Solving Linear Systems using the Adjoint

Martin Stoll

University of Oxford

Balliol College

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2008

To Anke and Hannes, my favorite people

Solving Linear Systems using the Adjoint

Martin Stoll, Balliol College, University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

Michaelmas 2008

Abstract

It is widely appreciated that the iterative solution of linear systems of equations with large sparse matrices is much easier when the matrix is symmetric. It is equally advantageous to employ symmetric iterative methods when a non-symmetric matrix is self-adjoint in a non-standard inner product. In order to satisfy this criterion the adjoint of the system matrix plays a crucial role, and in this thesis we will address three situations where the adjoint and adjointness of a matrix arise.

The role of the adjoint is illustrated in this thesis, and we give general conditions for such self-adjointness. A number of known examples for saddle point systems are surveyed and combined to make new combination preconditioners which are self-adjoint in different inner products. In particular, a new method related to the Bramble-Pasciak CG method is introduced, and it is shown that a combination of the two outperforms the widely used classical method in a number of examples. Furthermore, we combine Bramble and Pasciak's method with a recently introduced method by Schöberl and Zulehner. The result gives a new preconditioner and inner product that can outperform the original methods of Bramble-Pasciak and Schöberl-Zulehner.

The Bramble-Pasciak Conjugate Gradient algorithm is widely used in the finite element community. It uses the efficient implementation of CG in a non-standard inner product by using the self-adjointness of the system matrix. Motivated by a reformulation of the linear system in saddle point form, we introduce Bramble-Pasciak-like methods that can be used to solve problems coming from optimization. We illustrate that the eigenvalues for the preconditioned matrix in this setup have a very similar (sometimes equivalent) structure to the preconditioned matrix of a method which uses a constraint preconditioner. We furthermore give numerical results for optimization examples.

The simultaneous solution of $\mathcal{A}x = b$ and $\mathcal{A}^T y = g$ where $\mathcal{A}$ is a non-singular matrix is required in a number of situations. The algorithms presented in this thesis make use of the relationship between the matrix $\mathcal{A}$ and its adjoint $\mathcal{A}^T$. Darmofal and Lu have proposed a method based on the Quasi-Minimal residual algorithm (QMR) a method relating matrix and adjoint via the non-symmetric Lanczos process. Here, we introduce a technique for the same purpose based on the LSQR method and show how its performance can be improved when using the Generalized LSQR method. These strategies are based on the relationship with A and A^T to a (Block-)Lanczos method. We further show how preconditioners can be introduced to enhance the speed of convergence and discuss different preconditioners that can be used. The scattering amplitude $g^T x$, a widely used quantity in signal processing for example, has a close connection to the above problem since x represents the solution of the forward problem and g is the right hand side of the adjoint system. We show how this quantity can be efficiently approximated using Gauss quadrature and introduce a Block-Lanczos process that approximates the scattering amplitude and which can also be used with preconditioning.

CONTENTS

LIST OF ALGORITHMS

LIST OF FIGURES

ACKNOWLEDGMENTS

When writing these acknowledgments, I realized how much this thesis was not solely my adventure over the last three years and just how many people directly or indirectly contributed to it. It is easy to unintentionally forget some people and for that I apologize now.

First and foremost, I have to thank my supervisor Andy Wathen for his constant support, providing assistance and being a role model in more than one way. I will always admire his ability to ask the right questions. I would like to thank Peter Benner for introducing me to numerical linear algebra in the first place. I am very grateful for the luxury of having spent three months with Gene Golub in Oxford who generously shared his knowledge and became a good friend. This would not have been possible without the thriving environment that the Numerical Analysis group in Oxford offers. I also thank my coauthors Sue Dollar and Nick Gould for the invaluable discussions while writing the paper on preconditioners for optimization problems. I am greatly indebted to Laura Silver for reading this thesis carefully and commenting on my style of writing.

I also want to thank the students in the Numerical Analysis group. They became more than just colleagues and in particular I want to thank Thomas Schmelzer, Nachi Gupta, David Knezevic and Tyrone Rees. This would not be an Oxford thesis without thanking my friends in college, especially the Holywell Dinner Club – Anna Arnth-Jensen, Joanne Gale, Cameron Dunlop and Michael Kohl. I also want to thank the Court Place Gardens crowd for making Oxford such a fabulous place to live in.

I would like to acknowledge the financial support from the EPSRC and the Computing Laboratory.

Finally, I want to thank the three most important people in my life. First, my mom who brought me up and supported me whenever I needed her. Then I would like to thank Hannes, my little son, who showed me what life really is about and last but not least, I want to thank Anke, my wife, for keeping me grounded and for being such a fantastic partner.

CHAPTER 1 INTRODUCTION

1.1 The problem

With the increasing availability of computing power, mathematical models for processes coming from every application area have gained in popularity as well as complexity. The mathematical description of problems arising in biology, chemistry, automotive industry, etc., usually involves partial differential equations (PDEs), Ordinary Differential Equations (ODEs) or the solution of an optimization problem that might involve both ODEs and PDEs. In almost every case, at the heart of the computation, the solution to a non-singular linear system of the form

$$\mathcal{A}x = b \text{ with } \mathcal{A} \in \mathbb{R}^{N,N}, b \in \mathbb{R}^N \tag{1.1}$$

is required. The discretized linear system is typically of very large dimension; this could easily mean that $N \sim 10^9$, and the matrix $\mathcal{A}$ is in general sparse which means that only a small proportion of its entries will be non-zero. For apparent reasons, it is crucial to solve these systems as fast as possible, and the numerical methods should reflect this desire. For smaller systems, often the most effective methods are the direct methods based on the LU or Cholesky factorizations (see [20] for details). These methods perform impressively for systems up to certain dimensions. Eventually, these techniques will fail due to storage requirements depending on the underlying application and the corresponding sparsity pattern of the matrix. Recall also that the computational work of Gaussian elimination is $O(N^3)$ for a dense matrix. Hence, iterative techniques have to be employed. The most popular class is given by the so-called Krylov subspace solvers which compute approximations to the solution in a Krylov subspace

$$\mathcal{K}_k(\mathcal{A}, r_0) = span\left\{r_0, \mathcal{A}r_0, \ldots, \mathcal{A}^{k-1}r_0\right\} \tag{1.2}$$

with $r_0 = b - \mathcal{A}x_0$ an initial residual based on the initial guess x_0. There exists a vast amount of literature describing and analyzing Krylov subspaces; here we only refer to [56, 96, 21] and the references mentioned therein. We entirely focus on Krylov subspace solvers in this thesis but want to mention that there exist alternatives such as the multigrid method [115, 57] that can perform outstandingly for certain problems.

In most practical applications, we cannot simply solve the system with $\mathcal{A}$ since it would take too many iterations for the iterative scheme to converge. In such cases, preconditioning has to be used to obtain an approximation to x effectively. In more detail, a non-singular preconditioner $\mathcal{P}$ is chosen such that an appropriate iterative method applied to

$$\underbrace{\mathcal{P}^{-1}\mathcal{A}}_{\widehat{\mathcal{A}}} \; x = \; \mathcal{P}^{-1}b \tag{1.3}$$

has better convergence properties than with the original system. The system (1.3) is called the *left-preconditioned* system. It is also possible to introduce a *right-preconditioned* system $\mathcal{A}\mathcal{P}^{-1}\hat{x} = \mathcal{P}^{-1}b$ with $x = \mathcal{P}^{-1}\hat{x}$. Another variant is the so-called *centrally preconditioned* system where the system matrix is $\mathcal{P}_1^{-1}\mathcal{A}\mathcal{P}_2^{-1}$. Note that for $\mathcal{P} = \mathcal{P}_1\mathcal{P}_2$ all three systems are similar. The analysis of right versus left preconditioning can be found in [96,95]. In essence, for most problems the difference between left and right preconditioning is not significant which is why we only consider left preconditioning in this thesis.

For the preconditioner to be efficient, it has to be a 'good' approximation of $\mathcal{A}$ and a system with $\mathcal{P}$ needs to be easily solvable. A good preconditioner takes the structure of the problem into account and therefore reflects the underlying problem – this will be emphasized in later parts of this thesis. A general introduction to preconditioning can be found in [96].

So far, we have not talked about the properties of $\mathcal{A}$ apart from it being a non-singular, sparse matrix. In Section 1.3, we show two examples that result in linear systems of the form (1.1). The systems arising in each problem resemble the nature of the problem and will result in different properties of the matrix $\mathcal{A}$, such as symmetry and definiteness. The structure of the system matrix $\mathcal{A}$ also varies with the underlying application and also depends on numerical issues such as discretization. More importantly, we can pick iterative solvers based on the properties of $\mathcal{A}$; some solvers should be preferred to others (see Chapter 2). One of the most important solvers is the Conjugate Gradient method (CG) [59] that we carefully explain later. It needs not only symmetry but also definiteness of the matrix $\mathcal{A}$. This can be hard to achieve, especially if a certain type of preconditioner is used that is known to be a 'good' approximation to $\mathcal{A}$. In this thesis, we present examples where certain types of preconditioners can be used that destroy the symmetry of the matrix $\mathcal{A}$ but with a certain choice of inner product enable the use of CG.

One important question asked in the early 1980s was: for which matrices can we guarantee that certain desirable solvers can be used? In 1984, Faber and Manteuffel presented a milestone theorem that fully answers this question. In Section 1.2, we discuss the main theorem and recent developments based on it. It is not always easy to find a mathematical description that according to the Faber-Manteuffel theorem enables the use of the sought after solvers, e.g. when $\mathcal{A}$ is a symmetric matrix with $\mathcal{A}^T = \mathcal{A}$. Hence, a more general set of non-symmetric solvers has to be used, or the symmetry of the system matrix $\mathcal{A}$ in non-standard inner-products or bilinear forms

$$\langle x, y\rangle_{\mathcal{H}} = x^T\mathcal{H}y \tag{1.4}$$

with $\mathcal{H}^T = \mathcal{H}$, i.e.

$$\langle \mathcal{A}x, y\rangle_{\mathcal{H}} = \langle x, \mathcal{A}y\rangle_{\mathcal{H}}.$$

This will be underlined by the Faber-Manteuffel theory presented in 1.2.

In Chapter 2, some of the most important iterative methods are introduced with special emphasis on the possibility of using non-standard inner products or bilinear forms.

In Chapter 3, we discuss the concept of self-adjointness in non-standard inner products/bilinear forms. We present basic results and properties. Based on these, we introduce a technique called *Combination Preconditioning* that was recently published in [109]. Using this technique, we combine different known methods and also introduce a new method based on a classical result of Bramble and Pasciak [10].

The use of non-standard inner product solvers for problems coming from optimization is at the heart of Chapter 4. The standard formulation of the linear systems is reformulated with the alternative formulation representing a framework for many well-known methods. Based on this observation we analyze the convergence behavior of a non-standard inner product solver in comparison to the reformulation. This results in the rewriting of a method presented in [32] in terms of a non-standard inner product method. We show that this setup is more flexible than the original and look at the eigenvalue distribution. This work was submitted for publication [18].

Chapter 5 deals with the solution of the systems arising when one is interested in computing the scattering amplitude, a function of the solution of the linear system (1.1) (see Section 1.3). We present techniques well established in the literature to solve $Ax = b$ and $A^Ty = g$ simultaneously and introduce a method [52] based on a well-known algorithm by Saunders *et al.* in [86, 85, 99]. In addition to discussing preconditioning, we look at the approximation of the scattering amplitude without approximating the solution to the linear system first.

The iterative methods presented in earlier chapters of the thesis are all thoroughly tested and the numerical results are presented in Chapter 6. The methods are always compared to the best suited solvers available, and the examples are mostly taken from freely available data sets.

1.2 Faber-Manteuffel theorem

In 1981, Gene Golub posed the question: for which matrices $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$ do there exist short-term recurrence solution methods to solve (1.1) (cf. SIGNUM Newsletter, vol 16. no.4 1981)? This question was fully answered by Faber and Manteuffel in 1984 (cf. [27]). Unfortunately, the proof of the main theorem is complicated and non-constructive. Recently, Liesen and Saylor [71] attempted a constructive proof which led to a survey paper in 2006 by Liesen and Strakoš (see [72]). Following up on this Faber *et al.* [26] presented more accessible proofs for the Faber-Manteuffel theorem in terms of linear operators. Note that we use the preconditioned matrix $\widehat{\mathcal{A}}$ here but all results also hold for $\mathcal{A}$, i.e. by simply defining $\mathcal{P} = I$ for the preconditioner.

Here, we want to summarize the Faber-Manteuffel theorem and its application to the solution of preconditioned linear systems of the form (1.3) by staying close to the notation introduced in [72]. To define optimal short-term recurrences, we need several basic definitions. Let d be the dimension of the Krylov subspace $\mathcal{K}_k(\widehat{\mathcal{A}}, r_0)$ when it becomes invariant under $\widehat{\mathcal{A}}$; then d is called the *grade of* r_0 *with respect to* $\widehat{\mathcal{A}}$. Another important tool not only for the Faber-Manteuffel theorem is the bilinear form $\langle\cdot,\cdot\rangle_{\mathcal{H}} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ with $\mathcal{H}$ being symmetric. Whenever $\mathcal{H}$ is also positive definite then $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ defines an inner product. Let us assume for the remainder of this section that $\mathcal{H}$ defines an inner product. Then, for a given $\widehat{\mathcal{A}}$, r_0 and $\mathcal{H}$ under the above assumptions, we consider an $\mathcal{H}$-orthogonal basis $[v_1, \ldots, v_n]$ of $\mathcal{K}_k(\widehat{\mathcal{A}}, r_0)$ where

$$\begin{aligned} span\{v_1, \ldots, v_k\} &= \mathcal{K}_k(\widehat{\mathcal{A}}, r_0) \\ \langle v_i, v_j\rangle_{\mathcal{H}} &= 0 \qquad\qquad i \neq j \end{aligned}$$

for all k smaller than the grade d. Such a set of basis vectors can be generated by the Arnoldi algorithm with the $\mathcal{H}$-inner product. The *Arnoldi algorithm* [3,53] is a Gram-Schmidt process that creates an $\mathcal{H}$-orthogonal basis $v_1, v_2, \ldots$ for the Krylov space $\mathcal{K}_k(\widehat{\mathcal{A}}, r_0)$. It can also be viewed as a method to compute the QR decomposition of $[v_1, \widehat{\mathcal{A}}V_k]$ (see [83]). Its $\mathcal{H}$-inner product form is given by

$$\begin{aligned} v_1 &= r_0 \\ v_{k+1} &= \widehat{\mathcal{A}}v_k - \textstyle\sum_{j=1}^{k} h_{j,k}v_j \text{ with } h_{j,k} = \frac{(\widehat{\mathcal{A}}v_k, v_j)_{\mathcal{H}}}{(v_j, v_j)_{\mathcal{H}}}. \end{aligned}$$

where $k \leq d-1$. The normalization of the v_{k+1} is skipped for convenience. A standard implementation of the Arnoldi algorithm including normalization and classical Gram-Schmidt orthogonalization is given in Algorithm 1.1.

```
$v_1 = r_0/\|r_0\|$
for $j = 1, 2, \ldots, k$ do
    Compute $h_{ij} = \langle\widehat{\mathcal{A}}v_j, v_i\rangle_{\mathcal{H}}$ for $i = 1, 2, \ldots, j$
    Compute $w_j = \widehat{\mathcal{A}}v_j - \sum_{i=1}^{j} h_{ij}v_i$
    $h_{j+1,j} = \|w_j\|_2$
    $v_{j+1} = w_j/h_{j+1,j}$
end for
```

Algorithm 1.1: Arnoldi-Classical Gram-Schmidt

The implementation with modified Gram-Schmidt should be preferred in practice since it is more stable (see [9]). The matrix representation of an instance of the Arnoldi algorithm is given by

$$\widehat{\mathcal{A}}V_k = V_kH_k + h_{k+1,k}v_{k+1}e_k^T = V_{k+1}H_{k+1,k} \tag{1.5}$$

where

$$V_{k+1} = [V_k \; v_{k+1}]$$

and

$$H_{k+1,k} = \begin{bmatrix} H_k \\ h_{k+1,k} e_k^T \end{bmatrix}$$

is an upper Hessenberg matrix. Note that for a symmetric matrix $\widehat{\mathcal{A}}$ and $\mathcal{H} = I$ the Arnoldi algorithm reduces to the well-known *symmetric Lanczos method* [66], i.e.

$$\widehat{\mathcal{A}} V_k = V_k T_k + h_{k+1,k} v_{k+1} e_k^T = V_{k+1} T_{k+1,k} \tag{1.6}$$

where T_k is a symmetric tridiagonal matrix. Algorithm 1.2 gives a typical implementation of the Lanczos algorithm, with $\mathcal{H} = I$.

```
$v_1 = r_0/\|r_0\|$
for $j = 1, 2, \ldots, k$ do
  Compute $w_j = \widehat{\mathcal{A}} v_j - \beta_j v_{j-1}$
  $\alpha_j = \langle w_j, v_j \rangle$
  Compute $w_j := w_j - \alpha_j v_j$
  $\beta_{j+1} = \|w_j\|_2$
  if $\beta_{j+1} = 0$ then
    stop
  end if
  $v_{j+1} = w_j/\beta_{j+1}$
end for
```

Algorithm 1.2: Lanczos algorithm-Modified Gram-Schmidt

The matrix representation of the Arnoldi algorithm at step with the $\mathcal{H}$-inner product is now

$$\widehat{\mathcal{A}} V_{d-1} = V_d H_{d,d-1} \tag{1.7}$$

where $V_{d-1} = [v_1, \ldots, v_{d-1}]$, $V_d = [v_1, \ldots, v_d]$ have orthogonal columns in the

$\mathcal{H}$-inner product and

$$H_{d,d-1} = \begin{bmatrix} h_{1,1} & \cdots & h_{1,d-1} \\ 1 & \ddots & \\ & \ddots & h_{d-1,d-1} \\ & & 1 \end{bmatrix}.$$

If symmetry of $\widehat{\mathcal{A}}$ is given in the $\mathcal{H}$-inner product then an $\mathcal{H}$ symmetric Lanczos, $\mathcal{H}$-Lanczos can be implemented (see Algorithm 1.3).

```
Choose start vector $v_1 \in \mathbb{R}^n$ with $\|v_1\| = 1$.
Set $\beta_0 = 0$
for $k = 1, 2, \ldots$ do
    $\tilde{v}_{k+1} = \widehat{\mathcal{A}} v_k - \beta_{k-1} v_{k-1}$
    Compute $\alpha_k = \langle \tilde{v}_{k+1}, v_k \rangle_{\mathcal{H}}$
    $\tilde{v}_{k+1} := \tilde{v}_{k+1} - \alpha_k v_k$
    Set $\beta_k = \|\tilde{v}_{k+1}\|_{\mathcal{H}}$
    Set $v_{k+1} = \tilde{v}_{k+1} / \beta_k$
end for
```

Algorithm 1.3: Algorithm for $\mathcal{H}$-Lanczos

The band structure of $H_{d,d-1}$ is important for developing efficient algorithms. Note in the symmetric case, i.e. $\widehat{\mathcal{A}}^T = \widehat{\mathcal{A}}$, the matrix $H_{d,d-1}$ is a tridiagonal matrix [66, 88]. This leads to the following definition.

Definition 1.1. *An unreduced upper Hessenberg matrix is called a $(s+2)$-band Hessenberg, when its s-th superdiagonal contains at least one non-zero entry and all entries above the s-th superdiagonal are zero.*

Let $H_{d,d-1}$ be now an $(s+2)$-band Hessenberg matrix. Then the Arnoldi algorithm with $\mathcal{H}$-inner product (1.7) reduces to

$$v_{k+1} = \widehat{\mathcal{A}} v_k - \sum_{j=k-s}^{k} h_{j,k} v_j \text{ with } h_{j,k} = \frac{\langle \widehat{\mathcal{A}} v_k, v_j \rangle_{\mathcal{H}}}{\langle v_j, v_j \rangle_{\mathcal{H}}}.$$

The $\mathcal{H}$-orthogonal basis of the Krylov subspace is then generated by an $(s+2)$-term recurrence. We need precisely the latest $s+1$ basis vectors $v_k, \ldots, v_{k-s}$ in the given iteration step. Furthermore, only one matrix vector product per iteration with $\widehat{\mathcal{A}}$ is required. We call such $(s+2)$-term recurrences *optimal*. Note that in practice we want s to be small in order to obtain efficient algorithms and so that the algorithm 'deserves' the term optimal.

Definition 1.2 (Definition 2.4 in [72]) states the condition when we say the matrix $\widehat{\mathcal{A}}$ admits an optimal short-term recurrence.

Definition 1.2. *Let $\widehat{\mathcal{A}} \in \mathbb{R}^{n,n}$ be a non-singular matrix with the degree of the minimal polynomial[1] being $d_{min}(\widehat{\mathcal{A}})$. Let $\mathcal{H}$ be a symmetric positive definite matrix and let s be an integer with $s+2 \leq d_{min}(\widehat{\mathcal{A}})$.*

1. *If for an initial vector r_0 the matrix $H_{d,d-1}$ is $(s+2)$-band Hessenberg, then we say that $\widehat{\mathcal{A}}$ admits for the given $\mathcal{H}$ and r_0 an optimal $(s+2)$-term recurrence.*

2. *If $\widehat{\mathcal{A}}$ admits for the given $\mathcal{H}$ and any initial vector r_0 an optimal recurrence of length at most $s+2$, while it admits for the given $\mathcal{H}$ and at least one r_0 an optimal $(s+2)$-term recurrence, then we say that $\widehat{\mathcal{A}}$ admits for the given $\mathcal{H}$ an optimal $(s+2)$-term recurrence.*

The interesting question, answered by Faber and Manteuffel, is now which matrices $\widehat{\mathcal{A}}$ admit such an optimal $(s+2)$-term recurrence. Once these classes are identified we look for efficient solvers of the system (1.1). The Arnoldi algorithm cannot produce more vectors than the degree of the minimal polynomial of $\widehat{\mathcal{A}}$, $d_{min}(\widehat{\mathcal{A}})$, and therefore to look for $s+2 > d_{min}(\widehat{\mathcal{A}})$ is meaningless. Moreover, we are interested in finding a very small s for a given $\mathcal{H}$, i.e. $s \ll d_{min}(\widehat{\mathcal{A}})$ since $d_{min}(\widehat{\mathcal{A}})$ is usually very large. Before stating the main theorems, we have to introduce some further instruments. We define the $\mathcal{H}$-adjoint $\widehat{\mathcal{A}}^+$ of $\widehat{\mathcal{A}}$ by

$$\widehat{\mathcal{A}}^+ = \mathcal{H}^{-1}\widehat{\mathcal{A}}^T\mathcal{H}$$

[1] The minimal polynomial p_{min} of $\mathcal{A}$ is the monic polynomial of minimal degree such that $p_{min}(\mathcal{A}) = 0$.

and see that it satisfies

$$\langle \widehat{\mathcal{A}}x, y\rangle_{\mathcal{H}} = \langle x, \widehat{\mathcal{A}}^{+}y\rangle_{\mathcal{H}} \Leftrightarrow \langle \mathcal{H}\widehat{\mathcal{A}}x, y\rangle = \langle \mathcal{H}x, \widehat{\mathcal{A}}^{+}y\rangle.$$

Using this adjoint we get the following important definition

Definition 1.3. *Let $\widehat{\mathcal{A}} \in \mathbb{R}^{n,n}$ be non-singular with $d_{min}(\widehat{\mathcal{A}})$ the degree of the minimal polynomial and let $\mathcal{H}$ be a symmetric positive definite matrix. Suppose that*

$$\widehat{\mathcal{A}}^{+} = p_s(\widehat{\mathcal{A}}),$$

where $p_s(\widehat{\mathcal{A}})$ is a polynomial of smallest possible degree s having this property. Then $\widehat{\mathcal{A}}$ is called normal of degree s with respect to $\mathcal{H}$, or, $\mathcal{H}$-normal(s).

The equivalence between $\widehat{\mathcal{A}}$ is admitting for a given $\mathcal{H}$ an optimal $(s+2)$-term recurrence and $\widehat{\mathcal{A}}$ is $\mathcal{H}$-normal(s) is given by the following two theorems in [72] (see Lemma 2.7 and Theorem 2.9).

Theorem 1.4. *Let $\widehat{\mathcal{A}} \in \mathbb{R}^{n,n}$ be non-singular with $d_{min}(\widehat{\mathcal{A}})$ the degree of the minimal polynomial. Let $\mathcal{H}$ be a symmetric positive definite matrix and let s be an integer with $s + 2 < d_{min}(\widehat{\mathcal{A}})$. If $\widehat{\mathcal{A}}$ is $\mathcal{H}$-normal(s) then it admits for $\mathcal{H}$ and any given r_0 an optimal recurrence of length at most $s + 2$, while for any r_0 with grade with respect to $\widehat{\mathcal{A}}$ at least $s + 2$ an optimal $(s + 2)$-term recurrence.*

This theorem states the sufficient condition for $\widehat{\mathcal{A}}$ to admit for a given $\mathcal{H}$ an optimal $(s + 2)$-term recurrence. The necessary conditions will be given by the next theorem (Theorem 2.10 in [72]).

Theorem 1.5. *Let $\widehat{\mathcal{A}} \in \mathbb{R}^{n,n}$ be non-singular with $d_{min}(\widehat{\mathcal{A}})$ the degree of the minimal polynomial. Furthermore, let $\mathcal{H}$ be a symmetric positive definite matrix and let s be an integer with $s + 2 < d_{min}(\widehat{\mathcal{A}})$. If $\widehat{\mathcal{A}}$ admits for a given $\mathcal{H}$ an optimal $(s + 2)$-term recurrence, then $\widehat{\mathcal{A}}$ is $\mathcal{H}$-normal(s).*

For a long time, the only known proof was given by Faber and Manteuffel in [27]. Recently, Faber *et al.* presented two new and more accessible proofs (see [26]). In [71], an attempt to prove this theorem in a linear algebra based way is shown. Liesen and Saylor showed that the reducibility of $\widehat{\mathcal{A}}$ for a given $\mathcal{H}$ to $(s+2)$-band Hessenberg form is equivalent to $\widehat{\mathcal{A}}$ being $\mathcal{H}$-normal(s) in the case of a nonderogatory $\widehat{\mathcal{A}}$. We will now precisely define what the reducibility of the original matrix $\widehat{\mathcal{A}}$ is. The dimension of the Krylov subspace generated by $\widehat{\mathcal{A}}$ and r_0 is supposed to be d. Then we know by construction, $\widehat{\mathcal{A}}v_d \in \mathcal{K}_d(\widehat{\mathcal{A}}, v_1)$ and

$$\widehat{\mathcal{A}}v_d = \sum_{i=1}^{d} h_{i,d}v_i \text{ where } h_{i,d} = \frac{\langle \widehat{\mathcal{A}}v_d, v_i\rangle_{\mathcal{H}}}{\langle v_i, v_i\rangle_{\mathcal{H}}}, \quad i = 1, \ldots, d.$$

Reformulated in matrix terms this becomes

$$\widehat{\mathcal{A}}V_d = V_d H_d$$

where $V_d = [v_1, \ldots, v_d]$ and

$$H_d = \begin{bmatrix} h_{1,1} & \cdots & h_{1,d-1} & h_{1,d} \\ 1 & \ddots & \vdots & \vdots \\ & \ddots & h_{d-1,d-1} & h_{d-1,d} \\ & & 1 & h_{d,d} \end{bmatrix}.$$

All of this gives rise to the following definition.

Definition 1.6. *Let $\widehat{\mathcal{A}} \in \mathbb{R}^{N,N}$ be a non-singular matrix with a minimal polynomial of degree $d_{min}(\widehat{\mathcal{A}})$. Let $\mathcal{H}$ be a symmetric positive definite matrix and let s be an integer with $s + 2 \leq d_{min}(\widehat{\mathcal{A}})$.*

1. *If for any initial vector r_0 the matrix H_d is $(s+2)$-band Hessenberg, then we say that $\widehat{\mathcal{A}}$ is reducible for the given $\mathcal{H}$ and r_0 to $(s+2)$-band Hessenberg form.*

2. *If $\widehat{\mathcal{A}}$ is reducible for the given $\mathcal{H}$ and any initial vector r_0 to at most $(s+2)$-band Hessenberg form, while it is reducible for the given $\mathcal{H}$ and at least one r_0 to $(s+2)$-band Hessenberg form, then we say that $\widehat{\mathcal{A}}$ is reducible for the given $\mathcal{H}$ to $(s+2)$-band Hessenberg form.*

Unfortunately, Liesen and Strakoš [72] were not able to show that if $\widehat{\mathcal{A}}$ admits for the given $\mathcal{H}$ an optimal $(s+2)$-term recurrence then $\widehat{\mathcal{A}}$ is reducible for a given $\mathcal{H}$ to $(s+2)$-band Hessenberg form. This relatively easy sounding statement also reduces to an easy expression that has to be shown, i.e.

$$h_{i,d} = \frac{\langle \widehat{\mathcal{A}} v_d, v_i \rangle_{\mathcal{H}}}{\langle v_i, v_i \rangle_{\mathcal{H}}} = 0 \qquad \forall i = 1, \ldots, d-s-1.$$

We know this is true due to the proof given by Faber and Manteuffel [27] and the recently more accessible proof of Faber *et al.* [26] where $\widehat{\mathcal{A}}$ is analyzed as a linear operator. Faber *et al.* also present a linear algebra proof for the assumption $s+3 < d_{min}(\widehat{\mathcal{A}})$. In more detail, complex Givens rotations are used to show that under the assumption of $s+3 < d_{min}(\widehat{\mathcal{A}})$ and $h_{1,d} \neq 0$ the matrix $\widehat{\mathcal{A}}$ would admit a $d_{min}(\widehat{\mathcal{A}})-1$-term recurrence. But since we assumed that $\widehat{\mathcal{A}}$ admits an $s+2$-term recurrence, we get that $d_{min}(\widehat{\mathcal{A}}) - 1 \leq s+2$, which contradicts the assumption that $s+3 < d_{min}(\widehat{\mathcal{A}})$. Faber *et al.* also note that there seems to be no linear algebra proof including the 'missing case' $s+3 = d_{min}(\widehat{\mathcal{A}})$ but that for practical applications, it is desired to have $s+2 \ll d_{min}(\widehat{\mathcal{A}})$.

In the course of this section, we mentioned the Lanczos algorithm and that for symmetric matrices the Arnoldi algorithm reduces to the symmetric Lanczos process (cf. Algorithm 1.2). This also implies that for any symmetric matrix $\widehat{\mathcal{A}}$ we have to find a polynomial of degree s such that $\widehat{\mathcal{A}}^{+} = p_s(\widehat{\mathcal{A}})$ which is a trivial task since $\widehat{\mathcal{A}} = \widehat{\mathcal{A}}^T$. Hence $s = 1$ and every symmetric matrix admits a 3-term recurrence. The class of solvers where $s = 1$ plays a very important role since the methods based on a 3-term recurrence are very efficient and reliable (see Chapter 2. Faber and Manteuffel showed that the class of matrices that are diagonalizable and have eigenvalues on a straight line in the complex plane admit 3-term recurrence methods [27, 72]. As already mentioned this is true for every symmetric matrix $\widehat{\mathcal{A}}$ or $\mathcal{A}$, and it is easy to see that this also holds for skew-symmetric matrices, i.e. $\mathcal{A}^T = -\mathcal{A}$ and $p_1(x) = -x$. In [116,14] an iterative method for the matrix $A+I$, where A is skew symmetric, was proposed. It is easy to see that the eigenvalues of $A+\alpha I$ for skew-symmetric A and $\alpha \in \mathbb{R}$ have real part α and the imaginary part consists of the eigenvalues of A which are purely imaginary. We can now implement iterative methods for (shifted) skew-symmetric systems.

In many cases, it cannot be assumed that the preconditioned matrix $\widehat{\mathcal{A}}$ or even $\mathcal{A}$ itself is symmetric and we recall the Arnoldi process with $\mathcal{H}$-inner product. If a non-standard inner product matrix $\mathcal{H}$ can be found such that $\widehat{\mathcal{A}}$ is symmetric in the inner product $\langle x, y\rangle_{\mathcal{H}} = \langle \mathcal{H}x, y\rangle \; \forall x, y$, i.e.

$$\langle \widehat{\mathcal{A}}x, y\rangle_{\mathcal{H}} = \langle x, \widehat{\mathcal{A}}y\rangle_{\mathcal{H}} \quad \forall x, y \tag{1.8}$$

we can implement a short-term recurrence method.

Mathematically Equation (1.8) reduces to the relation

$$\widehat{\mathcal{A}}^T\mathcal{H} = \mathcal{H}\widehat{\mathcal{A}}. \tag{1.9}$$

Note that (1.9) relation also holds if $\mathcal{H}$ is a bilinear form (see Chapter 3). If (1.9) holds with the inner product defined by $\mathcal{H}$, a 3-term recurrence can always be applied, i.e.

$$\widehat{\mathcal{A}}^T\mathcal{H} = \mathcal{H}\widehat{\mathcal{A}} \Longrightarrow \widehat{\mathcal{A}}^+ = \mathcal{H}^{-1}\widehat{\mathcal{A}}^T\mathcal{H} = \widehat{\mathcal{A}}.$$

1.3 Motivating examples

In this section we want to give two motivating examples that will both result in a linear system of the form (1.1) and frequently appear in applications from various areas.

1.3.1 Stokes problem

The modelling of incompressible fluid flow problems is a problem arising in many applications and is described by PDEs. When considering steady flow problems, i.e. the time dependency of the velocity u is negligible, the resulting PDEs [24, Section 0] are called the Navier-Stokes equations in a region $\Omega \subset \mathbb{R}^d$

$$\begin{aligned} \nu\nabla^2 u + u\cdot\nabla u + \nabla p &= 0 \\ \nabla\cdot u &= 0 \end{aligned} \tag{1.10}$$

where u is the velocity of the fluid and p the pressure. The term ν is called the kinematic viscosity. The Navier-Stokes equations are nonlinear. In situations when the velocity is small or the flow is tightly confined, the Navier-Stokes equations can be simplified. Then the quadratic term in (1.10) can be dropped, and by absorbing the constant ν into the velocity u, we obtain the Stokes equations in a region Ω

$$\begin{aligned} -\nabla^2 u + \nabla p &= 0 \\ \nabla\cdot u &= 0. \end{aligned} \tag{1.11}$$

Hence, we are now given a set of equations that model the slow flow of a viscous fluid. On the boundary $\partial\Omega$ we assume Dirichlet boundary conditions

$$u = w \text{ on } \partial\Omega_D$$

and Neumann boundary conditions

$$\frac{\partial u}{\partial n} - np = s \text{ on } \partial\Omega_N$$

such that

$$\partial\Omega = \partial\Omega_D \cup \partial\Omega_N, \;\; \partial\Omega_D \cap \partial\Omega_N = \emptyset.$$

In order to find approximations to u and p we employ the mixed finite element method [24]. Here the weak formulation

$$\begin{array}{rcl} \int_\Omega \nabla u_h \cdot \nabla v_h - \int_\Omega p_h \nabla \cdot v_h & = & \int_{\partial\Omega_N} s \cdot v_h \\ \int_\Omega q_h \nabla \cdot u_h & = & 0 \end{array} \tag{1.12}$$

is defined using two different finite-dimensional spaces for all v_h and q_h appropriately chosen. In more detail, v_h and q_h are taken from independent spaces which leads to the nomenclature 'mixed approximation'. To find u_h and p_h in the right spaces, we introduce velocity basis functions $\{\phi_j\}$ such that

$$u_h = \sum_{j=1}^{n_u} u_j \phi_j + \sum_{j=n_u+1}^{n_u+n_\partial} u_j \phi_j$$

where the second sum ensures interpolation of the boundary data. Additionally, introducing a set of pressure basis functions $\{\psi_j\}$ and setting

$$p_h = \sum_{j=1}^{n_p} p_j \psi_j$$

the discrete formulation (1.12) can be expressed as a system

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}. \tag{1.13}$$

The matrix $A \in \mathbb{R}^{n,n}$ is called the *vector-Laplacian matrix* and the matrix $B \in \mathbb{R}^{m,n}$ is called the *divergence matrix*. Note that for the dimension of $\mathcal{A}$ $N = n_u + n_p$ holds. The matrix entries are given by

$$\begin{array}{rcl} a_{ij} & = & \int_\Omega \nabla \phi_i \cdot \nabla \phi_j \\ b_{ij} & = & -\int_\Omega \psi_i \nabla \cdot \phi_j \end{array}$$

and the right hand sides by

$$
\begin{array}{rcl}
f_i & = & \int_{\partial\Omega_N} s\cdot \phi_i - \sum_{j=n_u+1}^{n_u+n_\partial} u_j \int_\Omega \nabla\phi_i\cdot \nabla\phi_j \\
g_i & = & \sum_{n_u+1}^{n_u+n_\partial} u_j \int_\Omega \psi_i \nabla\cdot \phi_j .
\end{array}
$$

The system with these definitions given in (1.13) is referred to as the *discrete Stokes problem*. For stabilized elements (cf. [24]) the discrete Stokes problem becomes

$$
\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}. \tag{1.14}
$$

where $C \in \mathbb{R}^{m,m}$ is called *the stabilization matrix*. The matrix

$$
\mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}
$$

is called a *saddle point matrix* because problems of this kind arise in the analysis of saddle points of a given function (see [19, 82]). Matrices of this type play an important role in Numerical Linear Algebra and Numerical Analysis (see [6] for a comprehensive survey). In the case of the Stokes problem, the block A is usually symmetric and positive definite and C is symmetric and positive semi-definite, often zero. Hence, the matrix $\mathcal{A}$ is symmetric and indefinite. With the wide range of applications that can be described by the Stokes equations in mind, it is important to find good solvers and preconditioners that guarantee fast convergence.

