tend to approximate the extremal eigenvalues of A well, but can give poor approximations to the interior eigenvalues, see [Par98]. And, in our computations, we are interested in interior eigenvalues. Switching from A to A^{-1}, extremal and interior eigenvalues change their roles, so that the inverses of Ritz values of A^{-1} should give good approximations to the interior eigenvalues of A, see [Mor91]. We will now define harmonic Ritz values as the Ritz values of A^{-1} w.r.t. the subspace $A\mathcal{K}$, see [PPV95].

Definition 2.24. *Let $A \in \mathbb{C}^{n\times n}$ and $\mathcal{K} \subseteq \mathbb{C}^n$ be any subspace. Then $\tilde{y} \in A\mathcal{K}$ is a* **right** *harmonic Ritz vector of A with respect to the subspace $A\mathcal{K}$ with a harmonic Ritz value $1/\theta$ if*

$$A^{-1}\tilde{y} - \theta\tilde{y} \perp A\mathcal{K}. \tag{2.20}$$

Approximating the eigenvalues of A^{-1} using the subspace $A\mathcal{K}$ has the advantage that we do not need A^{-1} explicitly, as we will see later. As before, we propose an appropriate definition of the left harmonic Ritz pair, which seems to be new.

Definition 2.25. *Let $A \in \mathbb{C}^{n\times n}$ and $\mathcal{K} \subseteq \mathbb{C}^n$ be any subspace. Then $\tilde{z} \in A\mathcal{K}$ is a* **left** *harmonic Ritz vector of A with respect to the subspace $A\mathcal{K}$ with harmonic Ritz value $1/\eta$ if*

$$A^{-H}\tilde{z} - \bar{\eta}\tilde{z} \perp A\mathcal{K}. \tag{2.21}$$

Remark 2.26. In the Hermitian case, *regardless of the choice of the subspace*, right and left harmonic Ritz vectors are equal, since (2.20) and (2.21) become equivalent conditions.

Definition 2.25 will be very useful throughout this thesis. It will yield a cheap way of obtaining left harmonic Ritz vectors from the Krylov subspace generated by A. This way, we will avoid additional multiplications by A^H, as will be seen in Chapter 4.

In a similar fashion as Lemma 2.23, Lemma 2.27 states that left harmonic Ritz pairs have canonical properties. Let us denote with $\Theta = \{\theta_1, \dots, \theta_j\}$ the set of all right harmonic Ritz values and with $N = \{\eta_1, \dots, \eta_j\}$ the set of all left harmonic Ritz values of the matrix A w.r.t. the same subspace $A\mathcal{K}$, where, as before, j is the dimension of the subspace $\mathcal{K}$.

Lemma 2.27. *Let $(1/\theta, \tilde{y})$ be a right and $(1/\eta, \tilde{z})$ a left harmonic Ritz pair of A w.r.t. the same subspace $A\mathcal{K}$. Then, if $\theta \neq \eta$, we have $\langle \tilde{y}, \tilde{z}\rangle = 0$. In addition, $\Theta = N$.*

Proof. The proof is similar to the proof of Lemma 2.23. First, we prove that $\Theta = N$. Let us assume that the columns of $V \in \mathbb{C}^{n\times m}$ form an orthonormal basis of $\mathcal{K}$, and let $\tilde{y} = AVu$. Then (2.20) is equivalent to

$$V^H A^H A^{-1} AVu = \theta V^H A^H AVu.$$

It follows that $(V^H A^H AV)^{-1} V^H A^H Vu = \theta u$, which shows that the right harmonic Ritz vectors $\tilde{y}$ are of the form $\tilde{y} = AVu$ with u an eigenvector of $(V^H A^H AV)^{-1} V^H A^H V$, and θ the corresponding eigenvalue. Similarly, with $\tilde{z} = AVw$, (2.21) is equivalent to

$$V^H A^H A^{-H} AVw = \bar{\eta} V^H A^H AVw.$$

It follows that $(V^H A^H AV)^{-1} V^H AVw = \bar{\eta} w$, which shows that the left harmonic Ritz vectors $\tilde{z}$ are of the form $\tilde{z} = AVw$ with w an eigenvector of $(V^H A^H AV)^{-1} V^H AV$, and $\bar{\eta}$ the corresponding eigenvalue. From the fact that $((V^H A^H AV)^{-1} V^H AV)^H = V^H A^H V (V^H A^H AV)^{-1}$, we conclude

$$\begin{aligned} \overline{N} &= spec((V^H A^H AV)^{-1} V^H AV) \\ &= \overline{spec}(((V^H A^H AV)^{-1} V^H AV)^H) \\ &= \overline{spec}(V^H A^H V (V^H A^H AV)^{-1}). \end{aligned}$$

Using the well-known identity $spec(BC) = spec(CB)$, with $B = V^H A^H V$ and $C = (V^H A^H AV)^{-1}$, it follows

$$\begin{aligned}&\overline{spec}(V^H A^H V(V^H A^H AV)^{-1})\\ &= \overline{spec}((V^H A^H AV)^{-1} V^H A^H V)\\ &= \overline{\Theta},\end{aligned}$$

which proves $N = \Theta$.
Now let $\theta \neq \eta$ with $(1/\theta, \tilde{y})$ a right and $(1/\eta, \tilde{z})$ a left harmonic Ritz pair. Then, by (2.20) we have

$$\langle A^{-1}\tilde{y}, \tilde{z}\rangle = \langle \theta\tilde{y} + \tilde{s}, \tilde{z}\rangle \quad with \quad \tilde{s} \in (A\mathcal{K})^{\perp},$$

and thus

$$\langle A^{-1}\tilde{y}, \tilde{z}\rangle = \langle \theta\tilde{y}, \tilde{z}\rangle = \theta\langle \tilde{y}, \tilde{z}\rangle. \tag{2.22}$$

Similarly, using (2.21) we get

$$\langle \tilde{y}, A^{-H}z\rangle = \langle \tilde{y}, \bar{\eta}\tilde{z} + \tilde{t}\rangle \quad with \quad \tilde{t} \in (A\mathcal{K})^{\perp},$$

and thus

$$\langle \tilde{y}, A^{-H}\tilde{z}\rangle = \langle \tilde{y}, \bar{\eta}\tilde{z}\rangle = \eta\langle \tilde{y}, \tilde{z}\rangle. \tag{2.23}$$

From (2.22) and (2.23) we get

$$\theta\langle \tilde{y}, \tilde{z}\rangle = \langle A^{-1}\tilde{y}, \tilde{z}\rangle = \langle \tilde{y}, A^{-H}\tilde{z}\rangle = \eta\langle \tilde{y}, \tilde{z}\rangle,$$

and thus $\langle \tilde{y}, \tilde{z}\rangle = 0$ if $\theta \neq \eta$.

□

In this section we developed theory for an arbitrary subspace $\mathcal{K}$. Starting with the next chapter, $\mathcal{K}$ will be the m-th Krylov subspace $\mathcal{K}_m(A, r)$, if not stated otherwise.

3 GMRES, GCR and Deflation

In this chapter we recall some of the well-known methods for solving a single, non-symmetric linear system (2.6). A common choice is the GMRES method [SS86], a projection based method which minimizes the 2-norm of the residual vector in each step. Recent demands in simulations led to a significant increase in sizes of systems to solve, which made GMRES impractical due to the excessive storage or computational time requirements. Possible remedies are to restart, an approach that we discuss in Section 3.1.1, to truncate, etc. (see, e.g., [Saa03]).

The GCR method [EES83] is a mathematically equivalent method to GMRES. Although we do not consider GCR as a means to solve the system, the method framework plays an important role throughout this thesis. We also introduce the GMRESR method [VV94], which uses an inner method, which is GMRES (m steps of GMRES), and an outer method, which is GCR. The GMRESR method provides only suboptimal corrections to the solution and, in addition, it exhibits some of the problems of the restarted GMRES method. Preserving the GCR orthogonality relations within the inner method, i.e. within GMRES, leads to a more efficient method GCRO (GCR with inner orthogonalization) [Stu96a]. Although GCRO provides the optimal corrections to the solution, the storage requirements might be excessive. Hence, in the next chapter we will introduce the concept of optimal truncation which together with the GCRO framework results in the GCROT method [Stu99].

Much work has been invested in enhancing the robustness of the restarted GMRES method. Most of the methods derived since, are based on two concepts: *deflation*, i.e. removing the "problematic" (usually the smallest) eigenvalues from the spectrum and *augmentation*, i.e. enlarging the search space with carefully chosen vectors from previous cycles. The choice of the vectors and the way in which they are used gives rise to many different methods. We present GMRES-E (GMRES with eigenvectors) [Mor95] and GMRES-DR (Deflated and restarted GMRES) [Mor02], two methods developed by R. Morgan. Despite the fact that these two methods are algebraically equivalent, they are based on different approaches. As it will be shown, GMRES-E uses an augmentation approach and can be modified for solving sequences of linear systems, whereas GMRES-DR uses an orthogonalization approach and can be efficient

when solving one system, but cannot be adapted for solving sequences of systems. A somewhat similar idea was exploited to derive the "loose" GMRES method [BJM05], which we also introduce.

Deflation and augmentation techniques can often show their full potential, only if they are used together with preconditioning. Preconditioning plays only a minor role in this thesis. Accordingly, in Section 3.6 we only give a brief introduction to right preconditioning. In Chapter 5, we will expand this topic by introducing a certain preconditioner for the lattice QCD application.

Proper combination of the methods from this chapter will result in elegant and efficient methods for solving sequences of linear systems, which will be the topic of the next chapter.

3.1 GMRES

The Generalized Minimal Residual method (GMRES) [SS86] is a projection based Krylov subspace method which minimizes the 2-norm of the residual vector in each step. One way to derive the method is to consider the conditions (2.8) and (2.9) with $\mathcal{K} = \mathcal{K}_k$ and $\mathcal{L} = A\mathcal{K}_k$, where $\mathcal{K}_k$ is the k-th Krylov subspace. We know from Proposition 2.11 that this means that the 2-norm of the residual is minimized. We now focus on more algorithmic aspects, in particular on how to obtain the iterates from the Arnoldi process (Alg. 2.1).

Any approximate solution, extracted from an affine space $x_0 + \mathcal{K}_k$, can be written as $x_k = x_0 + V_k y$, where the columns of V_k form an orthonormal basis of $\mathcal{K}_k$ and $y \in \mathbb{C}^k$. Thus, the 2-norm of the residual can be rewritten as

$$\begin{aligned} \|b - Ax\|_2 &= \|b - A(x_0 + V_k y)\|_2 \\ &= \|r_0 - AV_k y\|_2 . \end{aligned}$$

Using the Arnoldi relation (2.15) and the fact that the matrix V_{k+1} is orthonormal, it follows

$$\begin{aligned} \|b - Ax\|_2 &= \|r_0 - AV_k y\|_2 \\ &= \left\|\beta v_1 - V_{k+1}\bar{H}_k y\right\|_2 \\ &= \left\|V_{k+1}(\beta e_1 - \bar{H}_k y)\right\|_2 \\ &= \left\|\beta e_1 - \bar{H}_k y\right\|_2 , \end{aligned}$$

where $\beta = \|r_0\|_2$, as defined in the Arnoldi process (Alg. 2.1), and e_1 is the first unit vector. Therefore, the initial problem is reduced to solving the least-

squares problem

$$\min_y \left\|\beta e_1 - \bar{H}_k y\right\|_2 \tag{3.1}$$

of smaller size $(k+1) \times k$. Having this in mind, the GMRES approximate solution is defined as the unique vector $x_k = x_0 + V_k y_k$, where $y_k = \operatorname{argmin}_y \left\|\beta e_1 - \bar{H}_k y\right\|_2$.

The typical way of solving the least-squares problem (3.1) is by computing the QR decomposition of the Hessenberg matrix $\bar{H}_k$, $\bar{H}_k = Q_k R_k$, which allows to compute the solution of the least-squares system as a solution of a triangular linear system, see (3.2). Considering the special structure of the matrix $\bar{H}_k$, i.e. the Hessenberg form, this can be done efficiently by multiplying the Hessenberg matrix $\bar{H}_k$ and the right-hand side βe_1 by a sequence of Givens rotations

$$G_i := \begin{bmatrix} I & & & \\ & c_i & s_i & \\ & -\bar{s}_i & c_i & \\ & & & I \end{bmatrix} \begin{matrix} \\ \leftarrow \ i\text{-th row} \\ \leftarrow \ (i+1)\text{-st row} \\ \\ \end{matrix}$$

where $|c_i|^2 + |s_i|^2 = 1$. The coefficients c_i and s_i are chosen such that in each step the element $\tilde{h}_{i+1,i}$ of the matrix $\tilde{H}_{i-1} = G_{i-1} \cdots G_1 \bar{H}_k$ is eliminated. The following choice achieves that:

$$c_i = \begin{cases} 0 & \text{, if } \tilde{h}_{i,i} = 0, \\ \frac{|\tilde{h}_{i,i}|}{\sqrt{|\tilde{h}_{i,i}|^2 + |\tilde{h}_{i+1,i}|^2}} & \text{, else} \end{cases}, \quad s_i = \begin{cases} 1 & \text{, if } \tilde{h}_{i,i} = 0, \\ \frac{\tilde{h}_{i,i}\bar{\tilde{h}}_{i+1,i}}{|\tilde{h}_{i,i}|\sqrt{|\tilde{h}_{i,i}|^2 + |\tilde{h}_{i+1,i}|^2}} & \text{, else.} \end{cases}$$

Defining $Q_k = G_1^H G_2^H \cdots G_k^H$ we obtain

$$\begin{aligned} \bar{H}_k &= Q_k R_k \\ \gamma_k &= Q_k^H (\beta e_1), \end{aligned}$$

where γ_k is the new right-hand side. Noticing that Q_k is a unitary matrix, the least-squares problem (3.1) can be rewritten as

$$\begin{aligned} \min_y \left\|\beta e_1 - \bar{H}_k y\right\|_2 &\Leftrightarrow \min_y \left\|Q_k \gamma_k - Q_k R_k y\right\|_2 \\ &\Leftrightarrow \min_y \left\|\gamma_k - R_k y\right\|_2 . \end{aligned} \tag{3.2}$$

The new least-squares problem (3.2) is solved by deleting the last row of the matrix R_k and the last element of the right-hand side γ_k, and then solving the resulting (upper triangular) linear system to obtain y_k and further, the next iterate $x_k = V_k y_k$.

Using Givens rotations is not only an elegant way for solving the least-squares problem (3.1); it also provides the 2-norm of the residual $\|r_k\|_2$ in each step

$$\begin{aligned}\|r_k\|_2 &= \|b - Ax_k\|_2 \\ &= \left\|\beta e_1 - \bar{H}_k y_k\right\|_2 \\ &= \|\gamma_k - R_k y_k\|_2 \\ &= |(\gamma_k)_{k+1}|.\end{aligned}$$

In other words, a stopping criterion is available in every step without having to compute the residual and its norm explicitly. We end up obtaining the GMRES method, as described in Algorithm 3.1.

Algorithm 3.1: GMRES

Input : $A \in \mathbb{C}^{n\times n}$ system matrix
$b \in \mathbb{C}^n$, $x_0 \in \mathbb{C}^n$ right-hand side and initial guess

Output: approximate solution x_k to $Ax = b$

1 $r_0 = b - Ax_0$
2 $\beta_0 = \|r_0\|_2$
3 $v_1 = r_0/\beta_0$
4 $\gamma_0 = \beta_0 e_1$
5 **for** $k = 1, 2, \ldots$ *until convergence* **do**
6 compute v_{k+1} and $\bar{H}_k$ // Arnoldi process (Alg. 2.1)
7 apply $G_{k-1} \cdots G_1$ to the last column of $\bar{H}_k$ yielding $\tilde{R}_k$
8 compute the Givens rotation G_k using $\tilde{R}_k$
9 apply G_k to the result $\tilde{R}_k$ of line 7 yielding R_k
10 apply G_k to the k-th and $(k+1)$-st entry of $\begin{pmatrix}\gamma_{k-1}\\ 0\end{pmatrix}$ yielding γ_k
11 $\beta_k = |\gamma_k(k+1)|$
12 solve $\gamma_k(1:k) = R_k(1:k, 1:k)y_k$ for y_k
13 set $x_k = x_0 + V_k y_k$

A major concern, when discussing iterative methods, is whether they may break down. In GMRES, the breakdown occurs when dividing by 0 in line 11 of the Arnoldi process (Alg. 2.1). The following proposition states that the GMRES method cannot break down, unless it has already converged to the exact solution, see [SS86].

Proposition 3.1. *The approximate solution x_k produced by GMRES at step k is exact if and only if the following three equivalent conditions hold:*

a) $w = 0$.

b) $h_{k+1,k} = 0$.

c) The grade of v_1 *is equal to* k.

This type of breakdown is referred to as a "lucky" breakdown in the literature. Since the grade of v_1 cannot exceed n, an immediate consequence is that GMRES terminates in at most n steps.

The convergence of GMRES can be described by a bound on the 2-norm of the residuals. We give here two well-known results. For proofs and further discussions see [Gre97].

Theorem 3.2. *Assume that* A *is diagonalizable and let* $A = V\Lambda V^{-1}$ *be an eigendecomposition, where* $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$. *Then it follows*

$$\begin{aligned}\|r_k\| &= \min_{p_k \in \overline{\Pi}_k} \|V p_k(\Lambda) V^{-1} r_0\| \\ &\leq \|V\| \cdot \|V^{-1}\| \cdot \|r_0\| \min_{p_k \in \overline{\Pi}_k} \|p_k(\Lambda)\|\end{aligned}$$

and the residuals of GMRES satisfy the equation

$$\frac{\|r_k\|}{\|r_0\|} \leq \kappa(V) \min_{p_k \in \overline{\Pi}_k} \max_{i=1,\dots,n} |p_k(\lambda_i)|, \tag{3.3}$$

where $\kappa(V) = \|V\| \cdot \|V^{-1}\|$ *is the condition number of the eigenvector matrix* V. *Moreover, if* A *is normal, then*

$$\frac{\|r_k\|}{\|r_0\|} \leq \min_{p_k \in \overline{\Pi}_k} \max_{i=1,\dots,n} |p_k(\lambda_i)|$$

The bound (3.3) is not sharp in general, in particular when the matrix V is ill-conditioned.

Theorem 3.3. *Let* $\mathcal{F}(A) = \{\frac{x^H A x}{x^H x} | x \in \mathbb{C}^n, x \neq 0\}$ *be the field of values of* $A \in \mathbb{C}^{n \times n}$. *Assume that* $\mathcal{F}(A)$ *is contained in a disk* $D = \{z \in \mathbb{C} : |z - a| \leq b\}$ *which does not contain the origin. Then the GMRES residuals satisfy*

$$\frac{\|r_k\|}{\|r_0\|} \leq 2 \left(\frac{b}{|a|} \right)^k.$$

The GMRES method becomes impractical as the number of iteration steps grows due to the excessive storage and computational time requirements: the k-th step of the underlying Arnoldi process requires the storage of k vectors and k inner products of vectors of length n. One remedy is to restart the method after a certain number of steps m, an approach that we will shortly discuss. There are other variants based on the truncation of the orthogonalization in the Arnoldi process. Both, restarted GMRES (GMRES(m)) and Quasi-GMRES (QGMRES) are covered in detail in [Saa03].

3.1.1 Restarted GMRES

The idea of restarting is rather easy to understand and can be implemented straightforwardly. Basically, after m steps of GMRES, where m is usually much smaller than the dimension of the system, the method is restarted choosing the approximate solution x_m as the initial guess for the next cycle. This leads to the restarted GMRES method [SS86], termed GMRES(m), as described in Algorithm 3.2.

Algorithm 3.2: GMRES(m)

Input : $A \in \mathbb{C}^{n\times n}$ system matrix
$b \in \mathbb{C}^n$, $x_0 \in \mathbb{C}^n$ right-hand side and initial guess
$m \in \mathbb{N}$ restart length

Output: approximate solution x_m to $Ax = b$

1 $r_0 = b - Ax_0$, $\beta_0 = \|r_0\|_2$, $v_1 = r_0/\beta_0$
2 Perform m steps of the Arnoldi process (Alg. 2.1) yielding V_m and $\bar{H}_m$
3 Compute y_m which minimizes $\|\beta e_1 - \bar{H}_m y\|_2$ and $x_m = x_0 + V_m y_m$
4 **if** *satisfied* **then**
5 | Stop
6 **else**
7 | set $x_0 = x_m$ and go to 1

It is well-known that restarted GMRES might converge significantly more slowly than full GMRES. In some cases, even stagnation can occur which means that there is no convergence towards the solution. A simple example that illustrates such a behaviour is to consider the matrix A and the right-hand side b

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

with $x_0 = 0$. Then GMRES(1) will always produce the same approximate solution, i.e. $x_1 = x_0$, and thus $x_k = x_0$ for all k, meaning that the method will stagnate. Much work has been done with the same goal of improving the convergence behaviour of the restarted GMRES method. Most of the methods try to overcome the fact that in restarted GMRES at the time of restart all information built up in the previous cycle is discarded. In later sections we present some of the methods that improve the convergence of subsequent cycles and also systems by keeping some relevant information from previous cycles (systems). The choice of which information should be kept and the way in which it is used give rise to many different methods.

3.2 GCR

The generalized conjugate residual method (GCR) [Stu96a] is a method algebraically equivalent to the GMRES method. It is a modification of the conjugate residual method for solving nonsymmetric systems, where the symmetric part of the matrix, i.e. $(A + A^H)/2$ is positive definite. If this is not the case, GCR may break down.

The basic idea of the method is to keep two bases U_k and C_k for $\mathcal{K}_k(A, r_0)$ and $A\mathcal{K}_k(A, r_0)$, respectively, such that

$$\operatorname{range}(U_k) = \mathcal{K}_k(A, r_0) \tag{3.4}$$
$$AU_k = C_k \tag{3.5}$$
$$C_k^H C_k = I. \tag{3.6}$$

The method solves the same minimization problem as GMRES

$$\begin{aligned} x_k = \underset{x \in x_0 + \operatorname{range}(U_k)}{\operatorname{argmin}} \|b - Ax\|_2 &\Leftrightarrow y_k = \underset{y \in \mathbb{C}^k}{\operatorname{argmin}} \|b - A(x_0 + U_k y)\|_2 \\ &\Leftrightarrow y_k = \underset{y \in \mathbb{C}^k}{\operatorname{argmin}} \|r_0 - C_k y\|_2 . \end{aligned} \tag{3.7}$$

Because of (3.6) the solution of the problem (3.7) is given by $y_k = C_k^H r_0$ and therefore

$$x_k = x_0 + U_k C_k^H r_0. \tag{3.8}$$

Updating the residual is straightforward

$$r_k = b - Ax_0 - AU_k C_k^H r_0 = r_0 - C_k C_k^H r_0, \tag{3.9}$$

and because of the minimization property and Proposition 2.11 we have

$$r_k \perp \operatorname{range}(C_k).$$

It is worth mentioning that within the GCR method we have constructed the inverse of A over the space $\operatorname{range}(C_k)$, in the sense that

$$A^{-1} C_k = U_k,$$

which one can regard as the underlying principle of the method, see Algorithm 3.3.

Throughout this thesis, we will not consider GCR as a means to solve a system because of possible breakdowns and its reduced numerical stability as compared to GMRES. We are rather interested in the framework of the

Algorithm 3.3: GCR

Input : $A \in \mathbb{C}^{n\times n}$ system matrix
$b \in \mathbb{C}^n$, $x_0 \in \mathbb{C}^n$ right-hand side and initial guess

Output: approximate solution x_k to $Ax = b$

1 $r_0 = b - Ax_0$, $k = 0$
2 **while** $\|r_k\|_2 > tol$ **do**
3 $\quad k = k + 1$
4 $\quad u_k = r_{k-1}, c_k = Au_k$
5 $\quad$ **for** $i = 1, \ldots, k-1$ **do**
6 $\quad\quad \alpha_i = c_i^H c_k$
7 $\quad\quad c_k = c_k - \alpha_i c_i$
8 $\quad\quad u_k = u_k - \alpha_i u_i$
9 $\quad c_k = c_k / \|c_k\|_2, u_k = u_k / \|c_k\|_2$
10 $\quad x_k = x_{k-1} + u_k c_k^H r_{k-1}$
11 $\quad r_k = r_{k-1} - c_k c_k^H r_{k-1}$

method, just described above, which will be the starting point for more advanced methods in later sections. Having that in mind, and the mathematical equivalence to GMRES, we conclude this section by commenting that, as long as GCR is feasible (which is guaranteed when $(A + A^H)/2$ is positive definite), the results of Theorem 3.2 and of Theorem 3.3 hold for the GCR method.

3.3 GMRESR

Examining the GCR method, we conclude that in step 4 (Alg. 3.3) the choice of u_k can be modified without affecting the rest of the algorithm, and replacing r_{k-1} with any other vector leads to a method that solves the minimization problem (3.7) with a modified range(U_k), where range$(U_k) \neq \mathcal{K}_k(A, r_0)$. The better the choice of u_k is, the faster the method will converge. The optimal choice would be $u_k = e_{k-1}$ (in this case we would retrieve the exact solution), where e_{k-1} is the error vector $e_{k-1} = x_{k-1} - x^*$. Therefore, a reasonable approach is to find the best possible approximation to the error e_{k-1}, when working on the residual equation

$$Ae_{k-1} = r_{k-1}. \tag{3.10}$$

One such approach is the recursive GMRES method (GMRESR) [VV94], see Algorithm 3.4. It consists of an inner and an outer method. The outer method

is GCR which computes the optimal approximation over a given set of direction vectors such that the residual is minimized. The inner method is GMRES, which computes an approximation to the solution of the residual equation (3.10), with initial guess 0. This gives the new direction vector needed in the outer loop.

Algorithm 3.4: GMRESR

Input : $A \in \mathbb{C}^{n\times n}$ system matrix
$b \in \mathbb{C}^n$, $x_0 \in \mathbb{C}^n$ right-hand side and initial guess
$m \in \mathbb{N}$ restart length

Output: approximate solution x_k to $Ax = b$

1 $r_0 = b - Ax_0$, $k = 1$
2 **while** $\|r_{k-1}\|_2 > tol$ **do**
3 $\quad u_k = p_m(A)r_{k-1}, c_k = Au_k$ // p_m - GMRES polynomial
4 $\quad$ **for** $i = 1, \ldots, k+1$ **do**
5 $\quad\quad \alpha_i = c_i^H c_k$
6 $\quad\quad c_k = c_k - \alpha_i c_i$
7 $\quad\quad u_k = u_k - \alpha_i u_i$
8 $\quad c_k = c_k / \|c_k\|_2 , u_k = u_k / \|c_k\|_2$
9 $\quad x_k = x_{k-1} + u_k c_k^H r_{k-1}$
10 $\quad r_k = r_{k-1} - c_k c_k^H r_{k-1}$
11 $\quad k = k + 1$

The convergence criteria and costs of the method will not be discussed here (see, e.g., [VV94]), as in this thesis we consider the GMRESR method only as a step between GCR and the GCRO method, which is the topic of the next section. Instead, we review some disadvantages of GMRESR (see, e.g., [Stu96a]). The main flaw of the method is that it solves the "wrong" minimization problem, which leads to suboptimal corrections to the solution. In the k-th step of GMRESR, in the inner loop GMRES solves (line 3)

$$\min_{y\in\mathbb{C}^m} \|r_{k-1} - AV_m y\|_2 , \tag{3.11}$$

with the columns of V_m spanning a basis of $\mathcal{K}_m(A, r_{k-1})$ and in the outer loop (lines $8-10$) we set

$$\begin{aligned} u_k &= (V_m y - U_{k-1}C_{k-1}^H AV_m y)/\left\|(I - C_{k-1}C_{k-1}^H)AV_m y\right\|_2 \\ c_k &= (I - C_{k-1}C_{k-1}^H)AV_m y/\left\|(I - C_{k-1}C_{k-1}^H)AV_m y\right\|_2 \\ x_k &= x_{k-1} + u_k c_k^H r_{k-1} \\ r_k &= r_{k-1} - c_k c_k^H r_{k-1}. \end{aligned}$$

In the outer loop, we compute the new vector c_k such that it is orthogonal to range(C_{k-1}). These orthogonalization relations are not taken into account by the inner method. It turns out that, in order to have optimal corrections (see Theorem 3.4 of Section 3.4), instead of (3.11) the following problem should be solved

$$\min_{y\in\mathbb{C}^m} \left\| r_{k-1} - (I - C_{k-1}C_{k-1}^H)AV_m y \right\|_2 = \min_{y\in\mathbb{C}^m} \left\| (I - C_{k-1}C_{k-1}^H)(r_{k-1} - AV_m y) \right\|_2 .$$

Another disadvantage is that we search the whole Krylov subspace $\mathcal{K}(A, r_{k-1}) = \text{span}\{r_{k-1}, Ar_{k-1}, A^2 r_{k-1}, ...\}$ for an approximation to the error e_{k-1}, whereas

$$e_{k-1} \in \mathcal{K}(A, r_{k-1}) \cap A^{-1}\,\text{range}(C_{k-1})^{\perp}, \tag{3.12}$$

and therefore

$$r_{k-1} \in \mathcal{K}(A, Ar_{k-1}) \cap \text{range}(C_{k-1})^{\perp}.$$

In addition, since only m steps of GMRES are performed in the inner method (not full GMRES), GMRESR can inherit the convergence problems of restarted GMRES. These disadvantages lead to the conclusion that it can be useful to keep the GCR orthogonality relations within the inner method.