1.3.2 Linear programming, scattering and more

In linear programming [42] the primal linear programming problem:

$$\begin{array}{ll} \min_{x\in\mathbb{R}^N} & g^T x \\ \text{s.t.} & \mathcal{A}x \geq b, x \geq 0 \end{array} \tag{1.15}$$

with $g \in \mathbb{R}^N$ always has a corresponding dual problem

$$\begin{array}{ll} \min_{y\in\mathbb{R}^M} & b^T y \\ \text{s.t.} & \mathcal{A}^T y \geq g, y \geq 0. \end{array} \tag{1.16}$$

Here, we assume that $\mathcal{A} \in \mathbb{R}^{N,N}$ is a square matrix and hence $N = M$. The duality theorem [82] states that if the primal problem has a finite optimal solution x^* then the dual problem has an optimal solution y^* and $g^T x^* = b^T y^*$. In the case of all constraints being active constraints that is $\mathcal{A}^T y = g$ and $\mathcal{A}x = b$ the problem of computing the minimiser of the objective function comes down to computing the solution to the linear system

$$\mathcal{A}x = b \tag{1.17}$$

or

$$\mathcal{A}^T y = g. \tag{1.18}$$

We will refer to (1.17) as the *forward problem* and to (1.18) as the *adjoint problem*. Instead of solving only one system of equations, the solution of the corresponding adjoint system can prove useful for further analysis of the problem such as a posteriori error estimation. Examples are given in [113, 93, 90, 63, 41, 22, 5]. Approximating the solutions to the systems (1.17) and (1.18) simultaneously is therefore desired. Solving

$$\mathcal{A}x = b \text{ and } \mathcal{A}^T y = g$$

at the same time can be reformulated as solving

$$\begin{bmatrix} 0 & \mathcal{A} \\ \mathcal{A}^T & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} b \\ g \end{bmatrix}. \tag{1.19}$$

Note that the system matrix in this case describes a degenerated saddle point and hence resembles the structure given in (1.14).

The solution of forward and adjoint system is not only important when looking at optimization problems but also in the world of signal processing. Here, we briefly discuss the scattering amplitude, which is a quantity that is important when one wants to understand the scattering of incoming waves. We assume that an incoming wave given by b impinges on an object. The outgoing wave g then has information about the object, and the matrix $\mathcal{A}$ relates the incoming and the scattered fields. The system $\mathcal{A}x = b$ determines the field from the signal b, and the system $\mathcal{A}^T y = g$ gives the field for the received signal g. In many applications, such as radar, one is of course interested in the amplitude of the scattered field which is given by the quantity $g^T x$ the so-called *scattering amplitude*. The scattering amplitude is also computed in optimization typically under the name primal linear output $J^{pr}(x) = g^T x$.

The problem of simultaneously solving forward and adjoint system and approximating the quantity $g^T x$ not only arises in signal processing and optimization but also in quantum mechanics [68], nuclear physics [2] and many more areas (see [90]).

CHAPTER 2 ITERATIVE SOLVERS

In this chapter, we introduce a number of solvers that are well-suited to solve a linear system of the form (1.1). The focus is here on methods that can be tailored to efficiently solve systems constructed in Section 1.3, i.e. matrices in saddle point form. This chapter neither represents a conclusive list nor will it give all the details for each method; instead, it is supposed to explain the idea behind each method and why this method could be chosen from a practitioners' point of view. The literature about iterative solvers is vast and many good books exists that we recommend for the interested reader, such as [56, 96, 29, 53, 76]. We also recommend [58] where Hageman and Young discuss how to symmetrize iterative methods.

2.1 Symmetric solvers

In this section, the symmetric methods classical Conjugate Gradient (CG) method of Hestenes and Stiefel [59] and MINRES [84] are introduced. They are the most popular choice when people want to solve symmetric linear systems, and their performance motivates one to look for symmetric formulations of mathematical models.

2.1.1 CG (Conjugate Gradient)

The Conjugate Gradient (CG) method introduced by Hestenes and Stiefel in [59] is *the* method for symmetric and positive definite systems of the form (1.1). Due to its property of converging in at most N steps (in infinite precision), many considered CG to be a direct method although even the original paper (see [59]) points out the use as an iterative solver. When Reid [92] analyzed it as an iterative solver for symmetric and positive definite systems, the popularity of CG as an iterative solver took off. In [46], the history of the CG algorithm and its rise to one of the most popular iterative solvers for linear systems is described.

For symmetric and positive definite $\mathcal{A}$, the solution of the linear system

$$\mathcal{A}x = b$$

can be identified with the unique minimiser x of the quadratic form

$$f(x) = \frac{1}{2}x^T\mathcal{A}x - b^Tx + c$$

($c \in \mathbb{R}^N$ is some constant vector) for which

$$f'(x) = \mathcal{A}x - b.$$

Note that for notational convenience we use the notation x for the actual solution of the linear system and quadratic form.

To introduce CG, we start with the method of *steepest descent*. In more detail, the steepest descent from a given point x_k is given in the direction of the negative gradient $-f'(x_k) = r_k = b - \mathcal{A}x_k$.

With a line search technique and the update $x_{k+1} = x_k + \alpha_k r_k$, we can compute α_k such that $f(x_{k+1})$ is minimal. Setting $\frac{\partial f(x)}{\partial \alpha_k} = 0$ yields

$$\alpha_k = \frac{\langle r_k, r_k\rangle}{\langle r_k, \mathcal{A}r_k\rangle}.$$

Note that if we premultiply $x_{k+1} = x_k + \alpha_k r_k$ by $\mathcal{A}$ and subtract b we get

$$r_{k+1} = r_k - \alpha_k \mathcal{A} r_k$$

which is used in Algorithm 2.1. The line search parameter α_k is also computed in such a way that the $\mathcal{A}$-norm of the error along this line is minimized, i.e. $\|e_{k+1}\|_{\mathcal{A}}$ with $e_{k+1} = x - x_{k+1}$. This can be shown with the same technique by considering

```
$r_k = b - \mathcal{A}x_k$
for $k = 0, 1, \ldots$ do
    $\alpha_k = \frac{\langle r_k, r_k \rangle}{\langle r_k, \mathcal{A}r_k \rangle}$
    $x_{k+1} = x_k + \alpha_k r_k$
    $r_{k+1} = r_k - \alpha_k \mathcal{A} r_k$
end for
```

Algorithm 2.1: Steepest descent

$$\frac{\partial e_{k+1}(\alpha_k)}{\partial \alpha_k} = 0$$

or by considering the relation

$$f(x_{k+1}) = f(x) + \frac{1}{2} e_{k+1}^T \mathcal{A} e_{k+1},$$

which identifies the equivalence of minimizing the $\mathcal{A}$-norm to minimizing $f(x_{k+1})$.

The method of steepest descent might often take steps into previously used search directions [103] and therefore introducing a set of orthogonal search directions is desirable. It would be desirable to choose search directions p_j that are orthogonal. Using the condition that the error e_{k+1} is orthogonal to the previous search direction, i.e. $\langle p_k, e_{k+1} \rangle = 0$, so we precisely make one step towards any search direction leads to $\alpha_k = -\frac{\langle p_k, e_k \rangle}{\langle p_k, p_k \rangle}$ which is not computable without knowing the solution x.

Unfortunately, one cannot use orthogonal search directions since their computation would require the knowledge of the solution x. Instead a set of $\mathcal{A}$-conjugate search directions $p_1, \ldots, p_N$ is used. The iterates can now be described by

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

We have to compute α_k such that e_{k+1} is $\mathcal{A}$-orthogonal to p_k since the orthogonality gives that we make only one step into this search direction. From $\langle e_{k+1}, p_k \rangle_{\mathcal{A}} = 0$, we get

$$\alpha_k = \frac{\langle p_k, r_k \rangle}{\langle p_k, Ap_k \rangle}.$$

To create the set of search directions p_k, we use a conjugate Gram-Schmidt process that takes a set of independent vectors u_k to create $\mathcal{A}$-orthogonal direction vectors using a standard Gram-Schmidt technique (see [103] for details). Recall that we introduced a Gram-Schmidt process for the Krylov space in (1.7). For general vectors u_k, this process is called the method of *conjugate directions*. With the particular choice of u_k being equal to the residual r_k we can derive the CG method. This choice gives that the residual r_k is also orthogonal to the previous search directions. To see this, we note that the error e_k is A-orthogonal to all previous search directions $p_j \ \forall j \neq k$, i.e. $\langle p_j, \mathcal{A} e_k \rangle = 0$ so only one step towards every direction is made, and because of $r_k = -Ae_k$, we get that $\langle p_j, r_k \rangle = 0$. Remembering that the p_k are generated from $u_k = r_k$ via a Gram-Schmidt process, we get that $p_j = r_j - \sum_{i=1}^{j-1} \beta_i p_i$. Now looking at

$$\langle p_j, r_k \rangle = \langle r_j, r_k \rangle - \sum_{i=1}^{j-1} \beta_i \langle p_i, r_k \rangle$$

with the knowledge that $\langle p_j, r_k \rangle = 0$, we see that $\langle r_j, r_k \rangle = 0 \ \forall j \neq k$. Having chosen the u_k to be equal to the residuals gives

$$span\left\{p_0, p_1, \ldots\right\} = span\left\{r_0, r_1, \ldots\right\},$$

and using the fact that

$$r_{k+1} = r_k - \alpha_k \mathcal{A} p_k, \tag{2.1}$$

it is clear that the search space is equal to the Krylov subspace $\mathcal{K}_k(\mathcal{A}, r_0) = span\left\{r_0, \mathcal{A}r_0, \ldots, \mathcal{A}^{k-1}r_0\right\}$. Because the Krylov subspaces define a nested sequence and the next residual r_{k+1} is orthogonal to the space spanned by $p_0, \ldots, p_k$ and because of (2.1), r_{k+1} is $\mathcal{A}$-orthogonal to the space spanned by $p_0, \ldots, p_{k-1}$. This means that the Gram-Schmidt process for r_{k+1} only has to orthogonalize against p_k. In more detail, the Gram-Schmidt recurrence becomes

$$p_{k+1} = r_{k+1} - \beta_{k+1}p_k$$

where $\beta_{k+1} = \frac{\langle r_{k+1}, \mathcal{A}p_k\rangle}{\langle p_k, \mathcal{A}p_k\rangle}$. This can be further rewritten when taking the inner product of (2.1) and r_{k+1} which gives

$$\langle r_{k+1}, r_{k+1}\rangle = \langle r_{k+1}, r_k\rangle - \alpha_k\langle r_{k+1}, \mathcal{A}p_k\rangle.$$

Since the residuals are orthogonal, this reduces to $\langle r_{k+1}, r_{k+1}\rangle = -\alpha_k\langle r_{k+1}, \mathcal{A}p_k\rangle$. Given the definition of α_k and

$$\langle r_{k+1}, \mathcal{A}p_k\rangle = -\frac{\langle r_{k+1}, r_{k+1}\rangle}{\alpha_k},$$

we get

$$\beta_{k+1} = \frac{\langle r_{k+1}, r_{k+1}\rangle}{\langle p_k, r_k\rangle}.$$

Using $p_k = r_k - \beta_k p_{k-1}$ and the previous results, we show that $\langle p_k, r_k\rangle = \langle r_k, r_k\rangle - \beta_k\langle p_{k-1}, r_k\rangle = \langle r_k, r_k\rangle$, and the parameters finally become

$$\beta_{k+1} = \frac{\langle r_{k+1}, r_{k+1}\rangle}{\langle r_k, r_k\rangle} \text{ and } \alpha_k = \frac{\langle r_k, r_k\rangle}{\langle p_k, Ap_k\rangle}.$$

Bringing all the pieces together, we get the CG method given in Algorithm 2.2.

Similar to the steepest descent method, CG minimizes the error in the $\mathcal{A}$-norm over the current Krylov subspace, i.e. $\|e_k\|_{\mathcal{A}}$. For a more detailed discussion of the further properties of CG we refer to [59]. This method is

```
$r_0 = b - \mathcal{A}x_0$
$p_0 = r_0$
for $k = 0, 1, \ldots$ do
    $\alpha_k = \frac{\langle r_k, r_k \rangle}{\langle p_k, \mathcal{A}p_k \rangle}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k \mathcal{A}p_k$
    $\beta_{k+1} = \frac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle}$
    $p_{k+1} = r_{k+1} - \beta_{k+1} p_k$
end for
```

Algorithm 2.2: Conjugate Gradient (CG) method

also closely connected to the symmetric Lanczos process given in Algorithm 1.2 or [66] which is carefully explained in [53]. Hence, as a method based on the Lanczos process, CG fulfills all the requirements of an optimal 3-term recurrence method that we described in Section 1.2.

So far, we have introduced CG as a solver for the system matrix $\mathcal{A}$, but in practice we would be interested in solving a preconditioned system (1.3). Since CG is a method only for positive definite symmetric systems, it is easy to see that the preconditioner $\mathcal{P}$ has to be positive definite to be applicable. In more detail, consider the system

$$\mathcal{P}^{-1}\mathcal{A}x = \mathcal{P}^{-1}b$$

and with $\mathcal{P}$ being symmetric and positive definite we can compute the Cholesky decomposition $\mathcal{P} = R^T R$. Then, we get the spectrally equivalent, centrally preconditioned system

$$R^{-T}\mathcal{A}R^{-1}\hat{x} = R^{-T}b, \text{ with } \hat{x} = Rx$$

with a symmetric and positive definite system matrix. The preconditioned version of the CG algorithm is given in Algorithm 2.3. The preconditioned CG (PCG) is the most common method to solve preconditioned symmetric and positive definite systems, and we refer to the literature for a more detailed discussion (see [24, 96, 56, 75]). The preconditioned CG still minimizes the

```
$r_0 = b - \mathcal{A}x_0$
Solve $\mathcal{P}z_0 = r_0$
for $k = 0, 1, \ldots$ do
    $\alpha_k = \frac{\langle r_k, z_k \rangle}{\langle p_k, \mathcal{A}p_k \rangle}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k \mathcal{A}p_k$
    Solve $\mathcal{P}z_{k+1} = r_{k+1}$
    $\beta_{k+1} = \frac{\langle r_{k+1}, z_{k+1} \rangle}{\langle r_k, z_k \rangle}$
    $p_{k+1} = z_{k+1} - \beta_{k+1} p_k$
end for
```

Algorithm 2.3: Preconditioned Conjugate Gradient (CG) method

$\mathcal{A}$-norm of the error regardless of the choice of $\mathcal{P}$ as long as it is symmetric and positive definite.

For some problems, the preconditioned matrix $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$ might not fulfill the requirements of the CG method, i.e. all matrix symmetries are destroyed when $\mathcal{A}$ is preconditioned, in which case the Faber-Manteuffel theorem (Section 1.2) suggests that no short-term recurrence method can be applied. Introducing an appropriate inner product $\langle ., . \rangle_{\mathcal{H}}$ the matrix $\widehat{\mathcal{A}}$ may be symmetric and positive definite in this alternative inner product, in which case the Hessenberg matrix of the $\mathcal{H}$-inner product Arnoldi algorithm (1.7) will be tridiagonal. This means that under these circumstances CG is applicable. Such methods play a major role in this thesis and we come back to CG with such non-standard inner products in Chapters 3 and 4. There, we also show that CG with a non-standard inner product is equivalent to a special PCG. A straightforward implementation of CG with non-standard inner product is given in Algorithm 2.4. In the case of CG with non-standard inner product, the preconditioner $\mathcal{P}$ is chosen such that the eigenvalues of the preconditioned matrix $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$ are clustered since the inner product does not influence the eigenvalues of preconditioned system.

At iteration k of Algorithm 2.4,

$$\text{span}\left\{p_0, p_1, \ldots, p_{k-1}\right\} = \text{span}\left\{r_0, r_1, \ldots, r_{k-1}\right\},$$

```
Given $x_0 = 0$, set $r_0 = \mathcal{P}^{-1}(b - \mathcal{A}x_0)$ and $p_0 = r_0$
for $k = 0, 1, \ldots$ do
    $\alpha = \frac{\langle r_k, p_k \rangle_{\mathcal{H}}}{\langle \mathcal{P}^{-1}\mathcal{A}p_k, p_k \rangle_{\mathcal{H}}}$
    $x_{k+1} = x_k + \alpha p_k$
    $r_{k+1} = r_k - \alpha \mathcal{P}^{-1}\mathcal{A}p_k$
    $\beta = \frac{\langle \mathcal{P}^{-1}\mathcal{A}r_{k+1}, p_k \rangle_{\mathcal{H}}}{\langle \mathcal{P}^{-1}\mathcal{A}p_k, p_k \rangle_{\mathcal{H}}}$
    $p_{k+1} = r_{k+1} - \beta p_k$
end for
```

Algorithm 2.4: Non-standard inner-product CG (variant 1)

$\langle r_k, r_j \rangle_{\mathcal{H}} = 0$, $\langle r_k, p_j \rangle_{\mathcal{H}} = 0$ and $\langle \mathcal{P}^{-1}\mathcal{A}p_k, p_j \rangle_{\mathcal{H}} = 0$ for all $j < k$, (see [70, Theorem 3.2]). This will lead to a simplification of Algorithm 2.4. To see this, we look at

$$\langle r_k, p_k \rangle_{\mathcal{H}} = \langle r_k, r_k + \beta p_{k-1} \rangle_{\mathcal{H}} = \langle r_k, r_k \rangle_{\mathcal{H}}$$

using the $\mathcal{H}$-orthogonality between r_k and p_{k-1}. The expression for β can be simplified by looking at

$$\langle r_{k+1}, r_{k+1} \rangle_{\mathcal{H}} = \langle r_{k+1}, r_k \rangle_{\mathcal{H}} - \alpha \langle r_{k+1}, \mathcal{P}^{-1}\mathcal{A}p_k \rangle_{\mathcal{H}}$$

using the definition of r_{k+1}. Furthermore, if we use the definition of α and the $\mathcal{H}$-orthogonality between r_{k+1} and r_k we get

$$\langle r_{k+1}, r_{k+1} \rangle_{\mathcal{H}} = -\frac{\langle r_k, p_k \rangle_{\mathcal{H}} \langle r_{k+1}, \mathcal{P}^{-1}\mathcal{A}p_k \rangle_{\mathcal{H}}}{\langle \mathcal{P}^{-1}\mathcal{A}p_k, p_k \rangle_{\mathcal{H}}}.$$

Finally, this gives

$$\beta = \frac{\langle r_{k+1}, r_{k+1} \rangle_{\mathcal{H}}}{\langle r_k, r_k \rangle_{\mathcal{H}}}$$

using the previous result $\langle r_k, r_k \rangle_{\mathcal{H}} = \langle r_k, p_k \rangle_{\mathcal{H}}$ and the update $p_{k+1} = r_{k+1} + \beta p_k$. Hence, we can now reformulate Algorithm 2.4 as Algorithm 2.5.

CG and PCG both minimize the $\mathcal{A}$-norm of the error. For CG with

```
Given $x_0 = 0$, set $r_0 = \mathcal{P}^{-1}(b - \mathcal{A}x_0)$ and $p_0 = r_0$
for $k = 0, 1, \ldots$ do
    $\alpha = \frac{\langle r_k, r_k \rangle_{\mathcal{H}}}{\langle \mathcal{P}^{-1}\mathcal{A}p_k, p_k \rangle_{\mathcal{H}}}$
    $x_{k+1} = x_k + \alpha p_k$
    $r_{k+1} = r_k - \alpha \mathcal{P}^{-1}\mathcal{A}p_k$
    $\beta = \frac{\langle r_{k+1}, r_{k+1} \rangle_{\mathcal{H}}}{\langle r_k, r_k \rangle_{\mathcal{H}}}$
    $p_{k+1} = r_{k+1} + \beta p_k$
end for
```

Algorithm 2.5: Non-standard inner-product CG (variant 2)

inner product defined by $\mathcal{H}$ the error, e_{k+1} is now minimized in the norm defined by the matrix $\mathcal{H}\widehat{\mathcal{A}}$, i.e. $\|e_{k+1}\|_{\mathcal{H}\widehat{\mathcal{A}}}$. Note that CG with inner product is only applicable if $\mathcal{H}\widehat{\mathcal{A}}$ is symmetric and positive definite and hence defines a norm. In (1.9) we showed that this matrix has to be symmetric for the preconditioned matrix to be symmetric in the inner product defined by $\mathcal{H}$. If we also want the matrix to be positive definite in this inner product, we get $\langle \widehat{\mathcal{A}}x, x \rangle_{\mathcal{H}} > 0 \Leftrightarrow \langle \mathcal{H}\widehat{\mathcal{A}}x, x \rangle > 0$, which is only true if $\mathcal{H}\widehat{\mathcal{A}}$ is a positive definite matrix itself. For more details, see [70, 24].

2.1.2 MINRES

The minimal residual method (MINRES) [84] is an iterative solver for symmetric systems which may be indefinite, such as the saddle point system (1.14) introduced in Section 1.3. MINRES is not the preferred method for symmetric and positive definite systems as it is marginally more expensive than CGand it does not minimize the $\mathcal{A}$-norm of the error. The MINRES method is based on the symmetric Lanczos procedure (see 1.6), which can be expressed as

$$AV_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^T = V_{k+1} T_{k+1,k}$$

with

$$T_{k+1,k} = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_k \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}.$$

The quantity that is minimized in MINRES, namely the 2-norm of the residual, differs from the one minimized in CG. The approximate solution x_k is of the form

$$x_k = x_0 + V_k z_k \tag{2.2}$$

for some vector z_k where the columns of V_k form an orthogonal basis of the Krylov subspace $\mathcal{K}_k(\mathcal{A}, r_0)$. We refer to the condition (2.2) as the space condition because the current approximation x_k is a linear combination of the starting vector x_0 and the actual basis of the Krylov space $\mathcal{K}_k(\mathcal{A}, r_0)$. The vector z_k is computed such that the 2-norm of the current residual $r_k = b - \mathcal{A}x_k$ is minimized. Mathematically, this is expressed as

$$\begin{aligned} \|r_k\|_2 &= \|b - \mathcal{A}x_k\|_2 \\ &= \|b - \mathcal{A}(x_0 + V_k z_k)\|_2 \\ &= \|r_0 - \mathcal{A}V_k z_k\|_2 \\ &= \|r_0 - V_{k+1}T_{k+1,k} z_k\|_2 \end{aligned} \tag{2.3}$$

and with the typical choice of $v_1 = r_0 / \left\| r_0 \right\|_2$ we get

$$\begin{aligned} \|r_k\|_2 &= \|V_{k+1}(\left\|r_0\right\|_2 e_1 - T_{k+1,k} z_k)\|_2 \\ &= \| \left\|r_0\right\|_2 e_1 - T_{k+1,k} z_k\|_2. \end{aligned} \tag{2.4}$$

The term V_{k+1} inside the norm can be ignored because its columns are orthogonal in exact arithmetic. In order to compute the vector z_k, we have to solve the least squares problem (2.4), i.e.

$$\min \|r_k\|_2 = \min \| \, \|r_0\|_2 \, e_1 - T_{k+1,k} z_k \|_2.$$

A well-known technique to solve such a least squares system is the QR decomposition (cf. [84]). Computing a QR decomposition at every step would pose serious computational cost to the algorithm, but, since the matrix $T_{k+1,k}$ changes from step to step simply by adding one column and one row, its QR decomposition can be updated at every step. The factorization can be updated at each step using just one Givens rotation. In more detail, we assume that the QR factorization of $T_{k,k-1} = Q_{k-1} R_{k-1}$ is given with

$$R_{k-1} = \begin{bmatrix} \hat{R}_{k-1} \\ 0 \end{bmatrix}$$

and $\hat{R}_{k-1}$ is an upper triangular matrix. To obtain the QR factorization of $T_{k+1,k}$ we eliminate the element β_{k+1} from

$$\begin{aligned} \begin{bmatrix} Q_{k-1}^T & 0 \\ 0 & 1 \end{bmatrix} T_{k+1,k} &= \begin{bmatrix} Q_{k-1}^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T_{k,k-1} & \alpha_k e_k \\ 0 & \beta_{k+1} \end{bmatrix} \\ &= \begin{bmatrix} R_{k-1} & Q_{k-1}^T \alpha_k e_k \\ 0 & \beta_{k+1} \end{bmatrix} \end{aligned} \tag{2.5}$$

by using one Givens rotation in rows $k, k+1$. There is no need to store the whole basis V_k in order to update the solution. The matrix R_k of the QR decomposition of the tridiagonal matrix $T_{k+1,k}$ has only three non-zero diagonals. Let us define $C_k = [c_0, c_1, \ldots, c_{k-1}] = V_k \hat{R}_k^{-1}$. Note that c_0 is a multiple of v_1 and we can compute successive columns using that $C_k\hat{R}_k = V_k$, i.e.

$$c_{k-1} = (v_k - \hat{r}_{k-1,k}c_{k-2} - \hat{r}_{k-2,k}c_{k-3})/\hat{r}_{k,k} \tag{2.6}$$

where the $\hat{r}_{i,j}$ are elements of $\hat{R}_k$. Therefore, we can update the solution

$$x_k = x_0 + \|r_0\|_2 \, C_k \left(Q_k^T e_1\right)_{k\times 1} = x_{k-1} + a_{k-1}c_{k-1} \tag{2.7}$$

where a_{k-1} is the kth entry of $\|r_0\| \, Q_k^T e_1$. The complete method is given in Algorithm 2.6. When considering MINRES in finite precision, we have to deal with the loss of orthogonality of the vectors generated by the Lanczos process. This can result in non-convergence of the method. In [56,77] the problems of dealing with MINRES in finite precision are explained very carefully. In [106] it is explained why MINRES is more prone to roundoff errors than its close relative, SYMMLQ [84].

```
$r_0 = b - \mathcal{A}x_0$
Set $v_1 = r_0/\|r_0\|_2$
while residual norm $>$ tolerance do
    Compute $v_{k+1}$, $T_{k+1,k}$ via Lanczos algorithm
    Update $QR$ decomposition
    Solve min $\|\|r_0\|_2 e_1 - T_{k+1,k}z_k\|_2$
    if Convergence criterion fulfilled then
        $x_k = x_0 + V_k w_k$ using (2.7)
        stop
    end if
end while
```

Algorithm 2.6: MINRES

Again, in practice the method is hardly ever used without preconditioning due to the conditioning of the system matrices and the poor clustering of the eigenvalues. Therefore, a preconditioner $\mathcal{P}$ is introduced and we are working with the system matrix $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$, which is spectrally equivalent to the centrally preconditioned system $R^{-T}\mathcal{A}R^{-1}$ with $\mathcal{P} = R^TR$. Note that the preconditioner has to be positive definite in order to be able to use a method for symmetric matrices because for indefinite $\mathcal{P}$ there is no spectrally equivalent centrally preconditioned symmetric system. A preconditioned MINRES method can be implemented such that one solve with $\mathcal{P}$ and one multiplication with $\mathcal{A}$ has to be performed (see the preconditioned Lanczos in Algorithm 2.7). The preconditioned implementation would differ from Algorithm 2.6 only in the first statement of the while loop where a preconditioned Lanczos process generates the Lanczos vector and the tridiagonal matrix. For more details, we refer to [24,56,84]. Note that the quantity minimized in the preconditioned MINRES is the $\mathcal{P}^{-1}$-norm of the residual or, equivalently, the *pseudoresidual* in the 2-norm. To see this, we look at norm of the pseudoresidual $\hat{r}_k = R^{-T}r_k$, i.e. $\|\hat{r}_k\|$, for the centrally preconditioned system

$$R^{-T}\mathcal{A}R^{-1}\hat{x} = R^{-T}b, \text{ with } \hat{x} = Rx$$

and get

$$\|\hat{r}_k\|_2 = \left\|R^{-T}(b - \mathcal{A}x_k)\right\|_2 = \|b - \mathcal{A}x_k\|_{R^{-T}R^{-1}} = \|r_k\|_{\mathcal{P}^{-1}}.$$

Note that in contrast to CG the minimized quantity changes when MINRES is considered with preconditioning.

```
Set $v_1 = r_0/\|r_0\|$, solve $\mathcal{P}\tilde{w}_1 = v_1$, compute $\beta_1 = \langle v_1\tilde{w}_1\rangle^{1/2}$
Set $q_1 = v_1/\beta_1$, $w_1 = \tilde{w}_1/\beta_1$ and $q_0 = 0$
for $k = 1, 2, \ldots$ do
    Compute $v_{k+1} = Aw_k - \beta_{k-1}q_{k-1}$
    Compute $\alpha_k = \langle v_k, w_k\rangle$
    Compute $v_{k+1} = v_{k+1} - \alpha_k q_k$
    Solve $\mathcal{P}\tilde{w}_{k+1} = v_{k+1}$
    Compute $\beta_{k+1} = \langle v_{k+1}, \tilde{w}_{k+1}\rangle^{1/2}$
    Set $q_{j+1} = v_{k+1}/\beta_{k+1}$ and $w_{k+1} = \tilde{w}_{k+1}/\beta_{k+1}$
end for
```

Algorithm 2.7: Preconditioned Lanczos (Modified Gram-Schmidt)

The preconditioned MINRES method is a very popular method when solving symmetric but indefinite systems such as the saddle point problem

(1.14). A typical preconditioner in that case is given by

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ 0 & S_0 \end{bmatrix}$$

where A_0 is a preconditioner for the $(1,1)$ block of the saddle point system (1.14) and S_0 a Schur-complement preconditioner (see [114, 104] for details). Note that A_0 and S_0 have to be positive definite in order for MINRES to be applicable. We will revisit this preconditioner in Chapter 6.

Again, an inner product defined by $\mathcal{H}$ as used in Section 1.2 about the Faber-Manteuffel theorem can also be used in MINRES. We call this method $\mathcal{H}$-MINRES and briefly discuss its properties here. We assume that the preconditioned matrix $\widehat{\mathcal{A}}$ is symmetric in the inner product induced by $\mathcal{H}$. Hence, we can use an $\mathcal{H}$-Lanczos version of the classical Lanczos method (Algorithm 1.3) to generate a basis for the Krylov subspace and then minimize the $\mathcal{H}$-norm of the preconditioned residual. Using the $\mathcal{H}$-Lanczos method we get

$$\begin{aligned}
\|r_k\|_{\mathcal{H}} &= \|b - Ax_k\|_{\mathcal{H}} \\
&= \|b - Ax_0 - AV_k y_k\|_{\mathcal{H}} \\
&= \|r_0 - V_{k+1}T_{k+1}y_k\|_{\mathcal{H}} \\
&= \left\|V_{k+1}(V_{k+1}^T\mathcal{H}r_0 - T_{k+1}y_k)\right\|_{\mathcal{H}} \\
&= \left\|V_{k+1}^T\mathcal{H}r_0 - T_{k+1}y_k\right\|_{\mathcal{H}} \\
&= \left\| \|r_0\|_{\mathcal{H}}\, e_1 - T_{k+1}y_k \right\|_{\mathcal{H}}.
\end{aligned} \tag{2.8}$$

Based on (2.8), a $\mathcal{H}$-MINRES process which minimizes the $\mathcal{H}$-norm of the preconditioned residual (2.8) can be implemented in complete analogy to the standard MINRES method given in Algorithm 2.6.

2.2 Non-symmetric solvers

Due to the effectiveness of solvers such as MINRES and CG obtaining symmetric matrices from models and applications is desirable. Unfortunately, there are many problems where it is not possible to obtain a symmetric representation of the mathematical problem apart from the normal equations. For these cases, non-symmetric solvers have to be introduced. In this Chapter, we only scratch the surface of methods available and point the interested reader to [96, 56, 24, 76, 86] for a more detailed description or a larger variety of solvers.

2.2.1 GMRES (Generalized Minimal Residual Method)

The most popular non-symmetric solver is probably the generalized minimal residual method (GMRES) which was introduced by Saad and Schultz in [97]. It is based on the Arnoldi matrix relation

$$\mathcal{A}V_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T = V_{k+1} H_{k+1,k},$$

which we introduced in (1.7). Again, using the space condition

$$x_k = x_0 + V_k z_k$$

the vector z_k is computed such that the 2-norm of the current residual $r_k = b - \mathcal{A}x_k$ is minimized. As mentioned earlier, MINRES represents a special case of GMRES in the same way the symmetric Lanczos process is a special case of the Arnoldi algorithm (see Section 1.2). In more detail, we can use the space condition and the Arnoldi process to obtain for the residual

$$\|r_k\|_2 \quad = \quad \|r_0 - V_{k+1} H_{k+1,k} z_k\|_2 \tag{2.9}$$

and with the choice $v_1 = r_0/\|r_0\|_2$ (2.9) becomes

$$\|r_k\|_2 = \|\|r_0\|_2 e_1 - H_{k+1,k} z_k\|_2. \tag{2.10}$$

Minimizing equation (2.10) means solving the least squares problem

$$\min \|r_k\|_2 = \min_{z_k} \|\|r_0\|_2 e_1 - H_{k+1,k} z_k\|_2.$$

In the same fashion as discussed for MINRES, the least squares problem can be solved by using an updated QR decomposition that needs one Givens rotation at every step. A very simple implementation of the GMRES method is shown in Algorithm 2.8. A drawback of this method is that the underlying Arnoldi process is expensive because it performs a full Gram-Schmidt orthogonalization process at every step. This entails more evaluations of scalar products and also more storage requirements. Therefore, restarting techniques have been introduced for GMRES (see [97] for details). Convergence of such restarted methods is, however, not guaranteed (see [25]).

```
Compute $r_0 = b - \mathcal{A}x_0$
Set $v_1 = r_0/\|r_0\|$
while residual norm > tolerance do
  Compute $v_{k+1}$, $H_{k+1,k}$ via Arnoldi algorithm
  Update $QR$ decomposition
  Solve $\min \|\|r_0\|_2 e_1 - H_{k+1,k} z_k\|_2$
  if Convergence criterion fulfilled then
    Compute $x_k = x_0 + V_k w_k$
    stop
  end if
end while
```

Algorithm 2.8: GMRES

Again, for most problems, preconditioning the linear system is necessary and can be incorporated without many difficulties into the algorithm. An implementation of the preconditioned GMRES that needs only one evaluation of the preconditioner and one multiplication with $\mathcal{A}$ can be found in [96]. In the context of the Faber-Manteuffel theorem it could be suggested to use the Arnoldi algorithm with an $\mathcal{H}$-inner product to obtain a GMRES version with a non-standard inner product. At this stage we are not aware of any research devoted to this particular problem and feel that this is an interesting project for further research.

2.2.2 QMR and ITFQMR

The disadvantage of GMRES is that orthogonalization against all the previous vectors in the Krylov subspace is needed, which means significant storage requirements for the method when more than just a few iterations are required. Other alternatives are based on the non-symmetric Lanczos process.

The non-symmetric Lanczos process (cf. [96, 35, 38, 39, 56]) for the preconditioned matrix $\widehat{\mathcal{A}}$ generates two sequences of vectors v_k and w_k that are bi-orthogonal, i.e. $\langle v_i, w_j \rangle = 0 \quad \forall i \neq j$ and are generated by

$$\rho_{k+1} v_{k+1} \quad = \quad \widehat{\mathcal{A}} v_k - \mu_k v_k - \nu_{k-1} v_{k-1} \tag{2.11}$$

for the first sequence and

$$\zeta_{k+1} w_{k+1} \quad = \quad \widehat{\mathcal{A}}^T w_k - \mu_k w_k - \tfrac{\nu_{k-1}\rho_k}{\zeta_k} w_{k-1} \tag{2.12}$$

for the second sequence with $\mu_k = w_k^T \widehat{\mathcal{A}} v_k / w_k^T v_k$ and $\nu_k = \zeta_k w_k^T v_k / w_{k-1}^T v_{k-1}$. There is more than one way to scale the two vectors in every iteration step and hence how to determine ζ_k and ρ_k. Note that the method is introduced for the preconditioned matrix $\widehat{\mathcal{A}}$.

Here, we use $\|v_j\| = 1$ and $\|w_j\| = 1$. The biorthogonality condition between W_k and V_k, i.e. $W_k^T V_k = D_k$, gives

$$D_k = diag(\delta_1, \delta_2, \ldots, \delta_k) \text{ where } \delta_j = \langle w_j, v_j \rangle. \tag{2.13}$$

Note that $D_k = I$ can also be chosen but then $\|v_j\| = 1 \neq \|w_j\| \neq 1$.