3.4 GCRO

The GCRO (GCR with inner orthogonalization) method [Stu96a] is a modification of the GMRESR method, which overcomes some of the disadvantages of GMRESR. It consists of the inner method (GMRES) which computes an approximation to the error in the outer method and the outer method (GCR) which computes the information used to speed up the convergence of the inner method.

Following the discussion on disadvantages of the GMRESR method in the previous section, one concludes that it can be useful to keep the GCR orthogonality relations in the inner method. This is done by using

$$A_{C_k} = (I - C_k C_k^H)A \tag{3.13}$$

instead of A as the operator in GMRES. This way, we compute corrections to the residual $c_{k+1} \in \mathcal{K}(A, Ar_k) \cap \text{range}(C_k)^{\perp}$, that correspond to corrections to the approximation

$$u_{k+1} = A^{-1}c_{k+1} \in \mathcal{K}(A, r_k) \cap A^{-1}\,\text{range}(C_k)^{\perp},$$

which coincides with (3.12). Introducing the inverse of A will not bring up issues, since over the space span $\left\{A_{C_k} r_k, A_{C_k}^2 r_k, ...\right\}$ the inverse is available

$$\begin{aligned} A^{-1}A_{C_k} &= A^{-1}(I - C_k C_k^H)A \\ &= I - A^{-1}C_k C_k^H A \\ &= I - U_k C_k^H A. \end{aligned}$$

A straightforward implementation of the method involves many vector updates, innerproducts and vector scalings. Therefore, a set of formulae is developed to reduce the work. Before we proceed with the implementation details, we give an important result, which is necessary for later work.

Theorem 3.4. *Let U_k and C_k be defined as in* (3.4), (3.5) *and* (3.6) *and x_k, r_k and A_{C_k} be defined as in* (3.8), (3.9) *and* (3.13), *respectively. Furthermore, let $\{r_k, A_{C_k} r_k, \ldots, A_{C_k}^m r_k\}$ be linearly independent and $\{v_1, \ldots, v_{m+1}\}$ be an orthonormal basis for $\mathcal{K}_{m+1}(A_{C_k}, r_k)$, with $v_1 = r_k / \|r_k\|_2$, generated by m steps of the Arnoldi process. This defines the Arnoldi relation $A_{C_k} V_m = V_{m+1}\bar{H}_m$. Let y be defined by*

$$y = \underset{y_m \in \mathbb{C}^m}{\operatorname{argmin}} \left\| r_k - A_{C_k} V_m y_m \right\|_2 = \underset{y_m \in \mathbb{C}^m}{\operatorname{argmin}} \left\| r_k - V_{m+1}\bar{H}_m y_m \right\|_2 . \tag{3.14}$$

Then, $r_k - V_{m+1}\bar{H}_m y_m$ is the residual of the approximate solution $A^{-1}A_{C_k}V_m y$ for the equation $Az = r_k$, and we put

$$x_{k+1} = x_k + A^{-1}A_{C_k}V_m y. \tag{3.15}$$

This x_{k+1} is also the solution to the global minimization problem

$$x_{\text{opt}} = \underset{\tilde{x} \in \text{range}(U_k) \oplus \text{range}(V_m)}{\operatorname{argmin}} \left\| r_0 - A\tilde{x} \right\|_2 .$$

Proof. Writing $x_{\text{opt}} = U_k z + V_m y$ we look at the solution of the minimization problem

$$(z, y) = \underset{z \in \mathbb{C}^k, y \in \mathbb{C}^m}{\operatorname{argmin}} \left\| r_0 - AU_k z - AV_m y \right\|_2 . \tag{3.16}$$

Using the Arnoldi relation $A_{C_k}V_m = V_{m+1}\bar{H}_m$, which can be rewriten as $AV_m = C_k C_k^H AV_m + V_{m+1}\bar{H}_m$, it follows that the minimization problem (3.16) is equivalent to

$$(z, y) = \underset{z \in \mathbb{C}^k, y \in \mathbb{C}^m}{\operatorname{argmin}} \left\| r_0 - C_k(z + C_k^H AV_m y) - V_{m+1}\bar{H}_m y \right\|_2 . \tag{3.17}$$

Writing $r_0 = (I - C_k C_k^H) r_0 + C_k C_k^H r_0$, and due to the fact that range$(C_k) \perp$ range(V_{m+1}), (3.17) can be solved by two separate minimizations:

$$z + C_k^H A V_m y = C_k^H r_0 \Leftrightarrow z = C_k^H (r_0 - A V_m y) \tag{3.18}$$

and

$$y = \underset{y_m \in \mathbb{C}^m}{\operatorname{argmin}} \left\| (I - C_k C_k^H) r_0 - V_{m+1} \bar{H}_m y_m \right\|_2 = \underset{y_m \in \mathbb{C}^m}{\operatorname{argmin}} \left\| r_k - V_{m+1} \bar{H}_m y_m \right\|_2 . \tag{3.19}$$

This results in

$$x_{\text{opt}} = U_k z + V_m y,$$

where y is defined by (3.19), which is equivalent to (3.14) and z is defined by (3.18). As we can see from (3.18), the minimization problem for z is solved directly, meaning that these components of the residual are annihilated. For x_{k+1} we have from (3.15)

$$\begin{aligned} x_{k+1} &= x_k + A^{-1} A_{C_k} V_m y \\ &= x_k + A^{-1} (I - C_k C_k^H) A V_m y \\ &= U_k C_k^H r_0 + V_m y - U_k C_k^H A V_m y \\ &= U_k C_k^H (r_0 - A V_m y) + V_m y \\ &= U_k z + V_m y, \end{aligned}$$

which completes the proof.

□

The consequence of this theorem is that the residual of the outer GCR method is equal to the residual of the inner GMRES method

$$r_{k+1} = b - A x_{k+1} = b - A x_k - A_{C_k} V_m y = r_k - A_{C_k} V_m y = r_k - V_{m+1} \bar{H}_m y = r_m^{\text{inner}},$$

and therefore, needs not be computed in the outer method.

Next, we proceed with the implementation details, where one has to use a diagonal scaling for the sake of numerical stability, see [Stu96a]. Instead of the notation used so far, we introduce the new notation $\hat{U}_k, \bar{C}_k, N_k, Z_k$ and d_k:

$$\begin{aligned} C_k &= \bar{C}_k N_k, \\ N_k &= \operatorname{diag}\left(\|\bar{c}_1\|_2^{-1}, \|\bar{c}_2\|_2^{-1}, \ldots, \|\bar{c}_k\|_2^{-1} \right), \\ \hat{U}_k &= U_k N_k^{-1} Z_k, \quad \text{such that } A\hat{U}_k = \bar{C}_k Z_k, \\ r_k &= r_0 - \bar{C}_k d_k, \end{aligned} \tag{3.20}$$

where Z_k is assumed to be upper-triangular. Following the new notation, the approximate solution can be represented as

$$x_k = x_0 + \hat{U}_k Z_k^{-1} d_k, \tag{3.21}$$

which allows us to update it at the end of the complete iteration. In the inner GMRES method, the vectors Av_i are first orthogonalized against $\bar{C}_k$, thus constructing V_{m+1} such that $\bar{C}_k^H V_{m+1} = 0$ and

$$AV_m = \bar{C}_k B_m + V_{m+1}\bar{H}_m,$$

where $B_m = N_k^2 \bar{C}_k^H A V_m$. Next, y is computed using (3.14). Setting

$$\begin{aligned} \bar{c}_{k+1} &= V_{m+1}\bar{H}_m y \\ \hat{u}_{k+1} &= V_m y, \end{aligned} \tag{3.22}$$

we obtain

$$A\hat{u}_{k+1} = AV_m y = \bar{C}_k B_m y + V_{m+1}\bar{H}_m y = \bar{C}_k B_m y + \bar{c}_{k+1}.$$

Choosing $z_{k+1} = \begin{bmatrix} B_m y \\ 1 \end{bmatrix}$, the relation (3.20) is satisfied. The new residual of the outer iteration is given by

$$r_{k+1} = r_k - \bar{c}_{k+1},$$

thus concluding that $d_{k+1} = \begin{bmatrix} d_k \\ 1 \end{bmatrix}$. Replacing the residual of the outer iteration with the residual of the inner method (Theorem 3.4), i.e.

$$r_{k+1} = r_m^{\text{inner}} = r_k - V_{m+1}\bar{H}_m y,$$

we acquire an important relation

$$\bar{c}_{k+1} = r_k - r_m^{\text{inner}}$$

for updating $\bar{c}_{k+1}$ (instead of using (3.22)) in the outer method. The only part, left in the outer method, is to compute the new coefficient $\|\bar{c}_{k+1}\|_2^{-1}$ and to update the solution at the end using (3.21). We end up obtaining the GCRO method, as described in Algorithm 3.5.

In line 3 of Algorithm 3.5 we refer to the inner method, i.e. m steps of GMRES with the operator A_{C_k}, which we also present as Algorithm 3.6. It is worth

Algorithm 3.5: GCRO

Input : $A \in \mathbb{C}^{n\times n}$ system matrix
$b \in \mathbb{C}^n$, $x_0 \in \mathbb{C}^n$ right-hand side and initial guess
$m \in \mathbb{N}$ restart length

Output: approximate solution x_k to $Ax = b$

$r_0 = b - Ax_0$, $k = 0$
while $\|r_k\|_2 > tol$ **do**
  call gmres($\rightarrow \hat{u}_{k+1}, r^{\text{inner}}, \|r^{\text{inner}}\|_2, Z_{1\ldots k,k+1}$)
  $\bar{c}_{k+1} = r_k - r^{\text{inner}}$
  $(N_{k+1})_{k+1} = (\|r_k\|_2^2 - \|r^{\text{inner}}\|_2^2)^{1/2}$
  $Z_{k+1,k+1} = 1$
  $d_{k+1} = 1$
  $r_{k+1} = r^{\text{inner}}$
  $k = k + 1$
$x_k = x_0 + \hat{U}_k Z_k^{-1} d_k$

mentioning that different methods (e.g., BICGSTAB) could be used as the inner method, see, e.g., [Stu96a].

As described, the GCRO method finds an optimal approximate solution over the space spanned by both the inner and the outer iteration vectors. But, it also introduces a possible problem, when iterating with the singular operator $A_{C_k} = (I - C_k C_k^H)A$ in the inner GMRES method. GMRES computes the minimal residual solution over the generated Krylov subspace, but the generation of the subspace may break down in case of a singular operator. The following example illustrates that. Let us assume that the matrix A_{C_k} is defined as

$$A_{C_k} = (e_2, e_3, e_4, 0) = (I - e_1 e_1^H)(e_2, e_3, e_4, e_1),$$

where $C_k = e_1$, and e_i is the i-th unit vector. Let us further assume that we want to solve $A_{C_k} x = e_3$. The solution of this system of equations exists, $x = e_2$. But, GMRES with 0 initial guess will not find it, because a breakdown of the Krylov space occurs, i.e.

$$\text{span}\left\{e_3, A_{C_k} e_3, A_{C_k}^2 e_3, \ldots\right\} = \text{span}\left\{e_3, e_4, 0, 0, \ldots\right\},$$

and the solution is not contained in it. Next, we give assumptions under which a breakdown occurs and show that such a breakdown is rare, since it only happens after a certain (usually large) number of iterations. First of all, we define a breakdown of the Krylov subspace in the inner GMRES method with the singular matrix A_{C_k}.

Algorithm 3.6: GMRES WITH A_{C_k} (m STEPS AFTER k OUTER ITERATIONS)

Input : $A \in \mathbb{C}^{n\times n}$ system matrix
$C_k = [c_1, \dots, c_k] \in \mathbb{C}^{n\times k}$ matrix of direction vectors
r_k residual

Output: $\hat{u}_{k+1}, r^{\text{inner}}, Z_{1\dots k,k+1}$

$v_1 = r_k / \|r_k\|_2$
for $j = 1, \dots, m$ **do**
  $v_{j+1} = Av_j$
  for $i = 1, \dots, k$ **do**
    $b_{i,j} = c_i^H v_{j+1}$
    $v_{j+1} = v_{j+1} - b_{i,j} c_i$
  for $i = 1, \dots, j$ **do**
    $h_{i,j} = v_i^H v_{j+1}$
    $v_{j+1} = v_{j+1} - h_{i,j} v_i$
  $h_{j+1,j} = \|v_{j+1}\|_2$
  $v_{j+1} = h_{j+1,j}^{-1} v_{j+1}$
Solve $\min_{y_m} \left\| \|r_k\|_2 e_1 - \bar{H}_m y_m \right\|_2$
$\hat{u}_{k+1} = V_m y_m$
$r^{\text{inner}} = r_k - V_{m+1} \bar{H}_m y_m$
$Z_{1\dots k,k+1} = B_m y_m$

Definition 3.5. *Let U_k, C_k, A_{C_k} and r_k be defined as in* (3.4), (3.5), (3.6), (3.13) *and* (3.9). *Let $\{r_k, A_{C_k} r_k, \dots, A_{C_k}^{m-1} r_k\}$ be linearly independent and $\{v_1, \dots, v_m\}$ be an orthonormal basis for $\mathcal{K}_m(A_{C_k}, r_k)$, with $v_1 = r_k / \|r_k\|_2$, generated by $m-1$ steps of GMRES. We say to have a breakdown of the Krylov subspace if $A_{C_k} v_m \in \text{range}(V_m)$, since this implies we can no longer expand the Krylov subspace. We call it a* ***lucky breakdown*** *if $v_1 \in \text{range}(A_{C_k} V_m)$, because we have then found the solution to $A_{C_k} y = v_1$, where $y \in \mathcal{K}_m(A_{C_k}, r_k)$. We call it a* ***true breakdown*** *if $v_1 \notin \text{range}(A_{C_k} V_m)$, because then the solution is not contained in the Krylov subspace.*

The following theorems we give without proofs. For proofs and further comments, as well as discussions on how to proceed with the GCRO method if a breakdown occurs, consult [Stu96a].

Theorem 3.6. *Let U_k, C_k, A_{C_k} and r_k be defined as in* (3.4), (3.5), (3.6), (3.13) *and* (3.9). *Let $\{r_k, A_{C_k} r_k, \dots, A_{C_k}^{m-1} r_k\}$ be linearly independent and $\{v_1, \dots, v_i\}$ be an orthonormal basis for $\mathcal{K}_i(A_{C_k}, r_k)$, $i = 1, \dots, m$, with $v_1 = r_k / \|r_k\|_2$, generated by $m-1$ steps of GMRES. Then at step m:*

1. *A true breakdown occurs if and only if* $\text{range}(A_{C_k} V_{m-1})$ *is an invariant subspace of* A_{C_k}.
2. *A true breakdown occurs if and only if* $A_{C_k} v_m \in \text{range}(A_{C_k} V_{m-1})$.
3. *A breakdown occurs if and only if there exists an upper Hessenberg matrix* $H_m \in \mathbb{C}^{m \times m}$ *such that* $A_{C_k} V_m = V_m H_m$*; furthermore it is a true breakdown if and only if* H_m *is singular.*

Theorem 3.7. *Let* U_k, C_k, A_{C_k} *and* r_k *be defined as in* (3.4), (3.5), (3.6), (3.13) *and* (3.9). *Let* $\{r_k, A_{C_k} r_k, \ldots, A_{C_k}^{m-1} r_k\}$ *be independent and* $\{v_1, \ldots, v_i\}$ *be an orthonormal basis for* $\mathcal{K}_i(A_{C_k}, r_k)$, $i = 1, \ldots, m$, *with* $v_1 = r_k / \|r_k\|_2$, *generated by* $m-1$ *steps of GMRES. A true breakdown occurs at step* m *if and only if*

$$\exists u \neq 0 \in \text{range}(V_m) : u \in \text{range}(U_k).$$

The next theorem indicates that a breakdown cannot occur before the total number of iterations exceeds the dimension of the Krylov subspace $\mathcal{K}(A, r_0)$, hence meaning that in practice, a breakdown will be rare.

Theorem 3.8. *Let* U_k, C_k, A_{C_k} *and* r_k *be defined as in* (3.4), (3.5), (3.6), (3.13) *and* (3.9). *Let* m *be the grade of* r_0 *w.r.t.* A. *Define* $P_s(A) r_0 = \sum_{i=0}^{s} \gamma_i A^i r_0$, *with* $\gamma_s \neq 0$, *and let*

$$\begin{aligned} u_i &= P_{l_i - 1}(A) r_0, \quad i = 1, \ldots, k \\ l &= \max_{i=1,\ldots,k} l_i < m. \end{aligned}$$

Then $r_k = (I - C_k C_k^H) r_0 = P_l(A) r_0$. *We call* l *the total number of iterations. Then, if* $j < m - l$, *the set* $\{r_k, A_{C_k} r_k, \ldots, A_{C_k}^j r_k\}$ *is independent, and therefore no breakdown occurs in the first* j *steps of GMRES.*

The GCRO method (as well as GMRESR) is preferably used in the truncated version, see [Stu96a]. In the next chapter, we will introduce the concept of *optimal truncation*, which will, together with the framework of the GCRO method, result in the GCROT method [Stu99], which can be modified for solving our original problem, i.e. a sequence of linear systems.

3.5 Augmented and Deflated Krylov Subspace Methods

We have discussed some of the flaws of the restarted GMRES method [SS86] in Section 3.1.1. Most importantly, the method might converge significantly slower than full GMRES or even stagnate in some cases. The obvious reason is that at the time of restart all information built up in the previous cycle is discarded. Many methods have been proposed since, with the same goal of improving the performance of GMRES(m). Two main categories of methods rise above all: *hybrid iterative methods* and *acceleration techniques*. Hybrid methods combine already existing methods with the goal of maintaining the advantages of these methods while equilibrating their disadvantages. Hybrid methods still extract their iterates from the standard Krylov subspace. An overview of this class of methods is given in [NRT92]. Methods in the remainder of this thesis take GMRES(m) as their basis and aim at improving its convergence. They belong to the class of acceleration techniques and work with "enriched" Krylov subspaces.

One technique is based on the fact that the approximation space should ideally contain the approximation to the error $e_k = x_k - x^*$. We have already introduced methods based on this technique: both, GMRESR [VV94] and GCRO [Stu96a], approximately solve the residual equation (3.10) in the inner method, and this approximate solution then becomes the next direction for the outer approximation space.

Another acceleration technique is *augmentation*. Throughout this thesis, augmentation will always come along together with *deflation*, and understanding these two concepts is crucial for apprehending the methods to be discussed. Therefore, we dedicate Section 3.5.1 to motivating concepts of augmentation and deflation, as well as providing some basic analysis and convergence results.

There is a huge variety of methods that are based on these acceleration techniques. Within this chapter, we will address three of these methods, namely GMRES-E [Mor95], GMRES-DR [Mor02] and LGMRES [BJM05].

3.5.1 Augmentation and Deflation

Augmented methods tend to improve the convergence behaviour of GMRES(m) by enlarging the search space in the current cycle with a nearly A-invariant subspace $\mathbb{U}$ kept from the previous cycle. Hence, the approximate solution is extracted from the subspace

$$x_m \in x_0 + \mathcal{K}_m + \mathbb{U}.$$

Naturaly, two questions arise: which subspace should be kept? And in which way should it be used? Although a few different possibilities have been proposed, the common choice is to take $\mathbb{U}$ as a space spanned by approximate eigenvectors corresponding to the eigenvalues of the smallest magnitude.

Deflating, in our work, simply means removing the "problematic" eigenvalues. In general, deflated methods can exploit the approximate eigenvectors in two ways. The first way is to solve the projected system

$$PAX = Pb,$$

where the projector P usually carries some spectral information (e.g., $P = I - vv^H$, where v is a normalized eigenvector). The second way is to add the desired eigenvectors to the Krylov subspace. We use the second way, which by construction shows that augmentation and deflation are correlated naturally.

Next, we motivate the choice of approximate eigenvectors for augmentation. In Section 3.1, we gave general convergence results for GMRES, bounding the norm of the residuals. Assuming that the matrix of the system is diagonalizable, $A = V\Lambda V^{-1}$, and has real and positive eigenvalues, from (3.3), using scaled Chebyshev polynomials, one obtains the following bound for the GMRES residuals (see, e.g., [Mor95])

$$\frac{\|r_m\|}{\|r_0\|} \leq 2\kappa(V)\left(1 - \frac{2}{\sqrt{\hat{\kappa}}+1}\right)^m, \tag{3.23}$$

where $\hat{\kappa} = \frac{\lambda_n}{\lambda_1}$ is an additional condition number, the ratio of largest to smallest eigenvalues, and it is not necessarily equal to the standard condition number $\kappa(V) = \|V\|_2 \cdot \|V^{-1}\|_2$. (3.23) resembles the standard bound for CG residuals when A is hpd. Hence, the convergence is slow if there is a small eigenvalue. In addition, the distribution of the other eigenvalues also influences convergence. Having that in mind, the main goal of deflated methods is to remove the eigenvalues of smallest magnitude, that hinder convergence the most. This is done by augmenting the search subspace in the current cycle with (approximate) eigenvectors computed in the previous cycle. The motivation for this is that when a converged eigenvector is added to the subspace the corresponding eigenvalue is effectively deflated and convergence proceeds according to the modified spectrum. The following result demonstrates this for the case of real and positive eigenvalues, see [Mor95].

Theorem 3.9. *Suppose A has spectral decomposition $A = V\Lambda V^{-1}$, with all the eigenvalues being real and positive and thus $Av_i = \lambda_i v_i$ with $0 < \lambda_1 \leq ... \leq \lambda_n$. Assume that the minimum residual solution is extracted from the subspace*

span $\{r_0, Ar_0, \dots, A^{m-1}r_0, v_1, \dots, v_k\}$. *Then*

$$\frac{\|r_m\|}{\|r_0\|} \leq 2\kappa(V)\left(1 - \frac{2}{\sqrt{\hat{\kappa}_e} + 1}\right)^m,$$

where $\hat{\kappa}_e = \frac{\lambda_n}{\lambda_{k+1}}$ *is the "effective condition number".*

In order to substantiate our claims, let us consider the example of a matrix whose eigenvalues are given as $1, 2, \dots, n$, which is similar to the spectrum of some matrices from real life applications (e.g., see the model problem from finite difference discretization of Poisson's equation [HY12]). Let us further assume that we deflate k eigenvalues. Then the convergence bound improves from

$$\frac{\|r_m\|}{\|r_0\|} \leq 2\kappa(V)\left(1 - \frac{2}{\sqrt{n} + 1}\right)^m = 2\kappa(V)C_1^m$$

to

$$\frac{\|r_m\|}{\|r_0\|} \leq 2\kappa(V)\left(1 - \frac{2}{\sqrt{\frac{n}{k+1}} + 1}\right)^m = 2\kappa(V)C_2^m.$$

We can roughly compare convergence by comparing the number of steps m needed for convergence. We usually want that $C^m \leq \varepsilon$, for some small ε. This means that without adding the eigenvectors to the subspace we have

$$m_1 \approx \frac{\log \varepsilon}{\log C_1},$$

whereras when we add k eigenvectors to the search space we obtain

$$m_2 \approx \frac{\log \varepsilon}{\log C_2}.$$

Furthermore,

$$\frac{m_1}{m_2} \approx \frac{\log C_2}{\log C_1} = \frac{\log\left(1 - \frac{2}{\sqrt{\frac{n}{k+1}}+1}\right)}{\log\left(1 - \frac{2}{\sqrt{n}+1}\right)} \approx \frac{\frac{2}{\sqrt{\frac{n}{k+1}}+1}}{\frac{2}{\sqrt{n}+1}} = \frac{\sqrt{n}+1}{\sqrt{\frac{n}{k+1}}+1} \approx \frac{\sqrt{n}}{\sqrt{\frac{n}{k+1}}} = \sqrt{k+1}.$$

This means that convergence is roughly $\sqrt{k+1}$ times faster when we add k eigenvectors to the subspace. To be fair, we should mention that adding eigenvectors to the search space is not always beneficial, e.g., if there are no small eigenvalues or if the distribution of the eigenvalues is not favorable.

The approximate eigenvectors can be beneficial even before they become very accurate, which is summarized in the following theorem for one eigenvector, see [Mor95].

Theorem 3.10. *Suppose A has spectral decomposition $A = V\Lambda V^{-1}$. Suppose we augment the search space with one approximate eigenvector y_1. Let $\psi = \angle(y_1, v_1)$, and let β_1 be the coefficient of v_1 in the expansion $r_0 = \sum_{i=1}^{n} \beta_i v_i$. Then*

$$\|r_m\| \leq \kappa(V) \max_{i \neq 1} |q(\lambda_i)| \, \|r_0\| + \frac{\|A\|}{\lambda_1} \tan(\psi) |q(\lambda_1)| |\beta_1|,$$

for any $q \in \overline{\Pi}_m$.

The second term occurs because of the inaccuracy of the approximate eigenvector. This term will not be significant, as long as the accuracy of the approximate eigenvector is greater than the amount of improvement brought by q.

Results in this subsection lead to the conclusion that it is reasonable to keep the approximate eigenvectors, built up in the previous cycle, corresponding to the eigenvalues of the smallest magnitude. Next, we proceed with a detailed description of the methods based on augmentation and deflation, which differ in the choice of the vectors that are kept and in the way they are used. Some general theoretical results are summarized in [CS97] and [Saa97].

3.5.2 GMRES with Eigenvectors

GMRES with eigenvectors (GMRES-E) [Mor95] is a method that is based on the augmentation approach. As it was shown, deflating some of the smallest eigenvalues can greatly improve the convergence of the method. In GMRES-E, the harmonic Ritz vectors from the previous cycle are chosen for the augmentation as they are good approximations to the wanted eigenvectors.

Let us assume that we want to deflate k eigenvalues and that we fix the overall dimension of the augmented subspace m. GMRES-E appends the harmonic Ritz vectors $\tilde{Y}_k = [\tilde{y}_1, \ldots, \tilde{y}_k]$ after the $m - k$ steps of the Arnoldi process leading to an Arnoldi-like recurrence

$$AW_m = V_{m+1}\bar{H}_m,$$

where $W_m = [V_{m-k} \quad \tilde{Y}_k]$. The first $m - k + 1$ columns of the matrix V_{m+1} (as well as the first $m - k$ columns of W_m) are the orthonormalized Arnoldi vectors, whereas the last k columns are formed by orthogonalizing the vectors $A\tilde{y}_i, i = 1, \ldots, k$ against the first $m-k+i$ columns of V_{m+1}. We should mention that, in the first cycle, m steps of standard GMRES are performed, since

the harmonic Ritz vectors are not available (this will also hold for GMRES-DR [Mor02] in Section 3.5.3 and LGMRES [BJM05] in Section 3.5.4). The algorithm is presented as Algorithm 3.7.

Algorithm 3.7: GMRES-E

Input : $A \in \mathbb{C}^{n\times n}$ system matrix
$b \in \mathbb{C}^n$, $x_0 \in \mathbb{C}^n$ right-hand side and initial guess
$m \in \mathbb{N}$ restart length
$k \in \mathbb{N}$ number of deflated ev's

Output: approximate solution x_m to $Ax = b$

1 $r_0 = b - Ax_0$, $\beta = \|r_0\|$, $v_1 = r_0/\beta$
2 **for** $j = 1, \dots, m$ **do**
3 **if** $j \leq m - k$ **then**
4 $w_j = v_j$
5 **else**
6 $w_j = \tilde{y}_{j-m+k}$
7 $z = Aw_j$
8 **for** $i = 1, \dots, j$ **do**
9 $h_{i,j} = \langle z, v_i \rangle$
10 $z = z - h_{i,j} v_i$
11 $h_{j+1,j} = \|z\|$, $v_{j+1} = z/h_{j+1,j}$
12 Define $W_m = [w_1, \dots, w_m]$
13 $x_m = x_0 + W_m y_m$, where $y_m = \operatorname{argmin}_y \|\beta e_1 - \bar{H}_m y\|$
14 **if** *Satisfied* **then**
15 Stop
16 **else**
17 Compute k harmonic Ritz vectors $\tilde{y}_1, \dots, \tilde{y}_k$ of A w.r.t. the subspace AW_m
18 Set $x_0 = x_m$ and go to 1

The augmented subspace built with GMRES-E is of the form

$$\operatorname{span}\left\{r_0, Ar_0, \dots, A^{m-k-1}r_0, \tilde{y}_1, \dots, \tilde{y}_k\right\}. \tag{3.24}$$

Chapman and Saad observed in [CS97] that GMRES-E works better with harmonic Ritz vectors than with Ritz vectors. This is due to the fact that the subspace (3.24) is itself a Krylov subspace, but with a different starting vector (see Section 3.5.3). It seems that the method (as well as GMRES-DR described in the next section) benefits from keeping the Krylov subspace structure.

In line 17 of Algorithm 3.7, one has to compute the harmonic Ritz vectors. From the proof of Lemma 2.27, we know that the harmonic Ritz pairs $(\frac{1}{\theta}, \tilde{y})$ can be computed by solving an m-by-m generalized eigenvalue problem

$$W_m^H A^H W_m \hat{y} = \theta W_m^H A^H A W_m \hat{y}.$$

The vectors $\hat{y}$ associated with the largest θ's are computed and the harmonic Ritz vectors belonging to the smallest harmonic Ritz values $1/\theta$'s are retrieved as $\tilde{y} = W_m \hat{y}$. This is known as the modified Rayleigh-Ritz procedure.

Methods from this chapter like GCR or GMRESR were interesting for our work, mainly for their framework. Similarly, we will not consider the GMRES-E method as a solver. We will rather use the framework of the method and combine it with some other ideas. Anyhow, it is worth mentioning that the method can be effective for certain problems with favorable distribution of the eigenvalues. In addition, using the fact that the method is based on the augmentation approach, since it appends the approximate eigenvectors after the Krylov subspace is built, GMRES-E can be modified for solving our main problem [Par+06], i.e. solving sequences of linear systems. More on this topic will be presented in the next chapter.

3.5.3 GMRES-DR

GMRES with deflated restarting (GMRES-DR) [Mor02] is a method mathematicaly equivalent to GMRES-E. Both methods build the same Krylov subspace at the end of the cycle, provided that they use the same harmonic Ritz vectors. The difference is that GMRES-E is based on the augmentation approach, meaning that the approximate eigenvectors are appended to the already built Krylov subspace, whereas GMRES-DR is based on the orthogonalization approach. To avoid confusion, we should say that GMRES-DR falls under the class of augmented and deflated methods. What differs is the way in which the approximate eigenvectors are used, and that is what is meant by the orthogonalization approach, which we now explain.

GMRES-DR uses a thick-restarting approach by Wu and Simon [WS98] for deflating the eigenvalues. The approach from [WS98] is modified for the non-symmetric case. The idea is to put the k orthonormalized harmonic Ritz vectors first in the subspace, and then to proceed with $m-k$ steps of the standard Arnoldi process (Alg. 2.1), thus leading to the Arnoldi-like relation

$$A[\tilde{\Upsilon}_k \quad V_{m-k}] = [\tilde{\Upsilon}_k \quad V_{m-k+1}]\bar{H}_m, \tag{3.25}$$

where $\tilde{\Upsilon}_k$ represents the k orthonormalized harmonic Ritz vectors, and $\bar{H}_m$ is upper-Hessenberg except for a leading $(k+1)\times(k+1)$ dense submatrix. The Arnoldi vectors, i.e. the columns of V_{m-k+1} formed this way are orthogonal to the harmonic Ritz vectors. Under the assumption that we have k distinct harmonic Ritz values, we present GMRES-DR as Algorithm 3.8 (see [Mor00] for a short discussion of the nondistinct case).

Algorithm 3.8: GMRES-DR

Input : $A \in \mathbb{C}^{n\times n}$ system matrix
$b \in \mathbb{C}^n$, $x_0 \in \mathbb{C}^n$ right-hand side and initial guess
$m \in \mathbb{N}$ restart length
$k \in \mathbb{N}$ number of deflated ev's

Output: approximate solution x_m to $Ax = b$

1 $r_0 = b - Ax_0$, $\beta = \|r_0\|$, $v_1 = r_0/\beta$
2 Perform m steps of GMRES (thus computing x_m and r_m)
3 Compute the k smallest eigenpairs $(1/\theta_i, \hat{y}_i)$ of $H_m + H_m^{-H} h_m h_m^H$
4 Orthonormalize $\hat{y}_i$'s, in order to form an m by k matrix P_k (real and imaginary parts are treated separately)
5 Form p_{k+1} by orthonormalizing the coefficients w.r.t. V_m of the GMRES residual, $\beta e_1 - \bar{H}_m y$, against previous columns of P_k (first we must append a row of zeros to P_k)
6 Form new V and H: $\bar{H}_k^{new} = P_{k+1}^H \bar{H}_m P_k$, $V_{k+1}^{new} = V_{m+1} P_{k+1}$ and set $\bar{H}_k = \bar{H}_k^{new}$ and $V_{k+1} = V_{k+1}^{new}$
7 Othogonalize v_{k+1} against previous columns of V_{k+1}
8 Perform $m - k$ steps of Arnoldi process to form the rest of V_{m+1} and $\bar{H}_m$, starting with $r_0 = r_m$ and $\beta = h_{m+1,m}$
9 Compute $x_m = x_0 + V_m d_m$, where $d_m = \operatorname{argmin}_d \|V_{m+1}^H r_0 - \bar{H}_m d\|_2$ and the corresponding residual
10 **if** *Satisfied* **then**
11 Stop
12 **else**
13 Compute the k smallest eigenpairs $(1/\theta_i, \hat{y}_i)$ of $H_m + H_m^{-H} h_m h_m^H$
14 Set $r_0 = r_m$ and go to 4

In line 7, reorthogonalization of the $(k+1)$-st vector is done for numerical reasons, see [Mor02]. Next, we allude to two essential parts of the method: computing the harmonic Ritz vectors, which is done in lines 3 and 13, and properties of the subspace built with GMRES-DR.

The following lemma shows how the harmonic Ritz pairs can be computed, provided that we have the Arnoldi-like relation (3.25).

Lemma 3.11. *The harmonic Ritz pairs of A with respect to the subspace $A\mathcal{K}$, where $\mathcal{K}$ is spanned by the columns of $\hat{V}_m$ from* (3.25)*, are given as $\left(1/\theta_i, A\hat{V}_m\hat{y}_i\right)$, where $(1/\theta_i, \hat{y}_i)$ are the eigenpairs of the matrix*

$$H_m + f_m h_m^H, \tag{3.26}$$

where $H_m^H f_m = h_m$ and $\bar{H}_m = \begin{bmatrix} H_m \\ h_m^H \end{bmatrix}$ is the Hessenberg matrix of the Arnoldi-like relation (3.25)*.*

Proof. Starting from Definition 2.24 and using the Arnoldi-like relation (3.25) with $\hat{V}_m = [\tilde{\Upsilon}_k \quad V_{m-k}]$ and $\hat{V}_{m+1} = [\tilde{\Upsilon}_k \quad V_{m-k+1}]$, we have that a harmonic Ritz pair $(\frac{1}{\theta}, \tilde{y})$ satisfies

$$(A\hat{V}_m)^H(A^{-1}\tilde{y} - \theta\tilde{y}) = 0 \quad \text{and} \quad \tilde{y} \in A\mathcal{K}.$$

Noticing that range$(V_{m-k}) \perp$ range$(\tilde{\Upsilon}_k)$ (by construction) it follows

$$\begin{aligned} \bar{H}_m^H \hat{V}_{m+1}^H A^{-1} A \hat{V}_m \hat{y} &= \theta \bar{H}_m^H \hat{V}_{m+1}^H A \hat{V}_m \hat{y} \\ \Leftrightarrow \bar{H}_m^H \hat{I}_m \hat{y} &= \theta \bar{H}_m^H \hat{V}_{m+1}^H \hat{V}_{m+1} \bar{H}_m \hat{y} \\ \Leftrightarrow \bar{H}_m^H \hat{I}_m \hat{y} &= \theta \bar{H}_m^H \bar{H}_m \hat{y}, \end{aligned}$$

where $\hat{I}_m = \begin{bmatrix} I_m \\ 0 \dots 0 \end{bmatrix}$. Writing $\bar{H}_m = \begin{bmatrix} H_m \\ h_m^H \end{bmatrix}$, we obtain

$$\begin{aligned} H_m^H \hat{y} &= \theta(H_m^H H_m + h_m h_m^H)\hat{y} \\ \frac{1}{\theta}\hat{y} &= (H_m + f_m h_m^H)\hat{y}, \end{aligned}$$

where $f_m = H_m^{-H} h_m$. Thus, $(\frac{1}{\theta}, \hat{y})$ is an eigenpair of the matrix $H_m + f_m h_m^H$. □

We see from Lemma 3.11, that we can compute the harmonic Ritz vectors in a cheap way, since we only have to solve the eigenvalue problem of small size m. The same result cannot be applied for GMRES-E, because the orthogonality of the Arnoldi vectors to the harmonic Ritz vectors, which is exploited within the proof of Lemma 3.11, does not hold for the GMRES-E method.

As mentioned in the previous section, choosing the harmonic Ritz vectors for the augmentation keeps the entire subspace used by GMRES-DR (GMRES-E) a Krylov subspace. To see this, we first give the following result, see [Mor02].

Theorem 3.12. *Suppose we have a subspace* $\mathcal{S} = \text{span}\{y_1, \dots, y_k, v\}$*, with the property that*

$$Ay_i - \alpha_i y_i = \gamma_i v, \tag{3.27}$$

for $\alpha_i, \dots, \alpha_k$ *distinct and for some nonzero* γ_i*'s. Then* $\mathcal{S}$ *is a Krylov subspace.*

Using the result of Theorem 3.12, we can prove the following statement.

Theorem 3.13. *The subspace used by GMRES-DR is the subspace*

$$\text{span}\{r_0, Ar_0, \dots, A^{m-k-1}r_0, \tilde{y}_1, \dots, \tilde{y}_k\}, \tag{3.28}$$

and it is a Krylov subspace.

Proof. The harmonic Ritz residual vectors are defined as $A\tilde{y}_i - \frac{1}{\theta_i}\tilde{y}_i$, and are multiples of the GMRES residual vector [Mor00]. Therefore, we obtain (3.27) with $v = r_0$ and with $1/\theta_i$ and $\tilde{y}_i$ instead of α_i and y_i. Using Theorem 3.12, it follows that span $\{\tilde{y}_1, \ldots, \tilde{y}_k, r_0\}$ is a Krylov subspace. Let $v_1, \ldots, v_{k+1}$ be the orthonormal basis for this subspace. GMRES-DR adds vectors to this basis with an Arnoldi process, so it clearly generates a Krylov subspace. Now, we will show that this subspace is (3.28). The vector v_{k+2} comes from orthonormalizing Av_{k+1}.

$$\begin{aligned} Av_{k+1} &= A(\text{a combination of } r_0 \text{ and } \tilde{y}_i's) \\ &= \text{a combination of } Ar_0 \text{ and } A\tilde{y}_i's \\ &= \text{a combination of } Ar_0, r_0 \text{ and } A\tilde{y}_i's \end{aligned}$$

using (3.27) on the last step. So

$$\text{span}\,\{v_1, \ldots, v_{k+2}\} = \text{span}\,\{\tilde{y}_1, \ldots, \tilde{y}_k, r_0, Ar_0\}\,.$$

Continuing in this way, we obtain the subspace (3.28). □

We mention that, moreover, the subspace (3.28) is equivalent to

$$\text{span}\,\{\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_k, A\tilde{y}_i, A^2\tilde{y}_i, \ldots, A^{m-k}\tilde{y}_i\}\,,$$

for $1 \leq i \leq k$, meaning that it contains Krylov subspaces with each of the harmonic Ritz vectors as starting vectors.

GMRES-DR *cannot* be adapted for solving sequences of linear systems, not even if the matrix does not change. This is due to the fact that the harmonic Ritz vectors $\tilde{y}_1, \ldots, \tilde{y}_k$ of A do not form a Krylov subspace for another matrix or even just another starting vector. In such situations, combining the GMRES-DR method with the GCRO method results in an effective solver GCRO-DR [Par+06], which reduces costs of subsequent *systems* by retaining some useful information from the previous system. This method will be covered in details in the next chapter.

3.5.4 Loose GMRES

We have discussed so far two methods which improve the convergence of GMRES(m) by augmenting the search space in the current cycle with the harmonic Ritz vectors computed in the previous cycle, thus deflating the eigenvalues of the smallest magnitude, which hamper the convergence. A somewhat similar idea was exploited in developing the loose GMRES (LGMRES) method [BJM05].

Another possibility is to augment the search space with k error approximations of the form

$$z_i = x_i - x_{i-1}. \tag{3.29}$$

The motivation for such a choice comes from the discussion regarding GMRESR (or GCRO). In Section 3.3, we have pointed out that an optimal choice for the next direction vector would be $u_{i+1} = e_i$, because in that case we would retrieve the exact solution. In addition, we notice that $z_i \in \mathcal{K}_m(A, r_{i-1})$, hence one could say that the i-th error approximation z_i represents the discarded space form the previous cycle. Following this discussion, it can be concluded that the last k error approximations of the form (3.29) make a good choice for augmentation.

LGMRES uses the framework of the GMRES-E method. It performs $m - k$ steps of the Arnoldi process (Alg. 2.1), and then it appends the k error approximations $Z_k = [z_1, \dots, z_k]$, leading to the Arnoldi-like relation

$$AW_m = V_{m+1}\bar{H}_m,$$

where $W_m = [V_{m-k} \quad Z_k]$. The first $m - k + 1$ columns of the matrix V_{m+1} (as well as the first $m - k$ columns of W_m) are the orthonormalized Arnoldi vectors, whereas the last k columns are formed by orthogonalizing the vectors $Az_i, i = 1, \dots, k$ against the previous $m - k + i$ columns of V_{m+1}. We end up obtaining LGMRES$(m - k, k)$, as described in Algorithm 3.9.

In Algorithm 3.9 we assume that all k error approximations are available, which is not always the case, i.e. in the first $i < k$ cycles, we have only i error approximations. The suggestion in [BJM05] for these cycles is to use $k - i$ additional Arnoldi vectors, hence the actual implementation is modified. In addition, in line 17, we state that upon a restart we only need to set $x_0 = x_m$, whereas we also have to take care of including the previously computed error approximation z_{k+1}, as well as removing z_1 used in this cycle, so that we still have the last k error approximations. The last, but not the least comment, is that in addition to the $m+1$ Arnoldi vectors, the vectors $z_1, \dots, z_k, Az_1, \dots, Az_k$ have to be stored, which increases storage requirements, but tests in [BJM05] (Section 4) show that the optimal values for k are typically small, i.e. $k \leq 3$.

We proceed with some useful properties of the LGMRES method [BJM05].

Theorem 3.14. *The error approximation vectors $z_j = x_j - x_{j-1}$ with which we augment the Krylov space in Algorithm 3.9 are $A^H A$-orthogonal.*

The result of Theorem 3.14 is not exploited in Algorithm 3.9 (within the **for** loop in line 7 we can reduce some computations), while for small k, the

Algorithm 3.9: LGMRES

Input : $A \in \mathbb{C}^{n\times n}$ system matrix
$b \in \mathbb{C}^n$, $x_0 \in \mathbb{C}^n$ right-hand side and initial guess
$m \in \mathbb{N}$ restart length
$k \in \mathbb{N}$ number of error approximations

Output: approximate solution x_m to $Ax = b$

1 $r_0 = b - Ax_0$, $\beta = \|r_0\|$, $v_1 = r_0/\beta$
2 **for** $j = 1, \ldots, m$ **do**
3 **if** $j \leq m - k$ **then**
4 $w = Av_j$
5 **else**
6 $w = Az_{j-m+k}$
7 **for** $i = 1, \ldots, j$ **do**
8 $h_{i,j} = \langle w, v_i \rangle$
9 $w = w - h_{i,j}v_i$
10 $h_{j+1,j} = \|w\|$, $v_{j+1} = w/h_{j+1,j}$
11 Define $W_m = [v_1, \ldots, v_{m-k}, z_1, \ldots, z_k]$, $V_{m+1} = [v_1, \ldots, v_{m+1}]$
12 $z_{k+1} = W_m y_m$, $Az_{k+1} = V_{m+1}\bar{H}_m y_m$, where $y_m = \operatorname{argmin}_y \|\beta e_1 - \bar{H}_m y\|$
13 $x_m = x_{m-1} + z_{k+1}$
14 **if** *Satisfied* **then**
15 Stop
16 **else**
17 Set $x_0 = x_m$ and go to step 1

improvement is negligible. In [BJM05] it serves as a part of a discussion regarding the resemblence of LGMRES with the full CG method with polynomial preconditioning.

It was observed in [BJM05] that the residual vectors at the end of each restarted GMRES cycle alternate direction in a cyclic fashion. In other words, every other residual points in nearly the same direction, i.e. $r_{i+1} \approx \alpha r_{i-1}$, thus slowing down the covergence. This behavior is observed also when the method converges rapidly, meaning that even in that case a faster convergence is possible if this fact is exploited for augmentation. Although it cannot be proved that such a behavior occurs in general (it was shown for $m = n-1$ in [BJM05]), it is reasonable to conclude that changing such a pattern would improve the convergence of the method. And that is exactly what the LGMRES method does.

If the two vectors point in nearly the same direction, this basically means that the angle between them is small. Next, we relate the angles between

residuals with the convergence behavior of GMRES(m) and LGMRES (see, e.g., [EE01] for more general results on how the angles between approximation and residual spaces determine the convergence of Krylov methods). Hence, the definition of *sequential* and *skip* angles.

Definition 3.15. *The angles between every two consecutive residual vectors, e.g.,* $\angle(r_{i+1}, r_i)$ *are referred to as* ***sequential*** *angles, and the angles between every other residual vectors, e.g.,* $\angle(r_{i+1}, r_{i-1})$ *as* ***skip*** *angles.*

The following result from [BJM05] is mathematically equivalent to a result given in [Sim00].

Theorem 3.16. *Let* r_{i+1} *and* r_i *be the residuals from GMRES restart cycles* $i+1$ *and* i*, respectively. Then the sequential angle between them is given by*

$$\cos(\angle(r_{i+1}, r_i)) = \frac{\|r_{i+1}\|_2}{\|r_i\|_2}.$$

The same result holds for the LGMRES method, see [BJM05].

Theorem 3.17. *Let* r_{i+1} *and* r_i *be the residuals from LGMRES restart cycles* $i+1$ *and* i*, respectively. Then the sequential angle between them is given by*

$$\cos(\angle(r_{i+1}, r_i)) = \frac{\|r_{i+1}\|_2}{\|r_i\|_2}. \tag{3.30}$$

The result of Theorem 3.16 explains the claim that small sequential angles lead to slow convergence. Of course, it would be optimal to find the next residual vector r_{i+1}, such that it is orthogonal to the current residual vector r_i, since by (3.30) this would infer $r_{i+1} = 0$, and we would have found the exact solution. Now we turn to skip angles, see [BJM05].

Theorem 3.18. *Let* r_{i+1} *and* r_{i-1} *be the residuals from GMRES restart cycles* $i+1$ *and* $i-1$*, respectively. Then the skip angle between them is given by*

$$\cos(\angle(r_{i+1}, r_{i-1})) = \frac{\|r_{i+1}\|_2}{\|r_{i-1}\|_2} - \frac{\langle A\delta_{i+1}, A\delta_i\rangle}{\|r_{i+1}\|_2 \|r_{i-1}\|_2}, \tag{3.31}$$

where $r_{i+1} = r_i - A\delta_{i+1}$ *and* $r_i = r_{i-1} - A\delta_i$*.*

Theorem 3.19. *Let* r_{i+1} *and* r_{i-1} *be the residuals from LGMRES restart cycles* $i+1$ *and* $i-1$*, respectively. Then the skip angle between them is given by*

$$\cos(\angle(r_{i+1}, r_{i-1})) = \frac{\|r_{i+1}\|_2}{\|r_{i-1}\|_2}. \tag{3.32}$$

The second term in (3.31) vanishes in (3.32), because the error approximations in LGMRES are $A^H A$-orthogonal by construction (Theorem 3.14). The result of Theorem 3.19 indicates that the pace of the iteration for LGMRES also correlates with the skip angles. In general, it can be proven that for any $0 \leq j \leq k$ and $i \geq k$

$$\cos(\angle(r_{i+1}, r_{i-j})) = \frac{\|r_{i+1}\|_2}{\|r_{i-j}\|_2}$$

for LGMRES.

We conclude this section by giving an example, that verifies the claims that when GMRES(m) experiences slow convergence, skip angles are small, whereas LGMRES prevents the alternating behavior, thus improving the convergence. The example (reproduced from [BJM05]) is given in Table 3.1. Four matrices, from the Matrix Market Collection, are considered: $add20$ ($n = 2395$), $orsirr\_1$ ($n = 1030$), $orsreg\_1$ ($n = 2205$) and $sherman\_1$ ($n = 1000$).

Table 3.1: Comparison of GMRES(30) and LGMRES(29, 1) for four test problems. We compare iterations required for $\|r_i\|_2 / \|r_0\|_2 \leq 10^{-9}$, median sequential angle (in degrees) and median skip angle (in degrees) for both methods.

Matrix	Iterations LGMRES (GMRES(m))	Median seq. angle LGMRES (GMRES(m))	Median skip angle LGMRES (GMRES(m))
$add20$	606 (1002)	63.0 (51.3)	79.0 (5.4)
$orsirr\_1$	2190 (6659)	41.0 (23.0)	55.4 (6.9)
$arsreg\_1$	515 (888)	72.2 (59.3)	84.6 (8.4)
$sherman\_1$	757 (3688)	61.7 (27.5)	76.4 (0.2)

3.6 Preconditioning

Preconditioning basically means transforming the original system into one with the same solution as the original system, but easier to solve with an iterative method. Deflation and augmentation techniques often work better when used with preconditioning. Therefore, we only give a brief introduction, since it

is not a significant part of this thesis, and show some numerical results in Section 5.5.1, so the reader gets a feeling of the power of preconditioning.

The main challenge in preconditioning is to find a nonsingular matrix M close to A in some sense, such that solving $Mx = b$ is not expensive. The last requirement is of outmost importance, since the iterative methods require the solution of the linear system with M at each step. Having the preconditioner M available, there are three different ways to apply it: *from the left*, *to the right* and *in a split form*. We introduce here the right preconditioning.

When the preconditioner is applied to the right, the change of variables $u = Mx$ is necessary, and we obtain the following system

$$AM^{-1}u = b. \tag{3.33}$$

For easier understanding, we proceed by using GMRES as an iterative solver. We will now show that u does not have to be computed explicitly. The initial residual $r_0 = b - AM^{-1}u_0$ does not actually require u_0 since $b - AM^{-1}u_0 = b - Ax_0$, and x_0 is the chosen starting vector. The GMRES approximate solution of (3.33) is given as

$$u_m = u_0 + \sum_{i=1}^{m} v_i y_i.$$

Multiplying the previous equation with M^{-1} we obtain

$$x_m = x_0 + M^{-1} \sum_{i=1}^{m} v_i y_i. \tag{3.34}$$

To summarize, from (3.33) and (3.34) it follows that the right preconditioned GMRES method differs from GMRES in two details: we build a Krylov subspace with the Arnoldi process with AM^{-1} and r_0 and we update the approximate solution in the following way $x_m = x_0 + M^{-1}(V_m y_m)$.

Right preconditioning allows the change of the preconditioner in each step. The idea is to split the computation of w in line 4 of the Arnoldi process (Algorithm 2.1) in two parts

$$\begin{aligned} s_k &= M_k^{-1} v_k \\ w &= A s_k, \end{aligned}$$

where we have a possibility to choose different M_k in each step. Then, storing the s_k we update the approximate solution as $x_m = x_0 + S_m y_m$.

The choice of the preconditioner, in general, depends on the problem. For this reason, we carry on with this topic in Chapter 5 only for the lattice QCD application for which we use a red-black multiplicative Schwarz method as a preconditioner.

4 Krylov Subspace Recycling

Many problems in computational science and engineering require the solution of a sequence of linear systems

$$A^{(i)}x^{(i)} = b^{(i)}, \quad A^{(i)} \in \mathbb{C}^{nxn}, \quad b^{(i)} \in \mathbb{C}^n, \quad i = 1, \ldots, m, \tag{4.1}$$

where the matrices $A^{(i)}$ and right-hand sides $b^{(i)}$ change slightly from one system to the next. Usually, at each step, the matrix and right-hand side depend on the previous solution and are not available simultaneously. In addition, the cumulative change might be significant, since one has to solve thousands of systems. Therefore, instead of solving each system separately, it would be beneficial to keep a judiciously selected subspace from the previous system, and use it to reduce the number of iterations for solving the next system. This process is known as *recycling*.

In the previous chapter we have introduced some methods that could be modified to solve (4.1). The GCRO method computes the optimal solution over the space spanned by the new and old search vectors. It is usually used in a truncated form. A straightforward truncation would mean keeping orthogonality only to the last p vectors. Although, this has been shown to be a reasonable approach, it would be better to have some sort of mechanism for choosing the vectors to keep. Therefore, de Sturler [Stu99] introduced the concept of *optimal truncation* which together with GCRO results in the GCROT method. GCROT can be easily modified to solve (4.1), which will be described in Section 4.1.

The GMRES-E method is based on an augmentation approach and can be used for solving (4.1). After solving the i-th system, we run $m - k$ steps of GMRES for the $(i + 1)$-st system, append the k approximate eigenvectors from the i-th system and proceed as described in Section 3.5.2. However, the subspace formed this way is **not** a Krylov subspace, so an efficient version as for GMRES-DR is not possible. Another disadvantage of this approach is that it extends the search space as in restarted GMRES, thus the method may suffer from stagnation. On the other hand, GMRES-DR **cannot** be modified or adapted for solving (4.1), because the harmonic Ritz vectors of A do not form a Krylov subspace for another matrix or even another right-hand side. However, using deflated restarting within the GCRO framework results in an effective

method, called GCRO-DR, which recycles a carefully chosen subspace from previous systems, thus reducing costs of subsequent systems. In Section 4.2, we give a detailed description of the method, as well as the convergence analysis. Recently, Niu, Lu and Liu successfully combined the idea exploited for the LGMRES method within the GCRO-DR framework, thus developing the new method Loose GCRO-DR (LGCRO-DR), which we present in Section 4.3.

One of the main characteristics of the GCRO-DR method is that it deflates the smallest eigenvalues by using an orthogonal projector (see, e.g., Section 4.2). Since we are dealing mostly with nonsymmetric systems, i.e. matrices for which left and right eigenspaces are not mutually orthogonal, and GMRES-like methods, we propose, in the spirit of [Gut12], a method that uses an oblique projector (see, e.g., Section 4.4 or [Gut12]). This way, we do not only annihilate eigenvalues, but also deflate the corresponding left and right eigenspaces. Therefore, in [Gut12] Gutknecht refers to this approach as the *truly deflated* GMRES method. In addition to Gutknecht's theory, we provide a cheap way of computing approximations to left eigenvectors, without having to build a Krylov subspace for A^H. We introduce our method, named Left-Right Deflated GMRES (LRDGMRES), in Section 4.4. We will show that our method is numerically comparable to the GCRO-DR method, where from the theoretical point of view this should be a better approach (see Section 4.4 for a more detailed discussion), since we deflate both eigenspaces. In Section 4.4.2, we show how an idea from [NLL13] can be also incorporated into our method, thus leading to an even more efficient method for solving (4.1), termed Loose Left-Right Deflated GMRES (LLRDGMRES).

We would like to state here that throughout this chapter we will use the following notation:

- m – restart length (or the number of iterations done by GMRES)
- k – dimension of the recycle space
- l – number of error approximations

Furthermore, within this chapter we distinguish *right* and *left* harmonic Ritz vectors. In the previous chapter we have already discussed harmonic Ritz vectors computed according to Lemma 3.11, and they will correspond to the right harmonic Ritz vectors. In Section 4.3 we will derive a formula using Definition 2.25 for computing the left harmonic Ritz vectors.

4.1 Optimal Truncation

In order to reduce costs of optimal methods like GMRES or GCR, we usually consider restarted or truncated forms of these methods (see, e.g., Chapter 3). Restarting basically means that we perform $m < n$ steps of the method and then start over discarding all the information built in the current cycle, whereas truncation means orthogonalizing only against a certain number of recently computed vectors. It is well known that both strategies slow down the convergence of methods. In addition, the optimality of these methods is lost, since the orthogonality constraints $r_i \perp A\mathcal{K}_i(A, r_0)$ do not hold anymore. We have already introduced in the previous chapter some of the methods that improve convergence by keeping a subspace between cycles. One class of methods containing GMRES-E; and GMRES-DR augments the Krylov subspace in the current cycle with approximate eigenvectors corresponding to the smallest eigenvalues from the previous cycle. This technique has proved to be efficient when the slow convergence is influenced by the smallest eigenvalues, which is not always the case (see, e.g., [Mor95]). On the other hand, methods like GMRESR or GCRO, are in essence truncated methods, since they keep a limited number of vectors from previous search spaces. This form of truncation, i.e. keeping orthogonality to part of the old Krylov subspace yields good convergence in many cases (see, e.g., [Stu96a]). However, none of these methods have a mechanism for choosing the best vectors to keep, since they merely keep the correction to the residual in the outer method. In order to have an efficient method in its restarted or truncated form, we should not restrict the choice of the vectors to keep (discard) only to the vectors computed in the iteration, but we should rather consider subspaces of the space spanned by all available vectors.