Furthermore, we can now write the recursions in terms of matrices and get

$$\begin{aligned} \widehat{\mathcal{A}}V_k &= V_{k+1}T_{k+1,k} \\ \widehat{\mathcal{A}}^T W_k &= W_{k+1}\Gamma_{k+1}^{-1}T_{k+1,k}\Gamma_{k+1} \end{aligned} \tag{2.14}$$

where $\Gamma_k = diag(1, \gamma_2, \ldots, \gamma_k)$. One advantage of the non-symmetric Lanczos process is that $T_{k+1,k}$ is a tridiagonal matrix

$$T_{k,k} = \begin{bmatrix} \mu_1 & \nu_2 & & \\ \rho_2 & \mu_2 & \ddots & \\ & \ddots & \ddots & \nu_k \\ & & \rho_k & \mu_k \end{bmatrix},$$

which is typically non-symmetric.

There are different cases where the non-symmetric Lanczos process can break down. The first case is the so-called *lucky breakdown*, that is, when v_j and/or w_j are zero. This indicates that the solution lies already in the current Krylov space. In the case of $\langle w_j, v_j \rangle = 0$ and neither v_j nor w_j are zero the so-called *serious breakdown* occurs. In these cases it might be possible to recover by increasing the number of vectors used to generate new Lanczos vectors. This gives the so-called look-ahead strategies where the next Lanczos vectors are computed without needing the existence of the current ones (see [89, 36] for more details). The drawback of this approach is that additional cost are imposed on the algorithm. There are also cases where the look-ahead strategies will not be of any use since no solution can be obtained, the so-called *incurable breakdowns*.

One method using the non-symmetric Lanczos process is the quasi minimal residual (QMR) algorithm. It was developed by Freund and Nachtigal in 1991 (cf. [37]). The method can be derived in a very similar way to GMRES by starting with the space condition

$$x_k = x_0 + V_k y_k.$$

Then, the residual can be expressed as

$$\begin{aligned} r_k &= b - \widehat{\mathcal{A}} x_k \\ &= b - \widehat{\mathcal{A}}(x_0 + V_k y_k) \\ &= r_0 - \widehat{\mathcal{A}} V_k y_k \\ &= r_0 - V_{k+1} T_{k+1,k} y_k \\ &= V_{k+1}(\|r_0\|_2 e_1 - T_{k+1,k} y_k). \end{aligned} \tag{2.15}$$

In the case of an orthonormal V_{k+1}, we obtain GMRES. In the case of a non-orthogonal matrix V_{k+1}, such as that generated by the non-symmetric Lanczos method, the idea of ignoring the V_{k+1}-part of (2.15) and solving the least squares problem

$$\min \| \|r_0\|_2 e_1 - T_{k+1,k} y_k \|_2 \tag{2.16}$$

seems reasonable. Here, $r_k^Q = \|r_0\| e_1 - T_{k+1,k} y_k$ is called the *quasi-residual*. If furthermore the columns of V_{k+1} are normalized we get that $\|V_{k+1}\|_2 \leq \sqrt{k+1}$ (see [56]). From [80] we also get that when no weights are used in the non-symmetric Lanczos process the relation

$$\left\| r_k^Q \right\|_2 \leq \kappa(V_{k+1}) \left\| r_k \right\|_2$$

holds, where $\kappa(V_{k+1})$ is the condition number of V_{k+1}. This shows that the for a reasonable basis the QMR residual r_k^Q is not too far away from the GMRES residual. It is easy to see that the residual for the solution with QMR can never be smaller than the residual coming from GMRES. On the other hand, QMR is much cheaper since it uses less storage and also fewer evaluation of inner products. The solution of the least squares problem (2.16) can be obtained in the same way as presented for MINRES, i.e. an updated QR decomposition can be computed using only one Givens rotation. An implementation of the QMR method is given in [37] and also in [96].

In [35, 34, 39] a simplified version of QMR based on a simplification of the non-symmetric Lanczos process is introduced. The resulting method is called *ideal transpose-free* QMR (ITFQMR) or *simplified* QMR. The basis for the simplification of the Lanczos process is when $\widehat{\mathcal{A}}$ is self-adjoint in the bilinear form defined by $\mathcal{H}$, i.e. where

$$\widehat{\mathcal{A}}^T\mathcal{H} = \mathcal{H}\widehat{\mathcal{A}}.$$

In [38] Freund and Nachtigal observe that for the Lanczos vectors the relation

$$v_j = \phi_j(\widehat{\mathcal{A}})v_1 \text{ and } w_j = \gamma_j\phi_j(\widehat{\mathcal{A}}^T)w_1 \tag{2.17}$$

holds where ϕ is the so-called *Lanczos polynomial* which is of a polynomial of degree $j-1$. Using Equation (2.17) and setting $w_1 = \mathcal{H}v_1$, we get

$$w_j = \gamma_j\phi_j(\widehat{\mathcal{A}}^T)w_1 = \gamma_j\phi_j(\widehat{\mathcal{A}}^T)\mathcal{H}v_1 = \gamma_j\mathcal{H}\phi_j(\widehat{\mathcal{A}})v_1 = \gamma_j\mathcal{H}v_j,$$

by repeatedly using $\widehat{\mathcal{A}}^T\mathcal{H} = \mathcal{H}\widehat{\mathcal{A}}$ to shift the matrix $\mathcal{H}$ from one side of the polynomial to the other. Hence, we can compute the vector w_j without multiplying by $\widehat{\mathcal{A}}^T$. Instead,

$$w_{j+1} = \gamma_{j+1}\mathcal{H}v_{j+1} \tag{2.18}$$

can be used. The parameter $\gamma_{j+1} = \gamma_j\rho_{j+1}/\zeta_{j+1}$ involves ζ_{j+1} which cannot be computed at that time. Thus the relation (2.18) has to be reformulated to

$$\tilde{w}_{j+1} = \zeta_{j+1}w_{j+1} = \gamma_j\rho_{j+1}\mathcal{H}v_{j+1} = \gamma_j\mathcal{H}\tilde{v}_{j+1}$$

which gives us now a computable version of the simplified Lanczos method (see Algorithm 2.9).

```
Choose $v_1$ and compute $w_1 = \mathcal{H}v_1$
Compute $\rho_1 = \|v_1\|$ and $\zeta_1 = \|w_1\|$
Set $\gamma_1 = \frac{\rho_1}{\zeta_1}$
for $k = 1, 2, \ldots$ do
    Compute $\mu_k = (w_k^T \widehat{\mathcal{A}} v_k)/(w_k^T v_k)$
    Set $\nu_k = \zeta_k(w_k^T v_k)/(w_{k-1}^T v_{k-1})$
    $v_{k+1} = Av_k - \mu_k v_k - \nu_k v_{k-1}$
    $w_{k+1} = \gamma_k \mathcal{H} v_{k+1}$
    Compute $\rho_{k+1} = \|v_{k+1}\|$ and $\zeta_{k+1} = \|w_{k+1}\|$
    Set $\gamma_{k+1} = \gamma_k \rho_{k+1}/\zeta_{k+1}$.
end for
```

Algorithm 2.9: Simplified Lanczos method

Freund's ITFQMR implementation is based on a QMR-from-BICG procedure and coupled two term recurrence relations (details can be found in [35, 39]). Another way of implementing ITFQMR is to omit multiplications with $\widehat{\mathcal{A}}^T$ and replace them by multiplications with the matrix $\mathcal{H}$ in the standard QMR implementation. When using QMR and similar methods, we have to keep in mind that the quantities minimized here, the quasi-residual in the case of ITFQMR, are not as well understood as the corresponding quantities used in MINRES and CG. Furthermore, as a method based on the non-symmetric Lanczos process ITFQMR can break down and look-ahead strategies have to be employed (see [89, 36] for more details). There are also incurable breakdowns, but from our experience, it is hard to find them in practical applications. The ITFQMR method is based on the simplification using the self-adjointness in $\mathcal{H}$. There are more methods based on the non-symmetric Lanczos process that avoid the multiplication with $\widehat{\mathcal{A}}^T$. One of the earliest to consider such methods is Sonneveld in [107].

2.2.3 BICG

The BICG algorithm was derived in [31,67] as a non-symmetric version of CG. It can be derived from the non-symmetric Lanczos process in the same way

```
Choose $x_0$ and compute $r_0 = b - \widehat{\mathcal{A}}x_0$
Set $v_1 = r_0/\left\|r_0\right\|$ and compute $w_1 = \mathcal{H}v_1$
for $k = 1, 2, \ldots$ do
    Perform one step of the simplified Lanczos (Algorithm 2.9)
    Update $QR$ decomposition
    Update solution
    Convergence check
end for
```

Algorithm 2.10: Non-symmetric Lanczos implementation ITFQMR

that CG can be derived from the symmetric Lanczos process (cf. [53]). Let us assume that no breakdowns occur in the non-symmetric Lanczos process. If x_k is taken to be of the form

$$x_k = x_0 + V_k z_k$$

there exist several ways to choose z_k. One choice is to force orthogonality between the residual r_k and the sequence associated with $\mathcal{A}^T$, i.e. $w_1, w_2, \ldots, w_k$ coming from the non-symmetric Lanczos process. In more detail, this results in

$$W_k^T r_k = W_k^T r_0 - W_k^T \mathcal{A} V_k z_k = 0. \tag{2.19}$$

Using the non-symmetric Lanczos process given in (2.14), the biorthogonality of the two generated sequences and the fact that $W_k^T r_0 = \delta_1 \left\|r_0\right\| e_1$ since we set $v_1 = r_0/\left\|r_0\right\|$, (2.19) becomes

$$T_{k,k} z_k = \left\|r_0\right\| e_1. \tag{2.20}$$

In the case of $\mathcal{A}$ being symmetric this would reduce to the CG algorithm (see Section 2.1.1 or [59,56]). For non-symmetric $\mathcal{A}$, an algorithm that calculates z_k in order to satisfy (2.20) is BICG (biconjugate gradient algorithm). Algorithm 2.11 shows an implementation of BICG. In addition to the residual for the forward problem r_k, BICG also uses the adjoint residual $s_k = g - \mathcal{A}^T y_k$.

Whenever the relation $\mathcal{H}\widehat{\mathcal{A}}^T = \mathcal{H}\widehat{\mathcal{A}}$ holds with $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$, the simplification of the non-symmetric Lanczos process that was used for QMR can also be used for BICG. Since the $\mathcal{H}$-symmetric version of BICG will not be discussed in this thesis, we refer the interested reader to [95].

```
for $k = 0, 1, \ldots$ do
    $\alpha_k = \frac{\langle s_k, r_k \rangle}{\langle q_k, \mathcal{A} p_k \rangle}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $y_{k+1} = y_k + \alpha_k q_k$
    $r_{k+1} = r_k - \alpha_k \mathcal{A} p_k$
    $s_{k+1} = s_k - \alpha_k \mathcal{A}^T q_k$
    $\beta_{k+1} = \frac{\langle s_{k+1}, r_{k+1} \rangle}{\langle s_k, r_k \rangle}$
    $p_{k+1} = r_{k+1} + \beta_{k+1} p_k$
    $q_{k+1} = s_{k+1} + \beta_{k+1} q_k$
end for
```

Algorithm 2.11: Biconjugate Gradient Method (BICG)

CHAPTER 3 COMBINATION PRECONDITIONING

In Section 1.2, we discussed that the ability to solve linear systems with short-term recurrence methods is desirable. In the case of the saddle point problem (1.14), we can always apply MINRES since this matrix is symmetric and indefinite but as we pointed out earlier it might be necessary to precondition the linear system (1.1). Some very effective preconditioners might destroy the (symmetric) structure of the original matrix and hence it is no longer possible to apply a short-term recurrence method. But whenever an inner product can be found in which the matrix is symmetric, we can apply $\mathcal{H}$-MINRES and if it is also positive definite, we can apply CG with this inner product. In this chapter, we discuss preconditioners of this type when used for saddle point problems and introduce a technique proposed by Stoll and Wathen in [109]. We give a careful analysis of the self-adjointness relations and then present the technique that allows the combination of different preconditioners and inner products; hence the name *combination preconditioning*.

3.1 Basic properties

We start by reviewing some of the basic mathematics. We consider here only real Euclidean vector spaces; we see no reason that our theory should not apply in the complex case or indeed for other vector spaces, but we have not tried to do so.

We say that

$$\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} \tag{3.1}$$

is a symmetric bilinear form if

- $\langle w, y \rangle = \langle y, w \rangle$ for all $w, y \in \mathbb{R}^n$

- $\langle \alpha w + y, z \rangle = \alpha \langle w, z \rangle + \langle y, z \rangle$ for all $w, y, z \in \mathbb{R}^n$ and all $\alpha \in \mathbb{R}$.

With the addition of a non-degeneracy condition, i.e. $\langle x, y \rangle_{\mathcal{H}} = 0 \forall y \Rightarrow x = 0$, Gohberg *et al.* (cf. [44]) use the term ‘indefinite inner product’; general properties of such forms can also be found here.

If additionally, the positivity conditions

$$\langle w, w \rangle > 0 \text{ for } w \neq 0 \quad \text{ with } \quad \langle w, w \rangle = 0 \text{ if and only if } w = 0$$

are satisfied, then (3.1) defines an inner product on $\mathbb{R}^n$ as mentioned in the Introduction and Chapter 2.

For any real symmetric matrix, $\mathcal{H}$, $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ defined by

$$\langle w, y \rangle_{\mathcal{H}} := w^T \mathcal{H} y \tag{3.2}$$

is easily seen to be a symmetric bilinear form which is an inner product if and only if $\mathcal{H}$ is positive definite.

A matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$ is self-adjoint in $\langle \cdot, \cdot \rangle$ if and only if

$$\langle \mathcal{A} w, y \rangle = \langle w, \mathcal{A} y \rangle \qquad \text{for all } w, y.$$

Self-adjointness of the matrix $\mathcal{A}$ in $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ thus means that

$$w^T \mathcal{A}^T \mathcal{H} y = \langle \mathcal{A} w, y \rangle_{\mathcal{H}} = \langle w, \mathcal{A} y \rangle_{\mathcal{H}} = w^T \mathcal{H} \mathcal{A} y$$

for all w, y so that

$$\mathcal{A}^T\mathcal{H} = \mathcal{H}\mathcal{A}$$

is the basic relation for self-adjointness of $\mathcal{A}$ in $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ as already mentioned in (1.9). Remember, this relation was needed when we introduced iterative solvers in Chapter 2.

Furthermore, we want to describe basic properties of bilinear forms and non-standard inner products. This can also be viewed in terms of real symmetric matrices since Equation 1.9 states that $\mathcal{A}^T\mathcal{H}$ is a real symmetric matrix. Here, we prefer the language of inner products since we feel it indicates more of the mathematical structure which leads to the development of new methods based on non-standard inner products.

We emphasize that $\langle \cdot, \cdot \rangle$, $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ must be symmetric bilinear forms here, but we do not require them to be inner products for the theory presented in this section. For practical reasons, we will consider positivity/non-positivity of symmetric bilinear forms and positive definiteness/indefiniteness of self-adjoint matrices separately from our considerations of symmetry and self-adjointness. Whenever we write $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, $\mathcal{H}$ will be symmetric.

Lemma 3.1. *If $\mathcal{A}_1$ and $\mathcal{A}_2$ are self-adjoint in $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ then for any $\alpha, \beta \in \mathbb{R}$, $\alpha\mathcal{A}_1 + \beta\mathcal{A}_2$ is self-adjoint in $\langle \cdot, \cdot \rangle_{\mathcal{H}}$.*

Proof. Using the self-adjointness of $\mathcal{A}_1$ and $\mathcal{A}_2$ we get that

$$\begin{aligned} \langle (\alpha\mathcal{A}_1 + \beta\mathcal{A}_2)x, x \rangle_h &= \alpha\langle \mathcal{A}_1 x, x \rangle_{\mathcal{H}} + \beta\langle \mathcal{A}_2 x, x \rangle_h \\ &= \alpha\langle x, \mathcal{A}_1 x \rangle_{\mathcal{H}} + \beta\langle x, \mathcal{A}_2 x \rangle_h \\ &= \quad \langle x, (\alpha\mathcal{A}_1 + \beta\mathcal{A}_2)x \rangle_h. \end{aligned}$$

□

Also

Lemma 3.2. *If $\mathcal{A}$ is self-adjoint in $\langle \cdot, \cdot \rangle_{\mathcal{H}_1}$ and in $\langle \cdot, \cdot \rangle_{\mathcal{H}_2}$ then $\mathcal{A}$ is self-adjoint in $\langle \cdot, \cdot \rangle_{\alpha\mathcal{H}_1 + \beta\mathcal{H}_2}$ for every $\alpha, \beta \in \mathbb{R}$.*

Proof. Using the definition of $\langle .,.\rangle_{\mathcal{H}_1}$ and $\langle .,.\rangle_{\mathcal{H}_1}$ as well as the self-adjointness of $\mathcal{A}$ in these two bilinear forms we get

$$\begin{aligned}\langle \mathcal{A}x, x\rangle_{\alpha\mathcal{H}_1+\beta\mathcal{H}_2} &= \langle \alpha\mathcal{H}_1\mathcal{A}+\beta\mathcal{H}_2\mathcal{A}x, x\rangle \\ &= \alpha\langle \mathcal{H}_1\mathcal{A}x, x\rangle + \beta\langle \mathcal{H}_2\mathcal{A}x, x\rangle \\ &= \alpha\langle x, \mathcal{A}x\rangle_{\mathcal{H}_1} + \beta\langle x, \mathcal{A}x\rangle_{\mathcal{H}_2} \\ &= \alpha\langle \mathcal{H}_1x, \mathcal{A}x\rangle + \beta\langle \mathcal{H}_2x, \mathcal{A}x\rangle \\ &= \langle \alpha\mathcal{H}_1x+\beta\mathcal{H}_2x, \mathcal{A}x\rangle \\ &= \langle x, \mathcal{A}x\rangle_{\alpha\mathcal{H}_1+\beta\mathcal{H}_2}\end{aligned}$$

□

Now if $\mathcal{A}$ is preconditioned on the left by $\mathcal{P}$, then from (1.9), $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ if and only if

$$(\mathcal{P}^{-1}\mathcal{A})^T\mathcal{H} = \mathcal{H}\mathcal{P}^{-1}\mathcal{A} \tag{3.3}$$

which is

$$\mathcal{A}^T\mathcal{P}^{-T}\mathcal{H} = \mathcal{H}\mathcal{P}^{-1}\mathcal{A}$$

or

$$\mathcal{A}^T(\mathcal{P}^{-T}\mathcal{H}) = (\mathcal{P}^{-T}\mathcal{H})^T\mathcal{A}$$

since $\mathcal{H}$ is symmetric. Thus if $\mathcal{A}$ is also symmetric we get

$$(\mathcal{P}^{-T}\mathcal{H})^T\mathcal{A} = \mathcal{A}(\mathcal{P}^{-T}\mathcal{H}) \tag{3.4}$$

and so

Lemma 3.3. *For symmetric $\mathcal{A}$, $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ if and only if $\mathcal{P}^{-T}\mathcal{H}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{A}}$.*

Proof. Follows directly from the above and (1.9). □

Remark 3.4. *Lemma 3.3 includes the even more simple situations that $\mathcal{P}^{-1}\mathcal{A}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{P}}$ and $\mathcal{A}\mathcal{P}^{-1}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{A}^{-1}}$ when both $\mathcal{A}$ and $\mathcal{P}$ are symmetric since I is trivially self-adjoint in any symmetric bilinear form. Clearly invertibility of $\mathcal{P}$ and $\mathcal{A}$ respectively are needed in these two cases.*

Now for symmetric $\mathcal{A}$, if $\mathcal{P}_1$ and $\mathcal{P}_2$ are such that $\mathcal{P}_i^{-1}\mathcal{A}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}_i}$, $i = 1, 2$ for symmetric matrices $\mathcal{H}_1$, $\mathcal{H}_2$, then

$$(\mathcal{P}_1^{-1}\mathcal{A})^T\mathcal{H}_1 = \mathcal{H}_1(\mathcal{P}_1^{-1}\mathcal{A}) \quad \text{and} \quad (\mathcal{P}_2^{-1}\mathcal{A})^T\mathcal{H}_2 = \mathcal{H}_2(\mathcal{P}_2^{-1}\mathcal{A}). \tag{3.5}$$

Using Lemma 3.3, $\mathcal{P}_i^{-T}\mathcal{H}_i$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{A}}$ for $i = 1, 2$ and thus by Lemma 3.1

$$\alpha\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{P}_2^{-T}\mathcal{H}_2$$

is also self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{A}}$ for any $\alpha, \beta \in \mathbb{R}$. Now, if for some α, β we are able to decompose the matrix $(\alpha\mathcal{P}_1^{-T}\mathcal{H}_1+\beta\mathcal{P}_2^{-T}\mathcal{H}_2) = \mathcal{P}_3^{-T}\mathcal{H}_3$ for some symmetric matrix $\mathcal{H}_3$, then $\mathcal{P}_3^{-T}\mathcal{H}_3$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{A}}$ and a further application of Lemma 3.3 yields that $\mathcal{P}_3^{-1}\mathcal{A}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}_3}$. We have proved

Theorem 3.5. *If $\mathcal{P}_1$ and $\mathcal{P}_2$ are left preconditioners for the symmetric matrix $\mathcal{A}$ for which symmetric matrices $\mathcal{H}_1$ and $\mathcal{H}_2$ exist with $\mathcal{P}_1^{-1}\mathcal{A}$ self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}_1}$ and $\mathcal{P}_2^{-1}\mathcal{A}$ self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}_2}$ and if*

$$\alpha\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{P}_2^{-T}\mathcal{H}_2 = \mathcal{P}_3^{-T}\mathcal{H}_3$$

for some matrix $\mathcal{P}_3$ and some symmetric matrix $\mathcal{H}_3$ then $\mathcal{P}_3^{-1}\mathcal{A}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}_3}$.

We want to emphasize that Theorem 3.5 shows a possible way to generate new preconditioners for $\mathcal{A}$. In Section 3.5 we show practical examples of its use.

The construction of $\mathcal{P}_3$, $\mathcal{H}_3$ in Theorem 3.5 also allows straightforward inheritance of positive definiteness — for this to be a useful property it is essential that $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ defines an inner product, i.e. that $\mathcal{H}$ is positive definite. It is trivial to construct examples of indefinite diagonal matrices $\mathcal{A}$ and $\mathcal{H}$ for which $\langle\mathcal{A}w,w\rangle_{\mathcal{H}} > 0$ for all non-zero w, but in order to be able to take advantage of positive definiteness, for example by employing Conjugate Gradients, it is important that $\langle w,w\rangle_{\mathcal{H}} = w^T\mathcal{H}w > 0$ for all non-zero w.

Lemma 3.6. *If the conditions of Theorem 3.5 are satisfied and additionally if $\mathcal{P}_i^{-1}\mathcal{A}$ is positive definite in $\langle\cdot,\cdot\rangle_{\mathcal{H}_i}$, $i = 1,2$ then $\mathcal{P}_3^{-1}\mathcal{A}$ is positive definite in $\langle\cdot,\cdot\rangle_{\mathcal{H}_3}$ at least for positive values of α and β.*

Proof. Positive definiteness of $\mathcal{P}^{-1}\mathcal{A}$ in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ means that

$$\langle\mathcal{P}^{-1}\mathcal{A}w,w\rangle_{\mathcal{H}} > 0, \quad \text{for } \ w \neq 0$$

i.e. that $w^T\mathcal{A}\mathcal{P}^{-T}\mathcal{H}w > 0$ so that $\mathcal{A}\mathcal{P}^{-T}\mathcal{H}$ is a symmetric matrix with all eigenvalues positive. Thus $\mathcal{A}\mathcal{P}_1^{-T}\mathcal{H}_1$ and $\mathcal{A}\mathcal{P}_2^{-T}\mathcal{H}_2$ are symmetric and positive definite and it follows that

$$\alpha\mathcal{A}\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{A}\mathcal{P}_2^{-T}\mathcal{H}_2 = \mathcal{A}\mathcal{P}_3^{-T}\mathcal{H}_3$$

must also be symmetric and positive definite at least for positive values of α and β. □

We note that there will in general be some negative values of α or β for which $\mathcal{P}_3^{-1}\mathcal{A}$ remains positive definite, but at least one of α and β needs to be positive in this case. The precise limits on the values that α and β can take whilst positive definiteness is preserved depend on the extreme eigenvalues of $\mathcal{A}\mathcal{P}_1^{-T}\mathcal{H}_1$ and $\mathcal{A}\mathcal{P}_2^{-T}\mathcal{H}_2$. Unfortunately, even if $\mathcal{H}_1$ and $\mathcal{H}_2$ are positive definite, there is no guarantee that $\mathcal{H}_3$ will be also.

We can also consider right preconditioning: if $\widehat{\mathcal{A}} = \mathcal{A}\mathcal{P}^{-1}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ then

$$(\mathcal{A}\mathcal{P}^{-1})^T\mathcal{H} = \mathcal{H}(\mathcal{A}\mathcal{P}^{-1})$$

or equivalently

$$\mathcal{P}^{-T}\mathcal{A}^T\mathcal{H} = \mathcal{H}\mathcal{A}\mathcal{P}^{-1} \text{ which is } (\mathcal{P}^{-1})^T(\mathcal{A}^T\mathcal{H}) = (\mathcal{A}^T\mathcal{H})^T\mathcal{P}^{-1}. \tag{3.6}$$

Thus

Lemma 3.7. *If the right preconditioner $\mathcal{P}$ is symmetric and $\widehat{\mathcal{A}} = \mathcal{A}\mathcal{P}^{-1}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ for some symmetric matrix $\mathcal{H}$, then $\mathcal{A}^T\mathcal{H}$ is self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{P}^{-1}}$.*

Lemma 3.7 shows that we could combine problem matrices and symmetric bilinear forms for the same preconditioner. This is obviously more a theoretical than a practical result compared to obtaining new preconditioners for a given problem as in the case of left preconditioning above.

One of the decompositions as $\mathcal{P}_3^{-T}\mathcal{H}_3$ introduced in Section 3.5 will provide not only a symmetric inner product matrix but also a symmetric preconditioner and therefore fulfills the conditions of Lemma 3.7.

We now want to discuss very briefly the eigenvalues of matrices which are self-adjoint according to our definition which allows indefinite symmetric bilinear forms. Assume that $\mathcal{A}^T\mathcal{H} = \mathcal{H}\mathcal{A}$ holds and that (λ, v) is a given eigenpair of $\mathcal{A}$. Thus,

$$\mathcal{A}v = \lambda v, \qquad v \neq 0. \tag{3.7}$$

Multiplying (3.7) from the left by $v^*\mathcal{H}$ where v^* is the conjugate transpose of v gives

$$v^*\mathcal{H}\mathcal{A}v = \lambda v^*\mathcal{H}v. \tag{3.8}$$

Notice that the left hand side of (3.8) is real since $\mathcal{H}\mathcal{A}$ is real symmetric. On the right-hand side, $v^*\mathcal{H}v$ is also real since $\mathcal{H}$ is real symmetric; therefore the eigenvalue must be real unless $v^*\mathcal{H}v = 0$. A matrix $\mathcal{H}$ always exists such that $\mathcal{A}^T\mathcal{H} = \mathcal{H}\mathcal{A}$ since any matrix is similar to its transpose (see for example Section 3.2.3 in [62]). The interesting case when $v^*\mathcal{H}v = 0$ is discussed in [12].

Note that the above arguments establish that there is no inner product in which $\mathcal{A}$ is self-adjoint unless $\mathcal{A}$ has real eigenvalues.

It is also known that for a real diagonalizable matrix $\mathcal{A}$ which has only real eigenvalues, inner products always exist in which $\mathcal{A}$ is self-adjoint.

Lemma 3.8. *If $\mathcal{A} = R^{-1}\Lambda R$ is a diagonalization of $\mathcal{A}$ with the diagonal matrix Λ of eigenvalues being real, then $\mathcal{A}$ is self-adjoint in $\langle \cdot, \cdot \rangle_{R^T\Theta R}$ for any real diagonal matrix Θ.*

Proof. The conditions (1.9) for self-adjointness of $\mathcal{A}$ in $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ are

$$R^T \Lambda R^{-T} \mathcal{H} = \mathcal{H} R^{-1} \Lambda R$$

which are clearly satisfied for $\mathcal{H} = R^T \Theta R$ whenever Θ is diagonal because then Θ and Λ commute. Clearly $\mathcal{H}$ is positive definite whenever the diagonal entries of Θ are positive. □

This result is not of great use in practice since knowledge of the complete eigensystem of $\mathcal{A}$ is somewhat prohibitive.

3.2 An example and implementation details

In 1988, Bramble and Pasciak [10] introduced a block triangular preconditioner for the discrete Stokes problem (1.14), i.e.

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}$$

where we assume for the remainder of this chapter that $A \in \mathbb{R}^{n,n}$ is symmetric and positive definite, B has full rank and $C \in \mathbb{R}^{m,m}$ is symmetric and positive semi-definite. This had the almost-magical effect of turning the original indefinite symmetric matrix problem into a non-symmetric matrix which is both self-adjoint and, in certain practical circumstances, positive definite in a non-standard inner product; thus the conjugate gradient method could be used in the non-standard inner product. To be precise, the symmetric saddle point problem (1.14) if preconditioned on the left by

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ B & -I \end{bmatrix} \text{ with } \mathcal{P}^{-1} = \begin{bmatrix} A_0^{-1} & 0 \\ BA_0^{-1} & -I \end{bmatrix} \tag{3.9}$$

results in the non-symmetric matrix

$$\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A} = \begin{bmatrix} A_0^{-1}A & A_0^{-1}B^T \\ BA_0^{-1}A - B & BA_0^{-1}B^T + C \end{bmatrix} \tag{3.10}$$

which turns out to be self-adjoint in the symmetric bilinear form defined by

$$\mathcal{H} = \begin{bmatrix} A - A_0 & 0 \\ 0 & I \end{bmatrix}. \tag{3.11}$$

If the block $A - A_0$ is positive definite, $\mathcal{H}$ obviously defines an inner product and we write $A - A_0 > 0$ or $A > A_0$. This means that the eigenvalues of $A - A_0$ are all positive. The positivity can be achieved by scaling the matrix A_0 appropriately. In more detail by computing the minimal eigenvalue of the matrix $A_0^{-1}A$ or an estimate to it, the matrix A_0 can be scaled in order to guarantee the definiteness of $A - A_0$. We will now show that under certain conditions on the preconditioner $\widehat{\mathcal{A}}$ is positive definite in the inner product defined by $\mathcal{H}$, i.e. $\langle \widehat{\mathcal{A}}w, w\rangle_{\mathcal{H}} > 0$ for all $w \neq 0$.

First, by using the definition of the inner product induced by $\mathcal{H}$ we notice, that the condition $\langle \widehat{\mathcal{A}}w, w\rangle_{\mathcal{H}} > 0$ is equivalent to $\langle \mathcal{H}\widehat{\mathcal{A}}w, w\rangle > 0$, which tells us that the matrix $\mathcal{H}\widehat{\mathcal{A}}$ has to be positive definite. To show that $\mathcal{H}\widehat{\mathcal{A}}$ is positive definite, we introduce a splitting of $\mathcal{H}\widehat{\mathcal{A}}$ also used by Klawonn in [65] where the matrix

$$\mathcal{H}\widehat{\mathcal{A}} = \begin{bmatrix} AA_0^{-1}A - A & AA_0^{-1}B^T - B^T \\ BA_0^{-1}A - B & BA_0^{-1}B^T + C \end{bmatrix} = \widehat{\mathcal{A}}^T\mathcal{H} \tag{3.12}$$

can be factorized as

$$\begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} AA_0^{-1}A - A & 0 \\ 0 & BA^{-1}B^T + C \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix} \tag{3.13}$$

which is a congruence transformation. Now using Sylvester's law of inertia [53] we know that the number of positive, negative and zero eigenvalues is determined by the eigenvalues of the diagonal blocks of

$$\begin{bmatrix} AA_0^{-1}A - A & 0 \\ 0 & BA^{-1}B^T + C \end{bmatrix}.$$

The Schur complement block $BA^{-1}B^T + C$ is obviously positive definite under the assumptions made earlier that A is positive definite and C positive semi-definite. The block $AA_0^{-1}A - A$ can be rewritten as

$$A(A_0^{-1} - A^{-1})A$$

which will be positive definite if $A_0^{-1} - A^{-1}$ is positive definite or equivalently

$$y^T A_0 y < y^T A y. \tag{3.14}$$

Note, the condition (3.14) is precisely that required for $\mathcal{H}$ to be positive definite in this case, which is needed if methods such as CG or $\mathcal{H}$-MINRES should be applied. Since both $BA^{-1}B^T + C$ and $AA_0^{-1}A - A$ are positive definite, Sylvester's Law of inertia applied to (3.13) guarantees that $\mathcal{H}\widehat{\mathcal{A}}$ is positive definite, i.e. that $\widehat{\mathcal{A}}$ is self-adjoint and positive definite in $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. Hence, the applicability of CG is guaranteed. If one is not willing to pay the price of the (sometimes) costly eigenvalue analysis to guarantee the definiteness of $\mathcal{H}$, it is always possible to employ ITFQMR, which only needs the self-adjointness in the bilinear form or inner product $\mathcal{H}$. This is also recommended in [105].

In Section 2.1.1, we introduced the CG method with non-standard inner product (see Algorithm 2.5). Here, we want to show that the Bramble-Pasciak CG can also be viewed as the Preconditioned Conjugate Gradient method (PCG) [56,24] applied to the matrix $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$. In more detail, solving system (1.14) is equivalent to solving the system

$$\mathcal{H}\mathcal{P}^{-1}\mathcal{A}x = \mathcal{H}\mathcal{P}^{-1}b. \tag{3.15}$$

The sequence of approximations $\{x_k\}$ generated by the Bramble-Pasciak CG method satisfies $x_k \in span\left\{\mathcal{P}^{-1}\mathcal{A}r_0, \ldots, (\mathcal{P}^{-1}\mathcal{A})^{k-1}r_0\right\}$. This again emphasizes the fact that $\mathcal{P}$ needs to be chosen to cluster the eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ since the inner product $\mathcal{H}$ does not effect the Krylov subspace. Applying the (unpreconditioned) conjugate gradient method to solve the linear system with $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$ will result in $x_k \in span\left\{\mathcal{H}\mathcal{P}^{-1}\mathcal{A}r_0, \ldots, (\mathcal{H}\mathcal{P}^{-1}\mathcal{A})^{k-1}r_0\right\}$. Thus, the Krylov subspaces will be different and a different sequence of iterates will be formed. Suppose we apply the preconditioned CG method with a symmetric and positive definite preconditioner L to solve (3.15). Using the classical PCG implementation given in [56, 24] we obtain Algorithm 3.1. Obviously, both algorithms are identical when $L = \mathcal{H}$.

We now want to examine whether we should work with Algorithm 2.5 and hence with the matrix $\mathcal{P}^{-1}\mathcal{H}$ or Algorithm 3.1 and hence with the matrix $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$ for computational purposes. The preconditioner $\mathcal{P}$ was constructed to alter the spectrum of the preconditioned matrix $\mathcal{A}$ such that good convergence can be achieved for $\widehat{\mathcal{A}}$. On the other hand, if we premultiply $\widehat{\mathcal{A}}$ by $\mathcal{H}$, a further convergence enhancement cannot necessarily be expected. Moreover,

```
Given $x_0 = 0$, set $z_0 = L^{-1}\mathcal{H}\mathcal{P}^{-1}(b - \mathcal{A}x_0)$ and $p_0 = z_0$
for $k = 0, 1, \ldots$ do
    $\alpha = \frac{\langle z_k, Lz_k\rangle}{\langle p_k, \mathcal{H}\mathcal{P}^{-1}\mathcal{A}p_k\rangle}$
    $x_{k+1} = x_k + \alpha p_k$
    $z_{k+1} = z_k - \alpha L^{-1}\mathcal{H}\mathcal{P}^{-1}\mathcal{A}p_k$
    $\beta = \frac{\langle z_{k+1}, Lz_{k+1}\rangle}{\langle z_k, Lz_k\rangle}$
    $p_{k+1} = z_{k+1} + \beta p_k$
end for
```

Algorithm 3.1: PCG for solving $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}x = \mathcal{H}\mathcal{P}^{-1}b$ with preconditioner L

we expect the convergence with $\mathcal{H}\widehat{\mathcal{A}}$ to be poorer since the premultiplication by $\mathcal{H}$ will destroy the eigenvalue structure achieved by applying the preconditioner $\mathcal{P}$.