Optimal truncation is a mechanism of retaining a subspace between cycles such that the loss of orthogonality with respect to the truncated space is minimized. Incorporating optimal truncation within the GCRO method leads to the GCROT method [Stu99]. GCROT, as well as GCRO, maintains matrices U_k and C_k, such that $AU_k = C_k$ and $C_k^H C_k = I$. Further on, it performs GMRES while keeping orthogonality to C_k, which corresponds to GMRES with the operator $A_{C_k} = (I - C_k C_k^H)A$. The method then computes the approximate solution and selects vectors via optimal truncation, appends them to U_k, and again updates U_k and C_k such that $AU_k = C_k$ and $C_k^H C_k = I$. At the end of the cycle, U_k and C_k are carried over to the next cycle (see, e.g., Section 3.4 for more detailed description).

Since all the implementation details have been described in Section 3.4, we dedicate the remainder of this section to explaining the concept of optimal

truncation. We will derive a set of formulas, which describe how we choose the best possible subspace. First, we need the following definition.

Definition 4.1. *Let the matrices $C \in \mathbb{C}^{n\times k}$, $F \in \mathbb{C}^{n\times m}$ and the vector $r \in \mathbb{C}^n$ be given, such that*

$$\begin{aligned} C^H C &= I_k, \\ C^H r &= 0, \\ \operatorname{rank}(F) &= m, \qquad (4.2) \\ \operatorname{rank}(F - CC^H F) &= m. \qquad (4.3) \end{aligned}$$

Furthermore, let

$$\begin{aligned} F &= CB + QR, \qquad (4.4) \\ Q^H Q &= I_m, \end{aligned}$$

where $B = C^H F$ and R is upper-triangular. Using B and R we define

$$Z = BR^{-1} = (C^H F)(Q^H F)^{-1},$$

and we denote the singular value decomposition of Z by

$$Z = Y_Z \Sigma_Z V_Z^H.$$

$Y_Z = \begin{bmatrix} y_1 & y_2 & \dots & y_k \end{bmatrix}$ and $V_Z = \begin{bmatrix} v_1 & v_2 & \dots & v_m \end{bmatrix}$ are ordered so as to follow the convention

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p,$$

for the diagonal entries σ_i of Σ_Z, where $p = \min(k, m)$. Also let

$$\begin{aligned} F &= WS \qquad (4.5) \\ W^H W &= I_m, \end{aligned}$$

and S be upper-triangular. Finally, let

$$\begin{aligned} r_1 &= (I - QQ^H)r, \qquad (4.6) \\ r_2 &= (I - WW^H)r. \qquad (4.7) \end{aligned}$$

Note that from (4.2) and (4.3) it follows that $\dim(\operatorname{range}(C) \oplus \operatorname{range}(F)) = \dim(\operatorname{range}(C)) + \dim(\operatorname{range}(F))$. Rewriting the equation (4.4), we see that $QR = F - CC^H F$ is the QR-decomposition of the matrix $F - CC^H F$.

This decomposition yields $QQ^H r$, the best approximation to r in the space range$(C) \oplus$ range(F). Furthermore, r_1 defined in (4.6) is the residual which corresponds to the best approximation to r in the space range$(C) \oplus$ range(F). On the other hand, equation (4.5) represents the QR-decomposition of the matrix F, and yields $WW^H r$, the best approximation to r in the subspace range(F). Therefore, r_2 defined in (4.7) represents the residual corresponding to the best approximation to r in the space range(F) *ignoring* orthogonality to range(C). The difference $e = r_2 - r_1$ between these two approximations is called the residual error. The geometric aspect is depicted in Figure 4.1, which is reproduced from [Stu96b].

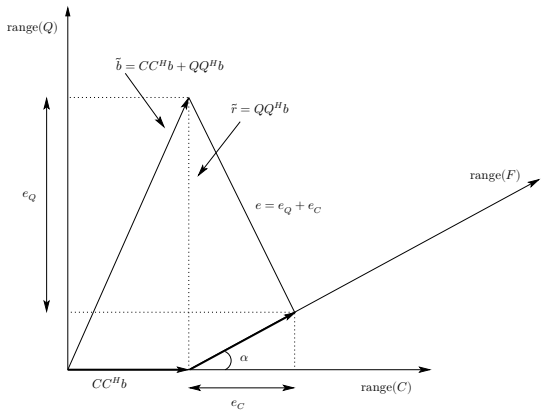


Figure 4.1: Loss of optimality from neglecting previous search directions. The vectors $\tilde{b}$ and $\tilde{r}$ are the projections of b and r respectively onto the space range$(C) \oplus$ range(F). The vector $CC^H b$ is the current best approximation to the right-hand side b, and the vector $QQ^H b$ is the best possible correction to the current residual. If we neglect the orthogonality to C, then the orthogonal projection on range(F) leads to the residual error e.

By studying the residual error, we can analyze the consequences of neglecting orthogonality to range(C). The residual error depends on the *principal angles* α between the subspaces range(C) and range(F), so one way to analyze the error is to look at the principal angles. However, we follow the work in [Stu99], and instead of the principal angles, we look at the length of the residual error e.

Theorem 4.2. *The residual error is given by*

$$r_2 - r_1 = \sum_{i=1}^{p} \left(\frac{\nu_i \sigma_i^2}{1+\sigma_i^2} Q v_i - \frac{\nu_i \sigma_i}{1+\sigma_i^2} C y_i \right),$$

where $\nu_i = v_i^H Q^H r$*, and the norm of the residual error is given by*

$$\|r_2 - r_1\|_2 = \left(\sum_{i=1}^{p} \frac{|\nu_i|^2 \sigma_i^2}{1 + \sigma_i^2} \right)^{1/2}.$$

For the proof see, e.g., [Stu99].

Remark 4.3. Theorem 4.2 indicates that for $\sigma_i = 0$, which corresponds to a direction in range(F) orthogonal to range(C), no error is made in that direction, whereas for $\sigma_i \to \infty$, corresponding to a direction in range$(C) \oplus$range(F), the error in this direction equals the optimal correction, and therefore no correction is made.

The result of Theorem 4.2 explains the consequences of neglecting the orthogonality to range(C), i.e. the consequences of discarding range(C) by truncation or restart. We will use this result to show which subspace of range(C) is the best possible subspace to keep or discard, such that we maintain good convergence at low cost. Let us consider computing the residual r_3 while maintaining orthogonality to an arbitrary subspace range(CT) and neglecting orthogonality to range(CT_C), where T_C is the complement of T. Different choices of T lead to different subspaces of range(C).

Definition 4.4. *Let* C, F, Q *and* r *be as in Definition 4.1, and let the matrix* $[T|T_C]$ *be a square, unitary matrix such that* $\operatorname{rank}([T|T_C]) = \operatorname{rank}(C) = k$*, i.e.* $T \in \mathbb{C}^{k \times l}$*. Now let* $\bar{F} = [CT|F]$*,* $\bar{C} = CT_C$*, and* $\bar{Q} = [CT|Q]$*, and let*

$$\bar{B} = \bar{C}^H \bar{F} = [0|T_C^H B],$$
$$\bar{R} = \bar{Q}^H \bar{F} = \begin{bmatrix} I & T^H B \\ 0 & R \end{bmatrix}.$$

Using $\bar{B}$ *and* $\bar{R}$*, we define*

$$\bar{Z} = \bar{B}\bar{R}^{-1} = [0|T_C^H Z].$$

We denote the singular value decomposition of $\bar{Z}$ *by*

$$\bar{Z} = Y_{\bar{Z}} \Sigma_{\bar{Z}} V_{\bar{Z}}^H,$$

where $Y_{\bar{Z}} = \begin{bmatrix} \bar{y}_1 & \bar{y}_2 & \dots & \bar{y}_{k-l} \end{bmatrix}$ *and* $V_{\bar{Z}} = \begin{bmatrix} \bar{v}_1 & \bar{v}_2 & \dots & \bar{v}_{m+l} \end{bmatrix}$ *are ordered so as to follow the convention*

$$\bar{\sigma}_1 \geq \bar{\sigma}_2 \geq \cdots \geq \bar{\sigma}_{\min(k-l,m+l)}.$$

Furthermore, let

$$\bar{F} = \bar{W}\bar{S},$$
$$\bar{W}^H\bar{W} = I_{m+l},$$

where $\bar{S}$ *is upper-triangular, and let*

$$r_3 = (I - \bar{W}\bar{W}^H)r.$$

Finally, let the residual error from discarding $\mathrm{range}(CT_C)$ *be given by* $\bar{e} = r_3 - r_1$.

The residual r_3 corresponds to the best approximation to r in the space $\mathrm{range}(CT) \oplus \mathrm{range}(F)$, while ignoring orthogonality to $\mathrm{range}(CT_C)$. Next we derive the equation for the residual error $\bar{e}$ depending on T, which follows from Theorem 4.2.

Theorem 4.5. *The residual error* $\bar{e}$ *is given by*

$$r_3 - r_1 = \sum_{i=1}^{\min(k-l,m+l)} \left(\frac{\bar{\nu}_i \bar{\sigma}_i^2}{1+\bar{\sigma}_i^2} \bar{Q}\bar{v}_i - \frac{\bar{\nu}_i \bar{\sigma}_i}{1+\bar{\sigma}_i^2} \bar{C}\bar{y}_i \right),$$

where $\bar{\nu}_i = \bar{v}_i^H \bar{Q}^H r$, *and its norm is given by*

$$\|r_3 - r_1\|_2 = \left(\sum_{i=1}^{\min(k-l,m+l)} \frac{|\bar{\nu}_i|^2 \bar{\sigma}_i^2}{1+\bar{\sigma}_i^2} \right)^{1/2}.$$

We see that the norm of the residual error $\bar{e}$ (as well as e) is determined by the singular values $\bar{\sigma}_i$ (σ_i) and the coefficients $\bar{\nu}_i$ (ν_i). Assuming that the coefficients $\bar{\nu}_i$ are not very small, we conclude that the smaller the singular values are, the smaller the residual error will be. Therefore, we would like to choose T_C, and hence T, such that the l largest singular values from Z are removed in $\bar{Z}$. This is achieved according to the following min-max theorem [Stu99], which is a variant of Theorems 3.1.2 and 3.3.15 in [HJ94].

Theorem 4.6. *Let Z and its singular value decomposition be as in Definition 4.1, and let T_C be as in Definition 4.4. Then*

$$\min_{\substack{S\subset\mathbb{C}^k,\\ \dim(S)=k-l}} \max_{\substack{x\in S,\\ \|x\|_2=1}} \left\|x^H Z\right\|_2 = \begin{cases} \sigma_{l+1}, & \text{if } l+1\leq p \\ 0, & \text{if } l+1>p \end{cases},$$

and the minimum is found for

$$S = \operatorname{span}\{y_{l+1},\dots,y_k\}.$$

This is equivalent to

$$\min_{\substack{T_C\in\mathbb{C}^{k\times(k-l)},\\ T_C^H T_C=I_{k-l}}} \max_{\substack{\xi\in\mathbb{C}^{k-l},\\ \|\xi\|_2=1}} \left\|(T_C\xi)^H Z\right\|_2 = \begin{cases} \sigma_{l+1}, & \text{if } l+1\leq p \\ 0, & \text{if } l+1>p \end{cases},$$

and the minimum is found for T_C, such that

$$\begin{aligned} \operatorname{range}(T_C) = S &= \operatorname{span}\{y_{l+1},\dots,y_k\}, \\ x &= T_C\xi. \end{aligned} \tag{4.8}$$

Since $\bar{Z} = [0|T_C^H Z]$, and $\left\|(T_C\xi)^H Z\right\|_2 = \left\|\xi^H(T_C^H Z)\right\|_2 = \left\|\xi^H\bar{Z}\right\|_2$, it follows that T_C from (4.8) minimizes the maximum singular value of $\bar{Z}$. Therefore, the obvious choice for the optimal truncation T and its complement T_C are

$$\begin{aligned} T &= [y_1,\dots,y_l] \\ T_C &= [y_{l+1},\dots,y_k]. \end{aligned} \tag{4.9}$$

With this choice, the following theorem about the residual error $\bar{e}$ and its norm holds, see, e.g., [Stu99].

Theorem 4.7. *The residual error $\bar{e}$ that results from the truncation defined by the matrix T from* (4.9) *is given by*

$$r_3 - r_1 = \sum_{i=l+1}^{p} \left(\frac{\nu_i\sigma_i^2}{1+\sigma_i^2} Q v_i - \frac{\nu_i\sigma_i}{1+\sigma_i^2} C y_i \right),$$

and its norm is given by

$$\|r_3 - r_1\|_2 = \left(\sum_{i=l+1}^{p} \frac{|\nu_i|^2 \sigma_i^2}{1+\sigma_i^2} \right)^{1/2}.$$

The purpose of this section was mainly to introduce the concept of optimal truncation, which serves as the mechanism for choosing the best possible subspace to keep, such that the effects of ignoring orthogonality to a discarded subspace are minimized. The author in [Stu99] has chosen to implement the optimal truncation within the GCRO method, because GCRO consists of the inner method GMRES, which is more efficient and more stable than GCR, and the outer method GCR, which offers the flexibility to optimize over arbitrary subspaces. For further implementation details we refer to [Stu99].

The GCROT method can be easily modified for solving (4.1), by carrying U_k from the i-th to the $(i+1)$-st system. Since for GCROT we have $A^{(i)}U_k = C_k$, the idea is to modify U_k and C_k with respect to $A^{(i+1)}$, such that the relation $A^{(i+1)}U_k^{\text{new}} = C_k^{\text{new}}$ holds. This is done in the following way:

$$A^{(i+1)}U_k^{\text{old}} = QR \tag{4.10}$$
$$C_k^{\text{new}} = Q \tag{4.11}$$
$$U_k^{\text{new}} = U_k^{\text{old}} R^{-1}. \tag{4.12}$$

Now, we can simply proceed with GCROT on the $(i+1)$-st linear system. In practice, computing $A^{(i+1)}U_k^{\text{old}} = C_k^{\text{old}} + \Delta A^{(i)} U_k^{\text{old}}$ is much cheaper than k matrix-vector products, since $\Delta A^{(i)}$ is substantially sparser than $A^{(i)}$ or has a special structure. Even if that is not the case, computing $A^{(i+1)}U_k^{\text{old}}$ directly can be faster than k matrix-vector multiplications.

4.2 GCRO-DR

The GCRO-DR method is a combination of the GMRES-DR method and the GCRO method. It uses deflated restarting within the GCRO framework, which in contrast to GMRES-DR, allows the method to easily be modified for solving (4.1). When solving a single linear system, GMRES-DR and GCRO-DR are mathematically equivalent.

GCRO-DR has been proved to be a significant advancement. It works under general assumptions, without assuming that matrices are pairwise close or that the sequence of matrices converges to a particular matrix and without any further assumptions on how right-hand sides change. The underlying idea

is to reduce the cost of solving subsequent systems in the sequence by keeping a carefully chosen subspace from previous systems. This is known as *Krylov subspace recycling*. An efficient method that uses recycling must satisfy certain properties:

- It has to determine and converge to an effective recycle space in a reasonable number of iterations.
- It must be able to converge to an effective recycle space over the solution of multiple linear systems.
- A convergence improvement should be obtained with a recycle space of a relatively small dimension.
- It must provide an effective and cheap mechanism for regularly updating the recycle space when solving the next system in the ensemble.

As it will be shown, GCRO-DR satisfies all these properties.

We now present the method. As we suggested, GCRO-DR uses the GCRO framework. This means that we need to keep two matrices U_k and C_k such that (recall (3.4) - (3.6)):

$$\begin{aligned} \text{range}(U_k) &= \mathcal{K}_k(A, r_0) \\ AU_k &= C_k \\ C_k^H C_k &= I. \end{aligned}$$

We distinguish two cases:

1. We are solving the first cycle of the first system, i.e. we do not have any previous information available.
2. We have solved the i-th system and we carried k right harmonic Ritz vectors $\tilde{Y}_k = [\tilde{y}_1, \ldots, \tilde{y}_k]$ to the $(i+1)$-st system.

In the first case, we have to build a Krylov subspace using m steps of GMRES, and to update the solution and the corresponding residual as with GMRES. Next, we compute k right harmonic Ritz vectors according to Lemma 3.11, i.e. $\tilde{Y}_k = AV_m\hat{Y}_k$, where the columns of $\hat{Y}_k$ are the eigenvectors of the matrix (3.26) corresponding to the k smallest eigenvalues. Using the Arnoldi relation (2.15), i.e. $AV_m = V_{m+1}\bar{H}_m$ and the fact that V_{m+1} is orthonormal, we obtain U_k and C_k in the following way:

$$\bar{H}_m\hat{Y}_k = QR \tag{4.13}$$

$$C_k = V_{m+1}Q \tag{4.14}$$

$$U_k = V_m\hat{Y}_k R^{-1}. \tag{4.15}$$

Now, let us assume that we have solved the i-th system and that we kept

k right harmonic Ritz vectors $\tilde{Y}_k$, by keeping their preimages U_k^{old}, for the $(i+1)$-st system. Then, the GCRO-DR method computes matrices U_k and C_k from $A^{(i+1)}$ and U_k^{old} using (4.10) - (4.12). The optimal solution over the space range(U_k) and the corresponding residual vector are computed using (3.8) and (3.9), respectively. In the remainder of the section we drop the superscript in $A^{(i+1)}$ for notational convenience.

After updating U_k and C_k (in both cases), we proceed in the same way as for the GCRO method. We perform $m-k$ steps of the Arnoldi process with a different operator $A_{C_k} = (I - C_k C_k^H)A$, which produces the Arnoldi relation

$$(I - C_k C_k^H)AV_{m-k} = V_{m-k+1}\bar{H}_{m-k}. \tag{4.16}$$

We can rewrite (4.16) as

$$A\begin{bmatrix} U_k & V_{m-k} \end{bmatrix} = \begin{bmatrix} C_k & V_{m-k+1} \end{bmatrix} \begin{bmatrix} I_k & B_k \\ 0 & \bar{H}_{m-k} \end{bmatrix}, \tag{4.17}$$

where $B_k = C_k^H A V_{m-k}$. Numerical tests suggest that the rightmost matrix in (4.17) is ill-conditioned. To reduce unnecessary ill-conditioning we can compute the diagonal matrix D_k, such that $\tilde{U}_k = U_k D_k$ has unit columns. Defining

$$\hat{V}_m = \begin{bmatrix} \tilde{U}_k & V_{m-k} \end{bmatrix}, \quad \hat{W}_{m+1} = \begin{bmatrix} C_k & V_{m-k+1} \end{bmatrix}, \quad G_m = \begin{bmatrix} D_k & B_k \\ 0 & \bar{H}_{m-k} \end{bmatrix},$$

the Arnoldi relation (4.17) becomes

$$A\hat{V}_m = \hat{W}_{m+1} G_m. \tag{4.18}$$

It is worth mentioning that the columns of $\hat{V}_m$ and $\hat{W}_{m+1}$ have unit norm. In addition, the columns of $\hat{W}_{m+1}$ are orthogonal (since range$(V_{m-k+1}) \perp$ range(C_k)), which is not true for $\hat{V}_m$.

We proceed as with GMRES, by solving the minimization problem

$$\xi_m = \underset{\xi \in \text{range}(\hat{V}_m)}{\text{argmin}} \ \|r_{i-1} - A\xi\|_2,$$

which reduces to the $(m+1) \times m$ least squares problem

$$\xi_m = \underset{\xi \in \text{range}(\hat{V}_m)}{\text{argmin}} \ \left\|\hat{W}_{m+1}^H r_{i-1} - G_m \xi\right\|_2.$$

The residual and solution are updated via

$$\begin{aligned} r_i &= r_{i-1} - A\hat{V}_m \xi_m = r_{i-1} - \hat{W}_{m+1} G_m \xi_m, \\ x_i &= x_{i-1} + \hat{V}_m \xi_m. \end{aligned} \tag{4.19}$$

At the end of each cycle, the method computes the k right harmonic Ritz vectors corresponding to the k smallest harmonic Ritz values by solving the generalized eigenvalue problem derived from Definition 2.25 and the Arnoldi relation (4.18), which is given in the following Lemma.

Lemma 4.8. *Solving the generalized eigenvalue problem*

$$G_m^H \hat{W}_{m+1}^H \hat{V}_m \hat{y}_i = \theta_i G_m^H G_m \hat{y}_i,$$

*one obtains the k right harmonic Ritz vectors (corresponding to the k harmonic Ritz values of smallest magnitude $1/\theta_i, i = 1, \ldots, k$) in the main loop of the GCRO-DR method (***while** *loop in Algorithm 4.1) as $\tilde{y}_i = A\hat{V}_m \hat{y}_i, i = 1, \ldots, k$.*

Proof. Using Definition 2.24 and the Arnoldi relation (4.18) we have with $\tilde{y}_i = A\hat{V}_m \hat{y}_i, i = 1, \ldots, k$

$$\begin{aligned}
&(A\hat{V}_m)^H (A^{-1} A\hat{V}_m \hat{y}_i - \theta_i A\hat{V}_m \hat{y}_i) = 0 \\
\Leftrightarrow\ &\theta_i \hat{V}_m^H A^H A\hat{V}_m \hat{y}_i = \hat{V}_m^H A^H \hat{V}_m \hat{y}_i \\
\Leftrightarrow\ &\theta_i G_m^H \hat{W}_{m+1}^H \hat{W}_{m+1} G_m \hat{y}_i = G_m^H \hat{W}_{m+1}^H \hat{V}_m \hat{y}_i \\
\Leftrightarrow\ &\theta_i G_m^H G_m \hat{y}_i = G_m^H \hat{W}_{m+1}^H \hat{V}_m \hat{y}_i,
\end{aligned}$$

which completes the proof. Note that we have used the fact that $\hat{W}_{m+1}$ has orthonormal columns. □

Before restarting, we need to compute the new matrices U_k and C_k, which is done similarly to (4.13)-(4.15)

$$G_m \hat{Y}_k = QR \tag{4.20}$$

$$C_k = \hat{W}_{m+1} Q \tag{4.21}$$

$$U_k = \hat{V}_m \hat{Y}_k R^{-1}. \tag{4.22}$$

After we have solved the $(i+1)$-st system, the matrix U_k is carried over to the next system. We give GCRO-DR as Algorithm 4.1.

Remark 4.9. So far, we have discussed deflating the eigenvalues of smallest magnitude. An interesting possibility is to combine a few right harmonic Ritz vectors corresponding to the harmonic Ritz values of smallest magnitude and a few right harmonic Ritz vectors corresponding to the harmonic Ritz values of largest magnitude, thus simultaneously deflating eigenvalues of both, smallest and largest, magnitude.

Algorithm 4.1: GCRO-DR

Input :
- $A^{(i)} \in \mathbb{C}^{n\times n}$ — i-th system matrix
- $b^{(i)} \in \mathbb{C}^n$, $x_0^{(i)} \in \mathbb{C}^n$ — right-hand side and initial guess for the i-th system
- $m \in \mathbb{N}$ — restart length
- $k \in \mathbb{N}$ — number of deflated approximate ev's
- U_k^{old} — recycle space from the previous system (if it is defined)

Output:
- approximate solution $x_m^{(i)}$ to $A^{(i)}x^{(i)} = b^{(i)}$ of the i-th system
- k approximate right eigenvectors $\tilde{Y}_k$

$r_0 = b - Ax_0$, $\beta = \|r_0\|$, $v_1 = r_0/\beta$, $i = 1$
if *We are solving the first system* **then**
  Perform m steps of the Arnoldi process, thus providing the Arnoldi relation $AV_m = V_{m+1}\bar{H}_m$
  Compute $x_1 = x_0 + V_m\xi_m$ and $r_1 = r_0 - V_{m+1}\bar{H}_m\xi_m$, where $\xi_m = \operatorname{argmin}_\xi \left\|\beta e_i - \bar{H}_m\xi\right\|_2$
  Compute k right harmonic Ritz vectors according to Lemma 3.11
  Compute U_k and C_k using (4.13)-(4.15)
else
  $AU_k^{\text{old}} = QR$, $C_k = Q$, $U_k = U_k^{\text{old}}R^{-1}$
  $x_1 = x_0 + U_kC_k^Hr_0$, $r_1 = r_0 - C_kC_k^Hr_0$
while *Convergence criteria not satisfied* **do**
  $i = i + 1$
  Perform $m - k$ steps of the Arnoldi process with an operator $(I - C_kC_k^H)A$ and starting with $v_1 = r_{i-1}/\|r_{i-1}\|_2$
  Define $\hat{V}_m$, $\hat{W}_{m+1}$ and G_m such that the Arnoldi relation (4.18) holds
  Compute $x_i = x_{i-1} + \hat{V}_m\xi_m$ and $r_i = r_{i-1} - \hat{W}_{m+1}G_m\xi_m$, where $\xi_m = \operatorname{argmin}_\xi \left\|\hat{W}_{m+1}^Hr_{i-1} - G_m\xi\right\|_2$
  Compute k right harmonic Ritz vectors according to Lemma 4.8
  Update U_k and C_k using (4.20)-(4.22)
Carry U_k to the next system

4.2.1 Convergence Analysis

We discuss here two theoretical results taken from [Par+06] and [KS07] and their implications. The first result concerns the convergence of GCRO-DR, whereas the second result concerns the perturbation of invariant subspaces associated with the smallest eigenvalues when the change in the matrix is concentrated in an invariant subspace corresponding to "large" eigenvalues. What is meant here by large eigenvalues are the remaining ones, i.e. all eigenvalues except for the smallest.

Let $\mathcal{Q}$ be an l-dimensional invariant subspace of A, and let $\mathcal{C} = \text{range}(C_k)$ be a k-dimensional space ($k \geq l$) selected to approximate $\mathcal{Q}$. We define $\Pi_{\mathcal{C}}$ and $\Pi_{\mathcal{Q}}$ to be the orthogonal projectors onto $\mathcal{C}$ and $\mathcal{Q}$, respectively. We also define $P_{\mathcal{Q}}$ to be the spectral projector onto $\mathcal{Q}$, according to the following definition.

Definition 4.10. *Let* $\text{range}(V_1)$ *and* $\text{range}(V_2)$ *be two complementary invariant subspaces of* A *such that*

$$\begin{aligned} AV_1 &= V_1\Lambda_1 \\ AV_2 &= V_2\Lambda_2, \end{aligned}$$

and

$$\text{range}(V_1) \oplus \text{range}(V_2) = \mathbb{C}^n.$$

Then

$$P = V_1(V_2^H V_1)^{-1} V_2^H$$

is called a spectral projector. In particular, if A *is diagonalizable, then*

$$\begin{aligned} A = X\Lambda X^{-1} = X\Lambda Y^H &= \begin{bmatrix} X_1 & X_2 \end{bmatrix} \Lambda \begin{bmatrix} Y_1^H \\ Y_2^H \end{bmatrix} \\ &= X_1\Lambda_1 Y_1^H + X_2\Lambda_2 Y_2^H, \end{aligned}$$

where $Y_1^H X_1 = I$, $Y_2^H X_2 = I$, *is the spectral representation of* A. *Here,* $\text{range}(X_1)$ *and* $\text{range}(X_2)$ *are complementary right eigenspaces, and* $\text{range}(Y_1)$ *and* $\text{range}(Y_2)$ *are complementary left eigenspaces corresponding to* $\text{range}(X_1)$ *and* $\text{range}(X_2)$, *respectively. Then, the spectral projector is given as (see [SG90])*

$$P_i = X_i Y_i^H, \quad i = 1, 2.$$

Furthermore, we define the *one-sided distance* from the subspace $\mathcal{Q}$ to the subspace $\mathcal{C}$ as

$$\delta(\mathcal{Q}, \mathcal{C}) \equiv \left\| (I - \Pi_{\mathcal{C}})\Pi_{\mathcal{Q}} \right\|_2,$$

which is equal to the sine of the largest principal angle between $\mathcal{Q}$ and $\mathcal{C}$. Next, we give the theorem which shows that the recycle space need not approximate an invariant subspace accurately to improve the convergence of GCRO-DR significantly, see, e.g., [Par+06].

Theorem 4.11. *Given a space $\mathcal{C}$, let $\mathcal{K} = \operatorname{range}(V_{m-k+1}\bar{H}_{m-k})$ be the $(m-k)$-dimensional Krylov subspace generated by GCRO-DR as in (4.16). Let $r_0 \in \mathbb{C}^n$, and let $r_1 = (I - \Pi_{\mathcal{C}})r_0$. Then, for each $\mathcal{Q}$ such that $\delta(\mathcal{Q}, \mathcal{C}) < 1$,*

$$\min_{d_1 \in \mathcal{K}\oplus\mathcal{C}} \|r_0 - d_1\|_2 \leq \min_{d_2 \in (I-P_{\mathcal{Q}})\mathcal{K}} \left\|(I - P_{\mathcal{Q}})r_1 - d_2\right\|_2 + \frac{\gamma}{1-\delta} \left\|P_{\mathcal{Q}}\right\|_2 \|(I - \Pi_{\mathcal{K}})r_1\|_2 ,$$

where $\gamma = \left\|(I - \Pi_{\mathcal{C}})P_{\mathcal{Q}}\right\|_2$. If, in addition, A is Hermitian, then we have

$$\min_{d_1 \in \mathcal{K}\oplus\mathcal{C}} \|r_0 - d_1\|_2 \leq \min_{d_2 \in (I-\Pi_{\mathcal{Q}})\mathcal{K}} \left\|(I - \Pi_{\mathcal{Q}})r_1 - d_2\right\|_2 + \frac{\delta}{1-\delta} \|(I - \Pi_{\mathcal{K}})r_1\|_2 .$$

The left-hand side in the above inequalities represents the residual norm after $m-k$ steps of GCRO-DR with the recycled space $\mathcal{C}$. On the right-hand side, the first term represents the best possible approximation of the deflated problem, where all components in the subspace $\mathcal{Q}$ have been removed, which typically leads to an improved rate of convergence, whereas the second term is a constant times the residual of $m-k$ iterations of GCRO-DR, solving for r_1. If the recycle space $\mathcal{C}$ contains $\mathcal{Q}$, then $\delta = \gamma = 0$, and GCRO-DR converges at least as fast as the deflated problem. In numerical experiments in [Par+06] it is shown that the method fairly quickly reaches values $\delta = O(10^{-2})$, for which GCRO-DR converges as fast as the deflated problem, as long as $\left\|P_{\mathcal{Q}}\right\|_2$ is not large in the non-Hermitian case. Note that $\delta = O(10^{-2})$ means that the invariant subspace $\mathcal{Q}$ is not approximated very accurately.

Now we turn to the second result. One of the main properties of a sequence of systems (4.1)—which is also the reason why we can use the recycling strategy—is that the matrices change slightly from one system to the next. If the magnitude of the change is smaller than the size of the gap between the smallest and the remaining eigenvalues, then the invariant subspace associated with the smallest eigenvalues is not significantly altered. For our work, this is a desirable property, since we want to recycle exactly that subspace. Before stating the next theorem, we need to define certain quantities.

For simplicity we deal with a Hermitian positive definite matrix A and a corresponding Hermitian perturbation E. Let A have the eigendecomposition

$$A = [Q_1\ Q_2\ Q_3] \operatorname{diag}(\Lambda_1, \Lambda_2, \Lambda_3) [Q_1\ Q_2\ Q_3]^H , \tag{4.23}$$

where $[Q_1\ Q_2\ Q_3]$ is a unitary matrix, $\Lambda_1 = \operatorname{diag}\left(\lambda_1^{(1)}, \ldots, \lambda_{j_1}^{(1)}\right)$, and Λ_2 and Λ_3 are defined analogously. Furthermore, let

$$\lambda_1^{(1)} \leq \cdots \leq \lambda_{j_1}^{(1)} < \lambda_1^{(2)} \leq \cdots \leq \lambda_{j_2}^{(2)} < \lambda_1^{(3)} \leq \cdots \leq \lambda_{j_3}^{(3)}.$$

Now consider the change in the invariant subspace range(Q_1) under a symmetric perturbation E of A. Let $\theta_1(.,.)$ denote the largest canonical angle between two spaces. It is not required that $\|E\|_F$ is small, but rather that the projection of E onto the subspace range($[Q_1\ Q_2]$) is small. We thus assume that $\left\|[Q_1\ Q_2]^H E\right\|_F \leq \varepsilon$ and that ε is small relative to $\lambda_1^{(2)} - \lambda_{j_1}^{(1)}$. In addition, assume that $\eta \equiv \left\|Q_3^H E\right\|_F$ is small relative to $\lambda_1^{(3)} - \lambda_{j_1}^{(1)}$. Also, let

$$\mu \equiv \min\left(\lambda_1^{(2)} - \varepsilon, \lambda_1^{(3)} - \eta\right) - 2\varepsilon - \left(\lambda_{j_1}^{(1)} + \varepsilon\right) > 2\varepsilon,$$
$$\hat{\mu} \equiv \mu\left(1 - \frac{2\varepsilon^2}{\eta^2}\right) + \lambda_{j_1}^{(1)} + \varepsilon.$$

Theorem 4.12. *Let A be Hermitian positive definite and have the eigendecompositin given in* (4.23)*, and let $E, \varepsilon, \eta, \mu$ and $\hat{\mu}$ be defined as above. Then there exists a matrix $\widehat{Q}_1$ conforming to Q_1, such that* range($\widehat{Q}_1$) *is an invariant subspace of $A + E$, and*

$$\tan\theta_1\left(\text{range}(Q_1), \text{range}(\widehat{Q}_1)\right) \leq \tfrac{\varepsilon}{\hat{\mu}}.$$

For the proof, see [KS07].

The result of Theorem 4.12 indicates that if the recycle space reasonably approximates an invariant subspace, then it will also provide a reasonable approximation to the slightly perturbed invariant subspace of the updated matrix.

4.3 Loose GCRO-DR

The GCRO-DR method has been a huge advancement with respect to the related work for solving (4.1). It works under general assumptions and has flexibility of choosing any subspace for recycling, as we have seen in the previous section. Recently, Niu, Lu and Liu in [NLL13] have showed that GCRO-DR experiences a similar alternating behaviour of the residual vectors at the end of every cycle as restarted GMRES (which is not surprising since GCRO-DR is a GMRES-like method). This implies that it is possible to further improve GCRO-DR by recycling a few error approximations $z_i = x_i - x_{i-1}$ in the same manner as in Section 3.5.4. This was the main motivation for developing the Loose GCRO-DR method (LGCRO-DR) [NLL13]. The new method combines

recycling an invariant subspace related to the approximate eigenvectors corresponding to the smallest eigenvalues and recycling a few error approximations in an easy and straightforward way, which we explain next.

Since LGCRO-DR differs from GCRO-DR only in the way how a recycle space is being formed, we focus here only on that part. For reasons that will become clear later, we distinguish two cases: recycling one error approximation and recycling $l \geq 2$ error approximations. Let us first assume that we are recycling one error approximation z_m. Noticing that $\tilde{Y}_k = A\hat{V}_m\hat{Y}_k$ (Lemma 4.8) and that $z_m = \hat{V}_m\xi_m$ (see (4.19)) it follows that recycling one error approximation is equivalent to appending ξ_m to $\hat{Y}_{k-1}$, i.e. $\hat{Y}_k = \begin{bmatrix}\hat{Y}_{k-1} & \xi_m\end{bmatrix}$, and proceeding according to (4.20)-(4.22)

$$G_m\hat{Y}_k = QR \tag{4.24}$$

$$C_k = \hat{W}_{m+1}Q \tag{4.25}$$

$$U_k = \hat{V}_m\hat{Y}_kR^{-1}. \tag{4.26}$$

This can be termed an economic approach, because all the computations are done on matrices of small size.

Now, let us consider recycling $l \geq 2$ error approximations. The above trick does not work in this case, because in each cycle we build a new Krylov subspace, which means that in each cycle we have a new matrix $\hat{V}_m$ and $z_m = \hat{V}_m\xi_m$ does not hold anymore for error approximations from previous cycles. There are two ways how to proceed in this case. One is to keep the l previous error approximations $z_i, i = m, \ldots, m-l+1$ and to append them to $\bar{Y}_{k-l} = \hat{V}_m\hat{Y}_{k-l}$, $\bar{Y}_k = \begin{bmatrix}\bar{Y}_{k-l} & z_m & \ldots & z_{m-l+1}\end{bmatrix}$. Then, we compute the reduced QR factorization of $A\bar{Y}_k$ explicitly, i.e. $QR = A\bar{Y}$ and continue according to (4.25)-(4.26). This approach is computationally inefficient. We will use a different approach, which simply retains $\xi_i, i = m, \ldots, m-l+1$, appends them at the end of $\hat{Y}_{k-l}$, thus forming $\hat{Y}_k = \begin{bmatrix}\hat{Y}_{k-l} & \xi_m & \ldots & \xi_{m-l+1}\end{bmatrix}$ and proceeds according to (4.24)-(4.26). Although, this is an inexact approach, it is straightforward and economic and numerical tests show that it significantly improves the convergence of GCRO-DR (see, e.g., Chapter 5). The LGCRO-DR method is given as Algorithm 4.2.

Remark 4.13. We have assumed that l error approximations are always available, which is not the case in the first $l-1$ cycles of each system. In this case we simply use additional harmonic Ritz vectors.

Recall, the main motivation for developing the LGCRO-DR method was the fact that GCRO-DR experiences alternating behaviour of the residual vectors at the end of every cycle, which basically means that the residual vectors at the

end of every other cycle point in nearly the same direction, i.e. $r_{i+1} \approx \alpha r_{i-1}$, thus slowing down the covergence. Next, we give two results from [NLL13] that reveal the skip and sequential angles for GCRO-DR, which correspond to Theorem 3.16 and Theorem 3.18 from Section 3.5.4. The proofs are similar to the ones in [BJM05], so we ommit them here.

Theorem 4.14. *Let r_{i+1} and r_i be the residuals from GCRO-DR restart cycles $i+1$ and i, respectively. Then the sequential angle between them is given by*

$$\cos(\angle(r_{i+1}, r_i)) = \frac{\|r_{i+1}\|_2}{\|r_i\|_2}.$$

Theorem 4.15. *Let r_{i+1} and r_{i-1} be the residuals from GCRO-DR restart cycles $i+1$ and $i-1$, respectively. Then the skip angle between them is given by*

$$\cos(\angle(r_{i+1}, r_{i-1})) = \frac{\|r_{i+1}\|_2}{\|r_{i-1}\|_2} - \frac{\langle Az_{i+1}, Az_i\rangle}{\|r_{i+1}\|_2 \|r_{i-1}\|_2}, \tag{4.27}$$

where $r_{i+1} = r_i - Az_{i+1}$ and $r_i = r_{i-1} - Az_i$.

A similar reasoning as for the LGMRES method can be applied here: the convergence of the GCRO-DR method is related to the sequential and skip angles in the sense that larger angles lead to faster convergence, and interfering the alternating behaviour will probably improve the convergence. This is done by recycling a few error approximations. The following two theorems give the improved results for the skip and sequential angles for LGCRO-DR.

Theorem 4.16. *Let r_{i+1} and r_i be the residuals from LGCRO-DR restart cycles $i+1$ and i, respectively. Then the sequential angle between them is given by*

$$\cos(\angle(r_{i+1}, r_i)) = \frac{\|r_{i+1}\|_2}{\|r_i\|_2}.$$

Theorem 4.17. *Let r_{i+1} and r_{i-1} be the residuals from LGCRO-DR restart cycles $i+1$ and $i-1$, respectively. Then the skip angle between them is given by*

$$\cos(\angle(r_{i+1}, r_{i-1})) = \frac{\|r_{i+1}\|_2}{\|r_{i-1}\|_2}. \tag{4.28}$$

The second term in (4.27) vanishes in (4.28), because the error approximations used by LGCRO-DR are $A^H A$-orthogonal by construction (Theorem 3.14).

Remark 4.18. We will tend always to use the economic approach. In this case, the computational costs, as well as the storage requirements, for LGCRO-DR are the same as for GCRO-DR [Par+06].

Algorithm 4.2: LGCRO-DR

Input : $A^{(i)} \in \mathbb{C}^{n\times n}$ — i-th system matrix
$b^{(i)} \in \mathbb{C}^n$, $x_0^{(i)} \in \mathbb{C}^n$ — right-hand side and initial guess for the i-th system
$m \in \mathbb{N}$ — restart length
$k \in \mathbb{N}$ — number of deflated approximate ev's
U_k^{old} — recycle space from the previous system (if it is defined)

Output: approximate solution $x_m^{(i)}$ to $A^{(i)}x^{(i)} = b^{(i)}$ of the i-th system
k approximate right eigenvectors $\tilde{Y}_k$

$r_0 = b - Ax_0$, $\beta = \|r_0\|$, $v_1 = r_0/\beta$, $i = 1$, $\Xi = [\quad]$
if *We are solving the first system* **then**
  Perform m steps of the Arnoldi process, thus providing the Arnoldi relation $AV_m = V_{m+1}\bar{H}_m$
  Compute $x_1 = x_0 + V_m\xi_m$ and $r_1 = r_0 - V_{m+1}\bar{H}_m\xi_m$, where $\xi_m = \operatorname{argmin}_\xi \left\|\beta e_i - \bar{H}_m\xi\right\|_2$
  $\Xi = [\xi_m \quad \Xi]$
  Compute k right eigenpairs $(1/\theta_i, \hat{y}_i)$ according to Lemma 3.11
  Set $\hat{Y}_k = [\hat{Y}_{k-1} \quad \Xi]$
  Compute U_k and C_k using (4.13)-(4.15)
else
  $AU_k^{\text{old}} = QR$, $C_k = Q$, $U_k = U_k^{\text{old}}R^{-1}$
  $x_1 = x_0 + U_kC_k^Hr_0$, $r_1 = r_0 - C_kC_k^Hr_0$
while *Convergence criteria not satisfied* **do**
  $i = i + 1$
  Perform $m - k$ steps of the Arnoldi process with an operator $(I - C_kC_k^H)A$ and starting with $v_1 = r_{i-1}/\|r_{i-1}\|_2$
  Define $\hat{V}_m$, $\hat{W}_{m+1}$ and G_m such that the Arnoldi relation (4.18) holds
  Compute $x_i = x_{i-1} + \hat{V}_m\xi_m$ and $r_i = r_{i-1} - \hat{W}_{m+1}G_m\xi_m$, where $\xi_m = \operatorname{argmin}_\xi \left\|\hat{W}_{m+1}^Hr_{i-1} - G_m\xi\right\|_2$
  Compute k right eigenpairs $(1/\theta_i, \tilde{y}_i)$ according to Lemma 4.8
  if $\dim(\Xi, 2) < l$ **then**
    $\Xi = [\xi_m \quad \Xi]$
    Set $\hat{Y}_k = [\hat{Y}_{k-\dim(\Xi,2)} \quad \Xi]$
  else
    $\Xi = [\xi_m \quad \Xi(:, 1:l-1)]$
    Set $\hat{Y}_k = [\hat{Y}_{k-l} \quad \Xi]$
  Update U_k and C_k using (4.24)-(4.26)
Carry U_k to the next system

4.4 Truly Deflated GMRES

Motivated by work from Gutknecht in [Gut12], we now develop the method Left-Right Deflated GMRES (LRDGMRES), which is comparable to GCRO-DR in terms of the number of the matrix-vector products, but represents a theoretically better approach, as it will be discussed later in this section. In addition, we have improved it further by exploiting the technique of recycling a few error approximations, which will be covered in Section 4.4.2. We call the second method Loose LRDGMRES (LLRDGMRES).

For a better understanding of our methods, we start by giving some additional details regarding GCRO-DR, that were left out in Section 4.2. Starting with $\hat{r}_0 = (I - C_k C_k^H) r_0$, and denoting $\mathcal{U} = \text{range}(U_k)$ and $\mathcal{C} = \text{range}(C_k)$, the GCRO-DR method constructs a sequence of approximate solutions

$$x_m \in x_0 + \mathcal{K}_m(A_{C_k}, \hat{r}_0) + \mathcal{U}, \tag{4.29}$$

such that the residuals

$$r_m \in r_0 + A\mathcal{K}_m(A_{C_k}, \hat{r}_0) + \mathcal{C} \tag{4.30}$$

are minimized. Using the operator $A_{C_k} = (I - C_k C_k^H)A$ rather than A, where the columns of C_k are the approximate right eigenvectors, a deflation of the smallest eigenvalues occurs, which usually leads to a better convergence behaviour of the method. To sum up, GCRO-DR is a minimum residual method for which the deflation of the matrix is based on an orthogonal projection $P = I - C_k C_k^H$.

However, since we are mostly working with non-Hermitian systems and since we are using GMRES-like methods, it would be preferable to have a method that uses the oblique projection $\hat{P} = I - C_k(\tilde{C}_k^H C_k)^{-1}\tilde{C}_k^H$, with the property that if the nullspace $\mathcal{C}$ of $\hat{P}$ is A-invariant, then so is its range $\tilde{\mathcal{C}}^\perp$, where $\tilde{\mathcal{C}} = \text{range}(\tilde{C}_k)$. Under the assumption that the columns of C_k and $\tilde{C}_k$ are, respectively, (approximate) right and left eigenvectors, this way we do not only annihilate the smallest eigenvalues, but also deflate both, the corresponding left and right eigenspaces, so that convergence depends only on the non-deflated eigenspaces. For this reason, Gutknecht states in [Gut12] that this approach, which is called *truly deflated GMRES*, is theoretically better. To elaborate further, we give the following proposition, which explains the effects of "true" deflation.

Proposition 4.19. *Let us assume that the columns of X_k and Y_k are, respectively, the k right and left eigenvectors of the matrix A corresponding to the k*

smallest eigenvalues. Then

$$\hat{P}A = A\hat{P} = \hat{P}A\hat{P},$$

where $\hat{P} = I - X_k(Y_k^H X_k)^{-1}Y_k^H$ *is an oblique projection. Furthermore, this implies*

$$\mathcal{K}(A, \hat{P}b) = \mathcal{K}(\hat{P}A, \hat{P}b). \tag{4.31}$$

Proof. First we prove that $\hat{P}A = A\hat{P}$. Let X_k and Y_k be, respectively, the k right and left eigenvectors of the matrix A corresponding to the k smallest eigenvalues, which are the diagonal elements of Λ_k. We have

$$\begin{aligned}
\hat{P}A = (I - X_k(Y_k^H X_k)^{-1}Y_k^H)A &= A - X_k(Y_k^H X_k)^{-1}Y_k^H A \\
&= A - X_k(Y_k^H X_k)^{-1}\Lambda_k Y_k^H \\
&= A - X_k(\Lambda_k^{-1}Y_k^H X_k)^{-1}Y_k^H \\
&= A - X_k(Y_k^H A^{-1}X_k)^{-1}Y_k^H \\
&= A - X_k(Y_k^H X_k\Lambda_k^{-1})^{-1}Y_k^H \\
&= A - X_k\Lambda_k(Y_k^H X_k)^{-1}Y_k^H \\
&= A - AX_k(Y_k^H X_k)^{-1}Y_k^H \\
&= A(I - X_k(Y_k^H X_k)^{-1}Y_k^H) = A\hat{P},
\end{aligned}$$

which completes the proof. The second part of the proof is straightforward since

$$\hat{P}A\hat{P} = \hat{P}\hat{P}A = \hat{P}A.$$

□

From (4.31) we conclude that once the deflation of the k smallest eigenvalues has occured, they are completely removed from the spectrum and we do not have to use the deflated operator $\hat{P}A$.

On top of Gutknecht's theory, we provide a cheap way of computing left approximate eigenvectors, without having to build a Krylov subspace for A^H. We now turn to our method.

4.4.1 Left-Right Deflated GMRES

Our method resembles GCRO-DR in most of its parts. It is important to mention that we augment the search space only with approximate right eigenvectors, meaning that both U_k and C_k are needed throughout the algorithm, whereas we need only $\tilde{C}_k$ to build an oblique projection. This information will be helpful in understanding the process of updating the mentioned matrices.

We distinguish two cases, like with GCRO-DR: solving the first system and solving the system $i > 1$. When solving the first system, we do not have any previous information available. Therefore, we perform m steps of the Arnoldi process with A and r_0, thus building a Krylov subspace and obtaining the Arnoldi relation

$$AV_m = V_{m+1}\bar{H}_m. \tag{4.32}$$

And, of course, we compute the new approximate solution and the corresponding residual. Next, we compute left and right approximate eigenvectors, i.e. left and right harmonic Ritz vectors. The right ones, that we denote with $\tilde{y}_i$, can be computed according to Lemma 3.11. As we have pointed out, we provide a cheap way of computing the left harmonic Ritz vectors. The following new lemma shows that we can obtain the left harmonic Ritz vectors (2.21), which we denote with $\tilde{z}_i$, by solving an eigenvalue problem of small size.

Lemma 4.20. *The left harmonic Ritz pairs of A with respect to the subspace $A\mathcal{K}$ are given as $\left(1/\eta_i, AV_m H_m^{-1}\underline{z}_i\right)$, where $\left(1/\bar{\eta}_i, \underline{z}_i\right)$ are the eigenpairs of the matrix*

$$H_m^H + h_m f_m^H,$$

where $\bar{H}_m = \begin{bmatrix} H_m \\ h_m^H \end{bmatrix}$ is the Hessenberg matrix of the Arnoldi relation and $H_m^H f_m = h_m$.

Proof. Starting from Definition 2.25 and using the Arnoldi relation (4.32), we have that a left harmonic Ritz pair $\left(\frac{1}{\eta}, \tilde{z}\right)$ satisfies

$$(AV_m)^H(A^{-H}\tilde{z} - \bar{\eta}\tilde{z}) = 0.$$

Since $\tilde{z} \in A\mathcal{K}$, we have

$$\begin{aligned} & V_m^H A^H A^{-H} A V_m \hat{z} = \bar{\eta} V_m^H A^H A V_m \hat{z} \\ \Leftrightarrow\ & V_m^H V_{m+1} \bar{H}_m \hat{z} = \bar{\eta} \bar{H}_m^H V_{m+1}^H V_{m+1} \bar{H}_m \hat{z} \\ \Leftrightarrow\ & \tilde{I} \bar{H}_m \hat{z} = \bar{\eta} \bar{H}_m^H \bar{H}_m \hat{z}, \end{aligned}$$

where $\tilde{I}_m = \begin{bmatrix} I_m & \mathbf{o} \end{bmatrix}$, and $\mathbf{o}$ is a column vector of zeros. Writing $\bar{H}_m = \begin{bmatrix} H_m \\ h_m^H \end{bmatrix}$, we obtain

$$\begin{aligned} & H_m \hat{z} = \bar{\eta}(H_m^H H_m + h_m h_m^H)\hat{z} \\ \Leftrightarrow\ & \frac{1}{\bar{\eta}} H_m \hat{z} = (H_m^H + h_m \left(H_m^{-H} h_m\right)^H) H_m \hat{z} \\ \Leftrightarrow\ & \frac{1}{\bar{\eta}} \underline{z} = (H_m^H + h_m f_m^H)\underline{z}. \end{aligned}$$

Thus, $(\frac{1}{\eta}, \underline{z})$ is an eigenpair of the matrix $H_m^H + h_m f_m^H$. □

Remark 4.21. From Lemma 3.11 and Lemma 4.20 we can conclude that in order to obtain left and right harmonic Ritz vectors, we first have to compute left and right eigenvectors of the *same matrix* $H_m + f_m h_m^H$. However, we recover the harmonic Ritz vectors in a different way, i.e. we recover the right harmonic Ritz vectors as

$$\tilde{y} = AV_m \hat{y} = V_{m+1} \bar{H}_m \hat{y},$$

whereas we get the left ones in the following way

$$\begin{aligned} \tilde{z} &= AV_m H_m^{-1} \underline{z} \\ &= V_{m+1} \bar{H}_m H_m^{-1} \underline{z} \\ &= V_{m+1} \begin{bmatrix} I_m \\ h_m^H H_m^{-1} \end{bmatrix} \underline{z}. \end{aligned} \tag{4.33}$$

The next step is to obtain matrices U_k, C_k and $\tilde{C}_k$, so we can proceed with the GCRO framework. Updating U_k and C_k is done in the same way as for GCRO-DR, i.e. according to (4.13)-(4.15). However, updating $\tilde{C}_k$ seems to be the tricky part. Due to Stewart [Ste11], for stability reasons, we would like to have *orthonormal* matrices C_k and $\tilde{C}_k$ when applying the oblique projection.

From (4.33) it follows that the left harmonic Ritz vectors are retrieved in a similar way as the right ones, i.e. AV_m times a vector. Therefore, we decided to treat them in the same way, meaning that we update $\tilde{C}_k$ *inexpensively* as follows

$$\bar{H}_m \hat{Z}_k = Q_L R_L \tag{4.34}$$

$$\tilde{C}_k = V_{m+1} Q_L. \tag{4.35}$$

Now, let us consider solving the system $i > 1$. The first question that arises is which matrices do we carry from the previous system? Since we need both matrices, U_k and C_k, and since we want to "copy" the framework of GCRO-DR, we keep the U_k^{old} from the previous system, computed the last, whose columns are preimages of approximate right eigenvectors. Furthermore, we obtain the new U_k and C_k like for GCRO-DR, i.e.

$$A^{(i+1)} U_k^{\text{old}} = QR \tag{4.36}$$

$$\begin{aligned} C_k^{\text{new}} &= Q \\ U_k^{\text{new}} &= U_k^{\text{old}} R^{-1}. \end{aligned} \tag{4.37}$$

On the other hand, we have discussed that only $\tilde{C}_k$ is required throughout the method. Therefore, we can avoid any additional work (updating $\tilde{C}_k$ similarly to (4.36)-(4.37) is not necessary) by keeping $\tilde{C}_k^{\text{old}}$ from the previous system, computed the last.

Now we have the whole setup for discussing the main loop of the method. For notational convenience, we drop the superscripts $(i+1)$, old and new in the remainder.

Starting with $\hat{r}_0 = (I - C_k(\tilde{C}_k^H C_k)^{-1}\tilde{C}_k^H)r_0$, we perform $m-k$ steps of the Arnoldi process with an operator $A_{C_k}^{\tilde{C}_k} = (I - C_k(\tilde{C}_k^H C_k)^{-1}\tilde{C}_k^H)A$, leading to the Arnoldi relation

$$(I - C_k(\tilde{C}_k^H C_k)^{-1}\tilde{C}_k^H)AV_{m-k} = V_{m-k+1}\bar{H}_{m-k},$$

which can be rewriten as

$$A\begin{bmatrix} U_k & V_{m-k}\end{bmatrix} = \begin{bmatrix} C_k & V_{m-k+1}\end{bmatrix}\begin{bmatrix} I_k & B_k \\ 0 & \bar{H}_{m-k}\end{bmatrix}, \tag{4.38}$$

where $B_k = (\tilde{C}_k^H C_k)^{-1}\tilde{C}_k^H AV_{m-k}$. As in GCRO-DR, to avoid unnecessary ill-conditioning of the rightmost matrix in (4.38), we can compute the diagonal matrix D_k, such that $\hat{U}_k = U_k D_k$ has unit columns. Defining

$$\tilde{V}_m = \begin{bmatrix}\hat{U}_k & V_{m-k}\end{bmatrix}, \quad \tilde{W}_{m+1} = \begin{bmatrix} C_k & V_{m-k+1}\end{bmatrix}, \quad \tilde{G}_m = \begin{bmatrix} D_k & B_k \\ 0 & \bar{H}_{m-k}\end{bmatrix},$$

the Arnoldi relation (4.38) becomes

$$A\tilde{V}_m = \tilde{W}_{m+1}\tilde{G}_m. \tag{4.39}$$

A relevant difference in comparison to GCRO-DR is that the matrix $\tilde{W}_{m+1}$ is not unitary anymore.

Now we want to update the solution as (recall (4.29))

$$x_m = x_0 + V_{m-k}s_{m-k} + U_k t_k, \tag{4.40}$$

such that the residual (recall (4.30))

$$r_m = r_0 - AV_{m-k}s_{m-k} - C_k t_k \tag{4.41}$$

is minimized. Denoting $\underline{C}_k = \tilde{C}_k(\tilde{C}_k^H C_k)^{-H}$, $\hat{P} = I - C_k\underline{C}_k^H$ and $\hat{Q} = C_k\underline{C}_k^H$ and due to the decomposition

$$\begin{aligned} r_0 &= \hat{P}r_0 + \hat{Q}r_0 \\ &= \hat{r}_0 + \hat{Q}r_0 \\ &= v_1\beta + \hat{Q}r_0, \end{aligned}$$

with $\beta = \|\hat{r}_0\|_2$, the residual (4.41) can be rewriten as

$$\begin{aligned} r_m &= v_1\beta + \hat{Q}r_0 - (\hat{P}+\hat{Q})AV_{m-k}s_{m-k} - C_k t_k \\ &= v_1\beta + C_k\underline{C}_k^H r_0 - V_{m-k+1}\bar{H}_{m-k}s_{m-k} - C_k\underline{C}_k^H AV_{m-k}s_{m-k} - C_k t_k \\ &= [C_k \quad V_{m-k+1}]\, q_m, \end{aligned}$$

where

$$q_m = \begin{bmatrix} q_m^0 \\ q_m^\perp \end{bmatrix} = \begin{bmatrix} \underline{C}_k^H r_0 \\ e_1\beta \end{bmatrix} - \begin{bmatrix} I & B_k \\ 0 & \bar{H}_{m-k} \end{bmatrix} \begin{bmatrix} t_k \\ s_{m-k} \end{bmatrix}$$

is the *truly deflated GMRES quasi-residual*. As we tend to point out differences to GCRO-DR throughout this section, we state here that for LRDGMRES $\|r_m\|_2 \neq \|q_m\|_2$, since the columns of C_k are not orthogonal to V_{m-k+1} anymore. However, taking into account that

$$r_m = C_k q_m^0 + V_{m-k+1}q_m^\perp, \quad \text{with} \quad C_k q_m^0 = \hat{Q}r_m \in \mathcal{C}, \quad V_{m-k+1}q_m^\perp = \hat{P}r_m \in \underline{\mathcal{C}}^\perp,$$

with $\underline{\mathcal{C}} = \mathrm{range}(\underline{C}_k)$, we have at least

$$\|q_m\|_2^2 = \|q_m^0\|_2^2 + \|q_m^\perp\|_2^2 = \left\|\hat{Q}r_m\right\|_2^2 + \left\|\hat{P}r_m\right\|_2^2.$$

Therefore, we will minimize $\|q_m\|_2$ instead of $\|r_m\|_2$, which amounts to solving the least-squares problem of small size $(m-k+1)\times(m-k)$ for s_{m-k}

$$s_{m-k} = \underset{s\in\mathbb{C}^{m-k}}{\mathrm{argmin}} \left\|e_1\beta - \bar{H}_{m-k}s\right\|_2, \tag{4.42}$$

and solving directly for t_k

$$t_k = \underline{C}_k^H r_0 - B_k s_{m-k}, \tag{4.43}$$

such that $q_m^0 = 0$. Having s_{m-k} and t_k computed, we can update the solution and the residual according to (4.40) and (4.41), respectively.