An alternative would be to use $\mathcal{H}$ as a preconditioner for $\mathcal{H}\widehat{\mathcal{A}}$, as explained above, which would result in the eigenstructure of the preconditioned matrix $\widehat{\mathcal{A}}$. In Figure 3.1, we plot the convergence history of the Bramble-Pasciak CG for (1.14) and the classical PCG without preconditioning and with preconditioning for (3.15) when applied to a Stokes problem of dimension 59 that was generated by IFISS [23]. Note that the matrix $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$ is explicitly formed for the small example used here. This is prohibitive for practical setups.

As predicted, the unpreconditioned CG method for $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$ is outperformed by the Bramble-Pasciak CG method. When the preconditioner $L = \mathcal{H}$ is used within PCG, the convergence curves are almost identical: the slight deviation in Figure 3.1 between the two dashed lines is due to round-off error.

Another issue that arises when implementing the Bramble-Pasciak CG method is whether multigrid preconditioners can be used. These preconditioners are never explicitly available, which means that we are only equipped with a function that represents the multiplication with the inverse. Preconditioners of this type are very often used in practice, e.g. when solving the Stokes problem (1.13). The action of multiplying with the inner product matrix $\mathcal{H}$ can still be implemented for that case and we will explain this procedure now in some detail. This is already mentioned in the original work by Bramble and Pasciak [10].

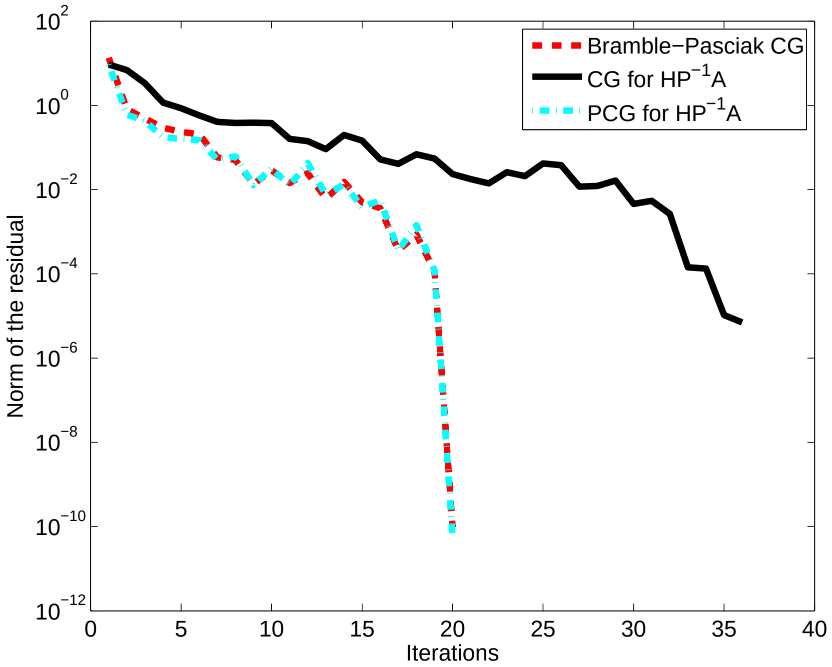

Figure 3.1: CG for $\mathcal{H}\widehat{\mathcal{A}}$ and the Bramble-Pasciak CG for a Stokes problem generated with IFISS [23] of dimension 59.

Let us assume that the preconditioner A_0 is not explicitly given and only the action of A_0^{-1} is available as a procedure. To compute the paremeters α and β in Algorithm 2.4, inner products with $\mathcal{H}$ have to be evaluated. In more detail, evaluating $\langle \mathcal{P}^{-1}\mathcal{A}r_{k+1}, p_k\rangle_{\mathcal{H}}$ reduces to expressing $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}r_{k+1}$ in full expanded form without using A_0. Hence, by introducing $\mathcal{A}r_{k+1} = \left[(\hat{r}_{k+1}^{(1)})^T(\hat{r}_{k+1}^{(2)})^T\right]^T$, where the blockdimensions of $\hat{r}_{k+1}$ correspond to the blockdimensions of the saddle point matrix (1.14), we get

$$\begin{aligned}\mathcal{H}\mathcal{P}^{-1}\mathcal{A}r_{k+1} &= \begin{bmatrix} A-A_0 & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} A_0^{-1}\hat{r}_{k+1}^{(1)} \\ BA_0^{-1}\hat{r}_{k+1}^{(1)} - \hat{r}_{k+1}^{(2)} \end{bmatrix} \\ &= \begin{bmatrix} AA_0^{-1}\hat{r}_{k+1}^{(1)} - \hat{r}_{k+1}^{(1)} \\ BA_0^{-1}\hat{r}_{k+1}^{(1)} - \hat{r}_{k+1}^{(2)} \end{bmatrix}.\end{aligned}$$

Note that for the last expression there is no need for the matrix A_0; only the application of its inverse A_0^{-1} is used. The same can be done whenever $\langle \mathcal{P}^{-1}\mathcal{A}p_k, p_k\rangle_{\mathcal{H}}$ has to be evaluated. Note that $\mathcal{A}p_k$ does not need to be evaluated explicitly since the relation $\mathcal{A}p_k = \mathcal{A}r_k + \beta\mathcal{A}p_{k-1}$ holds. The evaluation of $\langle r_k, p_k\rangle_{\mathcal{H}}$ can be similarly simplified by exploiting $r_k = \mathcal{P}^{-1}\tilde{r}_k$ where $\tilde{r}_k$ is the unpreconditioned residual. Finally, using the factorization $\tilde{r}_k = \left[(\tilde{r}_{k+1}^{(1)})^T(\tilde{r}_{k+1}^{(2)})^T\right]^T$, which is again split according to the dimensions of the saddle point matrix, we obtain

$$\mathcal{H}\mathcal{P}^{-1}\tilde{r}_k = \begin{bmatrix} A-A_0 & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} A_0^{-1}\tilde{r}_k^{(1)} \\ BA_0^{-1}\tilde{r}_k^{(1)} - \tilde{r}_k^{(2)} \end{bmatrix} = \begin{bmatrix} AA_0^{-1}\tilde{r}_{k+1}^{(1)} - \tilde{r}_k^{(1)} \\ BA_0^{-1}\tilde{r}_{k+1}^{(1)} - \tilde{r}_k^{(2)} \end{bmatrix}.$$

This shows that multigrid preconditioners can be used very efficiently within the Bramble-Pasciak CG.

3.3 More Saddle point examples

The first example was given by Bramble and Pasciak, and a straightforward extension of this method can be made by introducing a Schur-complement type preconditioner S_0, i.e. S_0 approximates the Schur-complement $C + BA^{-1}B^T$. This was done in [78, 65, 105]. The result of putting a Schur complement preconditioner S_0 into $\mathcal{P}$ is given by

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ B & -S_0 \end{bmatrix} \quad \text{and} \quad \mathcal{P}^{-1} = \begin{bmatrix} A_0^{-1} & 0 \\ S_0^{-1}BA_0^{-1} & -S_0^{-1} \end{bmatrix}; \tag{3.16}$$

under certain conditions positive definiteness of the preconditioned saddle-point system can still be guaranteed in a non-standard inner product similar to (3.11), i.e.

$$\mathcal{H} = \begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix}. \tag{3.17}$$

With this setup, we look at the matrix

$$\mathcal{H}\widehat{\mathcal{A}} = \begin{bmatrix} AA_0^{-1}A - A & AA_0^{-1}B^T - B^T \\ BA_0^{-1}A - B & BA_0^{-1}B^T + C \end{bmatrix},$$

which is symmetric and under certain conditions positive definite. Note that this is the same matrix as the one obtained for the Bramble-Pasciak case where no Schur complement preconditioner was used (cf. (3.12)). Therefore, the conditions imposed on A_0 are the same and we only need the positivity of S_0 to guarantee that $\mathcal{H}$ defines an inner product. Note that the Bramble-Pasciak method with Schur-complement preconditioner can still be used when S_0 is not explicitly given. We only show this for the inner product $\langle \mathcal{P}^{-1}\mathcal{A}r_{k+1}, p_k \rangle_{\mathcal{H}}$. As seen before, we have to evaluate $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}r_{k+1}$ without using A_0, which by introducing $\mathcal{A}r_{k+1} = \left[(\hat{r}_{k+1}^{(1)})^T(\hat{r}_{k+1}^{(2)})^T\right]^T$ gives

$$\begin{aligned}\mathcal{H}\mathcal{P}^{-1}\mathcal{A}r_{k+1} &= \begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix}\begin{bmatrix} A_0^{-1}\hat{r}_{k+1}^{(1)} \\ S_0^{-1}(BA_0^{-1}\hat{r}_{k+1}^{(1)} - \hat{r}_{k+1}^{(2)}) \end{bmatrix} \\ &= \begin{bmatrix} AA_0^{-1}\hat{r}_{k+1}^{(1)} - \hat{r}_{k+1}^{(1)} \\ BA_0^{-1}\hat{r}_{k+1}^{(1)} - \hat{r}_{k+1}^{(2)} \end{bmatrix}.\end{aligned}$$

Note, there is no need for the Schur-complement preconditioner S_0 at all. In an analogous way, we can analyze $\langle \mathcal{P}^{-1}\mathcal{A}p_k, p_k\rangle_{\mathcal{H}}$. Finally, by using the unpreconditioned residual, it can be shown that the Schur-complement preconditioner is not used for the evaluation of the inner product $\langle r_k, p_k\rangle_{\mathcal{H}}$.

A similar form to (3.17) was provided by Zulehner in 2002 (see [117]). Zulehner considered a preconditioner of the form (3.16) for an inexact Uzawa method which under certain conditions can admit the usability of a CG acceleration (see [76] for the connection of CG and the inexact Uzawa algorithm as a Richardson iteration method).

In 2006 Benzi and Simoncini gave a further example for the system (1.14) with $C = 0$ (see [7]) which is an extension of earlier work by Fischer *et al.* (cf. [30]). Namely,

$$\mathcal{P} = \mathcal{P}^{-1} = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \tag{3.18}$$

and

$$\mathcal{H} = \begin{bmatrix} A - \gamma I & B^T \\ B & \gamma I \end{bmatrix}. \tag{3.19}$$

The parameter γ depends on the eigenvalues of the block A and the Schur-complement $BA_0^{-1}B^T$.

Recently, Liesen and Parlett made an extension to this result taking a non-zero matrix C in (1.14) into account (see [69, 70]). In the language used here, the preconditioner is again

$$\mathcal{P} = \mathcal{P}^{-1} = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \tag{3.20}$$

but the symmetric bilinear form is now defined by

$$\mathcal{H} = \begin{bmatrix} A - \gamma I & B^T \\ B & \gamma I - C \end{bmatrix}. \tag{3.21}$$

There are certain conditions which must be satisfied by the parameter γ in order to guarantee positive definiteness of $\mathcal{H}$ so that CG in the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ can be reliably employed (see [7], [69, 70]).

Liesen and Parlett also show in [70] that the matrix $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$ is self-adjoint in every bilinear form of the type $\mathcal{H}p(\widehat{\mathcal{A}})$ where $\mathcal{H} = \mathcal{P}$ and $p(\widehat{\mathcal{A}})$ is any real polynomial in $\widehat{\mathcal{A}}$. The proof is based on a technique introduced by Freund (cf. [33]) where the matrix $\mathcal{H}$ can be shifted from one side of the polynomial $p(\mathcal{A})$ to the other side by successively using $\widehat{\mathcal{A}}^T\mathcal{H} = \mathcal{H}\widehat{\mathcal{A}}$, see also Section 2.2.2. Trivially, this observation holds for any real symmetric $\mathcal{H}$ whenever the condition $\widehat{\mathcal{A}}^T\mathcal{H} = \mathcal{H}\widehat{\mathcal{A}}$ is satisfied and not just for the matrix $\mathcal{H} = \mathcal{P}$. Through the choice of the polynomial p the approach presented by Liesen and Parlett provides a whole set of interesting bilinear forms that may give useful examples.

Another example was given in Zulehner [117] in the context of inexact Uzawa methods and in [102] by Schöberl and Zulehner where the saddle point problem with $C = 0$ is preconditioned by

$$\mathcal{P} = \begin{bmatrix} A_0 & B^T \\ B & BA_0^{-1}B^T - \hat{S} \end{bmatrix}$$

with A_0 and $\hat{S}$ being symmetric and positive definite. A preconditioner of this form is typically called *constraint preconditioner* [64] due to the presence of the blocks B and B^T in the preconditioner, which represent the constrains in problems coming from optimization. Then the preconditioned matrix is self-adjoint in the bilinear form defined by

$$\mathcal{H} = \begin{bmatrix} A_0 - A & 0 \\ 0 & BA_0^{-1}B^T - \hat{S} \end{bmatrix}.$$

The definiteness of $\mathcal{H}$ as well as the definiteness of $\widehat{\mathcal{A}}$ in the bilinear form defined by $\mathcal{H}$ depends again on the eigenvalues of the block A and the eigenvalues of the Schur-complement $BA_0^{-1}B^T$.

Another example using a constraint preconditioner was given by Dohrmann and Lehoucq in [17]. They consider the general saddle point problem given in (1.14) with the constraint preconditioner

$$\mathcal{P} = \begin{bmatrix} \hat{S}_A & B^T \\ B & \hat{C} \end{bmatrix}$$

where $\hat{S}_A$ is an approximation to a penalized primal Schur complement $S_A = A + B^T\hat{C}^{-1}B$ and $\hat{C}$ is symmetric and positive definite. The bilinear form in which the preconditioned matrix is self-adjoint in this case is given by

$$\mathcal{H} = \begin{bmatrix} S_A - \hat{S}_A & 0 \\ 0 & \hat{C} - C \end{bmatrix}.$$

Again, the obvious conditions are $S_A > \hat{S}_A$ and $\hat{C} - C > 0$ for $\mathcal{H}$ to define an inner product.

In this section we presented a number of examples where the preconditioned saddle point matrix $\widehat{\mathcal{A}}$ is self-adjoint in a non-standard inner product.

Hence, these examples represent potential candidates for combination preconditioning and we will show results in Section 3.5.

3.4 A modified Bramble-Pasciak preconditioner

The original Bramble-Pasciak CG method requires that the matrix

$$\mathcal{H} = \begin{bmatrix} A - A_0 & 0 \\ 0 & I \end{bmatrix}$$

is positive definite. The obvious drawback of this method is the necessity to scale the matrix A_0 such that $A - A_0$ is positive definite. Usually an eigenvalue problem for $A_0^{-1}A$, or at least an eigenvalue estimation problem has to be solved which can be costly (see [108] for a survey of methods that could be applied).

By contrast, we introduce the Bramble-Pasciak$^+$ preconditioner

$$\mathcal{P}^+ = \begin{bmatrix} A_0 & 0 \\ -B & S_0 \end{bmatrix} \quad \text{and} \quad \left(\mathcal{P}^+\right)^{-1} = \begin{bmatrix} A_0^{-1} & 0 \\ S_0^{-1}BA_0^{-1} & S_0^{-1} \end{bmatrix} \tag{3.22}$$

and obtain by left preconditioning with $\mathcal{P}^+$

$$\widehat{\mathcal{A}} = \left(\mathcal{P}^+\right)^{-1}\mathcal{A} = \begin{bmatrix} A_0^{-1}A & A_0^{-1}B^T \\ S_0^{-1}BA_0^{-1}A + S_0^{-1}B & S_0^{-1}BA_0^{-1}B^T - S_0^{-1}C \end{bmatrix}. \tag{3.23}$$

Simple algebra shows that $\widehat{\mathcal{A}}$ is self-adjoint in the inner product induced by

$$\mathcal{H}^+ = \begin{bmatrix} A + A_0 & 0 \\ 0 & S_0 \end{bmatrix} \tag{3.24}$$

where S_0 is a symmetric and positive definite Schur-complement preconditioner. An inner product of similar form to (3.24) was used by Zulehner in [117] in the analysis of inexact Uzawa methods. Note that for a positive definite preconditioner A_0, the matrix $\mathcal{H}^+$ is always positive definite due to the positive definiteness of the matrices A, A_0 and S_0. Thus, we are always equipped in this case with an inner product and not just a symmetric bilinear form whatever symmetric and positive definite A_0 is chosen, and so the appropriate Krylov subspace method can be used in this inner product.

The definiteness of $\mathcal{H}$ and the preconditioned matrix in the new inner product has to be shown for the new preconditioner $\mathcal{P}^+$ in order to use CG. Using the approach presented in Section 3.2, we find a splitting of

$$\widehat{\mathcal{A}}^T\mathcal{H}^+ = \begin{bmatrix} AA_0^{-1}A + A & AA_0^{-1}B^T + B^T \\ BA_0^{-1}A + B & BA_0^{-1}B^T - C \end{bmatrix} \tag{3.25}$$

as

$$\begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} AA_0^{-1}A + A & 0 \\ 0 & -BA^{-1}B^T - C \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix}. \tag{3.26}$$

By Sylvester's law of inertia this shows that $\widehat{\mathcal{A}}^T\mathcal{H}^+$ is indefinite since $-BA^{-1}B^T - C$ is always negative definite and $AA_0^{-1}A + A$ is positive definite. Therefore, the reliable applicability of the CG method cannot be guaranteed.

We want to mention that the Bramble-Pasciak$^+$ preconditioner can also be interpreted as the classical Bramble-Pasciak preconditioner applied to the matrix $\mathcal{J}\mathcal{A}$ where $\mathcal{J} = diag(I_n, -I_m)$ with I_j the identity of dimension $j = m, n$.

Different methods can be employed for solving the $\mathcal{P}^+$-preconditioned system. Since $\mathcal{H}^+$ defines an inner product, the $\mathcal{H}$-MINRES method given in Section 2.1.2 should be used. As was shown in (3.26), CG with $\mathcal{H}$-inner product cannot be reliably used, but applying Algorithm 2.5 quite often gives good results. For the applicability of ITFQMR not even the definiteness of $\mathcal{H}^+$ is needed (see Algorithm 2.10).

For a better understanding of the convergence behavior when using the $\mathcal{P}^+$ preconditioner, we analyze the eigenvalues of $(\mathcal{P}^+)^{-1}\mathcal{A}$ by looking at the generalized eigenvalue problem $\mathcal{A}v = \lambda\mathcal{P}^+v$, i.e.

$$\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \lambda \begin{bmatrix} A_0 & 0 \\ -B & S_0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}. \tag{3.27}$$

From (3.27) we get

$$Av_1 + B^T v_2 = \lambda A_0 v_1 \tag{3.28}$$

and

$$Bv_1 - C^T v_2 = -\lambda B v_1 + \lambda S_0 v_2. \tag{3.29}$$

We first analyze the case where $A_0 = A$ and get for $\lambda = 1$ from (3.28) that $B^T v_2 = 0$ and therefore $v_2 = 0$ under the condition that $Bv_1 = 0$. Since the kernel of B is $n - m$ dimensional, we have $\lambda = 1$ with multiplicity $n - m$. For the case $\lambda \neq 1$, (3.28) gives

$$v_1 = \frac{1}{\lambda - 1} A^{-1} B^T v_2$$

which we substitute into (3.29) to get

$$BA^{-1}B^T v_2 = \frac{\lambda(\lambda - 1)}{\lambda + 1} S_0 v_2 + \frac{\lambda - 1}{\lambda + 1} C v_2.$$

For $C = 0$ the remaining $2m$ eigenvalues of the preconditioned matrix $\widehat{\mathcal{A}}$ are given by the eigenvalues σ of

$$S_0^{-1} B A^{-1} B^T$$

and the relationship

$$\sigma = \frac{\lambda(\lambda-1)}{\lambda+1}.$$

Hence, the eigenvalues of $\widehat{\mathcal{A}}$ become

$$\lambda_{1,2} = \frac{1+\sigma}{2} \pm \sqrt{\frac{(1+\sigma)^2}{4} + \sigma}. \tag{3.30}$$

Obviously, $\sigma > 0$, and therefore, we have m negative eigenvalues given by (3.30) and $n - m + m = n$ positive eigenvalues. This shows that there are at most $2m+1$ different eigenvalues, and we expect the method to terminate in at most $2m + 1$ steps in finite precision. A similar analysis for the classical Bramble-Pasciak can be found in [105].

In contrast, the eigenvalues of the preconditioned saddle point problem $\mathcal{P}^{-1}\mathcal{A}$ in the case of $\mathcal{P}$ being the block diagonal preconditioner (cf. Section 2.1.2)

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ 0 & S_0 \end{bmatrix}$$

with $A_0 = A$ and $C = 0$ are given by $n - m$ unit eigenvalues and again the eigenvalues σ of

$$S_0^{-1} B A^{-1} B^T$$

via the relation

$$\sigma = \lambda(\lambda - 1) \text{ with } \lambda_{1,2} = \frac{1}{2} \pm \sqrt{\frac{1}{4} + \sigma}$$

Since we want S_0 to be a good preconditioner for $BA^{-1}B^T$ under the assumption that $C = 0$, we expect the eigenvalues not to differ too much from unit eigenvalues which would give a similar convergence for the block-diagonal and the Bramble-Pasciak$^+$ preconditioner. Figure 3.2 illustrates how the eigenvalues of the preconditioned matrix in the case of block-diagonal preconditioning (dashed line) and in the case of Bramble-Pasciak$^+$ preconditioning (solid line) depend on the eigenvalues σ of $BA^{-1}B^T$ in a region around 1.

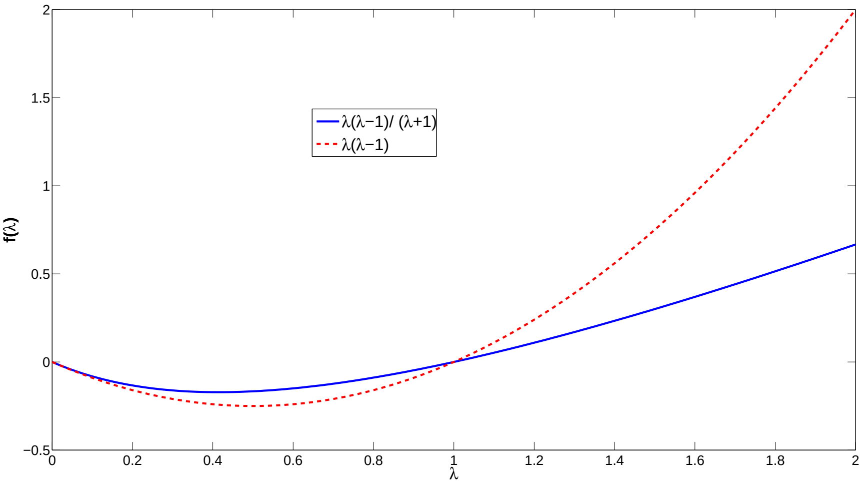

Figure 3.2: Eigenvalues of $\widehat{\mathcal{A}}$ as a function of the eigenvalues of $S_0^{-1}BA^{-1}B^T$

The indefiniteness of $(\mathcal{P}^+)^{-1}\mathcal{A}$ indicates that $\mathcal{H}$-MINRES should be used. We illustrate the convergence behavior in Chapter 6 by applying the presented methods to Stokes examples from the IFISS software [23].

Following the analysis presented in [105], we also want to analyze the case $A_0 \neq A$. The resulting bounds will only be of practical use if further knowledge about the preconditioner can be employed, e.g. eigenvalue information about A_0 and S_0 are at hand. We consider the symmetric and positive-definite block-diagonal preconditioner

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ 0 & S_0 \end{bmatrix}$$

and the generalized eigenvalue problem $\mathcal{A}u = \lambda\mathcal{P}^+u$. Using $v = \mathcal{P}^{1/2}u$ we get $\mathcal{P}^{-1/2}\mathcal{A}\mathcal{P}^{-1/2}v = \lambda\mathcal{P}^{-1/2}\mathcal{P}^+\mathcal{P}^{-1/2}v$. This gives rise to a new generalized eigenvalue problem $\tilde{\mathcal{A}}v = \lambda \tilde{P} v$ with

$$\tilde{\mathcal{A}} \equiv \begin{bmatrix} A_0^{-1/2} A A_0^{-1/2} & A_0^{-1/2} B^T S_0^{-1/2} \\ S_0^{-1/2} B A_0^{-1/2} & -S_0^{-1/2} C S_0^{-1/2} \end{bmatrix} \equiv \begin{bmatrix} \tilde{A} & \tilde{B}^T \\ \tilde{B} & -\tilde{C} \end{bmatrix}$$

and

$$\tilde{\mathcal{P}} \equiv \begin{bmatrix} I & 0 \\ -S_0^{-1/2} B A_0^{-1/2} & I \end{bmatrix} \equiv \begin{bmatrix} I & 0 \\ -\tilde{B} & I \end{bmatrix}.$$

The eigenvalue problem can hence be reformulated as

$$\begin{aligned} \tilde{A} v_1 + \tilde{B}^T v_2 &= \lambda v_1 && (3.31) \\ \tilde{B} v_1 - \tilde{C} v_2 &= -\lambda \tilde{B} v_1 + \lambda v_2. && (3.32) \end{aligned}$$

Assuming now that $v_2 = 0$ yields $\tilde{A} v_1 = \lambda v_1$ with λ an eigenvalue of the symmetric positive definite matrix $\tilde{A}$ if only $(1+\lambda)\tilde{B} v_1 = 0$. The case $v_1 = 0$ implies that $\tilde{B}^T v_2 = 0$, but since $\tilde{B}^T$ is of full rank $v_2 = 0$ based on the fact that we assumed B to have full rank. Thus, we assume that $v_1 \neq 0$ and $v_2 \neq 0$. If we multiply (3.31) on the left by the conjugate transpose v_1^*, we obtain

$$v_1^* \tilde{A} v_1 + v_1^* \tilde{B}^T v_2 = \lambda v_1^* v_1 \Longrightarrow v_1^* \tilde{B}^T v_2 = \lambda v_1^* v_1 - v_1^* \tilde{A} v_1 \qquad (3.33)$$

The conjugate transpose of (3.32) multiplied on the right by v_2 gives

$$v_1^* \tilde{B}^T v_2 - v_2^* \tilde{C} v_2 = -\bar{\lambda} v_1^* \tilde{B}^T v_2 + \bar{\lambda} v_2^* v_2. \qquad (3.34)$$

Using (3.33) gives for (3.34)

$$(1+\bar{\lambda})(\lambda v_1^* v_1 - v_1^* \tilde{A} v_1) - v_2^* \tilde{C} v_2 = \bar{\lambda} v_2^* v_2$$

which can be further simplified

$$(\lambda + |\lambda|^2)\left\|v_1\right\|^2 - (1+\bar{\lambda})v_1^*\tilde{A}v_1 - v_2^*\tilde{C}v_2 - \bar{\lambda}\left\|v_2\right\|^2 = 0. \tag{3.35}$$

Assuming that $\lambda = a + ib$, we can analyze the imaginary part of (3.35) and get

$$b(\left\|v_1\right\|^2 + v_1^*\tilde{A}v_1 + \left\|v_2\right\|^2) = 0$$

which implies that $b = 0$, and therefore we again see that eigenvalues must be real. This underlines the argument made earlier about the use of short-term recurrence methods such as MINRES since all eigenvalues of the preconditioned matrix are on the real line.

We analyze (3.35) further knowing that λ is real and under the assumption that $\|v\| = 1$ with $\left\|v_2\right\|^2 = 1 - \left\|v_1\right\|^2$ and get

$$(\lambda + \lambda^2)\left\|v_1\right\|^2 - \lambda v_1^*\tilde{A}v_1 - \lambda + \lambda\left\|v_1\right\|^2 - v_1^*\tilde{A}v_1 - v_2^*\tilde{C}v_2 = 0. \tag{3.36}$$

We then get for λ

$$\lambda_{\pm} = \frac{v_1^*\tilde{A}v_1 + 1 - 2\left\|v_1\right\|^2}{2\left\|v_1\right\|^2} \pm \sqrt{\frac{(v_1^*\tilde{A}v_1 + 1 - 2\left\|v_1\right\|^2)^2}{4\left\|v_1\right\|^4} + \frac{v_1^*\tilde{A}v_1 + v_2^*\tilde{C}v_2}{\left\|v_1\right\|^2}}. \tag{3.37}$$

Note that $v_1^*\tilde{A}v_1 + v_2^*\tilde{C}v_2 \geq 0$ for all v_1 and v_2. Since $\tilde{A}$ and $\tilde{C}$ are both symmetric matrices and $\|v_1\|\,\|v_2\| \leq 1$, we have the following bounds:

$$\hat{\mu}_{min}^{\tilde{C}} := v_2^*v_2\mu_{min}^{\tilde{C}} \leq v_2^*\tilde{C}v_2 \leq \mu_{max}^{\tilde{C}}v_2^*v_2 =: \hat{\mu}_{max}^{\tilde{C}}$$

and

$$\hat{\mu}_{min}^{\tilde{A}} := v_1^*v_1\mu_{min}^{\tilde{A}} \leq v_1^*\tilde{A}v_1 \leq \mu_{max}^{\tilde{A}}v_1^*v_1 =: \hat{\mu}_{max}^{\tilde{A}}$$

with $\mu_{min}^{\tilde{C}}$ and $\mu_{min}^{\tilde{A}}$ the minimal eigenvalue of $\tilde{C}$ and $\tilde{A}$ respectively and $\mu_{max}^{\tilde{C}}$ and $\mu_{max}^{\tilde{A}}$ the maximal eigenvalue of $\tilde{C}$ and $\tilde{A}$ respectively.

We first assume that $v_1^* \tilde{A} v_1 + 1 - 2\left\|v_1\right\|^2 > 0$ and get

$$\frac{\hat{\mu}_{min}^{\tilde{A}} + 1 - 2\left\|v_1\right\|^2}{2\left\|v_1\right\|^2} + \sqrt{\frac{(\hat{\mu}_{min}^{\tilde{A}} + 1 - 2\left\|v_1\right\|^2)^2}{4\left\|v_1\right\|^4} + \hat{\mu}_{min}^{\tilde{A}} + \hat{\mu}_{min}^{\tilde{C}}} \leq \lambda_+$$

and

$$\lambda_+ \leq \frac{\hat{\mu}_{max}^{\tilde{A}} + 1 - 2\left\|v_1\right\|^2}{2\left\|v_1\right\|^2} + \sqrt{\frac{(\hat{\mu}_{max}^{\tilde{A}} + 1 - 2\left\|v_1\right\|^2)^2}{4\left\|v_1\right\|^4} + \hat{\mu}_{max}^{\tilde{A}} + \hat{\mu}_{max}^{\tilde{C}}}$$

as well as

$$\frac{\hat{\mu}_{min}^{\tilde{A}} + 1 - 2\left\|v_1\right\|^2}{2\left\|v_1\right\|^2} - \sqrt{\frac{(\hat{\mu}_{max}^{\tilde{A}} + 1 - 2\left\|v_1\right\|^2)^2}{4\left\|v_1\right\|^4} + \hat{\mu}_{max}^{\tilde{A}} + \hat{\mu}_{max}^{\tilde{C}}} \leq \lambda_-$$

and

$$\lambda_+ \leq \frac{\hat{\mu}_{max}^{\tilde{A}} + 1 - 2\left\|v_1\right\|^2}{2\left\|v_1\right\|^2} - \sqrt{\frac{(\hat{\mu}_{min}^{\tilde{A}} + 1 - 2\left\|v_1\right\|^2)^2}{4\left\|v_1\right\|^4} + \hat{\mu}_{min}^{\tilde{A}} + \hat{\mu}_{min}^{\tilde{C}}}.$$

A similar analysis can be made for the case $v_1^* \tilde{A} v_1 + 1 - 2\left\|v_1\right\|^2 < 0$. The results here are rather complicated and only of practical use once a solid knowledge of the eigenvalues of the preconditioned blocks $\tilde{A}$ and $\tilde{C}$ is at hand.

3.5 Combination Preconditioning Examples

In this section we will present a few combinations of the methods that were introduced in Section 3.3. The possible combinations represent methods that might prove a good choice when solving practical problems in the future. At this point, we feel that the presented methods and the corresponding numerical results in Chapter 6 are a proof of concept that the combination preconditioning approach can give competitive methods.

3.5.1 Bramble-Pasciak Combination preconditioning

Using Theorem 3.5 we want to analyze the possibility of combining the classical Bramble-Pasciak configuration with the Bramble-Pasciak$^+$ preconditioner introduced in the last section. Therefore, we have the preconditioners

$$\mathcal{P}_1 = \begin{bmatrix} A_0 & 0 \\ B & -I \end{bmatrix} \quad \text{and} \quad \mathcal{P}_2 = \begin{bmatrix} A_0 & 0 \\ -B & I \end{bmatrix}$$

and for the inner products or bilinear forms

$$\mathcal{H}_1 = \begin{bmatrix} A - A_0 & 0 \\ 0 & I \end{bmatrix} \quad \text{and} \quad \mathcal{H}_2 = \begin{bmatrix} A + A_0 & 0 \\ 0 & I \end{bmatrix}.$$

Instead of $\alpha, \beta \in \mathbb{R}$ we use the combination parameters α and $1-\alpha$ and get

$$\alpha \mathcal{P}_1^{-T}\mathcal{H}_1 + (1-\alpha)\mathcal{P}_2^{-T}\mathcal{H}_2 = \begin{bmatrix} A_0^{-1}A + (1-2\alpha)I & A_0^{-1}B^T \\ 0 & (1-2\alpha)I \end{bmatrix}.$$

If we find a decomposition as described in Theorem 3.5 then a new preconditioner and bilinear form are given. One factorization possibility would be

$$P_3^{-T} = \begin{bmatrix} A_0^{-1} & A_0^{-1}B^T \\ 0 & (1-2\alpha)I \end{bmatrix} \Longrightarrow P_3 = \begin{bmatrix} A_0 & 0 \\ \frac{1}{(2\alpha-1)}B & \frac{1}{1-2\alpha}I \end{bmatrix}$$

as the new preconditioner and the bilinear form is then defined by

$$\mathcal{H}_3 = \begin{bmatrix} A + (1-2\alpha)A_0 & 0 \\ 0 & I \end{bmatrix}.$$

Note that for $\alpha = 1$, we obtain the classical Bramble-Pasciak configuration, and $\alpha = 0$ gives the Bramble-Pasciak^{+} setup. Note that for the choice $\alpha = 1/2$ the preconditioner degenerates. The obtained preconditioner can also be viewed as a special instance of an inexact Uzawa preconditioner (see [117]).

We now have to analyze if positivity in the new bilinear form can be achieved and if the bilinear form is an inner product which can be exploited for short-term recurrence methods. Hence, the matrix

$$\widehat{\mathcal{A}}^T\mathcal{H}_3$$

with $\widehat{\mathcal{A}} = \mathcal{P}_3^{-1}\mathcal{A}$ has to be analyzed. The matrix

$$\widehat{\mathcal{A}}^T\mathcal{H}_3 = \begin{bmatrix} AA_0^{-1}A + (1-2\alpha)A & AA_0^{-1}B^T + (1-2\alpha)B^T \\ BA_0^{-1}A + (1-2\alpha)B & BA_0^{-1}B^T - (1-2\alpha)C \end{bmatrix}$$

can, similarly to the Bramble-Pasciak case, be factorized as the congruence transform

$$\widehat{\mathcal{A}}^T\mathcal{H}_3 =$$

$$\begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} AA_0^{-1}A + (1-2\alpha)A & 0 \\ 0 & (2\alpha-1)(BA^{-1}B^T + C) \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix}.$$

The Sylvester Law of Inertia indicates that the number of positive and negative eigenvalues is determined by the eigenvalues of the matrix

$$\begin{bmatrix} AA_0^{-1}A + (1-2\alpha)A & 0 \\ 0 & (2\alpha-1)(BA_0^{-1}B^T + C) \end{bmatrix}$$

which we can analyze in a similar manner to (3.12), (3.13) above. It is easy to see that the block $(2\alpha-1)(BA_0^{-1}B^T+C)$ is positive for $\alpha > 1/2$. With this choice for α we have to find conditions such that the block $AA_0^{-1}A+(1-2\alpha)A$ is also positive definite. Similar to the analysis made in Section 3.4, we note the equivalence

$$A\left(A_0^{-1}+(1-2\alpha)A^{-1}\right)A$$

and therefore positivity is given if $y^TA_0y < (2\alpha-1)y^TAy$ which can also be written as

$$A_0 < (2\alpha-1)A.$$

In addition we want the matrix $\mathcal{H}_3$ to define an inner product which will be satisfied if the block $A+(1-2\alpha)A_0 > 0$ which is equivalent to

$$\frac{1}{2\alpha-1}A > A_0.$$

Again, the case $\alpha = 1$ gives the Bramble Pasciak configuration and $\alpha = 0$ shows that there is no configuration that makes the Bramble-Pasciak$^+$ setup positive definite and CG reliably applicable. It is still possible to obtain a reliable CG method in the combination preconditioning case, i.e. if

$$A_0 < \min\left\{(2\alpha-1)A, \frac{1}{2\alpha-1}A\right\}$$

which is a more general restriction on A_0 than the Bramble and Pasciak case, $\alpha = 1$.