What is left is to compute left and right harmonic Ritz vectors, and update U_k, C_k and $\tilde{C}_k$ so we can proceed with the next cycle (system). The following lemma provides a result similar to Lemma 4.8. The proof is left out, since it differs from the proof of Lemma 4.8 only by the fact that $\tilde{W}_{m+1}$ is not orthonormal anymore, and therefore the term $\tilde{W}_{m+1}^H\tilde{W}_{m+1}$ appears on the left-hand side.

Lemma 4.22. *Solving the generalized eigenvalue problem*

$$\theta_i\tilde{G}_m^H\tilde{W}_{m+1}^H\tilde{W}_{m+1}\tilde{G}_m\hat{y}_i = \tilde{G}_m^H\tilde{W}_{m+1}^H\tilde{V}_m\hat{y}_i,$$

one obtains the k right harmonic Ritz vectors (corresponding to the k harmonic Ritz values of smallest magnitude $1/\theta_i, i = 1, \dots, k$*) in the main loop of the LRDGMRES method (***while** *loop in Algorithm 4.3 below) as* $\tilde{y}_i = A\tilde{V}_m\hat{y}_i, i = 1, \dots, k$.

As discussed above, we continue to treat the left harmonic Ritz vectors as the right ones. We give the following result for their computation.

Lemma 4.23. *Solving the generalized eigenvalue problem*

$$\bar{\eta}_i \tilde{G}_m^H \tilde{W}_{m+1}^H \tilde{W}_{m+1} \tilde{G}_m \hat{z}_i = \tilde{V}_m^H \tilde{W}_{m+1} \tilde{G}_m \hat{z}_i,$$

one obtains the k left harmonic Ritz vectors (corresponding to the k harmonic Ritz values of smallest magnitude $1/\eta_i, i = 1, \dots, k$*) in the main loop of the LRDGMRES method (***while** *loop in Algorithm 4.3 below) as* $\tilde{z}_i = A\tilde{V}_m\hat{z}_i, i = 1, \dots, k$.

Proof. Using Definition 2.25 and the Arnoldi relation (4.39) we have with $\tilde{z}_i = A\tilde{V}_m\hat{z}_i, i = 1, \dots, k$

$$\begin{aligned} & (A\tilde{V}_m)^H (A^{-H} A\tilde{V}_m\hat{z}_i - \bar{\eta}_i A\tilde{V}_m\hat{z}_i) = 0 \\ \Leftrightarrow\ & \bar{\eta}_i \tilde{V}_m^H A^H A\tilde{V}_m\hat{z}_i = \tilde{V}_m^H A\tilde{V}_m\hat{z}_i \\ \Leftrightarrow\ & \bar{\eta}_i \tilde{G}_m^H \tilde{W}_{m+1}^H \tilde{W}_{m+1} \tilde{G}_m \hat{z}_i = \tilde{V}_m^H \tilde{W}_{m+1} \tilde{G}_m \hat{z}_i \end{aligned}$$

which completes the proof. □

Recall, updating U_k, C_k and $\tilde{C}_k$ requires the QR-decomposition of the matrices

$$\begin{aligned} A\tilde{V}_m\hat{Y} &= \tilde{W}_{m+1}\tilde{G}_m\hat{Y} \quad \text{and} \\ A\tilde{V}_m\hat{Z} &= \tilde{W}_{m+1}\tilde{G}_m\hat{Z}. \end{aligned}$$

Provided that $\tilde{W}_{m+1}$ is unitary, one could obtain the wanted updates in a similar fashion like for GCRO-DR, i.e. according to (4.20)-(4.22). This is a cheap way, since we need to compute the QR-decomposition of matrices of small size, i.e. of $\tilde{G}_m\hat{Y}, \tilde{G}_m\hat{Z} \in \mathbb{C}^{(m+1)\times k}$. Nonetheless, the unitarity requirement does not hold for LRDGMRES, which is one of the disadvantages of our method. Instead, we have to compute updates as follows

$$\tilde{W}_{m+1}\tilde{G}_m\hat{Y}_k = QR \tag{4.44}$$

$$C_k = Q \tag{4.45}$$

$$U_k = \tilde{V}_m\hat{Y}_k R^{-1} \tag{4.46}$$

for the right harmonic Ritz vectors and

$$\tilde{W}_{m+1}\tilde{G}_m\hat{Z}_k = Q_L R_L \tag{4.47}$$

$$\tilde{C}_k = Q_L \tag{4.48}$$

for the left ones, where as before, we handle them in the same way as the right ones. Formulas (4.44) and (4.47) represent the QR-decomposition of two matrices of size $n \times k$, where n is the size of the system. Considering that the restart length m is usually much smaller than n, it is obvious why these steps are computationally costly. At the end of each cycle (system) we keep U_k and $\tilde{C}_k$. We give LRDGMRES as Algorithm 4.3.

Remark 4.24. From Remark 2.26 it follows that in the Hermitian case, right and left harmonic Ritz vectors are equal. Hence, in this case the LRDGMRES method is equivalent to the GCRO-DR method.

Remark 4.25. For stability reasons, the application of the oblique projection to a vector x is not done straightforwardly. Instead, we use the set of formulas (for more details see, e.g., [Ste11])

1. $\tilde{C}_k^H C_k = QR$ (reduced QR-decomposition)
2. $f_1 = \tilde{C}_k^H x$
3. $f_2 = Q^H f_1$
4. Solve the system $Rf_3 = f_2$ for f_3
5. Compute $C_k f_3$.

Remark 4.26. LRDGMRES requires somewhat more computational effort and slightly higher storage requirements in comparison to GCRO-DR. We have to compute additionally the left harmonic Ritz vectors, meaning that we have to solve two generalized eigenvalue problems of size $m \times m$ per cycle. Furthermore, it is necessary to compute the QR-decomposition of two matrices of size $n \times k$, i.e. (4.44) and (4.47), where n is the size of the system. In contrast to the GCRO-DR method, where the QR-decomposition of a smaller matrix of size $m \times k$, $m << n$ is needed, these computations for LRDGMRES are quite costly. Additional storage is necessary for keeping $\tilde{C}_k$.

Algorithm 4.3: LRDGMRES

Input :	$A^{(i)} \in \mathbb{C}^{n\times n}$	i-th system matrix
	$b^{(i)} \in \mathbb{C}^n$, $x_0^{(i)} \in \mathbb{C}^n$	right-hand side and initial guess for the i-th system
	$m \in \mathbb{N}$	restart length
	$k \in \mathbb{N}$	number of deflated approximate ev's
	$U_k^{\text{old}}, \tilde{C}_k^{\text{old}}$	recycle spaces from the previous system (if they are defined)
Output:	approximate solution $x_m^{(i)}$ to $A^{(i)}x^{(i)} = b^{(i)}$ of the i-th system	
	k approximate left and right eigenvectors $\tilde{Z}_k$ and $\tilde{Y}_k$	

$r_0 = b - Ax_0$, $\beta = \|r_0\|$, $v_1 = r_0/\beta$, $i = 1$
if *We are solving the first system* **then**
  Perform m steps of the Arnoldi process, thus providing the Arnoldi relation $AV_m = V_{m+1}\bar{H}_m$
  Compute $x_1 = x_0 + V_m\xi_m$ and $r_1 = r_0 - V_{m+1}\bar{H}_m\xi_m$, where $\xi_m = \operatorname{argmin}_\xi \left\|\beta e_i - \bar{H}_m\xi\right\|_2$
  Compute k right harmonic Ritz vectors according to Lemma 3.11
  Compute k left harmonic Ritz vectors according to Lemma 4.20
  Compute U_k and C_k using (4.13)-(4.15)
  Compute $\tilde{C}_k$ using (4.34)-(4.35)
else
  $AU_k^{\text{old}} = QR$, $C_k = Q$, $U_k = U_k^{\text{old}}R^{-1}$
  $\tilde{C}_k = \tilde{C}_k^{\text{old}}$
  $x_1 = x_0 + U_k(\tilde{C}_k^H C_k)^{-1}\tilde{C}_k^H r_0$, $r_1 = r_0 - C_k(\tilde{C}_k^H C_k)^{-1}\tilde{C}_k^H r_0$
while *Convergence criteria not satisfied* **do**
  $i = i + 1$
  Perform $m - k$ steps of the Arnoldi process with an operator $\hat{P}A$ and starting with $v_1 = r_{i-1}/\left\|r_{i-1}\right\|_2$
  Define $\tilde{V}_m$, $\tilde{W}_{m+1}$ and $\tilde{G}_m$ such that the Arnoldi relation (4.39) holds
  Compute x_i and r_i according to (4.40), (4.41), (4.42) and (4.43)
  Compute k right harmonic Ritz vectors according to Lemma 4.22
  Compute k left harmonic Ritz vectors according to Lemma 4.23
  Update U_k and C_k using (4.44)-(4.46)
  Update $\tilde{C}_k$ using (4.47)-(4.48)
Carry U_k and $\tilde{C}_k$ to the next system

Theorem 4.12 also holds for our method, because it is equivalent to GCRO-DR in case that A is Hermitian.

Theorem 4.27. *Let A be Hermitian positive definite and have the eigendecomposition given in* (4.23)*, and let $E, \varepsilon, \eta, \mu$ and $\hat{\mu}$ be defined as in Section 4.2.1.*

Then there exists a matrix $\hat{Q}_1$ *conforming to* Q_1*, such that* $\mathrm{range}(\hat{Q}_1)$ *is a simple invariant subspace of* $A + E$*, and*

$$\tan\theta_1 \left(\mathrm{range}(Q_1), \mathrm{range}(\hat{Q}_1)\right) \leq \tfrac{\varepsilon}{\tilde{\mu}}.$$

Recall, Theorem 4.27 states that if the recycle space reasonably approximates an invariant subspace, then it will also provide a reasonable approximation to the slightly perturbed invariant subspace of the updated matrix. Since we are augmenting the Krylov subspace only with the approximate right eigenvectors, like in GCRO-DR, the previously stated result holds.

Further convergence analysis of LRDGMRES is beyond the scope of this thesis. A similar result to the one of Theorem 4.11 is not easily obtained, since we are dealing with oblique projections.

4.4.2 Loose Left-Right Deflated GMRES

In addition to LRDGMRES, we have developed the method Loose LRDGMRES (LLRDGMRES), which combines recycling approximate eigenvectors and error approximations, similarly to GCRO-DR. It is not surprising that LRDGMRES experiences alternating behaviour of residuals at the end of every cycle, since it is a GMRES-like method. Therefore, it is reasonable to conclude that changing such a behaviour would lead to a more efficient method. So far we have introduced two "loose" methods, meaning that most of the details have been provided in Section 4.3 and Section 3.5.4. Hence, we put our attention in this section only on the differences between our method and LGCRO-DR. We will, of course, use the economic approach discussed in Section 4.3.

Basically, there are two major differences that have to be considered. First, the matrix $\tilde{W}_{m+1}$ is no longer unitary. This means that instead of Lemma 4.8 we need to use Lemma 4.22 for updating the right harmonic Ritz vectors. The second point is that left harmonic Ritz vectors play an important role and that we should also augment the subspace spanned by them. We need $\tilde{C}_k$ only for building the oblique projection, and not for augmenting the search space. The immediate idea was to treat left harmonic Ritz vectors the same way like the right ones. After all, this is what we have been doing throughout the method (LRDGMRES). In other words, we proceed as follows

1. Append $\xi_i, i = 1, \ldots, m-l+1$, where $\xi_i = \begin{bmatrix} t_k^{(i)} \\ s_{m-k}^{(i)} \end{bmatrix}$, to both $\hat{Y}_{k-l}$ and $\hat{Z}_{k-l}$
2. Update U_k and C_k according to (4.44)-(4.46)

3. Update $\tilde{C}_k$ according to (4.47)-(4.48).

Numerical tests in the next chapter will validate that this seems to be a reasonable approach. We give LLRDGMRES as Algorithm 4.4.

Remark 4.28. In the economic approach, LLRDGMRES has the same storage and computational cost requirements as LRDGMRES.

We would like to conclude this section by stating that the results of Theorem 4.14, Theorem 4.15, Theorem 4.16 and Theorem 4.17 also hold for LLRDGMRES.

Algorithm 4.4: LLRDGMRES

Input : $A^{(i)} \in \mathbb{C}^{n\times n}$ — i-th system matrix
$b^{(i)} \in \mathbb{C}^n$, $x_0^{(i)} \in \mathbb{C}^n$ — right-hand side and initial guess for the i-th system
$m \in \mathbb{N}$ — restart length
$k \in \mathbb{N}$ — number of deflated approximate ev's
$U_k^{\text{old}}, \tilde{C}_k^{\text{old}}$ — recycle spaces from the previous system (if they are defined)

Output: approximate solution $x_m^{(i)}$ to $A^{(i)}x^{(i)} = b^{(i)}$ of the i-th system
k approximate left and right eigenvectors $\tilde{Z}_k$ and $\tilde{Y}_k$

$r_0 = b - Ax_0$, $\beta = \|r_0\|$, $v_1 = r_0/\beta$, $i = 1$, $\Xi = [\quad]$
if *We are solving the first system* **then**
  Perform m steps of the Arnoldi process, thus providing the Arnoldi relation $AV_m = V_{m+1}\bar{H}_m$
  Compute $x_1 = x_0 + V_m\xi_m$ and $r_1 = r_0 - V_{m+1}\bar{H}_m\xi_m$, where $\xi_m = \operatorname{argmin}_\xi \left\|\beta e_i - \bar{H}_m\xi\right\|_2$
  $\Xi = [\xi_m \quad \Xi]$
  Compute k right harmonic Ritz vectors according to Lemma 3.11
  Compute k left harmonic Ritz vectors according to Lemma 4.20
  Set $\hat{Y}_k = [\hat{Y}_{k-1} \quad \Xi]$ and $\hat{Z}_k = [\hat{Z}_{k-1} \quad \Xi]$
  Compute U_k and C_k using (4.13)-(4.15)
  Compute $\tilde{C}_k$ using (4.34)-(4.35)
else
  $AU_k^{\text{old}} = QR$, $C_k = Q$, $U_k = U_k^{\text{old}}R^{-1}$, $\tilde{C}_k = \tilde{C}_k^{\text{old}}$
  $x_1 = x_0 + U_k(\tilde{C}_k^H C_k)^{-1}\tilde{C}_k^H r_0$, $r_1 = r_0 - C_k(\tilde{C}_k^H C_k)^{-1}\tilde{C}_k^H r_0$
while *Convergence criteria not satisfied* **do**
  $i = i + 1$
  Perform $m - k$ steps of the Arnoldi process with an operator $\hat{P}A$ and starting with $v_1 = r_{i-1}/\|r_{i-1}\|_2$
  Define $\tilde{V}_m$, $\tilde{W}_{m+1}$ and $\tilde{G}_m$ such that the Arnoldi relation (4.39) holds
  Compute x_i and r_i according to (4.40), (4.41), (4.42) and (4.43)
  Compute k right harmonic Ritz vectors according to Lemma 4.22
  Compute k left harmonic Ritz vectors according to Lemma 4.23
  if $\dim(\Xi, 2) < l$ **then**
    $\Xi = [\xi_m \quad \Xi]$
    Set $\hat{Y}_k = [\hat{Y}_{k-\dim(\Xi,2)} \quad \Xi]$ and $\hat{Z}_k = [\hat{Z}_{k-\dim(\Xi,2)} \quad \Xi]$
  else
    $\Xi = [\xi_m \quad \Xi(:, 1:l-1)]$
    Set $\hat{Y}_k = [\hat{Y}_{k-l} \quad \Xi]$ and $\hat{Z}_k = [\hat{Z}_{k-l} \quad \Xi]$
  Update U_k and C_k using (4.44)-(4.46)
  Update $\tilde{C}_k$ using (4.47)-(4.48)
Carry U_k and $\tilde{C}_k$ to the next system

5 Applications and Numerical Results

For the purpose of comparing the methods introduced in the previous chapter, we have considered four different applications and one artificially contrived example. We compare the methods by the number of matrix-vector products needed until the method converges. All tests were done in MATLAB.

In order to support the claims of Remark 4.24, which states that LRDGMRES and GCRO-DR exhibit the same performance in the Hermitian case, we consider first an application in fracture mechanics. The simulation of crack propagation in a metal plate results in a sequence of symmetric positive definite matrices where both, the matrix and the right-hand side change slowly from one system to the next [Par+06]. We show results for 10 consecutive systems that were provided by M. Parks [Par06]. Typically, over 2000 systems have to be solved.

The rest of the applications involve non-Hermitian matrices. We start by a small 3×3 example that we constructed with an intent to show that the deflated approach used by GCRO-DR can encounter problems, such that the method converges slowly or even stagnates, whereas the true deflation, which we exploit in LRDGMRES, overcomes those problems and the respective methods converge significantly faster. Intuitively, vice versa cannot happen, i.e. if we encounter problems with the true deflation, then the methods based on the deflation should experience similar problems. However, this intuition is not backed by a theoretical proof.

Next, we consider a problem in fluid dynamics [Boi+96]. Modeling 2D fluid flow in a driven cavity results in a system with a nonsymmetric, indefinite and sparse matrix. Even though the true effects of recycling can be observed when solving a sequence of linear systems, this application found its way into this thesis, since it was used also in [NLL13] as an example where loose GCRO-DR outperforms GCRO-DR. We will show that our methods, i.e. LRDGMRES and loose LRDGMRES outperform the corresponding methods GCRO-DR and loose GCRO-DR. In addition, we will demonstrate that the skip angles for LGCRO-DR and LLRDGMRES are notably larger, thus (partly) explaining

why these methods perform better.

In Section 5.4 we consider systems arising in the Korringa-Kohn-Rostoker method [Bol+12] when performing electronic structure calculations based on the Schrödinger equation. The electronic structure is used to predict key physical properties, such as charge densities, total energy, force etc., in theoretical chemistry and solid-state physics.

Due to our involvement in the projects *Marie Curie Initial Training Network STRONGnet* and *SFB/Transregio 55 Hadronenphysik mit Gitter-QCD*, the application that was of the most interest for us was lattice QCD [DeT06; GL09; MM94]. QCD is a quantum field theory that describes interactions between quarks and gluons. Numerical simulations, requiring the discretization of the theory on the lattice, result in a sequence of slowly changing Dirac operators. Inverting the Dirac operator gives us information on the propagation of the quark. We describe the theory in Section 5.5 from the mathematical point of view [Bra+16; Fro+14]. In addition, we introduce the red-black multiplicative Schwarz method, which is used as a preconditioner in lattice QCD [FNZ12; Lüs04; Lüs07], and show results for the preconditioned case. Furthermore, we will briefly discuss the concept of *gauge fixing* for a Schwinger model [Sch62]. The Schwinger model is a quantum field theory in two dimensions, which can be solved exactly, in the sense that an analytical expression for observables (such as masses of particles) can be derived [Gat95; Smi97]. In the Schwinger model, the computations are cheaper and realistic systems are much smaller, and hence this model is used for tests. An important property of quantum field theories is *gauge invariance*, which on the level of the discretizations manifests itself by the fact that we can modify the so-called gauge links via conjugation without altering the theory. We can use such transformations to increase the similarity between the elements of a sequence of matrices as they typically arise in a numerical simulation. This "gauge fixing" will be discussed in detail in Section 5.5.3.

Proposition 4.19 suggests that in case of *exact* left and right eigenvectors, $\mathcal{K}(A, \hat{P}b) = \mathcal{K}(\hat{P}A, \hat{P}b)$, meaning that we do not need to work with an operator $\hat{P}A$, which might save us substantial costs. In Section 5.6 we show that proceeding in this manner with the approximate eigenvectors, i.e. not applying the oblique projection on A within the framework of LRDGMRES, can indeed result in a similar behavior to LRDGMRES, of course depending on the application.

5.1 Fatigue and Fracture Problems in Mechanics

This first example will serve to back up the claim of Remark 4.24, i.e. to verify that LRDMGRES and GCRO-DR are equivalent in the Hermitian case. We will consider a sequence of 10 linear systems provided by Michael Parks. They are taken from a finite element code that simulates crack propagation in a metal plate using *cohesive finite elements*. The plate mesh is shown in Figure 5.1 (the figure is provided by M. Parks, E. de Sturler et al. [Par+04]). The problem is symmetric about the x-axis and the crack propagates along the symmetry axis. The cohesive elements represent nonlinear springs connecting the surfaces that will define the crack location. As the crack propagates, the cohesive elements deform and eventually break. The element stiffness is set to zero for a broken element. This simulation results in a sequence of sparse, symmetric and positive definite stiffness matrices $A^{(i+1)} = A^{(i)} + \Delta A^{(i)}$ that change slowly from one system to the next. The matrices produced are of size 3988×3988, and have a condition number on the order of 10^4. Usually, over 2000 linear systems have to be solved to capture the fracture progression.

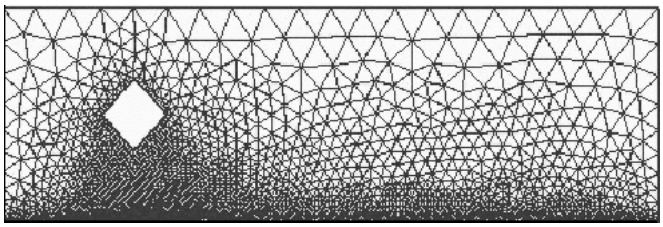


Figure 5.1: Two-dimensional plate mesh for the crack propagation problem.

We compare four methods for this problem: GCRO-DR(40,20), LRDGMRES(40,20), LGCRO-DR(40,18,2) and LLRDGMRES(40,18,2). The numbers in Table 5.1 represent the cumulative number of matrix-vector products necessary to solve the first i systems. In other words, the first number in the column for GCRO-DR means that the method requires 498 matrix-vector products to solve the first system, whereas the second number points out that the method needs 725 matrix-vector products for solving the first 2 systems, meaning that the second system requires $725 - 498 = 227$ matrix-vector products. We see that GCRO-DR and LRDGMRES perform exactly the same for this example, which substantiates the claims of Remark 4.24. In [NLL13], the authors use this example to show that LGCRO-DR converges faster than GCRO-DR, which can be also seen in Table 5.1. The overal gain is around 6%. We would expect that both loose methods behave the same, which seems not to be the case. One explanation could be that even though the matrix

is Hermitian, meaning that the left and right eigenvectors are the same, we still recompute the left harmonic Ritz vectors according to Lemma 4.20 and Lemma 4.23, and we still perform the QR-decompositions (4.44) and (4.47), which might result in a slightly different numerical behavior. We see this when comparing the numbers in the last row in Table 5.1.

Table 5.1: The total number of iterations required to solve 10 consecutive linear systems.

System	GCRO-DR	LRDGMRES	Loose GCRO-DR	Loose LRDGMRES
1	498	498	476	477
2	725	725	688	687
3	924	924	866	866
⋮	⋮	⋮	⋮	⋮
10	2241	2241	2110	2103

All four methods are advanced methods based on concepts of augmentation and deflation, and comparing them only between each other does not give an insight in the improvement with respect to the already existing methods like full GMRES, CG or even GMRES-DR. Therefore, we would like to point out that in [Par+06], the authors compared GCRO-DR with the aforementioned methods for 151 consecutive linear systems. For example, GCRO-DR needs 250 matrix-vector products less per system than full GMRES (about 55% matrix-vector products less per system), which is an amazing improvement considering that one has to solve over 2000 systems.

5.2 Comparison of Deflated and Truly Deflated GMRES

The purpose of this section is to show that a situation can occur for which GCRO-DR experiences stagnation or slow convergence, whereas LRDGMRES converges significantly faster, as a consequence of peforming "true" deflation. To illustrate this we simply compare GMRES with the operator $A_X = (I - XX^H)A$ (GMRES_ORTH) and GMRES with the operator $A_X^Y = (I - X(Y^HX)^{-1}Y^H)A$ (GMRES_OBL), where X and Y are built from exact right and left eigenvectors, which are in the basis of GCRO-DR and LRDGMRES, respectively. We will demonstrate on a small 3×3 example that GM-

RES_ORTH stagnates, whereas GMRES_OBL exhibits fast convergence. All computations are done in MATLAB. Consider the matrix

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Using *eig* in MATLAB, we compute the eigenvalues, which are the diagonal elements in the matrix D, and the right and left eigenvectors (columns of X and Y, respectively)

$$D = \begin{bmatrix} -0.618 & 0 & 0 \\ 0 & 1.618 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$X = \begin{bmatrix} -0.8507 & -0.5257 & 0 \\ 0 & 0 & 0.7071 \\ 0.5257 & -0.8507 & -0.7071 \end{bmatrix},$$

$$Y = \begin{bmatrix} -0.7529 & -0.4004 & 0 \\ 0.4653 & -0.6479 & 1 \\ 0.4653 & -0.6479 & 0 \end{bmatrix},$$

such that $AX = XD$ and $Y^H A = DY^H$ holds. Assuming that we want to deflate the eigenvalue of the smallest magnitude, i.e. $\lambda_1 = D(1,1)$, we obtain the orthogonal projection

$$P = I - x_1 x_1^H = \begin{bmatrix} 0.2764 & 0 & 0.4472 \\ 0 & 1 & 0 \\ 0.4472 & 0 & 0.7236 \end{bmatrix},$$

and the oblique projection

$$\hat{P} = I - x_1 (y_1^H x_1)^{-1} y_1^H = \begin{bmatrix} 0.2764 & 0.4472 & 0.4472 \\ 0 & 1 & 0 \\ 0.4472 & -0.2764 & 0.7236 \end{bmatrix},$$

where x_1 and y_1 are the right and the left eigenvector, corresponding to λ_1. Running GMRES(1) on two matrices PA and $\hat{P}A$, with a starting vector $x_0 = 0$, a tolerance of 10^{-8} and a carefully chosen right-hand side

$$b = \begin{bmatrix} 0.4472 \\ 0 \\ -0.2764 \end{bmatrix},$$

we get that GMRES_ORTH requires around 240000 matrix-vector products until it converged, whereas GMRES_OBL needs only 1400 matrix-vector products. In exact arithmetic this means that GMRES_ORTH stagnates. The reason for that lies in the choice of the right-hand side, which is chosen such that $b \in \text{null}(P) = \text{null}(\hat{P})$. Looking at line 6 of the Arnoldi process (Algorithm 2.1) with $v_1 = b/\beta, \beta = \|b\|_2$, we conclude that $h_{11} = v_1^H P A v_1 = (P^H v_1)^H A v_1 = 0$, since $P^H = P$. Furthermore, solving the least squares problem

$$y_1 = \underset{y}{\text{argmin}} \left\| \beta e_1 - \bar{H}_1 y \right\|_2 = \underset{y}{\text{argmin}} \left\| \begin{bmatrix} \beta e_1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ h_{21} \end{bmatrix} y \right\|_2$$

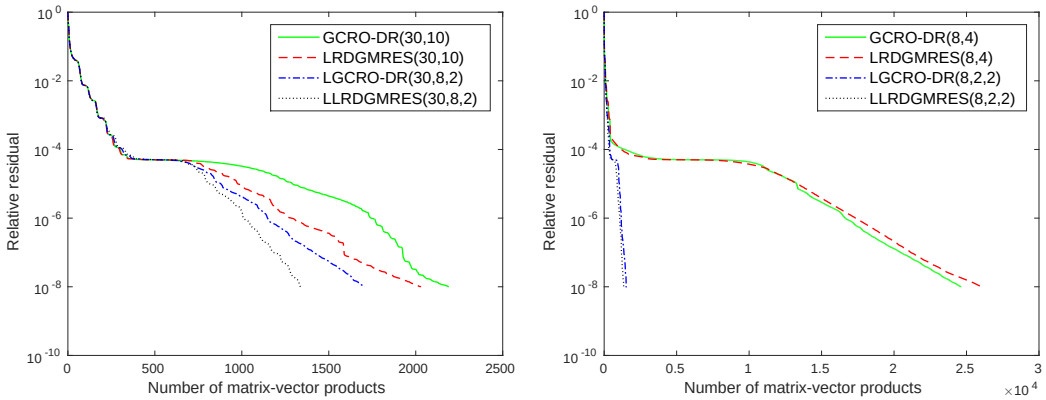
we obtain $y_1 = 0$, meaning that the new solution x_1 is equal to x_0, and hence the method stagnates. On the other hand, $\hat{P}^H \neq \hat{P}$, and as a consequence $h_{11} \neq 0$, meaning that GMRES_OBL does not suffer from stagnation.