3.5.2 Bramble Pasciak and Benzi-Simoncini

As a less practical example,in order to show how even very different methods can be combined, we consider $\mathcal{P}_1, \mathcal{H}_1$ defined by the classical Bramble-Pasciak method (3.9), (3.11) and $\mathcal{P}_2, \mathcal{H}_2$ defined by the Benzi-Simoncini approach

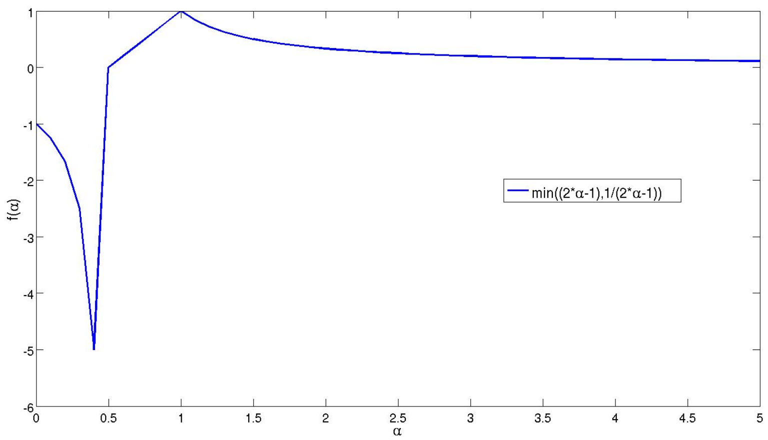

Figure 3.3: Value of $\min\left\{(2\alpha-1), \frac{1}{2\alpha-1}\right\}$

(3.18), (3.19). From Theorem 3.5 we get

$$(\alpha\mathcal{P}_1^{-T}\mathcal{H}_1+\beta\mathcal{P}_2^{-T}\mathcal{H}_2)=\begin{bmatrix}(\alpha A_0^{-1}+\beta I)A-(\alpha+\beta\gamma)I & (\alpha A_0^{-1}+\beta I)B^T\\ -\beta B & -(\alpha+\beta\gamma)I\end{bmatrix}$$

which is self-adjoint $\forall\alpha,\beta\in\mathbb{R}$ in $\langle\cdot,\cdot\rangle_{\mathcal{A}}$. If we are able to split this into a new preconditioner $\mathcal{P}_3$ and a symmetric matrix $\mathcal{H}_3$, Theorem 3.5 guarantees that $\mathcal{P}_3^{-1}\mathcal{A}$ will be self-adjoint in $\langle\cdot,\cdot\rangle_{\mathcal{H}_3}$.

One possibility is

$$\mathcal{P}_3^{-T}=\begin{bmatrix}\alpha A_0^{-1}+\beta I & 0\\ 0 & -\beta I\end{bmatrix} \tag{3.38}$$

and

$$\mathcal{H}_3=\begin{bmatrix}A-(\alpha+\beta\gamma)(\alpha A_0^{-1}+\beta I)^{-1} & B^T\\ B & \frac{\alpha+\beta\gamma}{\beta}I\end{bmatrix}. \tag{3.39}$$

Numerical results we have computed with this combination were less promising and we have omitted them. The bilinear form $\langle\cdot,\cdot\rangle_{\mathcal{H}_3}$ is not so convenient to work with.

3.5.3 Bramble-Pasciak and Schöberl-Zulehner

We now combine the Bramble-Pasciak CG and the method proposed by Schöberl and Zulehner in [102]. Therefore, we consider the preconditioners

$$\mathcal{P}_1 = \begin{bmatrix} A_0 & 0 \\ B & -S_0 \end{bmatrix} \quad \text{and} \quad \mathcal{P}_2 = \begin{bmatrix} A_0 & B^T \\ B & BA_0^{-1}B^T - \hat{S} \end{bmatrix}$$

and the inner products or symmetric bilinear forms

$$\mathcal{H}_1 = \begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix} \quad \text{and} \quad \mathcal{H}_2 = \begin{bmatrix} A_0 - A & 0 \\ 0 & BA_0^{-1}B^T - \hat{S} \end{bmatrix}.$$

Again, we look for a factorization of $\alpha\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{P}_2^{-T}\mathcal{H}_2$ as $\mathcal{P}_3^{-T}\mathcal{H}_3$. Setting

$$S_0 = BA_0^{-1}B^T - \hat{S}$$

yields

$$\begin{aligned} \alpha\mathcal{P}_1^{-T}\mathcal{H}_1 + \beta\mathcal{P}_2^{-T}\mathcal{H}_2 = {} & \alpha \begin{bmatrix} A_0^{-1} & A_0^{-1}B^TS_0^{-1} \\ 0 & -S_0^{-1} \end{bmatrix} \begin{bmatrix} A - A_0 & \\ & S_0 \end{bmatrix} \\ & + \beta \begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} A_0^{-1} & 0 \\ \hat{S}^{-1}BA_0^{-1} & -\hat{S}^{-1} \end{bmatrix} \begin{bmatrix} A_0 - A & \\ & S_0 \end{bmatrix} \end{aligned} \tag{3.40}$$

which can be reformulated using

$$\begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_0 - A & 0 \\ 0 & S_0 \end{bmatrix} = \begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix} \tag{3.41}$$

and

$$\begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} -A_0^{-1} & 0 \\ 0 & -S_0^{-1} \end{bmatrix} = \begin{bmatrix} -A_0^{-1} & A_0^{-1}B^T S_0^{-1} \\ 0 & -S_0^{-1} \end{bmatrix}. \tag{3.42}$$

Hence, (3.40) simplifies to

$$\begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I \end{bmatrix} \left(\alpha \begin{bmatrix} -A_0^{-1} & 0 \\ 0 & -S_0^{-1} \end{bmatrix} + \beta \begin{bmatrix} A_0^{-1} & 0 \\ \hat{S}^{-1}BA_0^{-1} & -\hat{S}^{-1} \end{bmatrix} \right)$$

$$\begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix}$$

with

$$\mathcal{P}_3^{-1} = \begin{bmatrix} (\beta - \alpha)A_0^{-1} & \beta A_0^{-1}B^T\hat{S}^{-1} \\ 0 & -(\alpha S_0^{-1} + \beta \hat{S}^{-1}) \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA_0^{-1} & I \end{bmatrix} \tag{3.43}$$

as the inverse of the new preconditioner, and as an inner product matrix we get

$$\mathcal{H}_3 = \begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix}.$$

The block $-(\alpha S_0^{-1} + \beta \hat{S}^{-1})$ of $\mathcal{P}_3^{-1}$ is not well suited for numerical purposes since it involves the inverse of $S_0 = BA_0^{-1}B^T - \hat{S}$. Therefore, we try a different approach combining Schöberl's and Zulehner's method with the Bramble-Pasciak CG. Thus, we consider the preconditioners

$$\mathcal{P}_1 = \begin{bmatrix} A_0 & 0 \\ B & -\hat{S} \end{bmatrix} \quad \text{and} \quad \mathcal{P}_2 = \begin{bmatrix} A_0 & B^T \\ B & BA_0^{-1}B^T - \hat{S} \end{bmatrix}$$

and the inner products

$$\mathcal{H}_1 = \begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S} \end{bmatrix} \quad \text{and} \quad \mathcal{H}_2 = \begin{bmatrix} A_0 - A & 0 \\ 0 & BA_0^{-1}B^T - \hat{S} \end{bmatrix}$$

where we chose $S_0 = \hat{S}$ rather than $S_0 = BA_0^{-1}B^T - \hat{S}$. Once more, we try to find a decomposition as $\mathcal{P}_3^{-T}\mathcal{H}_3$ of

$$\alpha \mathcal{P}_1^{-T}\mathcal{H}_1 + \beta \mathcal{P}_2^{-T}\mathcal{H}_2 = \alpha \begin{bmatrix} A_0^{-1} & A_0^{-1}B^T\hat{S}^{-1} \\ 0 & -\hat{S}^{-1} \end{bmatrix} \begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S} \end{bmatrix}$$

$$+\beta \begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} A_0^{-1} & 0 \\ \hat{S}^{-1}BA_0^{-1} & -\hat{S}^{-1} \end{bmatrix} \begin{bmatrix} A_0 - A & 0 \\ 0 & BA_0^{-1}B^T - \hat{S} \end{bmatrix}.$$

Using a simple modification of (3.41) then gives for the last expression

$$\left(\alpha\begin{bmatrix} A_0^{-1} & A_0^{-1}B^T\hat{S}^{-1} \\ 0 & -\hat{S}^{-1}\end{bmatrix} + \beta\begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I\end{bmatrix}\begin{bmatrix} A_0^{-1} & 0 \\ \hat{S}^{-1}BA_0^{-1} & -\hat{S}^{-1}\end{bmatrix}\right.$$

$$\left.\begin{bmatrix} -I & 0 \\ 0 & (BA_0^{-1}B^T - \hat{S})\hat{S}^{-1}\end{bmatrix}\right)\begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S}\end{bmatrix}.$$

This can be further simplified using a modification of (3.42) and the result is

$$\begin{bmatrix} I & -A_0^{-1}B^T \\ 0 & I\end{bmatrix}\begin{bmatrix} (\alpha-\beta)A_0^{-1} & 0 \\ -\beta\hat{S}^{-1}BA_0^{-1} & (\beta-\alpha)\hat{S}^{-1} + \beta\hat{S}^{-1}BA_0^{-1}B^T\hat{S}^{-1}\end{bmatrix}$$

$$\begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S}\end{bmatrix}.$$

The preconditioner is then given by

$$\mathcal{P}_3^{-1} = \begin{bmatrix} (\alpha-\beta)A_0^{-1} & -\beta A_0^{-1}B^T\hat{S}^{-1} \\ 0 & (\beta-\alpha)\hat{S}^{-1} - \beta\hat{S}^{-1}BA_0^{-1}B^T\hat{S}^{-1}\end{bmatrix}\begin{bmatrix} I & 0 \\ -BA_0^{-1} & I\end{bmatrix} \tag{3.44}$$

with

$$\mathcal{H}_3 = \begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S}\end{bmatrix} \tag{3.45}$$

defining the bilinear form. It is also possible to reformulate the preconditioner presented by Schöberl and Zulehner using (3.44), i.e. $\beta = 1$ and $\alpha = 0$. In

Chapter 6, results that show the competitiveness of the setup introduced in (3.44) and (3.45) are given. In order to achieve the inner product given in (3.45) one could try and replace S_0 by $\hat{S}$ in (3.40) but since the Schöberl and Zulehner method is not self-adjoint in

$$\begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S} \end{bmatrix}$$

we would have to introduce an extra factor in (3.40) to achieve the desired form. This means that we cannot simply replace S_0 by $\hat{S}$ in (3.43).

The method generated by combination preconditioning has a slightly more expensive preconditioner (3.44), i.e. one additional solve with $\hat{S}$, but the inner product matrix (3.45) is less expensive to apply than the one used by Schöberl and Zulehner because there is no need to solve with A_0. We here assume that $\hat{S}$ and A_0 are explicitly given, which might not be the case when working with multigrid preconditioning, for example.

CHAPTER 4

SADDLE POINT PROBLEM IN OPTIMIZATION

The properties of the saddle point system (1.14) can change when the underlying application changes, and hence, in this Chapter, we look at matrices with different definiteness properties coming from optimization problems. The strong connection between optimization and saddle point problems is beautifully explained in [19, 42]. The first major observation presented in this chapter is a general framework for saddle point problems that allows for many methods to be represented by it. The second new point presented is the reformulation of a recently introduced method as a method with non-standard inner products. This new method allows more flexibility than the known form. The results given in this Chapter were recently submitted in [18].

4.1 Reformulation

Assume that a saddle point problem of the form (1.14) is given where A is symmetric and positive definite on the kernel of B which we assume to be of full rank. The block C is assumed to be positive (semi)definite. It follows directly that any solution x to (1.14) also satisfies

$$\left(\sigma\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} + \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}\begin{bmatrix} D & F^T \\ F & E \end{bmatrix}\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}\right)\begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix}$$

$$= \sigma\begin{bmatrix} f \\ g \end{bmatrix} + \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}\begin{bmatrix} D & F^T \\ F & E \end{bmatrix}\begin{bmatrix} f \\ g \end{bmatrix} \tag{4.1}$$

for arbitrary σ, symmetric matrices $D \in \mathbb{R}^{n\times n}$ and $E \in \mathbb{R}^{m\times m}$ and any matrix $F \in \mathbb{R}^{m\times n}$. We denote the coefficient matrix and right-hand side of (4.1) as $K(\sigma, D, E, F)$ and $b(\sigma, D, E, F)$, respectively, and note that $K = K(1, 0, 0, 0)$ and $b = b(1, 0, 0, 0)$ gives the original saddle point system (1.14). Many well-known methods can be represented using this reformulation. For example,

- $K(0, I, I, 0)$ gives the normal equations for (1.14);
- $K(-1, A^{-1}, 0, 0)$ gives the Schur-complement method for finding y when A is nonsingular;
- $K(0, A^{-1}, C^{-1}, 0)$ gives the primal-dual Schur complement method for finding x and y simultaneously when both A and C are nonsingular; and
- $K(1, 0, (1+\nu)C^{-1}, 0)$ for a given ν (in particular $\nu = 1$) gives the system to which Forsgren, Gill, Griffin apply the preconditioned conjugate gradient (PCG) method (see [32]). The matrices C and $A + B^TC^{-1}B$ are assumed to be positive definite.

There are also a variety of methods that solve (1.14) by applying the conjugate gradient (CG) method within a non-standard inner-product. In Section 3.2 we introduced the Bramble-Pasciak preconditioner and showed that for the applicability of such a non-standard inner product method, it is important to look at the matrix $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$ and its definiteness properties. We saw that it is possible to apply PCG whenever this matrix is positive definite. The matrix $K(\sigma, D, E, F)$ given in the general framework (4.1) represents exactly the system matrix $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$ given in Equation (3.15), i.e.

$$\mathcal{H}\mathcal{P}^{-1}\mathcal{A}x = \mathcal{H}\mathcal{P}^{-1}b.$$

We now give a list of examples and in the next section discuss the representation in the reformulated form in more detail.

- The setup $K(-1, A_0^{-1}, 0, 0)$ gives (3.15) for the well-respected Bramble-Pasciak configuration for a given A_0 (see Section 3.2 for a detailed description). The matrices A and $BA^{-1}B^T + C$ are assumed to be positive definite, and A_0 is such that $A - A_0$ is also symmetric and positive definite.

- The setup $K(-\gamma, I, -I, 0)$ gives (3.15) for Liesen and Parlett's method for a given γ (see Section 3.3) [69, 70]. The matrix A is assumed to be positive definite and γ lies in the interval $[\lambda_{\max}(C), \lambda_{\min}(A)]$. This method extends that of Benzi and Simoncini presented in Section 3.3 to the case where $C \neq 0$.

- The setup $K(-(\alpha+\beta\gamma), \alpha A_0^{-1} + \beta I, -\beta I, 0)$ gives (3.15) for one of the combination preconditioners introduced in Chapter 3. The assumptions of both Bramble-Pasciak and Liesen *et al.* must hold, and α, β and γ must be chosen such that $K(-(\alpha + \beta\gamma), \alpha A_0^{-1} + \beta I, -\beta I, 0) = \alpha K(-1, A_0^{-1}, 0, 0) + \beta K(-\gamma, I, -I, 0)$ is positive definite.

- The setup $K(1, A_0^{-1}(B^T C_0^{-1} B - A_0^{-1})A_0^{-1}, C_0^{-1}, -C_0^{-1}BA_0^{-1})$ gives (3.15) of the method presented by Schöberl and Zulehner for the case $C = 0$ used in Sections 3.3 and 3.5 for given A_0 and C_0. The matrix A is assumed to be positive definite on the kernel of B, A_0 is such that $A_0 - A$ is symmetric and positive definite, and C_0 is such that $BA_0^{-1}B^T - C_0$ is symmetric and positive definite.

In Section 3.2, we showed that for the Bramble-Pasciak case solving system (1.3) is equivalent to solving (3.15) with preconditioner $\mathcal{H}$. In Section 4.2, we will clarify this point once more with regard to the general framework (4.1), since it may not be obvious why the above formulations produce algorithms that (in exact arithmetic) produce iterates which are equivalent to those produced by the CG methods within a non-standard inner-product.

Before considering the non-standard inner-product conjugate gradient methods, we will consider what properties need to hold to guarantee that $K(\sigma, D, E, F)$ is symmetric and positive definite (and thus one may use methods such as CG rather than MINRES). Clearly, D and E both need to be symmetric. Furthermore, we may factorize $K(\sigma, D, E, F)$ as

$$K(\sigma, D, E, F) = \begin{bmatrix} \Theta_1 & \Theta_2^T \\ \Theta_2 & \Theta_3 \end{bmatrix} =$$

$$\begin{bmatrix} I & \Theta_2^T\Theta_3^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} \Theta_1 - \Theta_2^T\Theta_3^{-1}\Theta_2 & 0 \\ 0 & \Theta_3 \end{bmatrix} \begin{bmatrix} I & 0 \\ \Theta_3^{-1}\Theta_2 & I \end{bmatrix} \tag{4.2}$$

or

$$K(\sigma, D, E, F) =$$

$$\begin{bmatrix} I & 0 \\ \Theta_2\Theta_1^{-1} & I \end{bmatrix} \begin{bmatrix} \Theta_1 & 0 \\ 0 & \Theta_3 - \Theta_2\Theta_1^{-1}\Theta_2^T \end{bmatrix} \begin{bmatrix} I & \Theta_1^{-1}\Theta_2^T \\ 0 & I \end{bmatrix}, \tag{4.3}$$

where

$$\begin{aligned}\Theta_1 &= \sigma A + ADA + B^T FA + AF^T B + B^T EB, && (4.4)\\ \Theta_2 &= \sigma B + BDA - CFA + BF^T B - CEB, && (4.5)\\ \Theta_3 &= BDB^T - CFB^T - BF^T C + CEC - \sigma C. && (4.6)\end{aligned}$$

Using Sylvester's law of inertia [53], we obtain the following theorem:

Theorem 4.1. *Let Θ_1, Θ_2 and Θ_3 be as defined in (4.4)–(4.6). $K(\sigma, D, E, F)$ is symmetric and positive definite if and only if*

- *D and E are symmetric,*
- *Θ_3 is positive definite and*
- *$\Theta_1 - \Theta_2^T \Theta_3^{-1} \Theta_2$ is positive definite.*

Alternatively, $K(\sigma, D, E, F)$ is symmetric and positive definite if and only if

- *D and E are symmetric,*
- *Θ_1 is positive definite and*
- *$\Theta_3 - \Theta_2 \Theta_1^{-1} \Theta_2^T$ is positive definite.*

Proof. The proof follows from the decompositions (4.2) and (4.3) and the use of Sylvester's law of inertia. □

Clearly, $K(\sigma, D, E, F)$ is symmetric and positive definite if and only if $\mathcal{A}^{-1} K(\sigma, D, E, F) \mathcal{A}^{-1}$ is symmetric and positive definite. This is equivalent to requiring that

$$\sigma \mathcal{A}^{-1} + \begin{bmatrix} D & F^T \\ F & E \end{bmatrix}$$

is symmetric and positive definite. We will consider different cases for A and C separately. Let

$$\sigma\mathcal{A}^{-1} + \begin{bmatrix} D & F^T \\ F & E \end{bmatrix} = \begin{bmatrix} \Omega_1 & \Omega_2^T \\ \Omega_2 & \Omega_3 \end{bmatrix} \tag{4.7}$$

for given matrices of Ω_1, Ω_2 and Ω_3. With Ω_3 invertible, we may factorize (4.7) as

$$\begin{bmatrix} \Omega_1 & \Omega_2^T \\ \Omega_2 & \Omega_3 \end{bmatrix} = \begin{bmatrix} I & \Omega_2^T\Omega_3^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} \Omega_1 - \Omega_2^T\Omega_3^{-1}\Omega_2 & 0 \\ 0 & \Omega_3 \end{bmatrix} \begin{bmatrix} I & 0 \\ \Omega_3^{-1}\Omega_2 & I \end{bmatrix}. \tag{4.8}$$

Using Sylvester's law of inertia,

$$\sigma\mathcal{A}^{-1} + \begin{bmatrix} D & F^T \\ F & E \end{bmatrix}$$

is positive definite if and only if Ω_3 and $\Omega_1 - \Omega_2^T\Omega_3^{-1}\Omega_2$ are both positive definite. Equivalently for invertible Ω_1, we may use the factorization

$$\begin{bmatrix} \Omega_1 & \Omega_2^T \\ \Omega_2 & \Omega_3 \end{bmatrix} = \begin{bmatrix} I & 0 \\ \Omega_2\Omega_1^{-1} & I \end{bmatrix} \begin{bmatrix} \Omega_1 & 0 \\ 0 & \Omega_3 - \Omega_2\Omega_1^{-1}\Omega_2^T \end{bmatrix} \begin{bmatrix} I & \Omega_1^{-1}\Omega_2^T \\ 0 & I \end{bmatrix}. \tag{4.9}$$

Again with Sylvester's law of inertia it is easy to see that

$$\sigma\mathcal{A}^{-1} + \begin{bmatrix} D & F^T \\ F & E \end{bmatrix}$$

is positive definite if and only if Ω_1 and $\Omega_3 - \Omega_2\Omega_1^{-1}\Omega_2^T$ are both positive definite.

Corollary 4.2. *If A is symmetric and nonsingular, and*

$$\begin{aligned} S_A &= C + BA^{-1}B^T, \\ \Upsilon_1 &= D + \sigma A^{-1} - \sigma A^{-1}B^T S_A^{-1} B A^{-1}, \\ \Upsilon_2 &= \sigma F + S_A^{-1} B A^{-1}, \\ \Upsilon_3 &= E - \sigma S_A^{-1}, \end{aligned}$$

then $K(\sigma, D, E, F)$ is symmetric and positive definite if and only if

- *D and E are symmetric,*
- *Υ_3 is positive definite and*
- *$\Upsilon_1 - \Upsilon_2^T \Upsilon_3^{-1} \Upsilon_2$ is positive definite.*

Proof. If A is nonsingular, then

$$\mathcal{A}^{-1} = \begin{bmatrix} A^{-1} - A^{-1}B^T S^{-1} B A^{-1} & A^{-1}B^T S^{-1} \\ S^{-1}BA^{-1} & -S^{-1} \end{bmatrix},$$

where $S = C + BA^{-1}B^T$. Use of factorization (4.8) completes the proof. □

Corollary 4.3. *If C is symmetric and nonsingular, and*

$$\begin{aligned} S_C &= A + B^T C^{-1} B, \\ \Delta_1 &= D + \sigma S_C^{-1}, \\ \Delta_2 &= F + \sigma C^{-1} B S_C^{-1}, \\ \Delta_3 &= E + \sigma C^{-1} B S_C^{-1} B^T C^{-1} - C^{-1}, \end{aligned}$$

then $K(\sigma, D, E, F)$ is symmetric and positive definite if and only if

- *D and E are symmetric,*
- *Δ_1 is positive definite, and*
- *$\Delta_3 - \Delta_2\Delta_1^{-1}\Delta_2^T$ is positive definite.*

Proof. If C is nonsingular, then

$$\mathcal{A}^{-1} = \begin{bmatrix} S^{-1} & S^{-1}B^TC^{-1} \\ C^{-1}BS^{-1} & C^{-1}BS^{-1}B^TC^{-1} - C^{-1} \end{bmatrix},$$

where $S = A + B^TC^{-1}B$. Use of factorization (4.9) completes the proof. □

Corollary 4.4. *If $C = 0$, the columns of $Z \in \mathbb{R}^{n\times(n-m)}$ span the nullspace of B, and if $B^\dagger$ is the Moore-Penrose inverse of B [53], then*

$$\begin{aligned} S_Z &= Z^TAZ, \\ \Gamma_1 &= D + \sigma Z S_Z^{-1} Z^T, \\ \Gamma_2 &= F + \sigma B^{\dagger T}\left(I - AZS_Z^{-1}Z^T\right), \\ \Gamma_3 &= E + \sigma B^{\dagger T}\left(AZS_ZZ^TA - A\right)B^\dagger. \end{aligned}$$

$K(\sigma, D, E, F)$ is symmetric and positive definite if and only if

- *D and E are symmetric,*
- *Γ_1 is positive definite,*
- *$\Gamma_3 - \Gamma_2\Gamma_1^{-1}\Gamma_2^T$ is positive definite.*

Proof. If $C = 0$, the columns of $Z \in \mathbb{R}^{n\times(n-m)}$ span the nullspace of B, and $B^\dagger$ be the Moore-Penrose inverse of B, then

$$\mathcal{A}^{-1} = \begin{bmatrix} ZS^{-1}Z^T & \left(I - ZS^{-1}Z^TA\right)B^\dagger \\ B^{\dagger T}\left(I - AZS^{-1}Z^T\right) & B^{\dagger T}\left(AZS^{-1}Z^TA - A\right)B^\dagger \end{bmatrix},$$

where $S = Z^T AZ$. Use of factorization (4.9) completes the proof. □

Conditions for the case where C is rank-deficient but nonzero may be derived by factoring C as

$$C = U^T \begin{bmatrix} \widehat{C} & 0 \\ 0 & 0 \end{bmatrix} U,$$

where $\widehat{C}$ is nonsingular and U is unitary. Premultiplying $\mathcal{A}$ by $\begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix}$ and post multiplying by the inverse of this matrix reveals a saddle point system to which either Corollary 4.3 or Corollary 4.4 could be applied.

4.2 Reformulation and non-standard inner products

We illustrated earlier in Section 3.2 that the CG method with non-standard inner product is equivalent to a preconditioned CG method. Figure 3.1 showed that only by preconditioning the matrix $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$ on the left with $\mathcal{H}^{-1}$ do we get the same convergence as the CG with the non-standard inner product. Nevertheless, it is very interesting to look at the matrix $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$ especially when comparing the reformulation (4.1) and methods with non-standard inner products as presented in Section 3.3. In the last section we noticed that in the presence of non-standard inner products the reformulation $K(\sigma, E, D, F)$ corresponds to the matrix $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$, and using this we see that the matrix $\mathcal{H}\mathcal{P}^{-1}$ is equivalent to

$$\sigma I + \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} D & F^T \\ F & E \end{bmatrix}.$$

As we saw earlier, for each non-standard inner product method (cf. Section 3.3) it has to be shown that the matrix $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$ is symmetric and positive definite. Once this is proven the applicability of CG for $\mathcal{P}^{-1}\mathcal{A}$ in $\langle .,. \rangle_{\mathcal{H}}$ is guaranteed. For the methods of Bramble-Pasciak, Benzi-Simoncini, etc. this means that the corresponding alternative formulation $K(\sigma, D, E, F)$, which corresponds to the matrix $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}$, must be positive definite as well. This guarantees the applicability of CG to the matrix $K(\sigma, D, E, F)$ (cf. Section 3.2).

In the last section we presented a number of non-standard inner product methods that can be represented in the reformulated form. We want to discuss this here for the Bramble-Pasciak CG, such that solving system (1.14) is equivalent to solving the system (3.15), i.e.

$$\mathcal{H}\mathcal{P}^{-1}\mathcal{A}\begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} = \mathcal{H}\mathcal{P}^{-1}\begin{bmatrix} f \\ g \end{bmatrix},$$

which can be obtained from (4.1) via

$$\begin{aligned} K(-1, A_0^{-1}, 0, 0) &= \left(-1 \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} + \begin{bmatrix} A \\ B \end{bmatrix} A_0^{-1} \begin{bmatrix} A & B^T \end{bmatrix} \right) \\ &= \begin{bmatrix} AA_0^{-1}A - A & AA_0^{-1}B^T - B^T \\ BA_0^{-1}A - B & BA_0^{-1}B^T + C \end{bmatrix} = \widehat{\mathcal{A}}^T\mathcal{H} = \mathcal{H}\widehat{\mathcal{A}}. \end{aligned}$$

Now this shows that the reformulation (4.1) gives the matrix $\mathcal{H}\widehat{\mathcal{A}}$ from the Bramble-Pasciak CG. Remember that we illustrated in Section 3.2 that PCG with preconditioner $\mathcal{H}$ applied to the system with the matrix $\mathcal{H}\widehat{\mathcal{A}} = K(-1, A_0^{-1}, 0, 0)$ gives the identical Krylov subspace to the Bramble-Pasciak setup (cf. Figure 3.1). Hence, the approximation to the solution will be the same for both methods. In a similar fashion, the non-standard inner product methods given in Section 3.3 can be obtained from (4.1). And PCG applied to the matrix $K(\sigma, D, E, F)$ with preconditioner $\mathcal{H}$ will give the same results as the ones obtained from the non-standard inner product version of the same method.

4.3 Using the reformulation

In Section 4.1, we illustrated that different methods for solving saddle point problems can be presented within the same framework (see Equation (4.1)). Furthermore, we showed that for a Bramble-Pasciak setup it would not be feasible to use the alternative formulation for numerical experiments due to the fact that we first multiply by $\mathcal{H}$ and then use it as a preconditioner (cf. Section 3.2). In this section we want to show how the alternative formulation can be used to generate a Bramble-Pasciak-like method for another method that lies within the same framework. We therefore quickly summarize the method of Forsgren *et al.* introduced in [32].

4.3.1 The method of Forsgren, Gill and Griffin (FGG)

Forsgren, Gill and Griffin work with a saddle point problem of the general form

$$\mathcal{A}(\nu)\begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} = \begin{bmatrix} A+(1+\nu)B^TC^{-1}B & -\nu B^T \\ -\nu B & \nu C \end{bmatrix}\begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \tag{4.10}$$

where $\nu \in \mathbb{R}$ which is $K(1, 0, (1+\nu)C^{-1}, 0)$ in our general setting. We want to emphasize the fact that C must be definite in this formulation, as already observed. For $\nu = -1$, $\mathcal{A}(\nu)$ gives the classical saddle point formulation, for $\nu = 0$ we obtain a condensed system, which is equivalent to the Schur-complement method for finding the solution, and for $\nu = 1$ the result is a doubly augmented system

$$\mathcal{A}(1) = \begin{bmatrix} A + 2B^TC^{-1}B & -B^T \\ -B & C \end{bmatrix}. \tag{4.11}$$

Using the splitting

$$\begin{bmatrix} A + (1+\nu)B^TC^{-1}B & -\nu B^T \\ -\nu B & \nu C \end{bmatrix} =$$

$$\begin{bmatrix} I & -B^TC^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A + B^TC^{-1}B & 0 \\ 0 & \nu C \end{bmatrix} \begin{bmatrix} I & 0 \\ -B^TC^{-1} & I \end{bmatrix}$$

Sylvester's law of inertia tells us that the matrix $A(\nu)$ is positive definite if $A + B^TC^{-1}B > 0$, $C > 0$ and $\nu > 0$. In addition, a general preconditioner

$$\mathcal{P}(\nu) = \begin{bmatrix} G + (1+\nu)B^TC^{-1}B & -\nu B^T \\ -\nu B & \nu C \end{bmatrix} \tag{4.12}$$

is introduced where G is an approximation to A and $G + B^TC^{-1}B > 0$. Again, $\mathcal{P}(\nu)$ represents different preconditioners for different instances of ν. In practice, it is often useful to use the decomposition

$$\begin{bmatrix} G + (1+\nu)B^TC^{-1}B & -\nu B^T \\ -\nu B & \nu C \end{bmatrix} = \begin{bmatrix} I & (1+\nu)B^TC^{-1} \\ 0 & -\nu I \end{bmatrix} \begin{bmatrix} G & B^T \\ B & -C \end{bmatrix}$$

to solve a system with the preconditioner $\mathcal{P}(\nu)$. The eigenvalues of the preconditioned system $\mathcal{P}(\nu)^{-1}\mathcal{A}(\nu)$ can be analyzed by assuming that an eigenpair $(\lambda, [x^T y^T]^T)$ is given. Hence, we get

$$\begin{array}{rcl} Ax + (1+\nu)B^TC^{-1}Bx - \nu B^Ty & = & \lambda Gx + \lambda(1+\nu)B^TC^{-1}Bx - \lambda\nu B^Ty \\ -\nu Bx + \nu Cy & = & -\lambda\nu Bx + \lambda\nu Cy \end{array} \tag{4.13}$$

and assuming that $\lambda = 1$ the first Equation in (4.13) reduces to $Ax = Gx$ which gives $x = 0$ and the second equation gives an arbitrary choice for y. Since y can be taken from an m-dimensional space, we have m eigenvalues at 1. Assuming now that $\lambda \neq 1$, we get from the second Equation in (4.13) that $y = C^{-1}Bx$, which put into the first equation results in

$$(A + B^TC^{-1}B)x = \lambda(G + B^TC^{-1}B)x.$$

Therefore, n eigenvalues of the preconditioned matrix $\mathcal{P}(\nu)^{-1}$ are given by the eigenvalues of

$$(G + B^TC^{-1}B)^{-1}(A + B^TC^{-1}B).$$

Therefore, in exact arithmetic, convergence is given in at most $n+1$ steps.

4.3.2 A Bramble-Pasciak-like approach

In this section we show the equivalence of the method proposed by Forsgren *et al.* and a Bramble-Pasciak-like method. In order to construct a Bramble-Pasciak-like method we consider the preconditioner

$$\mathcal{P}_- = \begin{bmatrix} A_0 & B^T \\ 0 & -C_0 \end{bmatrix} \text{ with } \mathcal{P}_-^{-1} = \begin{bmatrix} A_0^{-1} & A_0^{-1}B^TC_0^{-1} \\ 0 & -C_0^{-1} \end{bmatrix} \tag{4.14}$$

and the bilinear form

$$\mathcal{H}_{-} = \begin{bmatrix} A_0 & 0 \\ 0 & C - C_0 \end{bmatrix}. \tag{4.15}$$

It is easy to see that the preconditioned matrix $\widehat{\mathcal{A}} = \mathcal{P}_{-}^{-1}\mathcal{A}$ is self-adjoint in this bilinear form by verifying that $\widehat{\mathcal{A}}^T\mathcal{H}_{-} = \mathcal{H}_{-}\widehat{\mathcal{A}}$ holds. In more detail, we get that

$$\begin{aligned} \widehat{\mathcal{A}}^T\mathcal{H}_{-} &= \begin{bmatrix} AA_0^{-1} + B^TC_0^{-1}BA_0^{-1} & -B^TC_0^{-1} \\ BA_0^{-1} - CC_0^{-1}BA_0^{-1} & CC_0^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & C - C_0 \end{bmatrix} \\ &= \begin{bmatrix} A + B^TC_0^{-1}B & -B^TC_0^{-1}C + B^T \\ B - CC_0B & CC_0^{-1}C - C \end{bmatrix} \end{aligned}$$

is identical to

$$\begin{aligned} \mathcal{H}_{-}\widehat{\mathcal{A}} &= \begin{bmatrix} A_0 & 0 \\ 0 & C - C_0 \end{bmatrix} \begin{bmatrix} A_0^{-1}A + A_0^{-1}B^TC_0^{-1}B & A_0^{-1}B^T - A_0^{-1}B^TC_0^{-1}C \\ -C_0^{-1}B & C_0^{-1}C \end{bmatrix} \\ &= \begin{bmatrix} A + B^TC_0^{-1}B & -B^TC_0^{-1}C + B^T \\ B - CC_0B & CC_0^{-1}C - C \end{bmatrix}. \end{aligned}$$

The connection to the method by Forsgren *et al.* can be made by looking at

(4.1) in the setup

$$K(1,0,(1+\nu)C^{-1},0)=\begin{bmatrix} A+(1+\nu)B^TC^{-1}B & -\nu B^T \\ -\nu B & \nu C \end{bmatrix}.$$

This matrix can also be expressed as

$$\begin{aligned}\mathcal{H}_-\mathcal{P}_-^{-1}\mathcal{A} &= \begin{bmatrix} I & 0 \\ 0 & C-(1+\nu)^{-1}C \end{bmatrix}\begin{bmatrix} A+B^T(1+\nu)C^{-1}B & B^T-B^T(1+\nu)C^{-1}C \\ -(1+\nu)C^{-1}B & (1+\nu)C^{-1}C \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & C-(1+\nu)^{-1}C \end{bmatrix}\begin{bmatrix} I & (1+\nu)B^TC^{-1} \\ 0 & (1+\nu)C^{-1} \end{bmatrix}\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \\ &= \begin{bmatrix} A+(1+\nu)B^TC^{-1}B & -\nu B^T \\ -\nu B & \nu C \end{bmatrix}\end{aligned} \tag{4.16}$$

which corresponds to the Bramble-Pasciak-like setting with $C_0=(1+\nu)^{-1}C$ and $A_0=I$. We wish to stress the fact that for the method of Forsgren *et al.* the matrix C is assumed to be definite. Note that the symmetry and definiteness of C implies symmetry and definiteness of C_0.

First, we analyze the Bramble-Pasciak-like method for the case when C is definite and show that it is possible to choose A_0 and C_0 such that $\mathcal{H}_-$ defines an inner product and $\widehat{\mathcal{A}}$ is positive definite within this inner product. This would enable the use of CG for the Bramble-Pasciak-like equivalent of the method introduced [32].

The matrix

$$\mathcal{H}_-=\begin{bmatrix} A_0 & 0 \\ 0 & C-C_0 \end{bmatrix}$$

defines an inner product whenever A_0 is symmetric and positive definite and whenever the symmetric block $C - C_0$ becomes positive definite, i.e. $C - C_0 > 0$ where C is a positive definite matrix. In addition, we need all the eigenvalues of

$$\widehat{\mathcal{A}}^T\mathcal{H}_- = \begin{bmatrix} A + B^TC_0^{-1}B & -B^TC_0^{-1}C + B^T \\ B - CC_0B & CC_0^{-1}C - C \end{bmatrix}$$

to be positive. We use a technique employed in Section 3.4 where we split $\widehat{\mathcal{A}}^T\mathcal{H}_-$ as

$$\widehat{\mathcal{A}}^T\mathcal{H}_- = \begin{bmatrix} I & -B^TC^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A + B^TC^{-1}B & 0 \\ 0 & CC_0^{-1}C - C \end{bmatrix} \begin{bmatrix} I & 0 \\ -C^{-1}B & I \end{bmatrix}.$$

Since this is an congruence transformation, Sylvester's law of inertia gives that we only have to look at the eigenvalues of

$$\begin{bmatrix} A + B^TC^{-1}B & 0 \\ 0 & CC_0^{-1}C - C \end{bmatrix}.$$

Depending on the properties of A, the first block $A + B^TC^{-1}B$ will be positive definite and $CC_0^{-1}C - C$ is positive definite whenever $C_0 < C$. Note that optimality conditions usually imply that $A + B^TC^{-1}B$ should be positive definite. The block $CC_0^{-1}C - C$ is equivalent to $C(C_0^{-1} - C^{-1})C$, which gives that $C_0^{-1} - C^{-1}$ has to be positive definite. Hence, positivity is given whenever $C - C_0$ is a positive definite matrix.

Therefore, we are able to reliably apply the CG method to the linear system. The case given in [32] where $C_0 = (1 + \nu)^{-1}C$ fulfills this criterion if the matrix C is definite.

In Section 3.4, we introduced a preconditioner and bilinear form very similar to the classical Bramble-Pasciak one but with different numerical properties. The main motivation was to have a bilinear form

$$\mathcal{H}_+ = \begin{bmatrix} A_0 & 0 \\ 0 & C + C_0 \end{bmatrix} \tag{4.17}$$

that defines an inner product whenever the preconditioners A_0 and C_0 are positive definite. The preconditioner can also be modified and we get

$$\mathcal{P}_+ = \begin{bmatrix} A_0 & B^T \\ 0 & C_0 \end{bmatrix} \text{ with } \mathcal{P}_+^{-1} = \begin{bmatrix} A_0^{-1} & -A_0^{-1}B^TC_0^{-1} \\ 0 & C_0^{-1} \end{bmatrix}. \tag{4.18}$$

Hence, the preconditioned matrix

$$\widehat{\mathcal{A}} = \mathcal{P}_+^{-1}\mathcal{A} = \begin{bmatrix} A_0^{-1}A - A_0^{-1}B^TC_0^{-1}B & A_0^{-1}B^T + A_0^{-1}B^TC_0^{-1}C \\ C_0^{-1}B & -C_0^{-1}C \end{bmatrix}$$

is self-adjoint in the inner product $\mathcal{H}_+$.

The applicability of CG can be determined by studying the eigenvalues of

$$\widehat{\mathcal{A}}^T\mathcal{H}_+ = \begin{bmatrix} I & -B^TC^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A + B^TC^{-1}B & 0 \\ 0 & -CC_0^{-1}C - C \end{bmatrix} \begin{bmatrix} I & 0 \\ -C^{-1}B & I \end{bmatrix}.$$

Again Sylvester's law of inertia tells us that the eigenvalues of $A + B^TC^{-1}B$ and $-CC_0^{-1}C - C$ will determine the number of positive, negative and zero-eigenvalues of the matrix $\widehat{\mathcal{A}}^T\mathcal{H}$. The block $A + B^TC^{-1}B$ will be positive definite for all C_0 whereas the block $-(CC_0^{-1}C + C)$ will be negative for C_0 being positive definite. Therefore, we cannot reliably apply the CG method in this case. As an alternative, the $\mathcal{H}$-MINRES method given in Section 2.1.2

can always be implemented since an inner product is always at hand due to the definiteness of A_0 and C_0. Another possibility is to use the ITFQMR method introduced in Section 2.2.2.

In the case of the block C being positive semi-definite, e.g. $C = 0$, we can use $\mathcal{H}$-MINRES whenever $\mathcal{H}_\pm$ defines an inner product and ITFQMR whenever $\widehat{\mathcal{A}}^T\mathcal{H}_\pm = \mathcal{H}_\pm\widehat{\mathcal{A}}$ holds.

It should be mentioned here that the preconditioner A_0 in

$$\mathcal{P}_\pm = \begin{bmatrix} A_0 & B^T \\ 0 & \pm C_0 \end{bmatrix}$$

can be chosen such that A_0 resembles the structure given by Forsgren *et al.*, i.e. $A_0 = G + B^T C_0^{-1} B$ which we will call FGG setup. The preconditioner then becomes

$$\mathcal{P}_\pm = \begin{bmatrix} G + B^T C_0^{-1} B & B^T \\ 0 & \pm C_0 \end{bmatrix}$$

which is a block triangular matrix and therefore allows for the efficient solution of linear systems involving $\mathcal{P}_\pm$. In case a factorization of $A_0 = G + B^T C_0^{-1} B$ should be avoided, the preconditioner can be decomposed as

$$\mathcal{P}_- = \begin{bmatrix} G + B^T C_0^{-1} B & B^T \\ 0 & -C_0 \end{bmatrix} = \begin{bmatrix} G & B^T \\ B & -C_0 \end{bmatrix} \begin{bmatrix} I & 0 \\ C_0^{-1} B & I \end{bmatrix}$$

or

$$\mathcal{P}_+ = \begin{bmatrix} G + B^T C_0^{-1} B & B^T \\ 0 & C_0 \end{bmatrix} = \begin{bmatrix} G & B^T \\ -B & C_0 \end{bmatrix} \begin{bmatrix} I & 0 \\ C_0^{-1} B & I \end{bmatrix}.$$

4.4 Preconditioning

4.4.1 C positive definite

The case where C is positive definite can sometimes be found in optimization [32] (as well as other areas [6]) and usually occurs because of some explicit regularization [98]. Optimality conditions imply that $A + B^T C^{-1} B$ should be positive definite. Suppose that we set $C_0 = C$; then the eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ are given by the following theorem:

Theorem 4.5. *Let*

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \quad \textit{and} \quad \mathcal{P} = \begin{bmatrix} A_0 & B^T \\ 0 & -C \end{bmatrix}$$

with nonsingular A_0 and C. Then $\mathcal{P}^{-1}\mathcal{A}$ has

- *m eigenvalues at 1,*
- *the remaining n eigenvalues are defined by the generalized eigenvalue problem*
$$\left(A + B^T C^{-1} B\right) x = \lambda A_0 x.$$

Proof. It is straightforward to show that

$$\mathcal{P}^{-1}\mathcal{A} = \begin{bmatrix} A_0^{-1}\left(A + B^T C^{-1} B\right) & 0 \\ -C^{-1}B & I \end{bmatrix}.$$

Hence, there are m eigenvalues equal to 1 and the remaining eigenvalues satisfy

$$\left(A + B^T C^{-1} B\right) x = \lambda A_0 x.$$

□

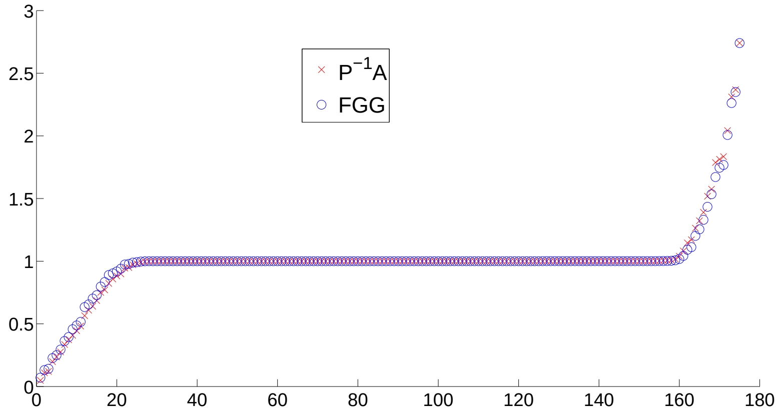

Figure 4.1: Eigenvalue distribution for $\mathcal{P}(\nu)^{-1}\mathcal{A}(\nu)$ and $\mathcal{P}^{-1}\mathcal{A}$, where $C_0 = C$ and $A_0 = G + B^T C^{-1} B$.

Note that if $A_0 = G + B^T C^{-1} B$, then $\mathcal{P}^{-1}\mathcal{A}$ will have the same eigenvalues as $\mathcal{P}(\nu)^{-1}\mathcal{A}(\nu)$, where $\mathcal{A}(\nu)$ and $\mathcal{P}(\nu)$ are defined by equations (4.10) and (4.12), respectively (see Section 4.3.1). We illustrate these results by considering the matrix CVXQP3_S of dimension 175 taken from the CUTEr [55] test set and comparing the eigenvalues of $\mathcal{P}(\nu)^{-1}\mathcal{A}(\nu)$ and $\mathcal{P}^{-1}\mathcal{A}$ (see Figure 4.1). In this example, $C = I$ and $G = diag(A)$.

However, if $C_0 = C$, then $\mathcal{H}_-$ in (4.15) will be singular and, therefore, Algorithms 2.4 and 2.5 may breakdown. Suppose that we instead choose $C_0 = (1+\nu)^{-1}C$, where $\nu \neq -1$. If A_0 is chosen to be a symmetric and positive definite matrix, $\mathcal{H}_-$ will be symmetric and positive definite if and only if $\nu > 0$ or $\nu < -1$. Applying Corollary 4.3 with $\sigma = 1$, $D = 0$, $E = C_0^{-1}$, and $F = 0$, we find that $\mathcal{H}_-\mathcal{P}^{-1}\mathcal{A}$ is positive definite if and only if $A + B^T C^{-1} B$ and $C_0^{-1} - C^{-1}$ are both positive definite. If $C_0 = (1+\nu)^{-1}C$, then $C_0^{-1} - C^{-1}$ is positive definite if and only if $\nu > 0$. This illustrates the result by Forsgren *et al.* that $\mathcal{A}(\nu)$ is positive definite if $A + B^T C^{-1} B > 0$ and $\nu > 0$ (see Section 4.3.1). Theorem 4.6 provides results on the eigenvalues of the resulting matrix $\mathcal{P}^{-1}\mathcal{A}$:

Theorem 4.6. *Let B have rank $r > 0$ and $Z \in \mathbb{R}^{n\times(n-r)}$ be such that its* *columns span the nullspace of B. Additionally, let*

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \quad \textit{and} \quad \mathcal{P} = \begin{bmatrix} A_0 & B^T \\ 0 & -(1+\nu)^{-1}C \end{bmatrix}$$

with nonsingular A_0 and C, where $\nu \neq 0$ and $\nu \neq -1$. Suppose that the generalized eigenvalue problem $Z^T AZx_z = \lambda Z^T A_0 Zx_z$ has j $(0 \leq j \leq n-r)$ eigenvalues equal to $1+\nu$. Then $\mathcal{P}^{-1}\mathcal{A}$ has

- *at least j eigenvalues at $1+\nu$,*
- *the remaining eigenvalues satisfy the quadratic eigenvalue problem (QEP)*

$$(\lambda^2 A_0 - \lambda \left(A + (1+\nu)\left(A_0 + B^T C^{-1} B\right)\right) + (1+\nu)\left(A + B^T C^{-1} B\right))x = 0$$

subject to $\lambda \neq 0$ and $\lambda \neq 1+\nu$.

Proof. Assume that $\left(\lambda, \begin{bmatrix} x^T & y^T \end{bmatrix}^T\right)$ represents an eigenpair of $\mathcal{P}^{-1}\mathcal{A}$. Then

$$\begin{aligned} Ax + B^T y &= \lambda\left(A_0 x + B^T y\right), && (4.19) \\ Bx - Cy &= -\frac{\lambda}{1+\nu} Cy. && (4.20) \\ & && (4.21) \end{aligned}$$

Let $\lambda = 1+\nu$. Equation (4.20) implies that $Bx = 0$. Let $Z \in \mathbb{R}^{n\times(n-r)}$ be such that its columns span the nullspace of B and $Y \in \mathbb{R}^{n\times r}$ be such that its columns span the range of the columns of B^T. If $x = Yx_y + Zx_z$, then $Bx = 0$ implies that $x_y = 0$. Premultiplying (4.19) by $\begin{bmatrix} Y & Z \end{bmatrix}^T$ and substituting in $x = Zx_z$ we obtain

$$\begin{aligned} Y^T AZx_z + (BY)^T y &= (1+\nu)\left(Y^T A_0 Zx_z + (BY)^T y\right), && (4.22) \\ Z^T AZx_z &= (1+\nu) Z^T AZx_z. \end{aligned}$$

Hence, $x_z \neq 0$ if and only if $1+\nu$ is an eigenvalue of the generalized eigenvalue problem $Z^T AZx_z = \lambda Z^T A_0 Zx_z$. Given such an x_z, y can be defined using (4.22).

Let $\lambda \neq 1+\nu$. Equation (4.20) implies that

$$y = \frac{1+\nu}{1+\nu-\lambda} C^{-1}Bx.$$

Substituting this into (4.19) and rearranging the result gives the quadratic eigenvalue problem

$$\lambda^2 A_0 x - \lambda \left(A + (1+\nu)\left(A_0 + B^T C^{-1} B\right)\right) x + (1+\nu)\left(A + B^T C^{-1} B\right) x = 0.$$

This completes the proof.

□

Figure 4.2 shows the eigendistribution for $\mathcal{P}(\nu)^{-1}\mathcal{A}(\nu)$ and $\mathcal{P}^{-1}\mathcal{A}$, where $\nu = 0.1$, $C_0 = (1+\nu)^{-1} C$ and $A_0 = diag(A) + B^T C^{-1} B$. The $2n$ eigenvalue predictions coming from the quadratic eigenvalue problem given in Theorem 4.6 are also plotted. As before, we consider the matrix CVXQP3_S from the CUTEr test set with $C = I$.

In summary, based on the eigenvalue analysis presented here we expect that for a positive definite C the convergence of the Bramble-Pasciak-like method will be similar to the convergence of the algorithm proposed by Forsgren and coauthors. Numerical results are shown in Chapter 6. In contrast to the method given in [32], we could work with a block-triangular preconditioner for the Bramble-Pasciak-like method, which should be favored over solving systems with a constraint preconditioner of the form (4.12). We illustrate the performance of this method by showing numerical results in Chapter 6.

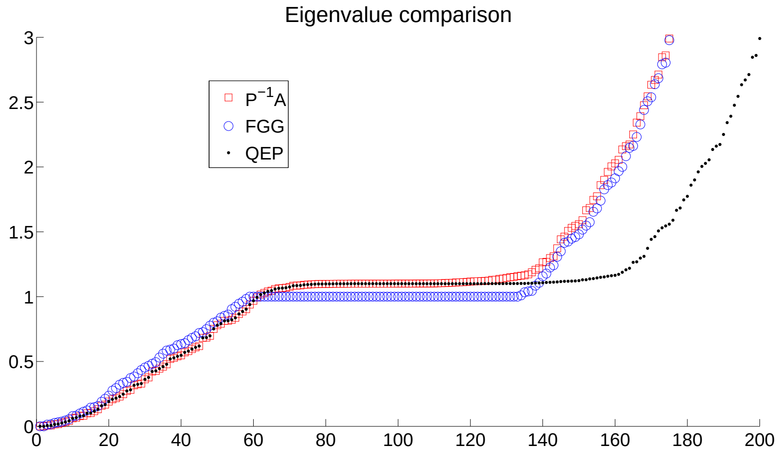

Figure 4.2: Eigenvalue distribution for $\mathcal{P}(\nu)^{-1}\mathcal{A}(\nu)$, $\mathcal{P}^{-1}\mathcal{A}$, and the $2n$ eigenvalues of QEP, see Theorem 4.6, where $\nu = 0.1$, $C_0 = (1+\nu)^{-1} C$ and $A_0 = G + B^T C^{-1} B$.

4.4.2 A positive definite and C positive semi-definite

If A is positive definite, then we may let $A_0 = A$; the analysis presented here is not based on the assumption that C is positive definite. The eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ are defined by Theorem 4.7.

Theorem 4.7. *Let*

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \quad \textit{and} \quad \mathcal{P} = \begin{bmatrix} A & B^T \\ 0 & -C_0 \end{bmatrix}$$

where A is symmetric and non-singular, C is symmetric and positive semi-definite, C_0 is symmetric and (positive or negative) definite, and $C - C_0$ is nonsingular. Then $\mathcal{P}^{-1}\mathcal{A}$ has

- *n eigenvalues at 1,*
- *the remaining m eigenvalues are defined by the generalized eigenvalue problem*

$$\left(C + BA^{-1}B^T\right) y = \lambda C_0 y.$$

Proof. Assume that $\left(\lambda, \begin{bmatrix} x^T & y^T \end{bmatrix}^T\right)$ represents an eigenpair of $\mathcal{P}^{-1}\mathcal{A}$. Then

$$\begin{aligned} Ax + B^T y &= \lambda \left(Ax + B^T y\right) & (4.23) \\ Bx - Cy &= -\lambda C_0 y. & (4.24) \end{aligned}$$

Let $\lambda = 1$. Equation (4.23) trivially holds. Equation (4.24) implies that

$$Bx = (C - C_0)\, y.$$

By assumption, $C - C_0$ is nonsingular and, hence, there are n linearly independent eigenvectors of the form

$$\begin{bmatrix} x \\ (C - C_0)^{-1} Bx \end{bmatrix}$$

associated with $\lambda = 1$.

Let $\lambda \neq 1$. Equation (4.23) implies that $Ax + B^T y = 0$. Therefore, $x = -A^{-1}B^T y$. Substituting this into (4.24) gives the generalized eigenvalue problem

$$\left(C + BA^{-1}B^T\right) y = \lambda C_0 y.$$

This completes the proof.

□

Theorem 4.7 implies that the convergence of the Bramble-Pasciak-like setup (4.14) with $A_0 = A$ is given in at most $m + 1$ steps. If $C + BA^{-1}B^T$ and C_0 are both positive definite, then all of the eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ will be positive, however, C_0 in this case must be chosen such that $C - C_0$ is positive definite in order to guarantee that H is positive definite. If $C + BA^{-1}B^T$ is positive definite and C_0 is negative definite, then $\mathcal{P}^{-1}\mathcal{A}$ will have m negative eigenvalues. Hence, CG with non-standard inner product cannot be applied reliably.

The case of A definite and C semi-definite typically occurs when working with the mixed finite element formulation of the Stokes problem (see Section 1.3 or [24]). Such examples can be easily generated using the IFISS package (cf. [23]). Instead of setting $A_0 = A$, A_0 is generally chosen to be a symmetric and positive definite approximation to A, e.g. an Incomplete Cholesky decomposition or a multigrid cycle [57] and C_0 an approximation to the positive or negative Schur-complement. A very general eigenvalue analysis for the Bramble-Pasciak-like setup is given in Appendix A; the results are only of practical use if a solid knowledge of the eigenvalues of A_0 and C_0 is at hand.

4.4.3 Neither A nor C are positive definite

The case where neither A nor C are positive definite is a more difficult case since we cannot set $A_0 = A$ or $C_0 = (1+\nu)^{-1}C$ and expect to obtain a positive definite matrix H. One remedy is to use a technique introduced by Gill *et al.* in [43] where a modified Bunch-Parlett factorization LDL^T can be used [11]. In more detail, we first compute the Bunch-Parlett factorization of the indefinite block $A = LDL^T$ where the D is a block diagonal matrix with 1×1 or 2×2 blocks on the main diagonal. It might not always be feasible to compute a Bunch-Parlett decomposition, in particular if the problems are very large. Note that we first assume that all of the eigenvalues of A are nonzero. Due to the indefiniteness of A, the elements of D represent inertia in the left and right half plane. As mentioned earlier, we are interested in a positive definite preconditioner for A which then also gives a Schur-complement preconditioner for further convergence improvement. Gill *et al.* [43] obtain a modified factorization $L\hat{D}L^T$ as follows. In the case of a 1×1 block α on the main diagonal with $\alpha < 0$, the sign of α is reversed. In the case of a 2 × 2 block

$$\begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}$$

we compute a Givens rotation such that

$$\begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}.$$

We then change the appropriate block in $\hat{D}$ to be

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} |\lambda_1| & 0 \\ 0 & |\lambda_2| \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

which gives a symmetric positive $\hat{D}$. Therefore, the preconditioner $A_0 = L\hat{D}L^T$ is symmetric and positive definite, and we can also use the Schur-complement approximation $C_0 = C + BA_0^{-1}B^T$.

In the case of A being indefinite with zero-eigenvalues, there are several strategies proposed in the literature (see [28] for an overview). We want to mention the method by Moré and Sorensen [79] where again a Bunch-Parlett factorization is used with D a block diagonal matrix. In more detail, the 1×1 block α is replaced by a modified block $\hat{\alpha}$ such that $\hat{\alpha} = \max(\delta, |\alpha|)$ and for the 2×2 block the spectral decomposition is computed

$$\begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

and then replaced by

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \hat{\lambda}_1 & 0 \\ 0 & \hat{\lambda}_2 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

where $\hat{\lambda}_i = \max(\delta, |\lambda_i|)\,\forall i = 1,2$. The result is a factorization $L\hat{D}L^T$. Moré and Sorensen choose $\delta = \epsilon_M$ with ϵ_M machine epsilon. Another choice for δ is given in [13] by $\delta = \sqrt{\epsilon_M/2}\,\|A\|_\infty$ where Cheng and Higham analyze a method similar to the one presented by Moré and Sorensen.

This setup completes the analysis of the Bramble-Pasciak-like method and emphasizes that it can be used for all practical setups of the saddle point matrix, which is indicated by the numerical results in Chapter 6.

4.4.4 Summary of the methods

Here, we quickly want to summarize the methods derived in this section and also mention the possible competitors. Starting with the setup where A is indefinite and C is positive definite, the methods of choice would be the Bramble-Pasciak-like method with P_- setup and the method of Forsgren *et al.* [32]. Note that if C_0 is a multiple of C the scaling of C_0 to guarantee positivity of $\mathcal{H}_-$ becomes trivial.

The second setup with A being positive definite and C being positive semi-definite can typically be solved with the classical Bramble-Pasciak method. Our setup with P_- is not guaranteed to work with CG but if one is not willing to scale in the classical Bramble-Pasciak method then both configurations can be used with ITFQMR.

The last setup is given when A is indefinite and C is positive semi-definite, for which case we can use MINRES with block-diagonal preconditioning. Furthermore, the Bramble-Pasciak-like method with the $\mathcal{P}^+$ preconditioner is always guaranteed to work with HMINRES. Numerical results for all three cases are given in Chapter 6.

CHAPTER 5

APPROXIMATING THE SCATTERING AMPLITUDE

In this Chapter we describe how to approximate the scattering amplitude introduced in Section 1.3, i.e. the scattering amplitude is $g^T x$ where x is the solution of $Ax = b$ and g is the right hand side of $A^T y = g$. First, we show how the scattering amplitude can be approximated via solving the associated linear systems (1.17) and (1.18). This can be done with a variety of methods, but our focus here is on a general form of the well-known LSQR method, the so-called GLSQR. In the second part, we show that the scattering amplitude can be approximated directly using the connection to Gauss quadrature. The methods proposed in this Chapter are published in [52]. Whereas in the Chapter illustrating combination preconditioning and the Chapter on using the Bramble-Pasciak-like method for optimization problems the adjoint of $\widehat{\mathcal{A}}$ was never explicitly used due to the self-adjointness in a non-standard inner product, we will make use of the adjoint in this Chapter in an explicit way.

5.1 Rewriting the problem and MINRES

In Section 1.3, the scattering amplitude was discussed as well as how it is important to approximate the systems

$$Ax = b \text{ and } A^T y = g$$

at the same time. Note that we slightly change the notation here and use A instead of $\mathcal{A}$ for the system matrix. This is due to the fact that for most parts of the thesis we used $\mathcal{A}$ for matrices in saddle point form. In Section 1.3, we saw that this problem can be reformulated to solve the special system with saddle point structure also given in (1.19)

$$\underbrace{\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} b \\ g \end{bmatrix}.$$

Having the Faber-Manteuffel theorem in mind, it is obvious that for the symmetric matrix

$$\mathcal{A} = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \in \mathbb{R}^{2N,2N}$$

a short-term recurrence methods can be applied. Note that the $2N$ eigenvalues of $\mathcal{A}$ are given by the singular values of A; i.e. N eigenvalues correspond to the singular values σ_i of A and the other N eigenvalues correspond to the N values $-\sigma_i$. For such a symmetric but indefinite system, we can use MINRES, but since the eigenvalues of $\mathcal{A}$ are symmetric about the origin, MINRES will only make progress in every other step. Intuitively, the poor convergence can be motivated by looking at the polynomial approximation that is part of MINRES where we want to find a polynomial that is small on all eigenvalues and has the value 1 at zero. Due to the symmetry of the eigenvalues about the origin, the odd order polynomials cannot benefit the convergence because they will not present a better approximation to the eigenvalues than the previous even order polynomial. The result is the typical staircasing behavior shown in Figure 5.1 where MINRES is applied to a random sparse matrix[1] A of dimension 100×100. In [73] Liesen and Tichý give residual bounds for a setup where the eigenvalues are on both side of the origin in intervals of equal length.

As mentioned earlier, an iterative scheme is not of great use in practice if preconditioning cannot be embedded unless the matrix is well conditioned. Therefore, we need to find a preconditioner that enables the use of MINRES and enhances the convergence properties of the system (1.19). Hence, we could look at

$$\begin{bmatrix} M_1^{-1} & 0 \\ 0 & M_2^{-T} \end{bmatrix} \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} M_1^{-T} & 0 \\ 0 & M_2^{-1} \end{bmatrix} = \begin{bmatrix} 0 & M_1^{-1} A M_2^{-1} \\ M_2^{-T} A^T M_1^{-T} & 0 \end{bmatrix} \tag{5.1}$$

with

$$\mathcal{P} = \begin{bmatrix} M_1 M_1^T & 0 \\ 0 & M_2 M_2^T \end{bmatrix}. \tag{5.2}$$

The preconditioner $\mathcal{P}$ given in (5.2) is symmetric and positive definite and hence MINRES can be used with this preconditioner. Again, the eigenvalues of the matrix

$$\begin{bmatrix} 0 & M_1^{-1} A M_2^{-1} \\ M_2^{-T} A^T M_1^{-T} & 0 \end{bmatrix}$$

[1] Creates a sparse randomly perturbed matrix where the random entries are normally distributed. The MATLAB command is *A=sprandn(n,n,0.2)+speye(n);*

are symmetric about the origin, which means that MINRES can only make progress every other step. We will therefore present different approaches in the remainder of this chapter.

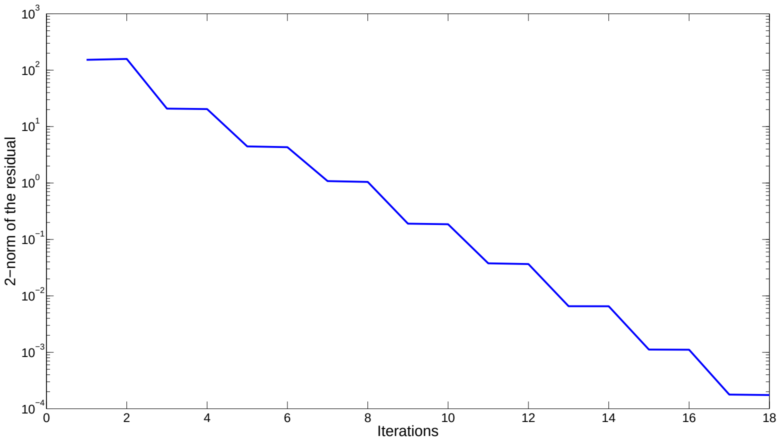

Figure 5.1: Staircasing for MINRES applied to (1.19)

5.2 Solving the linear systems

5.2.1 The QMR approach

In [74], Lu and Darmofal present a technique using the standard QMR method given in Section 2.2.2 to obtain an algorithm that would approximate the solution of the forward and the adjoint problem at the same time. As seen in Section 2.2.2, QMR is based on the non-symmetric Lanczos process, i.e.

$$\begin{aligned} AV_k &= V_{k+1}T_{k+1,k} \\ A^TW_k &= W_{k+1}\hat{T}_{k+1,k}, \end{aligned}$$

where $T_{k+1,k}$ and $\hat{T}_{k+1,k}$ are tridiagonal. The idea presented by Lu and Darmofal is to choose $v_1 = r_0/\left\|r_0\right\|$ and $w_1 = s_0/\left\|s_0\right\|_2$, where $s_0 = g - A^Ty_0$

and $y_k = y_0 + W_k d_k$, to obtain the norm of the adjoint quasi-residual

$$\left\| s_k^Q \right\|_2 = \left\| \left\| s_0 \right\|_2 e_1 - \hat{T}_{k+1,k} d_k \right\|_2 ,$$

in a similar fashion to the forward quasi-residual $\left\| r_k^Q \right\|_2$ given in Section 2.2.2. Again, the least-squares solutions for both quasi-residuals can be obtained via updated QR factorizations (see [84, 37] for details). It is also possible to introduce weights to improve the convergence behavior, though it is not always clear how these weights should be chosen [37].

5.2.2 The bidiagonalization or LSQR approach

We stressed earlier that the system matrix of (1.19)

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$$

is symmetric and indefinite. Furthermore, it is used when computing singular values of the matrix A (see MATLAB's *svds* command) and is also very important in the context of linear least squares problems. The main tool used for either purpose is the Golub-Kahan bidiagonalization (cf. [45]) which is also the basis for the well-known LSQR method introduced by Paige and Saunders in [86, 85]. LSQR is not necessarily considered a solver for linear systems and hence we introduce this method here and not in Chapter 2.

In more detail, we assume that the bidiagonal factorization

$$A = UBV^T \tag{5.3}$$

is given, where U and V are orthogonal and B is bidiagonal. Hence, we can express forward and adjoint systems as

$$UBV^Tx = b$$

and

$$VB^TU^Ty = g.$$

So far we have assumed that an explicit bidiagonal factorization (5.3) is given which is a rather unrealistic assumption for large sparse matrices. In practice, we need an iterative procedure that represents instances of the bidiagonalization process (cf. [53,45,86,85]). This is done using the following matrix relationships

$$\begin{array}{rcl} AV_k & = & U_{k+1}B_k \\ A^TU_{k+1} & = & V_kB_k^T + \alpha_{k+1}v_{k+1}e_{k+1}^T \end{array} \tag{5.4}$$

where $V_k = [v_1, \ldots, v_k]$ and $U_k = [u_1, \ldots, u_k]$ are orthogonal matrices and

$$B_k = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}.$$

The Golub-Kahan bidiagonalization is nothing than the Lanczos process applied to the matrix A^TA; i.e. we multiply the first part of (5.4) by A^T on the left and then use the second part to get the Lanczos relation for A^TA

$$A^TAV_k = A^TU_{k+1}B_k = \left(V_kB_k^T + \alpha_{k+1}v_{k+1}e_{k+1}^T\right)B_k = V_kB_k^TB_k + \hat{\alpha}_{k+1}v_{k+1}e_{k+1}^T$$

with $\hat{\alpha}_{k+1} = \alpha_{k+1}\beta_{k+1}$ (see [8,61] for details). Note that LSQR is algebraically equivalent to CG for the normal equations (CGNE) but with better numerical properties (see [86, 85] for details). Note that the convergence of CGNE is governed by the singular values of the matrix $\mathcal{A}$ [81]. The initial vectors of both sequences v_j and u_j are linked by the relationship

$$A^T u_1 = \alpha_1 v_1. \tag{5.5}$$

We now use the iterative process described in (5.4) to obtain approximations to the solutions of the forward and the adjoint problem. The residuals at step k can be defined as

$$r_k = b - Ax_k \tag{5.6}$$

and

$$s_k = g - A^T y_k \tag{5.7}$$

with

$$x_k = x_0 + V_k z_k$$

and

$$y_k = y_0 + U_{k+1} w_k.$$

A typical choice for u_1 would be the normalized initial residual $u_1 = r_0 / \|r_0\|$. Hence, we get for the residual norms that

$$\begin{aligned} \|r_k\|_2 &= \|b - Ax_k\|_2 \\ &= \|b - A(x_0 + V_k z_k)\|_2 \\ &= \|r_0 - AV_k z_k\|_2 \\ &= \|r_0 - U_{k+1} B_k z_k\|_2 \\ &= \| \|r_0\|_2 e_1 - B_k z_k\|_2 \end{aligned} \tag{5.8}$$

using (5.4) and the orthogonality of U_{k+1}. The adjoint residual can now be expressed as

$$\begin{aligned} \|s_k\|_2 &= \left\|g - A^T y_k\right\|_2 \\ &= \left\|g - A^T(y_0 + U_{k+1}w_k)\right\|_2 \\ &= \left\|g - A^T y_0 - A^T U_{k+1} w_k\right\|_2 \\ &= \left\|s_0 - V_k B_k^T w_k - \alpha_{k+1} v_{k+1} e_{k+1}^T w_k\right\|_2 . \end{aligned} \tag{5.9}$$

Notice that (5.9) cannot be simplified to the desired structure

$$\left\|\|s_0\|_2 \, e_1 - B_k^T w_k\right\|_2$$

since the initial adjoint residual s_0 is not in the span of the current and all the following v_j-s.

The classical LSQR [86] method is an algorithm created to obtain an approximation that minimizes only the residual for the forward problem $\|r_k\|_2 = \|b - Ax_k\|_2$. The method is very successful and widely used in practice but is limited due to the restriction given by (5.5) in the case of simultaneous iteration for the adjoint problem. In more detail, we are not able to choose the second starting vector independently and therefore cannot obtain the desired least squares structure for the adjoint problem as the one obtained for the forward residual. Figure 5.2 illustrates the behavior we could observe for all our examples with the LSQR method. Here, we are working with a random matrix[2] of dimension 100×100. Convergence for the forward solution could be observed when a large number of iteration steps was executed, whereas the convergence for the adjoint residual could not be achieved at any point which is illustrated by the stagnation of the adjoint solution. As already mentioned, this is due to the coupling of the starting vectors and hence the sequence v_j does not have any information about the right hand side of the adjoint equation. In the next section, we present a different approach that overcomes this drawback.

[2] A matrix with random normally distributed entries MATLAB command *randn*

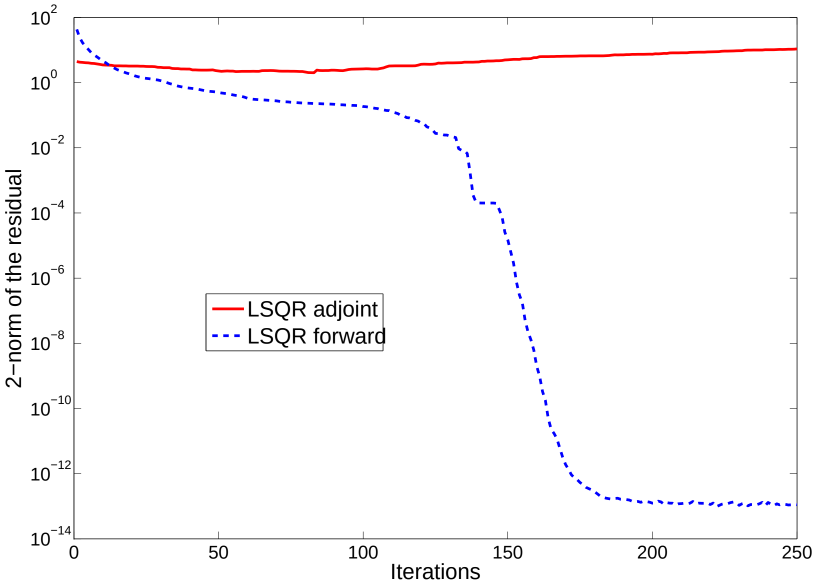

Figure 5.2: Solving a linear system of dimension 100 × 100 with the LSQR approach.