5.3 Matrix Market Collection

The true effects of recycling can be seen when solving a sequence of systems. However, for the purpose of comparing the methods introduced in the previous chapters, we consider first an application in fluid dynamics that results in solving just one system with a nonsymmetric matrix. For modeling 2D fluid flow in a driven cavity one can use the incompressible Navier Stokes equations. These are discretized using the Galerkin Finite Element method, and linearized using Newton's method. The resulting matrix is nonsymmetric, indefinite and sparse. We have chosen the matrix *cavity10* from the Matrix Market website at NIST [Boi+96], which was also used in [NLL13] as an example. The matrix is of size 2597×2597, has 76171 nonzeros, and is ill-conditioned (condition number $\approx 10^6$). The right-hand side is also provided.

In Figure 5.2a we see the convergence plot for four methods: GCRO-DR and LRDGMRES with a restart length $m = 30$ and recycling $k = 10$ approximate eigenvectors, as well as the loose variants which recycle $k - l = 8$ approximate eigenvectors and $l = 2$ error approximations. It can be concluded that LRDGMRES outperforms GCRO-DR, whereas LLRDGMRES turns out to be the best method in this example. Overall the gain by our loose method in comparison to GCRO-DR is around 40%.

We have pointed out in Section 4.2 that an efficient method based on the recycling strategy has to determine and converge to an effective recycle space in a reasonable number of iterations. If the (augmented) Krylov subspace



(a) $m = 30, k = 10, l = 2$ **(b)** $m = 8, k = 4, l = 2$

Figure 5.2: Comparison of GCRO-DR(m, k), LRDGMRES(m, k), LGCRO-DR$(m, k-l, l)$ and LLRDGMRES$(m, k-l, l)$ for cavity10

that we build is not large enough, as a consequence, the computed harmonic Ritz vectors might not be good enough approximations to the eigenvectors, and therefore they are not useful. In Figure 5.2b such a situation is depicted, where the restart length is $m = 8$. We see that GCRO-DR and LRDGMRES experience slow convergence, and present no advantage over GMRES. However, recycling two error approximations additionaly seems to have a great impact on the convergence behavior. Both loose methods show fast convergence, and in addition, LLRDGMRES needs around 10% less matrix-vector products than LGCRO-DR.

Recall, the main motivation for developing the loose methods was that the skip angles (see, e.g., Definition 3.15) from GMRES (as well as GCRO-DR and LRDGMRES) are too small. In other words, every other residual vector (i.e. r_{i-1} and r_{i+1}) points in nearly the same direction, which usually slows down the convergence, and it would be preferable to change such a behavior. In Table 5.2 we give the average sequential and skip angles (in degrees) for the four methods. We can see that the skip angles for LGCRO-DR and LLRDGMRES have improved considerably, which explains why these methods perform better.

Table 5.2: Comparison of average sequential and skip angles (in degrees) for GCRO-DR(30,10), LGCRO-DR(30,13,2), LRDGMRES(30,10) and LLRDGMRES(30,13,2).

	Average seq. angle	Average skip angle
GCRO-DR	25.2	18.7
LGCRO-DR	30.2	41.8
LRDGMRES	27	22.6
LLRDGMRES	32.7	45.6

5.4 Korringa-Kohn-Rostoker Method

Applications in theoretical chemistry and solid state physics often focus on solving the Schrödinger equation. The solution is later on used for predicting key physical properties of systems with a large number of atoms. While solving the Schrödinger equation, one has to compute the ground state density. In the Korringa-Kohn-Rostoker Green function method, the calculation of the density $n(r)$ is achieved by numerical integration over Kohn-Sham Green functions $G(r, r, ; E)$ for different energies E

$$n(r) = -\frac{2}{\pi}\mathrm{Im}\left(\int_{-\infty}^{E_F} G(r, r; E)dE\right). \tag{5.1}$$

To compute G one has to solve the following matrix equation

$$(I - TG^r)X = T, \tag{5.2}$$

where G^r, T and X are block diagonal matrices. In a system of $n_x \times n_y \times n_z$ cells with N_c atoms per cell, the matrices G^r, T and X are of dimension $N \times N$, where $N = n_x n_y n_z b$, with $b = N_c(l_{\max} + 1)^2$, and $l_{\max}$ is the highest angular momentum. Hence, (5.2) results in solving $n = n_x n_y n_z$ systems. If the number of the integration points in (5.1) is m, then in general we seek to solve mn systems.

We considered a crystal system of a nickel-palladium alloy, where some of the palladium atoms are replaced by nickel atoms and the atoms are slightly moved from their optimal positions, which is depicted in Figure 5.3. This system has a face-centered cubic (FCC) geometry and is broken into $4 \times 4 \times 4$ cells, with each cell containing 4 atoms. Choosing $l_{\max} = 3$, we have that the matrices are of size $4096^2 \times 4096^2$, with blocks $64^2 \times 64^2$. However, for

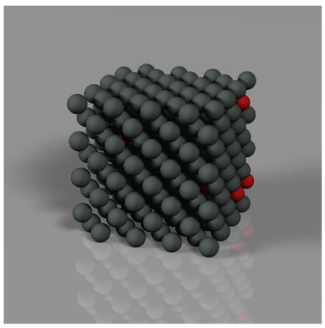


Figure 5.3: Structure of a nickel-palladium alloy.

the purposes of this thesis, we consider solving only $m = 27$ systems, (for 27 integration points), meaning that for the each integration point we solve one linear system corresponding to one $64^2 \times 64^2$ block with a random right-hand side.

For more technical details about the solution of the Schrödinger equation and the derivation of (5.2), we refer to [Bol+12]. I would like to thank Matthias Bolten who provided the matrices and Figure 5.3.

We show the results for the last 5 systems in the sequence (starting with system 23 as the first one), since the rest of the systems are easy to solve. In Figure 5.4 the convergence of GCRO-DR$(30, 5)$, LRDGMRES$(30, 5)$, LGCRO-DR$(30, 4, 1)$ and LLRDGMRES$(30, 4, 1)$ for these 5 systems is illustrated. As in Section 5.1, we present the cumulative number of matrix-vector products necessary for convergence. We see that for the first three systems (as well as for the previous 22 systems), there is no noticeable difference. The last two systems are ill-conditioned and difficult to solve, and that is where we notice a better performance of the new methods. Both our methods, LRDGMRES and LLRDGMRES, require about the same number of matrix-vector products for solving these 5 systems, which is in total around 6% less than for GCRO-DR and LGCRO-DR.

Remark 5.1. We would like to conclude this section by stating two observations we made while running tests for this application:

1. Recycling more than one error approximation does not improve the results for the loose methods.
2. If we increase parameters, e.g., for $m = 40, k = 20$, GCRO-DR becomes more efficient than the other 3 methods.

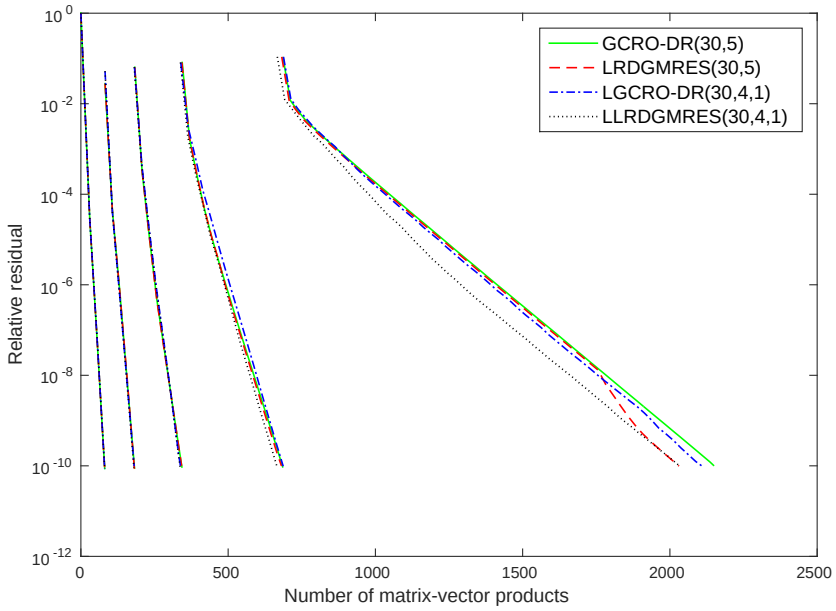


Figure 5.4: Convergence plot for 5 consecutive systems from the KKR method.

5.5 Lattice Quantum Chromo Dynamics

This, as well as the next, section is largely based on work in [Fro+14]. Quantum Chromodynamics (QCD) is a quantum field theory in four-dimensional space-time that describes how quarks and gluons bind together to form hadrons, e.g., neutrons and protons which make up all known matter [DeT06; GL09; MM94]. It is described by a path integral and thus involves infinitely many degrees of freedom. The interaction between quarks and gluons is given by the Dirac operator

$$D = \sum_{\mu=0}^{3} \gamma_\mu \otimes (\partial_\mu + A_\mu),$$

where $\otimes$ denotes the tensor product, $\gamma_\mu \in \mathbb{C}^{4\times 4}$ are Hermitian and unitary matrices which generate the Clifford algebra, i.e. they satisfy

$$\begin{aligned} \gamma_\mu\gamma_\nu + \gamma_\nu\gamma_\mu &= 0 \\ \gamma_\mu^2 &= I, \end{aligned}$$

$\partial_\mu = \frac{\partial}{\partial x_\mu}$ is the partial derivative in direction μ, and A_μ are anti-Hermitian traceless matrices, i.e. $A_\mu \in su(3)$. The set $\{A_\mu : \mu = 0, 1, 2, 3\}$ defines a

gauge field, i.e. a gluonic counterpart of a quark field. A quark field is defined by the twelve component vector (called spinor)

$$\psi(x) = (\psi_{10}(x), \psi_{20}(x), \psi_{30}(x), \psi_{11}(x), \dots, \psi_{33}(x))^T.$$

The twelve components come from properties of quarks that have a color charge $1, 2$ or 3 (or red, green and blue which are the colors of QCD) and a spin $0, 1, 2$ or 3. For future reference, we introduce the color space

$$\mathcal{C} = \{1, 2, 3\},$$

and the spin space

$$\mathcal{S} = \{0, 1, 2, 3\}.$$

One of the main purposes of QCD theory is to compute observables, e.g., masses of particles, so we can use the theory for predictions. All these computations require a discretization onto a four-dimensional euclidean space-time lattice $\mathcal{L}$ with, typically, periodic boundary conditions. Hence the definition of a lattice $\mathcal{L}$.

Definition 5.2. *A lattice $\mathcal{L}$ with lattice spacing a is a subset of a four-dimensional torus such that for any $x, y \in \mathcal{L}$ there exists $p \in \mathbb{Z}$ satisfying*

$$y = x + a \cdot p.$$

A quark field $\psi : \mathcal{L} \to \mathbb{C}^{12}$ is defined at each lattice point, the value $\psi(x)$ still being spinors.

In continuum, the gauge fields $A_\mu(x)$ are connecting infinitesimally close space-time points. On the lattice they have to be replaced by discretized gauge links U_μ^x which connect points at finite distances. The discretized gauge links are members of the $SU(3)$ group, meaning that U_μ^x are 3×3 unitary complex matrices with determinant 1. They are defined by the path ordered integral along the link $(x, x + \mu)$

$$U_\mu^x = e^{-\int_x^{x+\mu} A_\mu(s)ds} \approx e^{-aA_\mu(x+\frac{1}{2}\mu)},$$

where a is the lattice spacing (typically, $a \approx 0.1fm$). The set of all links $\mathcal{U} = \{U_\mu^x : x \in \mathcal{L}, \mu = 0, 1, 2, 3\}$ is called a gauge configuration. For the illustration and easier understanding we refer to Figure 5.5 which depicts a 2D slice of the lattice with the naming conventions.

The discretization of quantum field theory onto the lattice is done numerically using the Hybrid Monte Carlo (HMC) method. A fictitious time is introduced and at each time step HMC yields a new configuration $\mathcal{U}$, i.e. a

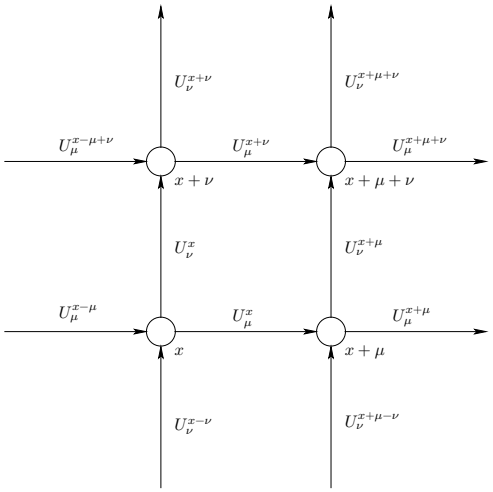


Figure 5.5: 2D representation of the lattice with naming conventions.

collection of links. In order to proceed to the next time step, one has to solve the Dirac equation with a discretized version of the continuum Dirac operator. The discretization of the Dirac operator involves the discretization of the covariant derivatives, which can be discretized by finite differences. Therefore, we define the forward covariant finite differences as

$$(\Delta^\mu \psi)(x) = \frac{U_\mu^x \psi(x+\mu) - \psi(x)}{a}$$

and the backward covariant finite differences as

$$(\Delta_\mu \psi)(x) = \frac{\psi(x) - (U_\mu^{x-\mu})^H \psi(x-\mu)}{a}.$$

Next, we define the Wilson discretization of the Dirac operator [Wil77], i.e. the Wilson-Dirac operator, as

$$D_W = \frac{m_0}{a} I_{12n_\mathcal{L}} + \frac{1}{2} \sum_{\mu=0}^{3} (\gamma_\mu \otimes (\Delta_\mu + \Delta^\mu) - a I_4 \otimes \Delta_\mu \Delta^\mu),$$

where $n_\mathcal{L}$ is the number of lattice sites and m_0 is a mass parameter which tunes the physical mass of the simulated quark. The naive discretization of the Dirac operator does not involve the term $aI_4 \otimes \Delta_\mu \Delta^\mu$. Wilson introduced it as a stabilization term, thus overcoming the problem that the naive discretization experiences by generating unphysical eigenvectors. This phenomenon is known as the "species doubling effect" or as the "red-black instability" [Sus77; Wil77].

Now we can proceed with the HMC method. From time step to time step, we change only a small amount of links, thus obtaining the next configuration

which does not differ too much from the previous one. Looking at the definition of the Wilson-Dirac operator and its dependence on configurations, we can conclude that at each time step $i = 1, 2, ...$ we have to solve the Dirac equation

$$D_W^{(i)} \psi^{(i)} = \eta^{(i)}, \tag{5.3}$$

where the Wilson-Dirac operators *change slightly* from one step to the next. Hence, we obtain a sequence of linear systems with slowly changing matrices, which we will solve with methods introduced in the previous chapter. Solving systems (5.3) is a common problem in the lattice QCD community. The column of the inverse of the Wilson-Dirac operator is called a quark propagator and gives the information on the dynamics of the quark.

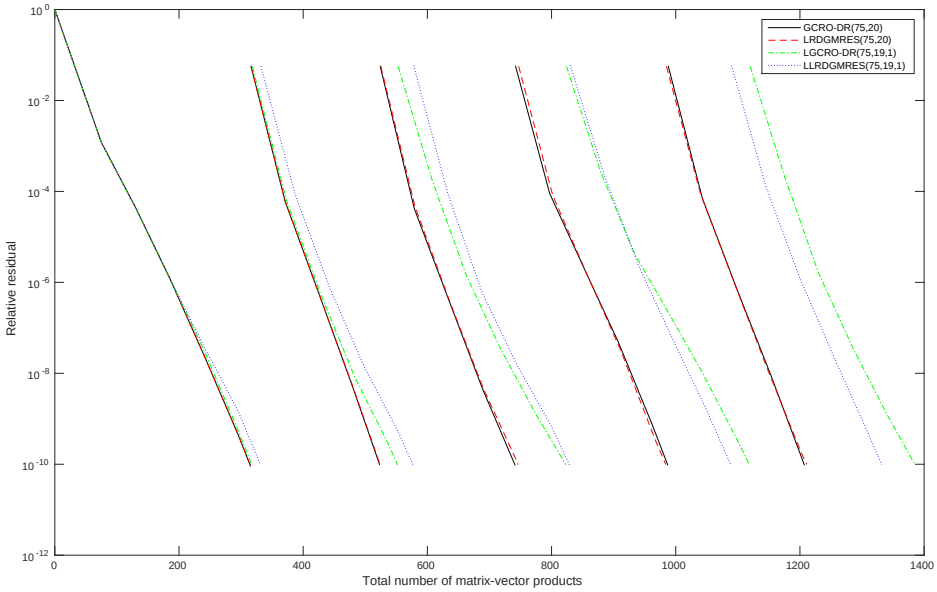


Figure 5.6: Convergence plot for 5 consecutive systems arising in lattice QCD.

In our simulations, we used an 8^4 lattice and we ran 10 steps of the HMC method, where only 10% of links are changed in each next time step. This simulation results in a sequence of 10 slowly changing Wilson-Dirac operators D_W of size 49152×49152 ($8^4 \times 12$). We compared GCRO-DR$(75, 20)$, LRDGMRES$(75, 20)$, LGCRO-DR$(75, 19, 1)$ and LLRDGMRES$(75, 19, 1)$. We notice that independent of the parameters, results are always similar, not in the total number of matrix-vector products, but in how the methods compare to each other. From Figure 5.6 one can draw several conclusions. First, the

loose methods are not suitable for this application. Second, LRDGMRES and GCRO-DR basically require about the same total number of matrix-vector products, again for any choice of m and k. Third, recycling reduces the costs for the systems $i > 1$ by about 30% when compared to the first system, where we start without any deflating subspace.

In these computations, our new methods do not show any improvement over GCRO-DR. However, due to our extensive work with systems arising in this particular application, we will show in Section 5.5.1 a related work for the preconditioned case. Recall, the choice of the preconditioner depends on the application. In the lattice QCD community, the commonly used preconditioner is the red-black multiplicative Schwarz method, which we briefly introduce and show results for the same simulation, thus demonstrating advantages of using a preconditioner. The choice of parameters in this section does not change the overal impression. In order to allow for a comparison with preconditioning, they are chosen such that the preconditioned methods exhibit the best possible gain when compared to the non-preconditioned case.

Furthermore, in Section 5.5.3 we describe *gauge fixing* for the 2D model of quantum field theory, which results in a sequence of diagonally similar matrices. Exploiting this fact, we present quite promising results.

5.5.1 Red-Black Multiplicative Schwarz Method

The example that we used in the previous section with an 8^4 lattice is considered a problem of small size. Nowadays, numerical simulations on the lattice tend to have hundreds of millions of unknowns. Therefore, the systems arising in lattice QCD calculations have to be solved on a parallel machine. The lattice has to be decomposed into blocks (see Definition 5.3) and each processor obtains a block of lattice sites, together with the corresponding variables, and performs calculations associated with these sites. Hence the definition of a block decomposition of the lattice $\mathcal{L}$.

Definition 5.3. *Assume that* $\{\mathcal{T}_1^0, \ldots, \mathcal{T}_{l_0}^0\}$ *is a partitioning of* $\{1, \ldots, N_t\}$ *into* l_0 *blocks of consecutive time points,*

$$\mathcal{T}_j^0 = \{t_{j-1} + 1, \ldots, t_j\}, \quad j = 1, \ldots, l_0, \quad 0 = t_0 < t_1 \ldots < t_{l_0} = N_t,$$

and similarly for the spatial dimensions with partitionings $\{\mathcal{T}_1^\mu, \ldots, \mathcal{T}_{l_\mu}^\mu\}, \mu = 1, 2, 3$.

A block decomposition of $\mathcal{L}$ *is a partitioning of* $\mathcal{L}$ *into* $l = l_0 l_1 l_2 l_3$ *lattice blocks* $\mathcal{L}_i$*, where each lattice block is of the form*

$$\mathcal{L}_i = \mathcal{T}_{j_0(i)}^0 \mathcal{T}_{j_1(i)}^1 \mathcal{T}_{j_2(i)}^2 \mathcal{T}_{j_3(i)}^3.$$

Accordingly, we define a block decomposition of all $12n_{\mathcal{L}}$ *variables in* $\mathcal{V} = \mathcal{L} \times \mathcal{C} \times \mathcal{S}$ *into* l *blocks* $\mathcal{V}_i$ *by grouping all spin and color components corresponding to the lattice block* $\mathcal{L}_i$*, i.e.*

$$\mathcal{V}_i = \mathcal{L}_i \times \mathcal{C} \times \mathcal{S}.$$

Finding a good preconditioner is not always an easy and straightforward task. However, decomposition of the lattice into blocks fits quite naturally with the idea of a domain decomposition. Domain decomposition methods, as the name suggests, split a physical domain into blocks and solve repeatedly smaller block restricted linear systems. A well-known method from this class of methods is the Schwarz method. A variant of the Schwarz method, i.e. the red-black multiplicative Schwarz method [FNZ12; Lüs04; Lüs07] (also known as Schwarz Alternating Procedure) was introduced to lattice QCD as a preconditioner, and is used since. Therefore, we will explain SAP in a few details and run some tests using it as a preconditioner. First, we need the following definition.

Definition 5.4. *Let* $\mathcal{V}_i \subset \mathcal{V}$ *be a lattice block. We define the corresponding trivial embedding*

$$I_{\mathcal{V}_i} : \mathcal{V}_i \to \mathcal{V}$$

as the restriction of the identity on $\mathcal{V}$ *to* $\mathcal{V}_i$*, i.e.*

$$I_{\mathcal{V}_i} = (id_{\mathcal{V}})|_{\mathcal{V}_i}.$$

The corresponding block inverse is defined as

$$B_i = I_{\mathcal{V}_i} D_i^{-1} I_{\mathcal{V}_i}^H \quad \textit{where} \quad D_i = I_{\mathcal{V}_i}^H D I_{\mathcal{V}_i}.$$

One step of the Schwarz method, for a given decomposition $\{\mathcal{V}_i : i = 1, \dots, k\}$ of $\mathcal{V}$, for solving (5.3) consists of solving each of the block systems

$$D_i e_i = I_{\mathcal{V}_i}^H r \tag{5.4}$$

and applying the corrections

$$\psi = \psi + B_i r \quad \text{where} \quad B_i r = I_{\mathcal{V}_i} e_i \quad i = 1, \dots, k. \tag{5.5}$$

We distinguish two versions of the Schwarz method: *additive* and *multiplicative*. In additive Schwarz, the residual is not updated during the computation (5.5) for all i. This means that all the block systems can be solved simultaneously, leading to a straightforward parallelization. We give the method as Algorithm 5.1. On the other hand, in multiplicative Schwarz, a residual update is done every time one of the block systems has been solved, i.e. after each

Algorithm 5.1: Additive Schwarz (one iteration)

Input : ψ, η

Output: ψ

$r = \eta - D\psi$
for $i = 1, \ldots, k$ **do**
  $\psi = \psi + B_i r$

Algorithm 5.2: Multiplicative Schwarz (one iteration)

Input : ψ, η

Output: ψ

for $i = 1, \ldots, k$ **do**
  $r = \eta - D\psi$
  $\psi = \psi + B_i r$

i in (5.5). Consequently, the multiplicative Schwarz method (Algorithm 5.2) is sequential.

The multiplicative Schwarz method can be modified such that it is suitable for parallelization. This modification is known as the SAP method. It is a colored version of multiplicative Schwarz. To this purpose, we have to color the blocks in a way that the blocks of the same color are decoupled, as illustrated in Figure 5.7. Now, after a residual update, all block systems of a given color can be solved simultaneously before the next residual update. Since the Wilson-Dirac operator is based only on the nearest-neighbor couplings, decoupling requires only two colors. The resulting method is given as Algorithm 5.3.

We use the same example as in the previous section. The 8^4 lattice is decomposed into 2^4 blocks of size 4^4. For the solution of the local systems (5.4) we use a direct method, i.e. backslash \ in MATLAB (although one could also

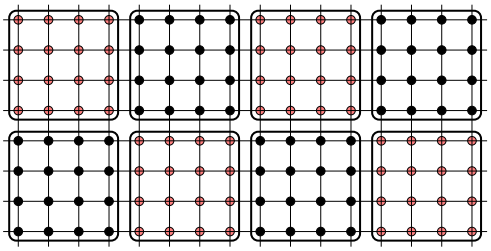


Figure 5.7: 2D block decomposition of the lattice with 2 colors.

Algorithm 5.3: RED-BLACK MULTIPLICATIVE SCHWARZ

Input : ψ, η, ν

Output: ψ

for $i = 1, \ldots, \nu$ **do**
  $r = \eta - D\psi$
  for *all red* i **do**
    $\psi = \psi + B_i r$
  $r = \eta - D\psi$
  for *all black* i **do**
    $\psi = \psi + B_i r$

use a few steps of an iterative method like flexible GMRES). In Figure 5.8 we display the convergence of PGCRO-DR$(75, 20)$, PLRDGMRES$(75, 20)$, PLGCRO-DR$(75, 19, 1)$ and PLLRDGMRES$(75, 19, 1)$ for 5 consecutive systems. In comparison to the non-preconditioned case, we observe a tremendous improvement. By using the preconditioner the costs are reduced by $70\%-75\%$, which demonstrates the potential of preconditioning.

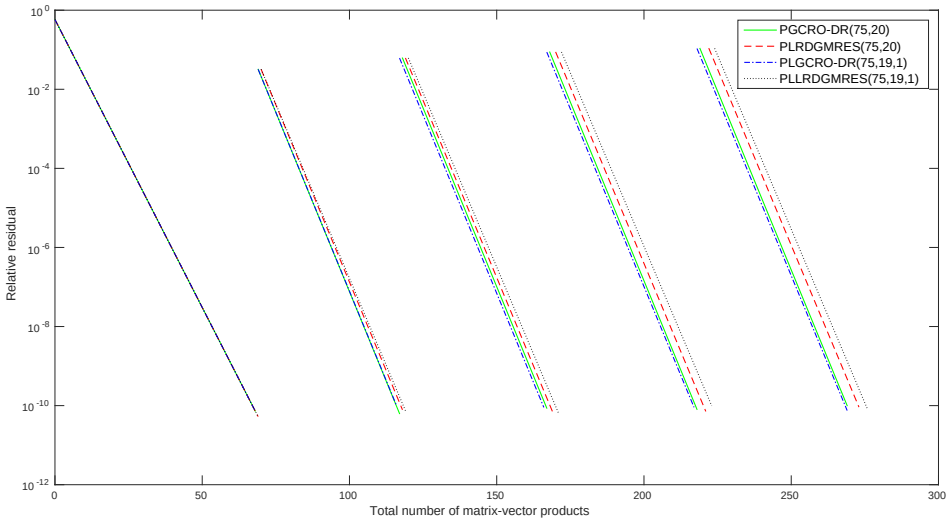


Figure 5.8: Convergence plot of preconditioned methods for 5 consecutive systems arising in lattice QCD.

5.5.2 Diagonal Similarity Transformations

In our research, we tested and employed many ideas. Some of them were definitely worth investigating further. Therefore, we present one of these ideas in this section, and discuss it further in the next section for an application in lattice QCD.

Assume that we have to solve two consecutive systems using the idea of recycling. After solving the first system, we keep the approximate eigenspace computed the last, and use it to reduce costs of the second system. This turns out to be a reasonable approach, if the matrices change slowly, and hence share some spectral properties. The logical question is: can we do better if we have more information about the matrices? And the answer is yes, at least in the case when the next matrix in the sequence can be obtained from the previous one by a diagonal similarity transformation, i.e.

$$A^{(i+1)} = DA^{(i)}D^{-1}, \tag{5.6}$$

where D is a diagonal matrix. As we will show in Section 5.5.3, this is a realistic scenario in real life applications.

For easier understanding, we explain further details for the GCRO-DR method. Recall, the GCRO framework used by GCRO-DR requires the following relation

$$AU = C. \tag{5.7}$$

Plugging in (5.6) in (5.7) for the matrix $A^{(i+1)}$, we get

$$\begin{aligned} DA^{(i)}D^{-1}U^{(i+1)} &= C^{(i+1)} \\ \Leftrightarrow A^{(i)}D^{-1}U^{(i+1)} &= D^{-1}C^{(i+1)}, \end{aligned}$$

which leads to the conclusion that $U^{(i+1)} = DU^{(i)}$. Thus, instead of using $U^{(i)}$ in the next system, an obviously better choice is $DU^{(i)}$.

In the next section we demonstrate how this concept actually works out for a particular application. This is the idea of *gauge fixing* in lattice QCD, where one can obtain a sequence of slowly changing Dirac operators that satisfy (5.6).