5.2.3 Generalized LSQR (GLSQR)

The simultaneous computation of forward and adjoint solutions based on the classical LSQR method is not very successful in the context of solving both problems together since the starting vectors u_1 and v_1 depend on each other through (5.5). In [99] Saunders *et al.* introduced a more general LSQR method that was also recently analyzed by Reichel and Ye (see [91]). Saunders and coauthors also mention in their paper that the method presented can be used to solve forward and adjoint problem at the same time. We discuss this here in more detail. We present further analysis of the method described in [99, 91]. The method of interest makes it possible to choose the starting vectors u_1 and v_1 independently, e.g. $u_1 = r_0/\left\|r_0\right\|_2$ and $v_1 = s_0/\left\|s_0\right\|_2$. The algorithm stated in [99, 91] is based on the following factorization

$$\begin{aligned} AV_k &= U_{k+1}T_{k+1,k} &= U_kT_{k,k} + \beta_{k+1}u_{k+1}e_k^T \\ A^TU_k &= V_{k+1}S_{k+1,k} &= V_kS_{k,k} + \eta_{k+1}v_{k+1}e_k^T \end{aligned} \tag{5.10}$$

where

$$V_k = [v_1, \ldots, v_k]$$

and

$$U_k = [u_1, \ldots, u_k]$$

are orthogonal matrices and

$$T_{k+1,k} = \begin{bmatrix} \alpha_1 & \gamma_1 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{k-1} \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}$$

as well as

$$S_{k+1,k} = \begin{bmatrix} \delta_1 & \theta_1 & & \\ \eta_2 & \delta_2 & \ddots & \\ & \ddots & \ddots & \theta_{k-1} \\ & & \eta_k & \delta_k \\ & & & \eta_{k+1} \end{bmatrix}.$$

In the case of no *breakdown*[3], the following relation holds

$$S_{k,k}^T = T_{k,k}.$$

[3] We discuss breakdowns later in this section.

The matrix factorization given in (5.10) can be used to produce simple algorithmic statements of how to obtain new iterates for u_j and v_j:

$$\begin{array}{rcl} \beta_{k+1}u_{k+1} & = & Av_k - \alpha_k u_k - \gamma_{k-1}u_{k-1} \\ \eta_{k+1}v_{k+1} & = & A^T u_k - \delta_k v_k - \theta_{k-1}v_{k-1} \end{array}. \tag{5.11}$$

The parameters $\alpha_k, \gamma_{k-1}, \delta_k, \theta_{k-1}$ can be determined via the Gram-Schmidt orthogonalization process in the classical version given by

$$\alpha_k = \langle Av_k, u_k \rangle, \gamma_{k-1} = \langle Av_k, u_{k-1} \rangle, \delta_k = \langle A^T u_k, v_k \rangle \text{ and } \theta_{k-1} = \langle A^T u_k, v_{k-1} \rangle.$$

It is also possible to employ a modified Gram-Schmidt process. Furthermore, β_{k+1} and η_{k+1} are determined from the normalization of the vectors in (5.11).

Since it is well understood that the classical Golub-Kahan bidiagonalization process introduced in [45] can be viewed as the Lanczos algorithm applied to the matrix A^TA, we want to analyze whether a similar connection can be made for the GLSQR method given in [99, 91]. Note that if the Lanczos process is applied to the matrix

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$$

with starting vector $[u_1, 0]^T$ we get equivalence to the Golub-Kahan bidiagonalization (see [8, 61] for details).

The generalized LSQR method (GLSQR) given in [99, 91] looks very similar to the Lanczos process applied to the matrix

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$$

and we will now show that in general GLSQR can not be seen as a Lanczos process applied to this matrix. The Lanczos iteration then gives

$$\nu_{k+1}\begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}\begin{bmatrix} u_k \\ v_k \end{bmatrix} - \xi_k\begin{bmatrix} u_k \\ v_k \end{bmatrix} - \varrho_{k-1}\begin{bmatrix} u_{k-1} \\ v_{k-1} \end{bmatrix} \tag{5.12}$$

and the resulting recursions are then

$$\begin{aligned} \nu_{k+1}u_{k+1} &= Av_k - \xi_k u_k - \varrho_{k-1}u_{k-1} \\ \nu_{k+1}v_{k+1} &= A^T u_k - \xi_k v_k - \varrho_{k-1}v_{k-1}. \end{aligned} \tag{5.13}$$

The parameters ϱ_{k-1}, ξ_k and ν_{k+1} are related to the parameters from the GLSQR process via

$$\xi_k = u_k^T A v_k + v_k^T A^T u_k = \alpha_k + \delta_k$$

$$\varrho_{k-1} = u_{k-1}^T A v_k + v_{k-1}^T A^T u_k = \gamma_{k-1} + \eta_{k-1}$$

and since the Lanczos process generates a symmetric tridiagonal matrix we also get

$$\nu_{k+1} = \varrho_k = \gamma_k + \eta_k.$$

The orthogonality condition imposed by the symmetric Lanczos process ensures that

$$\begin{bmatrix} u_{k+1}^T & v_{k+1}^T \end{bmatrix}\begin{bmatrix} u_k \\ v_k \end{bmatrix} = 0$$

which reduces to $u_{k+1}^T u_k + v_{k+1}^T v_k = 0$. This criteria would be fulfilled by the vectors coming from the GLSQR method because it creates two sequences of orthonormal vectors. In general, the vectors coming from the symmetric Lanczos process do not satisfy $u_{k+1}^T u_k = 0$ and $v_{k+1}^T v_k = 0$.

In the following, we study the similarity of GLSQR and a special Block-

Lanczos method. In [99], a connection to a Block-Lanczos for the matrix A^TA was made. Here we will discuss a method based on

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}.$$

Hence, we assume the complete matrix decompositions

$$AV = UT \qquad \text{and} \qquad A^TU = VT^*$$

with $S = T^*$ are given. Using this we can rewrite the linear system (1.19) as

$$\begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \begin{bmatrix} 0 & T \\ T^* & 0 \end{bmatrix} \begin{bmatrix} U^T & 0 \\ 0 & V^T \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} b \\ g \end{bmatrix}. \tag{5.14}$$

We now introduce the perfect shuffle permutation

$$\Pi = [e_1, e_3, \ldots, e_2, e_4, \ldots] \tag{5.15}$$

and use Π to modify (5.14) which becomes

$$\begin{bmatrix} U & \\ & V \end{bmatrix} \Pi^T\Pi \begin{bmatrix} 0 & T \\ T^* & 0 \end{bmatrix} \Pi^T\Pi \begin{bmatrix} U^T & 0 \\ 0 & V^T \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} b \\ g \end{bmatrix}. \tag{5.16}$$

We now further analyze the matrices given in (5.16). The first two matrices can also be written as

$$\left[\begin{array}{ccc|ccc} | & | & | & | & | & | \\ u_1 & u_2 & \vdots & 0 & 0 & 0 \\ | & | & | & | & | & | \\ \hline | & | & | & | & | & | \\ 0 & 0 & 0 & v_1 & v_2 & \vdots \\ | & | & | & | & | & | \end{array}\right]\Pi^T = \left[\begin{array}{cccccc} | & | & | & | & | & | \\ u_1 & 0 & u_2 & 0 & \vdots & \vdots \\ | & | & | & | & | & | \\ \hline | & | & | & | & | & | \\ 0 & v_1 & 0 & v_2 & \vdots & \vdots \\ | & | & | & | & | & | \end{array}\right] = \mathcal{U}.$$

Next, we are going to study the similarity transformation on

$$\begin{bmatrix} 0 & T \\ T^* & 0 \end{bmatrix}$$

using Π which results in

$$\mathcal{T} = \Pi \begin{bmatrix} 0 & T \\ T^* & 0 \end{bmatrix} \Pi^T = \begin{bmatrix} \Theta_1 & \Psi_1^T & & \\ \Psi_1 & \Theta_2 & \Psi_2^T & \\ & \Psi_2 & \ddots & \ddots \\ & & \ddots & \ddots \end{bmatrix} \tag{5.17}$$

with

$$\Theta_i = \begin{bmatrix} 0 & \alpha_i \\ \alpha_i & 0 \end{bmatrix} \text{ and } \Psi_i = \begin{bmatrix} 0 & \beta_{i+1} \\ \gamma_i & 0 \end{bmatrix}.$$

It is easy to see using the properties of the Reichel and Ye LSQR method that the matrix $\mathcal{U}$ is an orthogonal matrix and furthermore that if we write $\mathcal{U} = [\mathcal{U}_1, \mathcal{U}_2, \cdots]$ where

$$\mathcal{U}_i = \left[\begin{array}{cc} | & | \\ u_i & 0 \\ | & | \\ \hline | & | \\ 0 & v_i \\ | & | \end{array}\right]$$

that $\mathcal{U}_i^T \mathcal{U}_i = I$ for all i. Thus, one particular instance at step k of the reformulated Block-method reduces to

$$\mathcal{U}_{k+1}\Psi_{k+1} = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \mathcal{U}_k - \mathcal{U}_k\Theta_k - \mathcal{U}_{k-1}\Psi_{k-1}^T.$$

Hence, we have shown that the GLSQR method can be viewed as a special Block-Lanczos method with stepsize 2 (see [53, 76] for more details on Block-methods).

5.2.4 GLSQR and linear systems

The GLSQR process analyzed above can be used to obtain approximate solutions to the linear and the adjoint systems. With GLSQR, we are now able to set u_1 and v_1 independently and choose for initial guesses x_0, y_0 with residuals $r_0 = b - Ax_0$, $s_0 = g - A^T y_0$

$$u_1 = \frac{r_0}{\|r_0\|_2}$$

and

$$v_1 = \frac{s_0}{\|s_0\|_2}.$$

Hence, our approximations for the solution at each step are given by

$$x_k = x_0 + V_k z_k \tag{5.18}$$

for the forward problem and

$$y_k = y_0 + U_k w_k \tag{5.19}$$

for the linear system involving the adjoint. Using this and (5.10) we can express the residual at step k as follows: for the forward problem

$$\begin{aligned} \|r_k\|_2 &= \|b - Ax_k\|_2 \\ &= \|b - A(x_0 + V_k z_k)\|_2 \\ &= \|r_0 - AV_k z_k\|_2 \\ &= \|r_0 - U_{k+1} T_{k+1,k} z_k\|_2 \\ &= \left\|U_{k+1}^T r_0 - T_{k+1,k} z_k\right\|_2 \\ &= \left\|\|r_0\|_2 e_1 - T_{k+1,k} z_k\right\|_2 \end{aligned} \tag{5.20}$$

and in complete analogy

$$\begin{aligned} \|s_k\|_2 &= \left\|g - A^T y_k\right\|_2 \\ &= \left\|V_{k+1}^T s_0 - S_{k+1,k} w_k\right\|_2 \\ &= \left\|\|s_0\|_2 e_1 - S_{k+1,k} w_k\right\|_2 . \end{aligned} \tag{5.21}$$

The solutions z_k and w_k can be obtained by solving the least squares systems (5.20) and (5.21) respectively. We established earlier that such a least squares system can be solved using the updated QR factorization (see also [84]). Hence, we have to compute two Givens rotations at every step to solve the systems (5.20) and (5.21) efficiently. Remembering Section 2.1.2, there is no need to store the whole basis V_k or U_k in order to update the solution as described in (5.18) and (5.19).

The storage requirements for the GLSQR method are similar to the storage requirements for a method based on the non-symmetric Lanczos process as proposed by Lu and Darmofal in [74]. We need to store the vectors u_j, v_j, u_{j-1} and v_{j-1} to generate the basis vectors for the next Krylov space. Furthermore, we need to store the sparse matrices $T_{k+1,k}$ and $S_{k+1,k}$. This can be done in a parameterized fashion–remember that they are tridiagonal matrices– and since, until the first breakdown occurs, $T_{k,k} = S_{k,k}^T$ holds the storage requirement can be reduced even further. The triangular factors of $T_{k+1,k}$ and $S_{k+1,k}$ can also be stored very efficiently since they only have three nonzero diagonals. According to (2.6) the solutions x_k and y_k can be updated with only storing two vectors c_{k-2} and c_{k-3} for the forward problem and another two vectors for the adjoint solution. Thus the solutions can be obtained by storing only a minimal amount of data in addition to the original problem.

In [91], Reichel and Ye solve the forward problem and introduce the term *breakdown* in the case that the matrix $S_{k+1,k}$ associated with the adjoint problem has a zero entry on the subdiagonal. Note that until a breakdown occurs it is not necessary to distinguish between the parameters of the forward and adjoint sequence, since $T_{k,k} = S_{k,k}^T$. We will discuss these breakdowns and show that they are indeed *lucky breakdowns* which means that the solution can be found in the current space. When the breakdown occurs, we assume that the parameter $\beta_{k+1} = 0$ whereas $\eta_{k+1} \neq 0$, in which case Reichel and Ye proved in Theorem 2.2 that the solution x_k for the forward problem can be obtained via $x_k = x_0 + \|r_0\|_2 V_k T_{k,k}^{-1} e_1$. The same holds if $\beta_{k+1} \neq 0$ whereas $\eta_{k+1} = 0$ in which case the solution y_k can be obtained via $y_k = y_0 + \|s_0\|_2 U_k S_{k,k}^{-1} e_1$. Note that this is in contrast to the breakdowns that can occur in the non-symmetric Lanczos process.

In both cases, we have to continue the algorithm since only the solution to one of the two problems is found. Without loss of generality, we assume that $\beta_{k+1} = 0$ whereas $\eta_{k+1} \neq 0$ which means that the forward problem has already been solved. A strategy implicitly proposed by Reichel and Ye is to compute u_{k+1} using

$$\beta_{k+1}u_{k+1} = 0 = Av_k - \alpha_k u_k - \gamma_{k-1}u_{k-1}$$

which gives

$$\alpha_{k+1}u_{k+1} = Av_{k+1} - \gamma_k u_k.$$

In terms of matrices this would result in an upper bidiagonal part of $T_{k+1,k}$ from the iteration where the breakdown occurred. There is no need to update the solution x_k in further steps of the method. The vectors u_{k+1} generated by this two-term recurrence are used to update the solution for the adjoint problem in a way we will now describe. First, we obtain a new basis vector v_{j+1}

$$\eta_{j+1}v_{j+1} = A^T u_j - \delta_j v_j - \theta_{j-1}v_{j-1}$$

and then update the QR factorization of $S_{k+1,k}$ to get a new iterate y_k. If the parameter $\eta_{j+1} = 0$, the solution for the adjoint problem is found and the method can be terminated. In the case of the parameter α_{k+1} becoming zero the solution for the adjoint problem can be obtained using the following Theorem which stands in complete analogy to Theorem 2.3 in [91].

Theorem 5.1. *We assume that* GLSQR *does not break down until step m of the algorithm. At step m we get $\beta_{m+1} = 0$ and $\eta_{m+1} \neq 0$, which corresponds to the forward problem being solved. The process is continued with the update*

$$\alpha_{k+1}u_{k+1} = Av_{k+1} - \gamma_k u_k.$$

The solution of the adjoint problem can now be obtained from one of the following two cases if the breakdown occurs at step k

1. *If the parameter $\eta_{k+1} = 0$ then the adjoint solution is given by $y_k = y_0 + \|s_0\|_2 U_k S_{k,k}^{-1} e_1$.*

2. *If the parameter $\alpha_{k+1} = 0$ and $\eta_{k+1} \neq 0$ then the adjoint problem can be recovered using $y_k = y_0 + U_k w_k$.*

Proof. The proof of the first point is trivial since for $\eta_{k+1} = 0$ the least squares error in

$$\min_{w \in \mathbb{R}^k} \left\| \left\| r_0 \right\|_2 e_1 - S_{k+1,k} w_k \right\|_2$$

is equal to zero. For the second point we not that the solution w_k to the least squares problem

$$\min_{w \in \mathbb{R}^k} \left\| \left\| r_0 \right\|_2 e_1 - S_{k+1,k} w_k \right\|_2$$

satisfies the following relation

$$S_{k+1,k}^T \left(\left\| r_0 \right\|_2 e_1 - S_{k+1,k} w_k \right) = 0. \tag{5.22}$$

The breakdown with $\alpha_{k+1} = 0$ results in

$$\alpha_{k+1} u_{k+1} = 0 = A v_{k+1} - \gamma_k u_k$$

which means that no new u_{k+1} is generated in this step. In matrix terms we get

$$A V_{k+1} = U_k T_{k,k+1}$$

and

$$A^T U_k = V_{k+1} S_{k+1,k}.$$

This results in,

$$\begin{aligned}
A(g - A^T y) &= A(s_0 - A^T U_k w_k) \\
&= A(s_0 - V_{k+1} S_{k+1,k} w_k) \\
&= As_0 - AV_{k+1} S_{k+1,k} w_k \\
&= \|s_0\|_2 AV_{k+1} e_1 - AV_{k+1} S_{k+1,k} w_k \\
&= \|s_0\|_2 U_k T_{k,k+1} e_1 - U_k T_{k,k+1} S_{k+1,k} w_k \\
&= U_k T_{k,k+1} \left(\|s_0\|_2 e_1 - S_{k+1,k} w_k \right) \\
&= U_k S_{k+1,k}^T \left(\|s_0\|_2 e_1 - S_{k+1,k} w_k \right) \\
&= 0
\end{aligned}$$

using the fact that $S_{k+1,k}^T = T_{k,k+1}$ (see Theorem 2.1 in [91]). Due to the assumption that A is nonsingular, the solution for the adjoint problem is given by $y_k = y_0 + U_k w_k$. □

This shows that the GLSQR method is well-suited to find the solution of forward and adjoint problem at the same time. The breakdowns that occur in the process of the algorithm are all benign breakdowns which underlines the difference from methods–such as BICG or QMR– based on the non-symmetric Lanczos process. In order to give better reliability of the method based on the non-symmetric Lanczos process, look-ahead strategies have to be implemented (cf. [36,89]), though there can still be incurable breakdowns.

5.2.5 Preconditioned GLSQR

In practice, the GLSQR method can show slow convergence and therefore has to be enhanced using preconditioning techniques. The convergence of GLSQR has not yet been analyzed, but we feel that using the connection to the Block-Lanczos process for $\mathcal{A}^T\mathcal{A}$ [99] we can try to look for similarities to the convergence of CG for the normal equations (CGNE). It is well known [81] that the convergence of CGNE is governed by the singular values of the matrix $\mathcal{A}$. This fact will be illustrated in Chapter 6. We assume the preconditioner $M = M_1 M_2$ is given. Note that in general $M_1 \neq M_2$. The preconditioned matrix is now

$$\widehat{A} = M_1^{-1} A M_2^{-1},$$

and its corresponding transpose is given by

$$\widehat{A}^T = M_2^{-T} A^T M_1^{-T}.$$

Since we do not want to compute the matrix $\widehat{A}$, we have to rewrite the GLSQR method

$$\begin{array}{rcl} \beta_{j+1} u_{j+1} & = & M_1^{-1} A M_2^{-1} v_j - \alpha_j u_j - \gamma_{j-1} u_{j-1} \\ \eta_{j+1} v_{j+1} & = & M_2^{-T} A^T M_1^{-T} u_j - \delta_j v_j - \theta_{j-1} v_{j-1} \end{array} \tag{5.23}$$

to obtain an efficient implementation of the preconditioned procedure, i.e.

$$\begin{array}{rcl} \beta_{j+1} M_1 u_{j+1} & = & A M_2^{-1} v_j - \alpha_j M_1 u_j - \gamma_{j-1} M_1 u_{j-1} \\ \eta_{j+1} M_2^T v_{j+1} & = & A^T M_1^{-T} u_j - \delta_j M_2^T v_j - \theta_{j-1} M_2^T v_{j-1}. \end{array} \tag{5.24}$$

If we set $p_j = M_1 u_j$, $M_2 \hat{q}_j = v_j$, $q_j = M_2^T v_j$ and $M_1^T \hat{p}_j = u_j$ we get

$$\begin{array}{rcl} \beta_{j+1} p_{j+1} & = & A \hat{q}_j - \alpha_j p_j - \gamma_{j-1} p_{j-1} \\ \eta_{j+1} q_{j+1} & = & A^T \hat{p}_j - \delta_j q_j - \theta_{j-1} q_{j-1} \end{array} \tag{5.25}$$

with the following updates

$$\hat{q}_j = M_2^{-1} v_j = M_2^{-1} M_2^{-T} q_j \tag{5.26}$$

and

$$\hat{p}_j = M_1^{-T} u_j = M_1^{-T} M_1^{-1} p_j. \tag{5.27}$$

We also want to compute the parameters α_j, γ_{j-1}, δ_j and θ_{j-1} The parameters α_j, γ_{j-1}, δ_j and θ_{j-1} can also be expressed in terms of the vectors $\hat{p}_j$, $\hat{q}_j$, p_j and q_j. Namely, we get

$$\begin{array}{lclcl}
\alpha_j & = & \langle \widehat{A} v_j, u_j \rangle & = & \langle A\hat{q}_j, \hat{p}_j \rangle \\
\gamma_{j-1} & = & \langle \widehat{A} v_j, u_{j-1} \rangle & = & \langle A\hat{q}_j, \hat{p}_{j-1} \rangle \\
\delta_j & = & \langle \widehat{A}^T u_j, v_j \rangle & = & \langle A^T\hat{p}_j, \hat{q}_j \rangle \\
\theta_{j-1} & = & \langle \widehat{A}^T u_j, v_{j-1} \rangle & = & \langle A^T\hat{p}_j, \hat{q}_{j-1} \rangle
\end{array}$$

which can be computed cheaply. Note that we need to evaluate $A^T\hat{p}_j$ and $A\hat{q}_j$ once in every iteration step. The parameters β_{j+1} and η_{j+1} can be computed using Equations (5.26) and (5.27) (see Algorithm 5.1 for a summary of this method).

```
for $k = 0, 1, \dots$ do
    Solve $(M_2^T M_2)\hat{q}_j = q_j$
    Solve $(M_1 M_1^T)\hat{p}_j = p_j$
    Compute $A\hat{q}_j$.
    Compute $\alpha_j = \langle A\hat{q}_j, \hat{p}_j \rangle$ and $\gamma_{j-1} = \langle A\hat{q}_j, \hat{p}_{j-1} \rangle$.
    Compute $\beta_{j+1}$ and $p_{j+1}$ via $\beta_{j+1} p_{j+1} = A\hat{q}_j - \alpha_j p_j - \gamma_{j-1} p_{j-1}$
    Compute $A^T\hat{p}_j$
    Compute $\delta_j = \langle A^T\hat{p}_j, \hat{q}_j \rangle$ and $\theta_{j-1} = \langle A^T\hat{p}_j, \hat{q}_{j-1} \rangle$.
    Compute $\eta_{j+1}$ and $q_{j+1}$ via $\eta_{j+1} q_{j+1} = A^T\hat{p}_j - \delta_j q_j - \theta_{j-1} q_{j-1}$
end for
```

Algorithm 5.1: Preconditioned GLSQR

The above formulae enable us to compute the matrices $T_{k+1,k}$ and $S_{k+1,k}$ efficiently. As for the other methods, we can update the QR factorizations in every step using one Givens rotation for the forward problem and one Givens rotation for the adjoint problem. The solutions x_k and y_k can then be updated without storing the whole Krylov space but with a recursion similar to (2.7). The norm of the preconditioned residual $\hat{r}_k$ can be computed via the well known recursion [84]

$$\|\hat{r}_k\|_2 = |sin(\theta_k)| \, \|\hat{r}_{k-1}\|_2$$

where $sin(\theta_k)$ is associated with the Givens rotation at step k for the forward problem. The adjoint residual can be updated similarly.

There are different preconditioning strategies for enhancing the spectral properties of A to make the GLSQR method converge faster. One possibility would be to use an Incomplete LU factorization of A and then set $M_1 = L$ and $M_2 = U$ (see [96] for more details).

Another technique is to use the fact that the GLSQR method is also a Block-Lanczos method for the normal equations [99]; i.e. the system matrix that has to be preconditioned is now A^TA. We therefore consider preconditioning techniques that are well suited for the normal equations.

One possibility would be to compute an Incomplete Cholesky factorization of A^TA, but since the matrix A^TA is typically less sparse than A and we never want to form the matrix A^TA explicitly, we consider preconditioners coming from an LQ decomposition of $A = LQ$. In [96], Incomplete LQ preconditioners based on Incomplete Gram-Schmidt factorizations such as IMGS are discussed and used as a preconditioner to solve the system with AA^T. This strategy can be adopted when trying to find a solution to a system with A^TA.

Another approach is based on Incomplete Orthogonal factorizations, i.e. $A = QR + E$ with Q orthogonal, R is a sparse upper triangular matrix and E is the error term. There are different variants of this decomposition [4, 87] which result in a different structure of the matrix R. In the simple case of the so-called cIGO (column-Incomplete Givens Orthogonalization) method where entries are only dropped based upon their position, we restrict R to have the same sparsity pattern as the original matrix A. We now use Q and R from the incomplete factorization and set $M_1 = Q$ and $M_2 = R$ which gives $\widehat{A} = Q^TAR^{-1}$ for the normal equations $\widehat{A}^T\widehat{A} = R^{-T}A^TQQ^TAR^{-1} =$

$R^{-T}A^TAR^{-1}$. Hence, we can use R as a preconditioner for the normal equations and therefore for the GLSQR method; i.e. we have an incomplete Cholesky factorization for A^TA via an incomplete orthogonal factorization for A. The same holds for the incomplete LQ factorization and the Q factor can be omitted for preconditioning.

5.3 Approximating the scattering amplitude

In the following section, we discuss how to approximate the scattering amplitude without computing a solution to the linear system. The principle reason for this approach rather than computing x_k and then the inner product of g with x_k relates to numerical stability: the analysis in section 10 of [110] for Hermitian systems and the related explanation in [112] for non-Hermitian systems shows that approach to be sensitive in finite precision arithmetic, whereas our approach based on Gauss quadrature is more reliable. In [112] Strakoš and Tichý give a survey of methods that can be used to approximate the scattering amplitude directly.

5.3.1 Matrices, Moments and Quadrature: An Introduction

In [49, 50], Golub and Meurant show how Gauss quadrature can be used to approximate

$$u^Tf(W)v \tag{5.28}$$

where W is a symmetric and positive definite matrix and f is some smooth (possibly C^∞) function, not necessarily a polynomial. Here, we understand the function of the matrix as given by the definition by substitution, e.g. $f(z) = z^{-1}$ becomes $f(W) = W^{-1}$ (see [60] for details on functions of matrices).

This can be done using the eigendecomposition $W = Q\Lambda Q^T$ with orthogonal Q and we assume $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$. As a result we get

$$u^T f(W)v = u^T Q f(\Lambda) Q^T v. \tag{5.29}$$

By introducing $\alpha = Q^T u$ and $\beta = Q^T v$, we can rewrite (5.29) as

$$u^T f(W)v = \alpha^T f(\Lambda)\beta = \sum_{i=1}^{N} f(\lambda_i)\alpha_i\beta_i. \tag{5.30}$$

The expansion (5.30) can be viewed as a Riemann-Stieltes integral

$$I[f] = u^T f(W)v = \int_a^b f(\lambda)d\alpha(\lambda) \tag{5.31}$$

where the piecewise constant measure α is defined as follows

$$\alpha(\lambda) = \begin{cases} 0 & \text{if } \lambda < a = \lambda_1 \\ \sum_{j=1}^{i} \alpha_j\beta_j & \text{if } \lambda_i \leq \lambda < \lambda_{i+1} \\ \sum_{j=1}^{n} \alpha_j\beta_j & \text{if } b = \lambda_n < \lambda \end{cases}$$

We can now express (5.31) as the quadrature formula

$$\int_a^b f(\lambda)d\alpha(\lambda) = \sum_{j=1}^{k} \omega_j f(t_j) + \sum_{i=1}^{M} v_i f(z_i) + R[f], \tag{5.32}$$

where the weights ω_j, v_i and the nodes t_j are unknowns and the nodes z_i are prescribed. The remainder can be expressed as

$$R[f] = \frac{f^{(2N+M)}(\eta)}{(2N+M)!}\int_a^b \prod_{k=1}^{M}(\lambda - z_k)\left[\prod_{j=1}^{N}(\lambda - t_j)\right]^2 d\alpha(\lambda), \quad a < \eta < b. \tag{5.33}$$

Following the analysis presented in [49], $R[f]$ can be rewritten for Gauss ($M = 0$), Gauss-Radau ($M = 1$, $z_1 = a$ or $z_1 = b$) or Gauss-Lobatto ($M = 2$, $z_1 = a$ and $z_2 = b$) quadrature formulas. A more detailed description can be found in [54, 40, 48, 47, 15, 16].

We will see in the next section that in the case of $u = v$, we can compute the weights and nodes of the quadrature rule by simply applying the Lanczos process to the symmetric matrix W, see [54]. Then, the eigenvalues of the tridiagonal matrix will represent the nodes of the quadrature rule and the first component of the corresponding eigenvector can be used to compute the weights.

5.3.2 The Golub-Kahan bidiagonalization

The scattering amplitude or primal output $J^{pr}(x) = g^T x$ (cf. Section 1.3) can now be approximated using the connection between Gauss quadrature and the Lanczos process given in Algorithm 1.2. To be able to apply the theory of Golub and Meurant, we need the system matrix W in (5.28) to be symmetric, which can be achieved by

$$J^{pr}(x) = g^T A^{-1} b = g^T (A^T A)^{-1} A^T b = g^T (A^T A)^{-1} p = g^T f(A^T A) p \qquad (5.34)$$

where f is the reciprocal function and using the fact that $x = A^{-1}b$ and $p = A^T b$. In order to use the Lanczos process to obtain nodes and weights of the quadrature formula, we need a symmetrized version of (5.34)

$$J^{pr}(x) = \frac{1}{4}\left[(p+g)^T (A^T A)^{-1} (p+g) - (g-p)^T (A^T A)^{-1} (g-p)\right]. \qquad (5.35)$$

Good approximations to $(p+g)^T(A^T A)^{-1}(p+g)$ and $(p-g)^T(A^T A)^{-1}(p-g)$ will result in a good approximation to the scattering amplitude. Here, we present the analysis for the Gauss rule (i.e. $M = 0$ in (5.32)) where we apply the Lanczos process to A^TA and get

$$A^TAV_k = V_kT_k + r_ke_k^T \tag{5.36}$$

with orthogonal V_k and

$$T_k = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_N \\ & & \beta_N & \alpha_N \end{bmatrix}.$$

A well-known result by Golub and Welsch [54] is that the eigenvalues t_j of T_k determine the nodes in the quadrature formula

$$\int_a^b f(\lambda)d\alpha(\lambda) = \sum_{j=1}^{k} \omega_j f(t_j) + R_G\,[f]\,, \tag{5.37}$$

where the remainder $R_G\,[f]$ for the function $f(x) = \frac{1}{x}$ is given by

$$R_G\,[f] = \frac{1}{\eta^{2k+1}} \int_a^b \left[\prod_{j=1}^{k}(\lambda - t_j)\right]^2 d\alpha(\lambda).$$

Notice, $R_G\,[f]$ will always be positive and therefore the Gauss rule will always give an underestimation of the scattering amplitude.

The weights for the Gauss rule are given by the squares of the first elements of the normalized eigenvectors of T_k [54] in the same order as the eigenvalues. Instead of applying the Lanczos process to A^TA, we can simply use the Golub-Kahan bidiagonalization procedure presented in Section 5.2.2. Trivially, the matrix T_k can be obtained from (5.4) via $T_k = B_k^TB_k$. Since the matrix T_k is a tridiagonal, symmetric matrix, it is relatively cheap to compute its eigenvalues and eigenvectors, for example, by using a divide and conquer method as given in [53, Section 8.5].

The expression

$$\sum_{j=1}^{k} \omega_j f(t_j)$$

can be simplified to

$$\sum_{j=1}^{k} \omega_j f(t_j) = e_1^T f(T_k) e_1, \tag{5.38}$$

see [51] for a proof of this. Hence, for $f(x) = 1/x$ (5.38) reduces to $e_1^T T_k^{-1} e_1$. The last expression simply states that we have to find a good approximation for the $(1,1)$ element of the inverse of T_k. If we can find such a good approximation for $(T_k^{-1})_{(1,1)}$, the computation becomes much more efficient since no eigenvalues or eigenvectors have to be computed to determine the Gauss quadrature rule. Another possibility is to solve the system $T_k z = e_1$. To solve with T_k is relatively cheap since it is tridiagonal.

Golub and Meurant [49, 50] give bounds on the elements of the inverse using Gauss, Gauss-Radau, Gauss-Lobatto rules depending on the Lanczos process. These bounds can then be used to give a good approximation to the scattering amplitude without solving a linear system with T_n or using its eigenvalues and eigenvectors. We will only give the bounds connected to the Gauss-Radau rule ($M = 1$, $z_1 = a$ or $z_1 = b$) since then we get both upper and lower bounds, i.e.

$$\frac{t_{1,1} - b + \frac{s_1^2}{b}}{t_{1,1}^2 - t_{1,1} b + s_1^2} \leq (T_k^{-1})_{1,1} \leq \frac{t_{1,1} - a + \frac{s_1^2}{a}}{t_{1,1}^2 - t_{1,1} a + s_1^2}$$

with $s_1^2 = \sum_{j \neq 1} t_{j1}^2$ and $t_{i,j}$ the elements of T_k. These bounds are not sharp since they will improve with the number of Lanczos steps and the approximation to the scattering amplitude will improve as the algorithm progresses. The one-sided bounds for the Gauss rule and the Gauss-Lobatto rule are given in [49]. It is also possible to obtain the given bounds using variational principles (see [94]). In the case of CG applied to a positive definite matrix

A, the $(1,1)$-element of T_k^{-1} can be easily approximated using

$$(T_k^{-1})_{(1,1)} = 1/\left\|r_0\right\|^2 \sum_{j=0}^{N-1} \alpha_j \left\|r_j\right\|^2$$

where α_j and $\|r_j\|$ are given at every CG step. This formula is discussed in [111, 110, 1] where it is shown to be numerically stable. From [110] we get that the remainder $R_G\left[f\right]$ in the Gauss quadrature where f is the reciprocal function is equal to the error at step k of CG for the normal equations, i.e.

$$\left\|x - x_k\right\|_{A^TA} / \left\|r_0\right\| = R_G\left[f\right].$$

Hence, the Golub-Kahan bidiagonalization can be used to approximate the error for CG for the normal equations [112].

5.3.3 Approximation using GLSQR (the block case)

We now want to use a block method to estimate the scattering amplitude using GLSQR. The 2×2 matrix integral we are interested in is now

$$\int_a^b f(\lambda)d\alpha(\lambda) = \begin{bmatrix} b^T & 0 \\ 0 & g^T \end{bmatrix} \begin{bmatrix} 0 & A^{-T} \\ A^{-1} & 0 \end{bmatrix} \begin{bmatrix} b & 0 \\ 0 & g \end{bmatrix} = \begin{bmatrix} 0 & b^TA^{-T}g \\ g^TA^{-1}b & 0 \end{bmatrix}. \tag{5.39}$$

In [49], Golub and Meurant show how a block method can be used to generate quadrature formulae. In more detail, the integral $\int_a^b f(\lambda)d\alpha(\lambda)$ is now a 2×2

symmetric matrix and the most general quadrature formula is of the form

$$\int_a^b f(\lambda) d\alpha(\lambda) = \sum_{i=1}^{k} C_j f(H_j) C_j + R[f] \tag{5.40}$$

with H_j and C_j being symmetric 2×2 matrices. Expression (5.40) can be simplified using

$$H_j = Q_j \Lambda_j Q_j^T$$

where Q_j is the eigenvector matrix and Λ_j the 2×2 diagonal matrix containing the eigenvalues. Hence,

$$\sum_{i=1}^{k} C_j Q_j^T f(\Lambda_j) Q_j C_j$$

and if we write $C_j Q_j^T f(\Lambda_j) Q_j C_j$ as

$$f(\lambda_1) z_1 z_1^T + f(\lambda_2) z_2 z_2^T$$

where z_j is the j-th column of the matrix $C_j Q_j^T$. Hence, we get for the quadrature rule

$$\sum_{i=1}^{2k} f(\lambda_j) z_j z_j^T$$

where λ_j is a scalar and $z_j = \left[z_j^{(1)}, z_j^{(2)} \right]^T \in \mathbb{R}^2$. In [49], it is shown that there exist orthogonal matrix polynomials such that

$$\lambda p_{j-1}(\lambda) = p_j(\lambda) B_j + p_{j-1}(\lambda) D_j + p_{j-2}(\lambda) B_{j-1}^T$$

with $p_0(\lambda) = I_2$ and $p_{-1}(\lambda) = 0$. We can write the last equation as

$$\lambda \left[p_0(\lambda), \ldots, p_{N-1}(\lambda) \right] = \left[p_0(\lambda), \ldots, p_{k-1}(\lambda) \right] \mathcal{T}_k + \left[0, \ldots, 0, p_N(\lambda) B_k \right]^T$$

with

$$\mathcal{T}_k = \begin{bmatrix} D_1 & B_1^T & & & \\ B_1 & D_2 & B_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & B_{k-2} & D_{k-1} & B_{k-1}^T \\ & & & B_{k-1} & D_k \end{bmatrix}$$

which is a block-tridiagonal matrix. Therefore, we can define the quadrature rule as

$$\int_a^b f(\lambda)d\alpha(\lambda) = \sum_{i=1}^{2k} f(\theta_i)u_i u_i^T + R[f] \tag{5.41}$$

where $2k$ is the order of the matrix $\mathcal{T}_k$, θ_i eigenvalues of $\mathcal{T}_k$ and u_i is the vector consisting of the first two elements of the corresponding normalized eigenvector. The remainder $R[f]$ can be approximated using a Lagrange polynomial and we get

$$R[f] = \frac{f^{(2k)}(\eta)}{(2k)!}\int_a^b s(\lambda)d\alpha(\lambda)$$

where $s(x) = (x-\theta_1)(x-\theta_2)\dots(x-\theta_{2N})$. The sign of the function s is not constant over the interval $[a, b]$. Therefore, we cannot expect that the Block-Gauss rule always underestimates the scattering amplitude. This might result in a rather oscillatory behavior. In [49], it is also shown that

$$\sum_{i=1}^{2k} f(\theta_i)u_i u_i^T = e^T f(\mathcal{T}_k)e$$

with $e = (I_2, 0, \ldots, 0)$. In order to use the approximation (5.41), we need a Block-Lanczos algorithm for the matrix

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}.$$

The GLSQR algorithm represents an implementation of a Block-Lanczos method for this matrix and can therefore be used to create a block-tridiagonal matrix $\mathcal{T}_k$ as introduced in Section 5.2.3. Using this we show in the second part of this Section that we can then compute an approximation to the integral given in (5.39). Hence, the scattering amplitude is approximated via

$$\sum_{i=1}^{2k} f(\theta_i) u_i u_i^T \approx \begin{bmatrix} 0 & g^T x \\ g^T x & 0 \end{bmatrix}$$

without computing an approximation to x directly.