5.5.3 Gauge Fixing

For simplicity, we will not consider the 4D lattice QCD theory, but rather give details for the Schwinger model. The Schwinger model is the 2D (1 spatial + 1 time dimension) lattice gauge theory, which is suitable for testing, since the

computations are much cheaper than in four dimensions and realistic systems are of smaller size. In this model, the links U_μ^x are members of the $SU(1)$ group, which means that they can be written as

$$U_\mu^x = e^{i\theta_\mu^x},$$

where θ_μ^x is the only free parameter, and represents the angle in which the link from x points out in the direction μ. The angles come from dynamical simulations, i.e. they evolve according to the HMC method. Geometrically, going from one step of HMC to the next, we only rotate each link of the gauge field, i.e. compute the new angles that satisfy certain criteria.

The idea behind gauge fixing is to rotate back the gauge field by diagonal similarity transformations, such that the rotated field is in some sense "close" to the one from the previous step. The diagonal matrix has the form

$$T = \begin{bmatrix} \begin{pmatrix} e^{i\phi_1} & 0 \\ 0 & e^{i\phi_1} \end{pmatrix} & & \\ & \ddots & \\ & & \begin{pmatrix} e^{i\phi_N} & 0 \\ 0 & e^{i\phi_N} \end{pmatrix} \end{bmatrix},$$

where N is the number of lattice sites. For a random link $e^{i(\theta_\mu^x)'}$, the similarity transformation with the matrix T leads to a Schwinger model matrix with links

$$e^{i(\theta_\mu^x)''} = e^{i\phi(x)} e^{i(\theta_\mu^x)'} e^{-i\phi(x+\mu)},$$

where $e^{i(\theta_\mu^x)''}$ is the rotated link. Hence, we obtain the relation

$$(\theta_\mu^x)'' = \phi(x) + (\theta_\mu^x)' - \phi(x+\mu). \tag{5.8}$$

It becomes obvious that having θ_μ^x and $(\theta_\mu^x)'$ from HMC, one wants to compute the ϕ's, such that $(\theta_\mu^x)''$ is "close" to θ_μ^x. This cannot be achieved exactly, but minimized in a least squares sense, since there are $2N^2$ equations and only N^2 variables. This falls under the optimization problems. The typical approach in quantum field theory is to maximize the function

$$\chi = \sum_{x,\mu} \cos^2((\theta_\mu^x)'' - \theta_\mu^x),$$

using the steepest descent method, where $(\theta_\mu^x)''$ is defined in (5.8).

We performed tests on a small 32×32 lattice, which results in matrices of size $2 \times 32 \times 32 = 2048$. Instead of using $TU^{(i)}$ as the recycled subspace for the

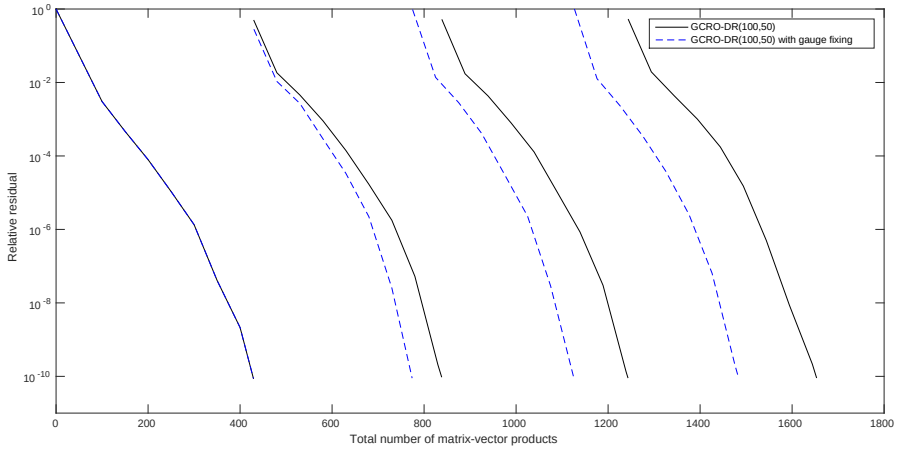


Figure 5.9: Effects of gauge fixing for GCRO-DR.

next matrix, we rather explicitly formed the matrix $TA^{(i+1)}T^{-1}$ and used $U^{(i)}$ for that matrix. We compared GCRO-DR$(100, 50)$, LRDGMRES$(100, 50)$, LGCRO-DR$(100, 49, 1)$ and LLRDGMRES$(100, 49, 1)$ with their modifications that exploit the concept of gauge fixing.

The results are illustrated in Figure 5.9, Figure 5.10, Figure 5.11 and Figure 5.12. As we can see, all four methods perform similarly, i.e. they need around 1600 matrix-vector products to solve four consecutive systems. On the other hand, if we compare the methods only to their gauge fixing variants, we

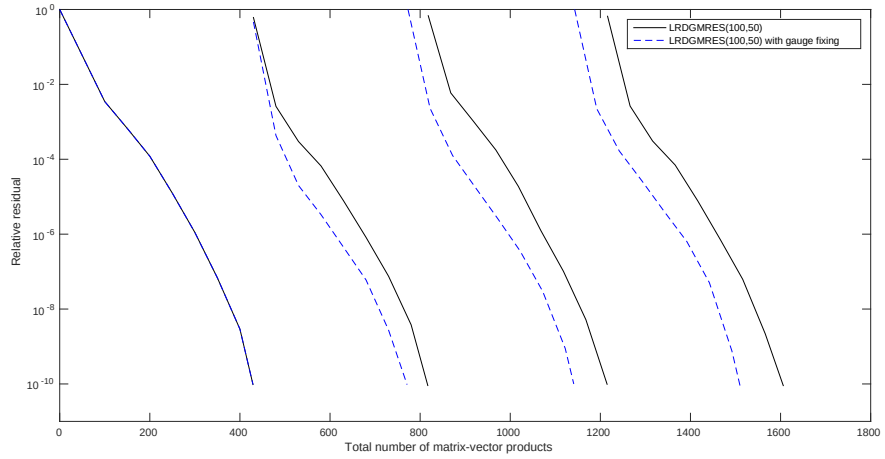


Figure 5.10: Effects of gauge fixing for LRDGMRES.

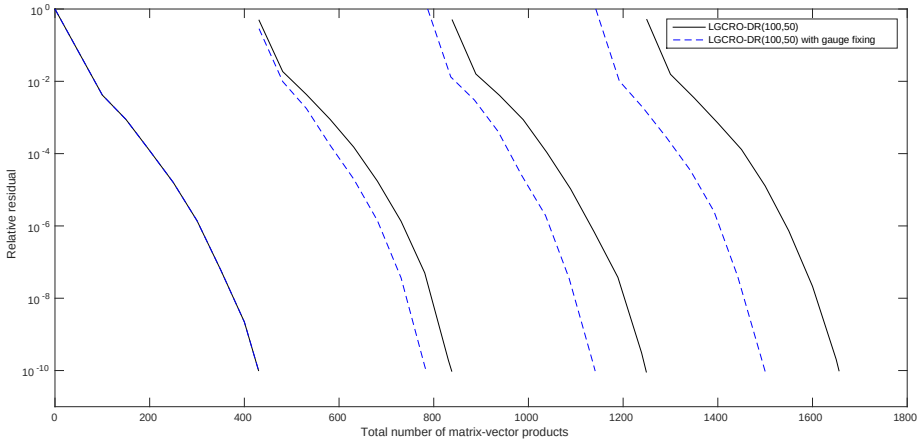


Figure 5.11: Effects of gauge fixing for LGCRO-DR.

can observe a noteworthy improvement when the methods are used together with gauge fixing. This is a very optimistic result since we ran tests only for the Schwinger model. If this idea works out for the 4D case, we could expect a tremendous gain.

At this point I would like to thank Karsten Kahl who provided the code for the HMC method for the Schwinger model from which the sequence of Schwinger matrices used in our computations was obtained.

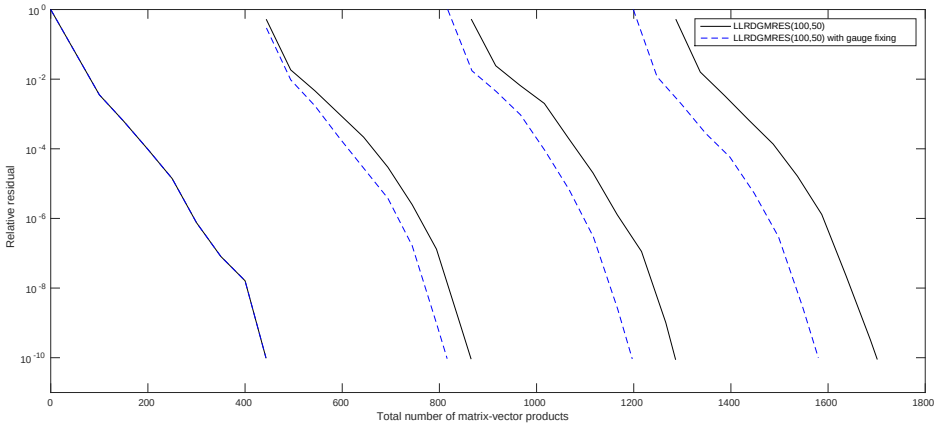
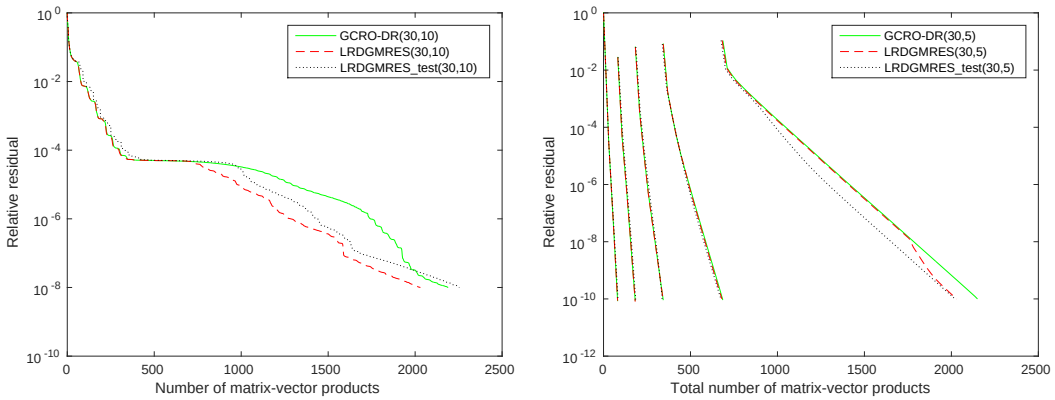


Figure 5.12: Effects of gauge fixing for LLRDGMRES.

5.6 A Variation of LRDGMRES

The main property of LRDGMRES is that the method deflates both, left and right, eigenspaces by using a different operator $(I - C_k(\tilde{C}_k^H C_k)^{-1}\tilde{C}_k^H)A$ rather than A, where the columns of C_k and $\tilde{C}_k$ are the right and left approximate eigenvectors, respectively. Hence, we argued in Section 4.4 that this should be a theoretically better approach than the one used in GCRO-DR. Furthermore, in case of k *exact* left and right eigenvectors corresponding to the k smallest eigenvalues, we stated Proposition 4.19 which shows that once the deflation has occured, the k smallest eigenvalues are completely removed from the spectrum and there is no need to apply additionally the oblique projection $I - C_k(\tilde{C}_k^H C_k)^{-1}\tilde{C}_k^H$ on A. In case of large k, this would save substantial costs.

Since we never compute the exact eigenvectors, we ran tests for a method that uses the framework of LRDGMRES, in a sense that we still augment the Krylov subspace with the *approximate* eigenvectors (harmonic Ritz vectors), but we do not work with the deflated operator. We compared the modified method (LRDGMRES_test) to GCRO-DR and LRDGMRES for three different applications. Surprisingly, for two applications, i.e. fluid dynamics (Section 5.3) and the KKR method (Section 5.4), LRDGMRES_test is comparable to GCRO-DR and LRDGMRES, as we can see in Figure 5.13a for the *cavity10* matrix and in Figure 5.13b for the sequence of 5 matrices arising in the KKR method.



(a) cavity10: $m = 30, k = 10$ **(b)** KKR method: $m = 30, k = 5$

Figure 5.13: Comparison of GCRO-DR(m, k), LRDGMRES(m, k), LRDGMRES_test(m, k) for cavity10 and the sequence of 5 matrices arising in the KKR method

The explanation for such a behavior is that the computed left and right ap-

proximate eigenvectors are quite close to the exact left and right eigenvectors, which makes LRDGMRES_test almost equivalent to LRDGMRES. On the other hand, for the systems arising in lattice QCD (Section 5.5), the results are much worse, as we can see in Figure 5.14.

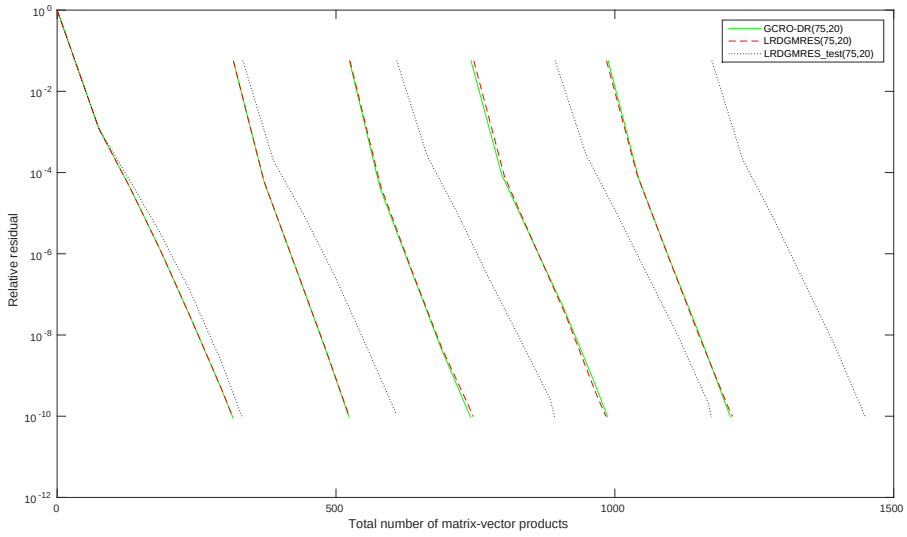


Figure 5.14: Convergence plot for 5 consecutive systems arising in lattice QCD.

To conclude this section, judging by the results we presented, one might benefit from using LRDGMRES_test, of course depending on the application and parameters used.

6 Conclusion and Outlook

The main goal of this thesis was the development of new Krylov solvers for the solution of (1.1). We proposed and derived two new methods that are based on the concepts of augmentation and deflation. Furthermore, we chose several real-life applications to compare our methods with the already existing ones.

The GCRO-DR method [Par$^+$06] was a starting point of our research. Combining the best features of two advanced Krylov solvers, GCRO and GMRES-DR, GCRO-DR proved to be an efficient method for solving (1.1). It is based on the idea of recycling, which means keeping a subspace between systems (cycles) and using it to reduce costs of subsequent systems (cycles). The subspace is chosen as an (approximate) eigenspace corresponding to the (approximate) eigenvalues of the smallest magnitude. Therefore, it is favorable to consider systems for which the convergence depends on the eigenvalue distribution, since in this case augmenting the Krylov subspace for the next system (cycle) with the approximate eigenvectors leads naturally to deflation of the corresponding eigenvalues, which further results in faster convergence of the method. The motivation for our work came from the fact that in GCRO-DR one deflates only the right eigenspace. Realizing an approach sketched in [Gut12], we derived a method which uses the framework of GCRO-DR, but performs true deflation, i.e. deflates both, left and right, approximate eigenspaces. Obtaining the right approximate eigenvectors, i.e. computing the right harmonic Ritz vectors, is done cheaply according to a well-known result (see Lemma 3.11), since it requires only the solution of an eigenproblem for a matrix of small size $m << n$. In our work, we managed to derive a formula for the cheap computation of left approximate eigenvectors, i.e. the left harmonic Ritz vectors, without having to build a Krylov subspace for A^H. Our method, called Left-Right Deflated GMRES (LRDGMRES), has somewhat higher costs and storage requirements than GCRO-DR, due to loss of some orthogonalization properties and having to keep additionally left harmonic Ritz vectors (see Section 4.4.1).

For quite some time, GCRO-DR was the best choice for solving (1.1). In 2013 the new method, Loose GCRO-DR (LGCRO-DR), was proposed by Niu, Lu and Liu. They suggested recycling a few error approximations, in addition to recycling the right eigenspace. This technique already demonstrated

to be effective when used in [BJM05] for the development of loose GMRES. For a number of applications, LGCRO-DR turned out to be a better choice than GCRO-DR. Moreover, using the economic approach (see Section 4.3), the implementation of the method is rather straightforward, following the implementation of GCRO-DR. This approach fits directly into our method. We named the new method Loose LRDGMRES (LLRDGMRES).

Numerical tests show that our methods are comparable to the already existing ones, i.e. GCRO-DR and LGCRO-DR, for any choice of parameters. However, there is no clear overall advantage over the existing methods – the results of a comparison depend on the application considered. This leads to the conclusion that there is still room for further work. An artificially constructed example in Section 5.2 is an indicator of the true power of our methods, which we have not seen yet in the real-life applications that we considered in this thesis. Nonetheless, exploring more real-life applications, we will find the ones for which our methods are better suitable. Moreover, the choice of parameters for each application plays an important role. Therefore, it would be desirable to derive a result for the best choice of parameters, depending on the application.

Furthermore, we tested the theory given by Proposition 4.19, which suggests that in case of exact left and right eigenvectors the corresponding eigenspace is completely removed, meaning that we do not have to use the modified operator $\hat{P}A$, which in case of the large enough k might save us substantial costs. It turns out that in case of approximate left and right eigenvectors, working with A instead of $\hat{P}A$, we can benefit (depending on the application), since the resulting method (see, e.g., Section 5.6) performs similarly to LRDGMRES.

Another obvious goal for future research is to work out a result similar to the one of Theorem 4.11 regarding the convergence for LRDGMRES. This might not be an easy task, since we are using oblique projections within our method. On the other hand, such a result would surely give us some further insights and explanations for the behavior of the method.

In Section 5.5.3 we presented some promising results when the four methods introduced in Chapter 4 are combined with the technique of gauge fixing. The next step in this direction would be to consider gauge fixing for the 4D lattice QCD theory, since we considered only a toy problem.

List of Figures

List of Tables

List of Algorithms

Bibliography

[Arn51] W. E. Arnoldi. "The principle of minimized iteration in the solution of the matrix eigenvalue problem". In: *Quart. Appl. Math.* 9 (1951), pp. 17–29 (cit. on p. 13).

[BJM05] A. H. Baker, E. R. Jessup, and T. Manteuffel. "A technique for accelerating the convergence of restarted GMRES". In: *SIAM J. Matrix Anal. Appl.* 26.4 (2005), pp. 962–984 (cit. on pp. 20, 35, 39, 43–47, 66, 106).

[Boi+96] R. F. Boisvert, R. Pozo, K. A. Remington, R. F. Barrett, and J. Dongarra. "Matrix Market: a web resource for test matrix collections." In: *Quality of Numerical Software*. Ed. by Ronald F. Boisvert. Vol. 76. IFIP Conference Proceedings. Chapman & Hall, 1996, pp. 125–137 (cit. on pp. 81, 86).

[Bol+12] M. Bolten, A. Thiess, I. Yavneh, and R. Zeller. "Preconditioning systems arising from the KKR Green function method using block-circulant matrices". In: *Linear Algebra Appl.* 436.2 (2012), pp. 436–446 (cit. on pp. 82, 89).

[Bra+16] J. J. Brannick, A. Frommer, K. Kahl, B. Leder, M. Rottmann, and A. Strebel. "Multigrid preconditioning for the overlap operator in lattice QCD." In: *Numer. Math.* 132.3 (2016), pp. 463–490 (cit. on p. 82).

[CS97] A. Chapman and Y. Saad. "Deflated and augmented Krylov subspace techniques". In: *Numer. Linear Algebra Appl.* 4.1 (1997), pp. 43–66 (cit. on pp. 38, 39).

[DeT06] C. DeTar T. DeGrand. *Lattice Methods for Quantum Chromodynamics*. World Sci., 2006 (cit. on pp. 82, 90).

[EE01] M. Eiermann and O. G. Ernst. "Geometric aspects of the theory of Krylov subspace methods". In: *Acta Numer.* 10 (2001), pp. 251–312 (cit. on p. 46).

[EES83] S. C. Eisenstat, H. C. Elman, and M. H. Schultz. "Variational iterative methods for nonsymmetric systems of linear equations". In: *SIAM J. Numer. Anal.* 20.2 (1983), pp. 345–357 (cit. on p. 19).

[Fro+14] A. Frommer, K. Kahl, S. Krieg, B. Leder, and M. Rottmann. "Adaptive Aggregation-Based Domain Decomposition Multigrid for the Lattice Wilson-Dirac Operator." In: *SIAM J. Sci. Comp.* 36.4 (2014) (cit. on pp. 82, 90).

[FNZ12] A. Frommer, A. Nobile, and P. Zingler. *Deflation and Flexible SAP-Preconditioning of GMRES in Lattice QCD Simulation*. Tech. rep. arXiv:1204.5463. Comments: 26 pages. 2012 (cit. on pp. 82, 95).

[Gat95] C. Gattringer. "QED_2 and U(1)-Problem". In: *ArXiv High Energy Physics - Theory e-prints* (1995) (cit. on p. 82).
eprint: `hep-th/9503137`.

[GL09] C. Gattringer and C. B. Lang. In: 788 (2009) (cit. on pp. 82, 90).

[Gre97] A. Greenbaum. *Iterative methods for solving linear systems.* Vol. 17. Frontiers Appl. Math. SIAM, Philadelphia, PA, 1997, pp. xiv+220 (cit. on p. 23).

[Gut12] M. H. Gutknecht. "Spectral deflation in Krylov solvers: a theory of coordinate space based methods". In: *Electron. Trans. Numer. Anal.* 39 (2012), pp. 156–185 (cit. on pp. 2, 50, 68, 105).

[HY12] L. A. Hageman and D. M. Young. *Applied iterative methods.* Courier Corporation, 2012 (cit. on p. 37).

[HJ94] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis.* Cambridge; New York: Cambridge University Press, 1994 (cit. on p. 55).

[KS07] M. E. Kilmer and E. de Sturler. "Recycling Subspace Information for Diffuse Optical Tomography." In: *SIAM J. Sci. Comp.* 27.6 (2007), pp. 2140–2166 (cit. on pp. 61, 64).

[Lüs04] M. Lüscher. "Solution of the Dirac equation in lattice QCD using a domain decomposition method". In: *Comput. Phys. Commun.* 156 (2004), pp. 209–220 (cit. on pp. 82, 95).
eprint: `hep-lat/0310048`.

[Lüs07] M. Lüscher. "Local coherence and deflation of the low quark modes in lattice QCD". In: *J. High Energy Phys.* 07.arXiv:0706.2298. CERN-PH-TH-2007-096 (2007), 081. 23 p (cit. on pp. 82, 95).

[MM94] I. Montvay and G. Münster. *Quantum fields on a lattice.* Cambridge Monogr. Math. Phys. Cambridge University Press, 1994 (cit. on pp. 82, 90).

[Mor91] R. B. Morgan. "Computing interior eigenvalues of large matrices". In: *Linear Algebra Appl.* 154/156 (1991), pp. 289–309 (cit. on p. 16).

[Mor95] R. B. Morgan. "A restarted GMRES method augmented with eigenvectors". In: *SIAM J. Matrix Anal. Appl.* 16.4 (1995), pp. 1154–1171 (cit. on pp. 19, 35–38, 51).

[Mor00] R. B. Morgan. "Implicitly restarted GMRES and Arnoldi methods for non-symmetric systems of equations". In: *SIAM J. Matrix Anal. Appl.* 21.4 (2000), pp. 1112–1135 (cit. on pp. 40, 43).

[Mor02] R. B. Morgan. "GMRES with deflated restarting". In: *SIAM J. Sci. Comput.* 24.1 (2002), pp. 20–37 (cit. on pp. 19, 35, 39–42).

[NRT92] N. M. Nachtigal, L. Reichel, and L. N. Trefethen. "A hybrid GMRES algorithm for nonsymmetric linear systems". In: *SIAM J. Matrix Anal. Appl.* 13.3 (1992), pp. 796–825 (cit. on p. 35).

[NLL13] Q. Niu, L.-Z. Lu, and G. Liu. "Accelerated GCRO-DR method for solving sequences of systems of linear equations". In: *J. Comput. Appl. Math.* 253 (2013), pp. 131–141 (cit. on pp. 50, 64, 66, 81, 83, 86).

[PPV95] C. C. Paige, B. N. Parlett, and H. A. van der Vorst. "Approximate solutions and eigenvalue bounds from Krylov subspaces". In: *Numer. Linear Algebra Appl.* 2.2 (1995), pp. 115–133 (cit. on p. 16).

[Par06] M. L. Parks. `http://www.sandia.gov/~mlparks/software.html/`. [Online; accessed 05-July-2011]. 2006 (cit. on p. 81).

[Par+04] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti. `https://www.researchgate.net/publication/32964251_Recycling_Krylov_Subspaces_for_Sequences_of_Linear_Systems/`. [Online; accessed 10-March-2017]. 2004 (cit. on p. 83).

[Par+06] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti. "Recycling Krylov subspaces for sequences of linear systems". In: *SIAM J. Sci. Comput.* 28.5 (2006), pp. 1651–1674 (cit. on pp. 40, 43, 61–63, 66, 81, 84, 105).

[Par98] B. N. Parlett. *The symmetric eigenvalue problem*. Vol. 2nd. Classics Appl. Math. SIAM, 1998 (cit. on p. 16).

[Saa97] Y. Saad. "Analysis of augmented Krylov subspace methods". In: *SIAM J. Matrix Anal. Appl.* 18.2 (1997), pp. 435–449 (cit. on p. 38).

[Saa03] Y. Saad. *Iterative methods for sparse linear systems*. Second. SIAM, Philadelphia, PA, 2003, pp. xviii+528 (cit. on pp. 5, 7, 9, 11, 12, 19, 23).

[SS86] Y. Saad and M. H. Schultz. "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems". In: *SIAM J. Sci. Stat. Comput.* 7.3 (1986), pp. 856–869 (cit. on pp. 19, 20, 22, 24, 35).

[Sch62] J. Schwinger. "Gauge Invariance and Mass. II". In: *Phys. Rev.* 128 (5 1962), pp. 2425–2429 (cit. on p. 82).

[Sim00] V. Simoncini. "On the convergence of restarted Krylov subspace methods". In: *SIAM J. Matrix Anal. Appl.* 22.2 (2000), pp. 430–452 (cit. on p. 46).

[Smi97] A. V. Smilga. "Critical amplitudes in two-dimensional theories". In: *ArXiv High Energy Physics - Theory e-prints* 55 (1997), R443–R447 (cit. on p. 82).
eprint: `hep-th/9607154`.

[Ste11] G. W. Stewart. "On the numerical analysis of oblique projectors". In: *SIAM J. Matrix Anal. Appl.* 32.1 (2011), pp. 309–348 (cit. on pp. 71, 75).

[SG90] G. W. Stewart and J. Guang Sun. *Matrix Perturbation Theory*. Academic Press, 1990 (cit. on p. 62).

[Stu96a] E. de Sturler. "Nested Krylov methods based on GCR". In: *J. Comput. Appl. Math.* 67.1 (1996), pp. 15–41 (cit. on pp. 19, 25, 27, 28, 30, 32–35, 51).

[Stu96b] E. de Sturler. "Truncation strategies for optimal Krylov subspace methods". In: *Tech. Report* TR-96-38 (1996) (cit. on p. 53).

[Stu99] E. de Sturler. "Truncation strategies for optimal Krylov subspace methods". In: *SIAM J. Numer. Anal.* 36.3 (1999), pp. 864–889 (cit. on pp. 19, 34, 49, 51, 53–57).

[Sus77] L. Susskind. "Lattice fermions". In: 16 (1977), pp. 3031–3039 (cit. on p. 92).

[VV94] H. A. van der Vorst and C. Vuik. "GMRESR: a family of nested GMRES methods". In: *Numer. Linear Algebra Appl.* 1.4 (1994), pp. 369–386 (cit. on pp. 19, 26, 27, 35).

[Wal88] H. F. Walker. "Implementation of the GMRES method using Householder transformations". In: *SIAM J. Sci. Stat. Comput.* 9.1 (1988), pp. 152–163 (cit. on p. 13).

[Wil77] K. G. Wilson. "Quarks and Strings on a Lattice". In: *New Phenomena in Subnuclear Physics*. Springer, 1977 (cit. on p. 92).

[WS98] K. Wu and H. D. Simon. “Thick-restart Lanczos method for symmetric eigenvalue problems”. In: *IRREGULAR*. Ed. by A. Ferreira, J. D. P. Rolim, H. D. Simon, and S.-H. Teng. Vol. 1457. Lecture Notes in Comput. Sci. Springer, 1998, pp. 43–55 (cit. on p. 40).