Further simplification of the above can be achieved following a result in [112]: since from (5.17)

$$\mathcal{T}_k = \Pi_{2k} \begin{bmatrix} 0 & T_k \\ T_k^T & 0 \end{bmatrix} \Pi_{2k}^T$$

where Π_{2k} is the permutation (5.15) of dimension $2k$, in the case of the reciprocal function

$$\begin{aligned} e^T \mathcal{T}_k^{-1} e &= e^T \Pi_{2k} \begin{bmatrix} 0 & T_k^{-T} \\ T_k^{-1} & 0 \end{bmatrix} \Pi_{2k}^T e \\ &= \begin{bmatrix} 0 & e_1^T T_k^{-T} e_1 \\ e_1^T T_k^{-1} e_1 & 0 \end{bmatrix}. \end{aligned}$$

Note that with the settings $r_0 = b - Ax_0$ and $s_0 = g - A^T y_0$ the scattering amplitude can be written as

$$g^T A^{-1} b = s_0^T A^{-1} r_0 + s_0^T x_0 + y_0^T b.$$

With our choice of $x_0 = y_0 = 0$, we get that the scattering amplitude is equal by $s_0^T A^{-1} r_0$. Starting the GLSQR Block-Lanczos process with

$$\begin{bmatrix} u_1 & 0 \\ 0 & v_1 \end{bmatrix}$$

where $u_1 = r_0 / \left\|r_0\right\|_2$ and $v_1 = s_0 / \left\|s_0\right\|_2$ results in $v_1^T A^{-1} u_1 = e_1^T T_N^{-1} e_1$. An approximation to the scattering amplitude $g^T A^{-1} b$ is thus obtained via

$$s_0^T A^{-1} r_0 \approx \left\|r_0\right\|_2 \left\|s_0\right\|_2 e_1^T T_N^{-1} e_1.$$

5.3.4 Preconditioned GLSQR

In Section 5.2.5, the preconditioned GLSQR method was introduced and we now show that we can use this method to approximate the scattering amplitude directly. In the above, we showed that GLSQR gives an approximation to scattering amplitude using

$$\int_a^b f(\lambda) d\alpha(\lambda) = \begin{bmatrix} 0 & g^T A^{-1} b \\ b^T A^{-T} g & 0 \end{bmatrix}.$$

Reformulating this in terms of the preconditioned method gives,

$$\begin{aligned}
\hat{g}^T\hat{x} &= \hat{g}^T\widehat{A}^{-1}\hat{b}\\
&= (M_2^{-T}g)^T(M_1^{-1}AM_2^{-1})^{-1}(M_1^{-1}b)\\
&= g^TM_2^{-1}M_2A^{-1}M_1M_1^{-1}b\\
&= g^TA^{-1}b\\
&= g^Tx
\end{aligned}$$

which shows that the scattering amplitude for the preconditioned system $\widehat{A}\hat{x} = \hat{b}$ with $\widehat{A} = M_1^{-1}AM_2^{-1}$, $\hat{x} = M_2x$ and $\hat{b} = M_1^{-1}b$ is equivalent to the scattering amplitude of the original system. The scattering amplitude can therefore be approximated via

$$\int_a^b f(\lambda)d\alpha(\lambda) = \begin{bmatrix} 0 & \hat{g}^T\hat{x}\\ \hat{x}^T\hat{g} & 0 \end{bmatrix}.$$

5.3.5 BICG and the scattering amplitude

The methods we presented so far are based on Lanczos methods for A^TA. The algorithm introduced in this Section connects BICG (see Algorithm 2.11 and [31]), a method based on the non-symmetric Lanczos process, and the scattering amplitude.

Using $r_j = b - Ax_j$ and $s_j = g - A^Ty_j$, the scattering amplitude can be expressed as

$$g^TA^{-1}b = \sum_{j=0}^{N-1}\alpha_js_j^Tr_j + s_N^TA^{-1}r_N, \qquad (5.42)$$

where N is the dimension of A (cf. [112]). To show this, we use $r_0 = b$, $s_0 = g$

and

$$\begin{aligned}
& s_j^T A^{-1} r_j - s_{j+1}^T A^{-1} r_{j+1} \\
= \; & (g - A^T y_j)^T A^{-1} (b - A x_j) - s_{j+1}^T A^{-1} r_{j+1} \\
= \; & (g - A^T y_j + A^T y_{j+1} - A^T y_{j+1})^T A^{-1} (b - A x_j + A^T x_{j+1} - A^T x_{j+1}) \\
& - s_{j+1}^T A^{-1} r_{j+1} \\
= \; & (s_{j+1} + A^T (y_{j+1} - y_j))^T A^{-1} (r_{j+1} + A(x_{j+1} - x_j)) - s_{j+1}^T A^{-1} r_{j+1} \\
= \; & \alpha_j (q_j^T r_{j+1} + s_{j+1}^T p_j + \alpha_j q_j^T A p_j) \\
= \; & \alpha_j s_j^T r_j,
\end{aligned}$$

where we use $\alpha_j = \frac{\langle s_j, r_j \rangle}{\langle q_j, Ap_j \rangle}$ (cf. Section 2.2.3). An approximation to the scattering amplitude at step k is then given by

$$g^T A^{-1} b \approx \sum_{j=0}^{k} \alpha_j s_j^T r_j.$$

Considering the preconditioned systems

$$\widehat{A} \hat{x} = \hat{b} \text{ and } \widehat{A}^T \hat{y} = \hat{g}$$

with $\widehat{A} = M_1^{-1} A M_2^{-1}$, $\hat{x} = M_2 x$, $\hat{y} = M_1^T y$, $\hat{b} = M_1^{-1} b = \hat{r}_0$, $\hat{g} = M_2^{-T} g = \hat{s}_0$ and the relation $g^T A^{-1} b = \hat{s}_0^T \widehat{A}^{-1} \hat{r}_0$, we can use (5.42) for the preconditioned BICG method.

Another way of approximating the scattering amplitude via BICG was given by Saylor and Smolarski [101, 100] in which the scattering amplitude is connected to Gaussian quadrature in the complex plane. The scattering amplitude is then given by

$$g^T A^{-1} b \approx \sum_{i=1}^{k} \frac{\omega_i}{\zeta_i}$$

where ω_i and ζ_i are eigenvector component and eigenvalue, respectively, of the tridiagonal matrix associated with the appropriate formulation of BICG(see [101, 100] for details). The derivation of Saylor's and Smolarski approach is not as straightforward as the one given in (5.42) coming from [112].

CHAPTER 6

NUMERICAL RESULTS

This chapter shows the numerical experiments carried out to illustrate the theoretical results presented in previous chapters. The focus is on showing that the methods not only have a solid theoretical foundation but can also compete and even outperform existing methods.

6.1 Bramble-Pasciak$^+$ and Combination Preconditioning

Bramble-Pasciak$^+$ preconditioner

In this section, we look at matrices coming from the Stokes problem (1.11). In particular, the individual examples are generated using the IFISS package [23]. Namely, we consider the flow over the channel domain (Example 5.1.1 in [24]) and the flow over a backward facing step (Example 5.1.2 in [24]). As shown in Section 1.3, the Stokes equation (1.11) can be transformed using a weak formulation which can then be treated using the finite element method. The governing linear system (1.14) is in saddle point form. Here, we have to compare our Bramble-Pasciak$^+$ preconditioner to other suitable methods. One candidate would be the block diagonal preconditioning already introduced in Section 2.1.2 (see [114, 104] for more details). This enables us to use the $\mathcal{H}$-MINRES method with $\mathcal{H}^+$ introduced in Section 2.1.2. We compare this method to the classical MINRES algorithm with the block-diagonal preconditioner. In the IFISS implementation, the preconditioner S_0 is chosen to be the positive-definite pressure mass matrix (see Section 6.2 in [24]). The right hand side for each example is also given by IFISS.

Example 6.1. *The first example is based on the flow over a backward facing step. The size of the system matrix $\mathcal{A}$ is 6659×6659 with $m = 769$ and $n = 5890$. The results shown in Figure 6.1 are obtained by using $\mathcal{H}$-*MINRES *with $\mathcal{H}^+$ and the classical preconditioned* MINRES *as given in [114, 104] as well as* CG *for the classical Bramble-Pasciak setup. The preconditioner A_0 is given by the Incomplete Cholesky factorization, in particular we use MATLAB's implementation with no additional fill-in (see [96] for details). S_0 is given by IFISS as the pressure mass matrix. The blue (dashed) curve shows the results of Preconditioned* MINRES *with a block-diagonal preconditioner. The corresponding preconditioned residual is given in the 2-norm. The black (dash-dotted) line shows the 2-norm preconditioned residuals computed by the $\mathcal{H}$-*MINRES *algorithm. The red (solid) curve shows the preconditioned residuals for* CG *with the Bramble-Pasciak setup. As expected from the eigenvalue analysis in Section 3.4 the results for* MINRES *and $\mathcal{H}$-*MINRES *are very similar and are both outperformed by the Bramble-Pasciak* CG *except for rather large convergence tolerances. It should be mentioned that* MINRES *and $\mathcal{H}$-*MINRES *with $\mathcal{H}_+$ will work as long as the positivity of the inner product for $\mathcal{H}$-*MINRES *and positivity of the preconditioner for* MINRES *are given. In contrast, reliability of* CG *for Bramble-Pasciak cannot be guaranteed if the matrix A_0 is not appropriately scaled. Nevertheless, it quite often works in practice as is shown in this example. We only show the 2-norm of the residuals here and hence monotonicity cannot be expected. The norms in which the residuals are minimized are given in Chapter 2.*

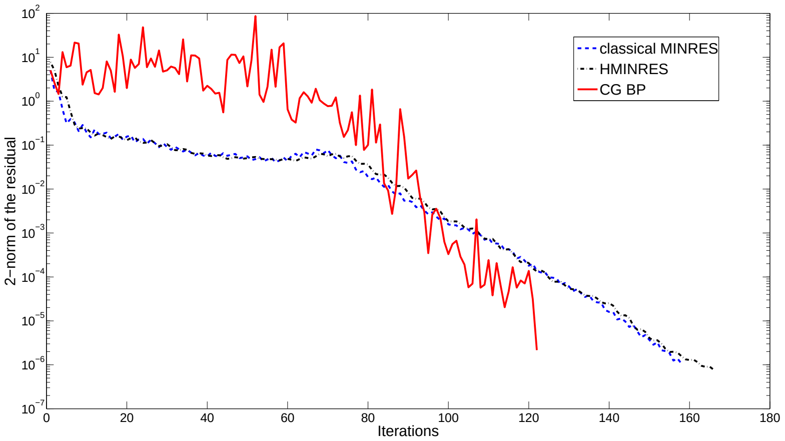

Figure 6.1: Results for $\mathcal{H}$-MINRES, classical preconditioned MINRES and CG for classical Bramble-Pasciak.

Example 6.2. *This example is again taken from* IFISS *and represents the flow over a channel domain. The size of the system matrix $\mathcal{A}$ is given by* 9539×9539 *with* $m = 1089$ *and* $n = 8450$. *The results shown in Figure 6.2 are obtained by using* $\mathcal{H}$-MINRES *with* $\mathcal{H}^{+}$ *and the classical preconditioned* MINRES *as given in [114, 104] as well as* ITFQMR *for the classical Bramble-Pasciak setup. The preconditioners are chosen such that* $A_0 = A$ *and* S_0 *is given by* IFISS *as the pressure mass matrix. With the choice of* $A_0 = A$ *the Bramble-Pasciak* CG *is bound to fail, but here we illustrate that* ITFQMR *can work in this setup. The blue (dashed) curve shows the results of the Preconditioned* MINRES *with a block-diagonal preconditioner. The corresponding preconditioned residual is given in the* 2*-norm. The black (dash-dotted) line shows the* 2*-norm preconditioned residuals computed by the* $\mathcal{H}$-MINRES *algorithm. The red (solid) curve shows the preconditioned residuals for* ITFQMR *with the Bramble-Pasciak setup. Again, the results for* MINRES *and* $\mathcal{H}$-MINRES *are very similar.*

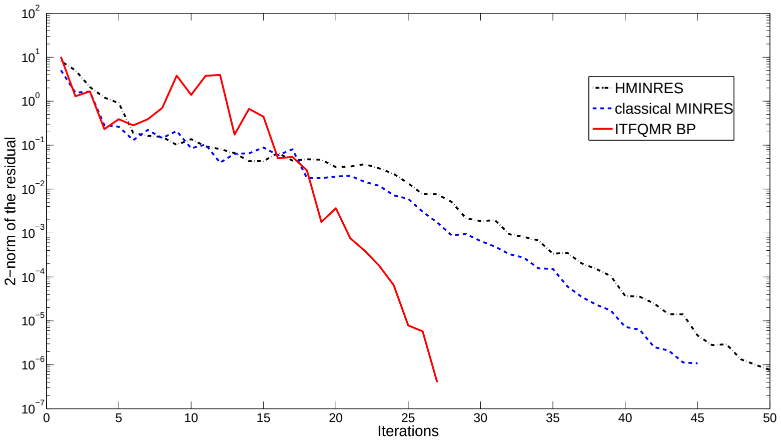

Figure 6.2: Results for $\mathcal{H}$-MINRES, classical Preconditioned MINRES and ITFQMR for classical Bramble-Pasciak.

Combination preconditioning

We now show results for the combination preconditioning with the Bramble-Pasciak and the Bramble-Pasciak^{+} setup presented in Chapter 3.

Example 6.3. *In this example, the matrix represents the flow over a channel domain and is of size* 9539×9539*. Our choice for* A_0 *is again the Incomplete Cholesky decomposition with zero fill-in and* S_0 *the pressure mass matrix. Figure 6.3 shows the results for different values of* α*. The choice for* $\alpha = 2/3$ *shown in the black (solid) curve performs better than original Bramble-Pasciak method reflected by* $\alpha = 1$ *in the blue (dashed) line. For comparison, we also show the results for the preconditioned* MINRES *in the red (dashed) line. Further values of* α *are shown in Figure 6.3.*

Example 6.4. *The setup for this example is identical to the one described in Example 6.3, only the underlying matrix now comes from the flow over the backward facing step. The dimension of* $\mathcal{A}$ *is* 6659×6659*. As can be seen from the results in Figure 6.4, the combination preconditioning again outperforms* ITFQMR *with Bramble-Pasciak setup for* $\alpha = 2/3$*.*

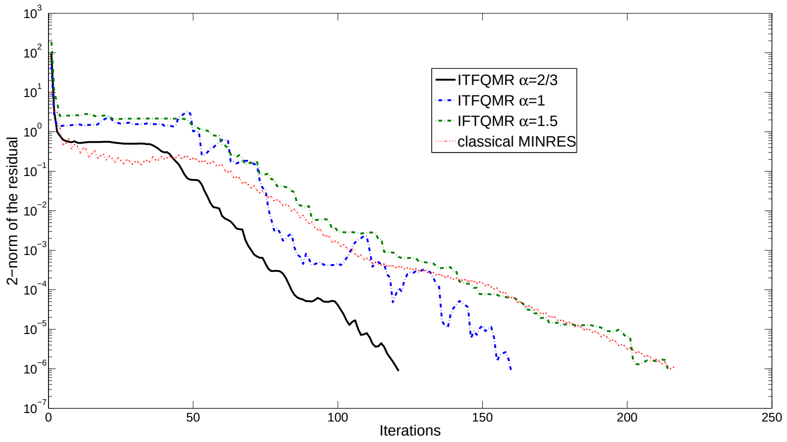

Figure 6.3: ITFQMR results for combination preconditioning with different values for α.

The combination of the Bramble-Pasciak setup and the method of Schöberl and Zulehner as presented in Section 3.5.3 is given by the preconditioner

$$\mathcal{P}_3^{-1} = \begin{bmatrix} (\alpha-\beta)A_0^{-1} & -\beta A_0^{-1}B^T\hat{S}^{-1} \\ 0 & (\beta-\alpha)\hat{S}^{-1} - \beta\hat{S}^{-1}BA_0^{-1}B^T\hat{S}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA_0^{-1} & I \end{bmatrix}$$

and inner product

$$\mathcal{H}_3 = \begin{bmatrix} A - A_0 & 0 \\ 0 & \hat{S} \end{bmatrix}.$$

Example 6.5. *In this example, we apply* CG *with the combination preconditioning setup for Schöberl-Zulehner and Bramble-Pasciak to a linear system coming from the flow over a backward facing step of dimension* 6659*. The preconditioner A_0 is chosen to be the zero fill-in Incomplete Cholesky factorization and $\hat{S}$ is the pressure mass matrix given in* IFISS*. For the parameter choice $\alpha = -.3$ and $\beta = .5$, the combination marginally outperforms the method of Schöberl and Zulehner as shown in Figure 6.5.*

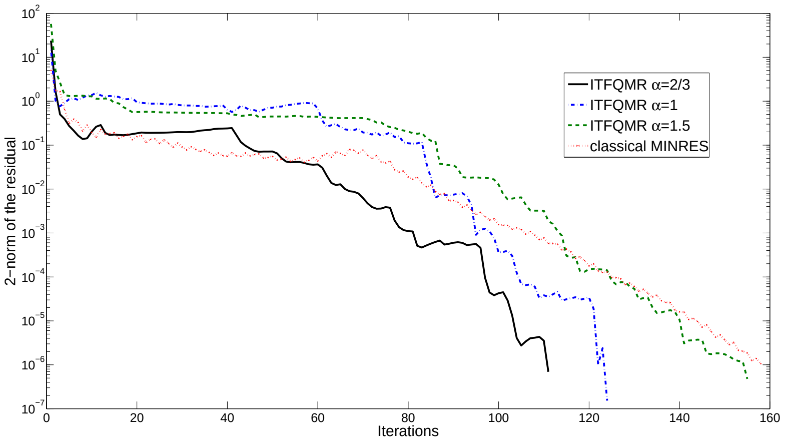

Figure 6.4: ITFQMR results for combination preconditioning with different values for α.

We want to mention here that at this point in time the choice of the parameters α and β is more or less determined from numerical experiments. We want to use these results as a proof of concept that combination preconditioning can give competitive results. Further research should investigate the choice of α and β.

Example 6.6. *In this example, we apply* CG *with the combination preconditioning setup for Schöberl-Zulehner and Bramble-Pasciak to a linear system coming from the flow over the channel domain of dimension* 9539*. In addition, we show the results for* CG *with Bramble-Pasciak setup. The preconditioner A_0 is chosen to be the zero fill-in Incomplete Cholesky factorization, and $\hat{S}$ is the pressure mass matrix given in* IFISS*. Again Figure 6.6 shows that for the parameter choice $\alpha = -.3$ and $\beta = .5$, the combination is able to outperform the method of Schöberl and Zulehner.*

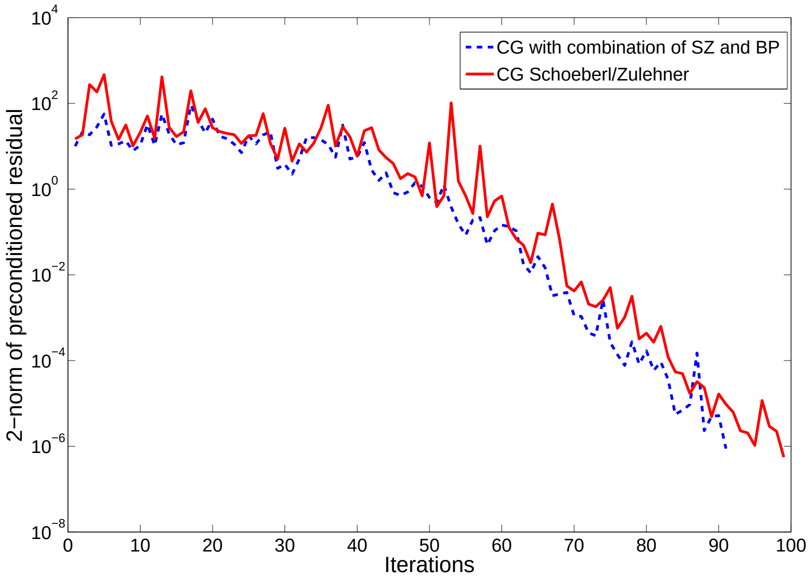

Figure 6.5: CG for Schöberl-Zulehner and Combination preconditioning for $\alpha = -.3$ and $\beta = .5$.

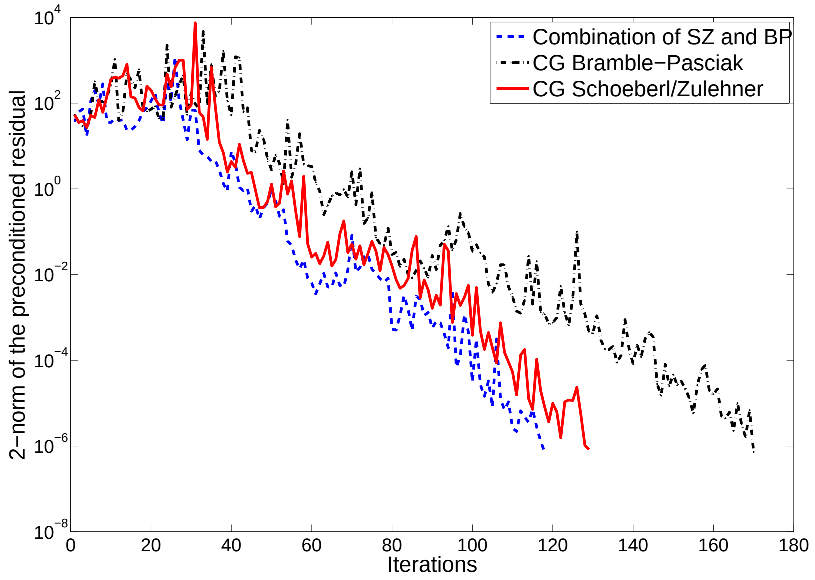

Figure 6.6: CG for Schöberl-Zulehner, Bramble-Pasciak and Combination preconditioning for $\alpha = -.3$ and $\beta = .5$.

The results shown in this section indicate that particular combinations outperform widely used methods. The first example for the combination of classical and modified Bramble-Pasciak method requires fewer iterations while the work per iteration is the same. The combination preconditioning method was able to outperform both the Bramble-Pasciak CG and Schöberl and Zulehner's CG method. Here the application of the preconditioner is slightly more expensive than in Schöberl and Zulehner's CG method but the application of the inner product is cheaper. The choice of parameters is not fully understood for all the methods presented and provides interesting research directions for the future.

6.2 The Bramble-Pasciak-like method

In this Section, we want to give examples of how the Bramble-Pasciak like method presented in Chapter 4 can be applied to different problems. The examples in this section are taken either from the CUTEr test set [55] or are generated using the IFISS software package. We will again use the structure presented in Section 4.4 where different setups of the original matrix were analyzed. The methods we compare in this section are the CG of Forsgren, Gill and Griffin and the Bramble-Pasciak-like CG with $\mathcal{P}_-$ and $\mathcal{H}_-$. Once more we compare our methods to MINRES with block-diagonal preconditioning,

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ 0 & C_0 \end{bmatrix}.$$

Note that the $(2,2)$ block in $\mathcal{P}$ is now called C_0 since this will not always be a Schur-complement type preconditioner.

A definite and C semi-definite

This is a typical setup arising when treating the Stokes problem with Mixed Finite Elements (see Section 1.3) and we again look at examples generated by IFISS.

Example 6.7. *The first test matrix is of size* 6659×6659 *and describes the flow over a backward facing step. The preconditioner* A_0 *is taken to be the Incomplete Cholesky factorization with zero fill-in [96]. The preconditioner* C_0 *is generated by* IFISS *as the positive-definite pressure mass matrix. It can be seen from the results in Figure 6.7 that the classical Bramble-Pasciak and the Bramble-Pasciak-like method in the* $\mathcal{P}_-$ *have a similar convergence behavior and they both outperform the preconditioned* MINRES *and the Bramble-Pasciak-like method in the* $\mathcal{P}_+$ *setup.*

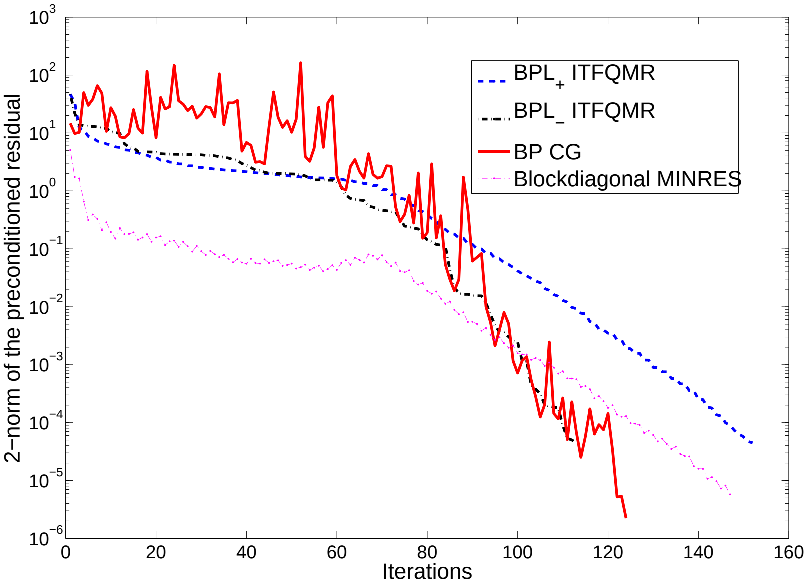

Figure 6.7: Results for the backward facing step

Example 6.8. *The second test matrix is of size* 9539×9539 *and describes the flow over a channel domain [24]. The preconditioner* A_0 *is chosen such that* $A_0 = .9A$ *and* C_0 *is again generated by* IFISS *as the positive-definite* *pressure mass matrix. The results given in Figure 6.8 again show that the Bramble-Pasciak* CG *and* ITFQMR *in the Bramble-Pasciak-like setup with* $\mathcal{P}_-$ *outperform* MINRES *and the* $\mathcal{P}_+$ *setup. Additionally, we show the results if* CG *is applied with the* $\mathcal{P}_-$ *configuration which is not guaranteed to work.*

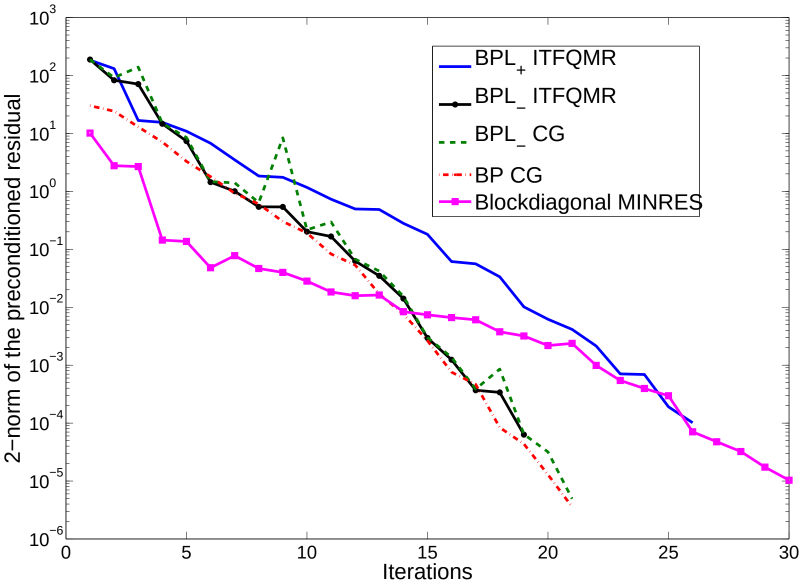

Figure 6.8: Flow over the channel domain

A indefinite and C definite

This setup typically arises when one is interested in solving optimization problems and hence the test matrices are coming from CUTEr.

Example 6.9. *In this example, we are looking at the matrix CVXQP1_M from CUTEr which is of size* 1500×1500*.* C *will either be a diagonal matrix with entries of the form* 10^{-k} *on the diagonal where* $2 \leq k \leq 10$ *or it is created using the MATLAB command* `C=1e-1*sprandsym(m,.3)+speye(m);` *which is an identity matrix plus a random sparse perturbation. The preconditioners are defined by* $C_0 = 0.9C$ *and* $A_0 = diag(A) + B^T C_0^{-1} B$*. The results for the Bramble-Pasciak-like method with the* $-$ *setup and the Forsgren-Gill-Griffin method are shown in Figures 6.9 and 6.10.*

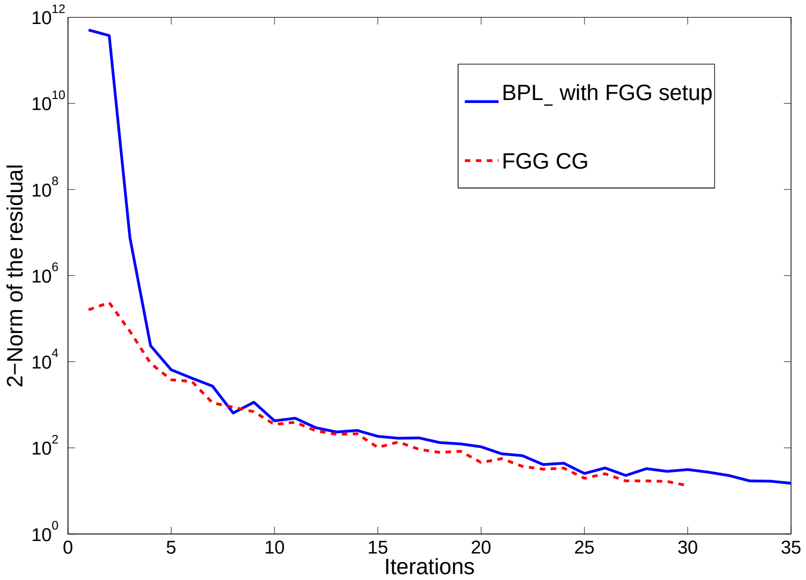

Figure 6.9: Diagonal C

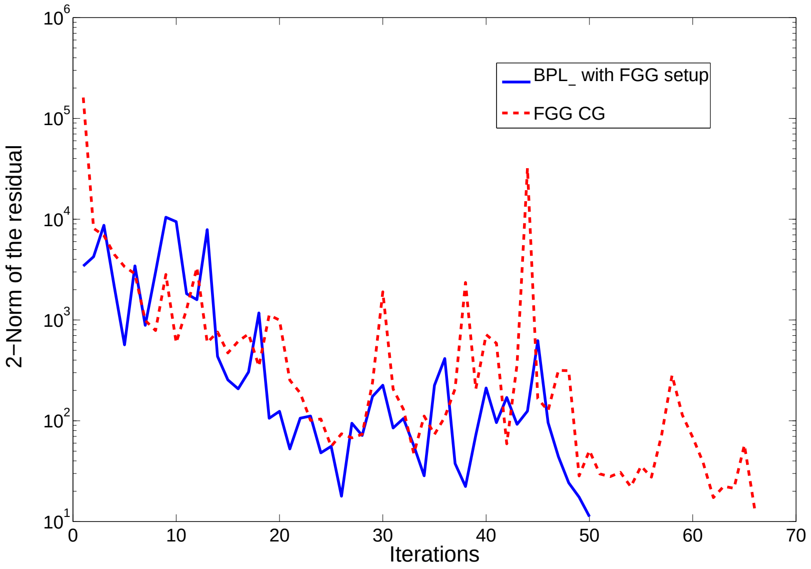

Figure 6.10: Randomly perturbed C

The last example shows that the performance in the case of diagonal C is very similar to that given by the FGG method. It should be noted here that we are able to work with a block-triangular preconditioner in the case of the Bramble-Pasciak method which makes the solution of a linear system with this preconditioner easier than for the FGG case. In the case that the decomposed form of the Bramble-Pasciak-like preconditioner (cf. Section 4.3.2) is used, we would expect similar timings.

A indefinite and C semi-definite

In this part, we again consider examples from the CUTEr testset where the block A is typically indefinite with zero eigenvalues and the matrix C is positive semi-definite. In [32], it is assumed that the matrix C if semi-definite has a zero block in the lower corner. In order to guarantee this structure for real world examples, some preprocessing might be necessary.

Example 6.10. *In this example, we consider the CUTEr matrix CVXQP1_M of size* 1500×1500 *with the block*

$$C = \begin{bmatrix} Z & 0 \\ 0 & \tilde{C} \end{bmatrix} \in \mathbb{R}^{m \times m}$$

where Z *is a matrix with eigenvalues at zero and* $\tilde{C}$ *is generated using the MATLAB command* `1e-1*sprandsym(p,.1)+1e1*speye(p);` *with* $p = m - 3$*. We use the modified Cholesky preconditioner* A_0 *for the block* A *as presented in Section 4.4 and then create a Schur-complement type preconditioner* $C_0 = C + BA_0^{-1}B^T$*. Note that we can always reliably apply* MINRES *if* $\mathcal{H}$ *defines an inner product (cf. Section 2.1.2). We also show results for* CG *which is not guaranteed to work in the case of semi-definite* C *and results using* ITFQMR *for the* $\mathcal{P}_-$ *configuration. We compare this setup to the block-diagonal preconditioned* MINRES*. From the results given in Figure*

6.11 it can be observed that the preconditioned MINRES *needs more iterations than the Bramble-Pasciak-like method with the $\mathcal{P}_+$ setup to achieve the given relative tolerance of* 10^{-4}.

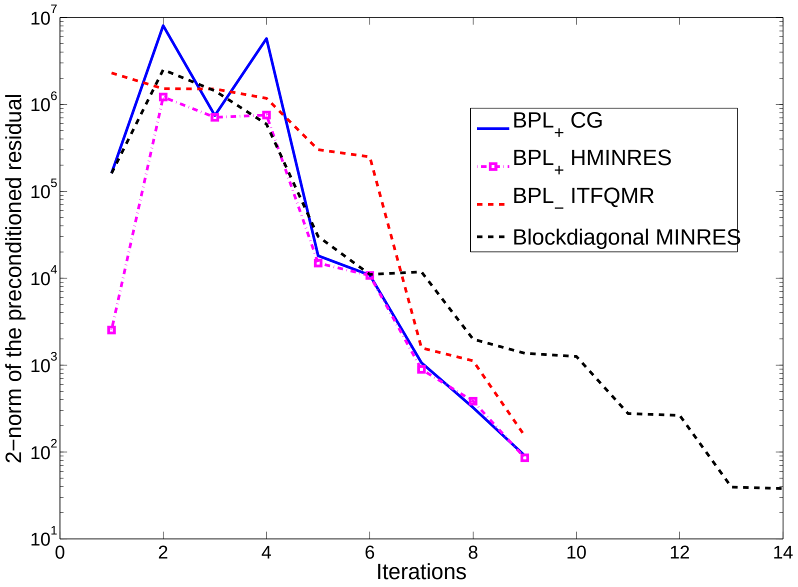

Figure 6.11: Indefinite A and semi-definite C for *CVXQP1_M*

It should be noted that the convergence of MINRES for the optimization problems presented in this Section is not based on a solid analysis. This is in contrast to the problems coming from the Stokes problem (Section 1.3) where the convergence behavior of MINRES was fully analyzed in [114,104].

Example 6.11. *The second example in this section is again taken from CUTEr. In particular, we use the matrix CONT050 which is of size* 4998×4998*. The setup for* C *is the same as for CVXQP1_M and we compute a modified Cholesky preconditioner* A_0 *for* A *which we then use to generate a Schur-complement-type preconditioner* $C_0 = C + B\,\mathrm{diag}(A_0)^{-1}B^T$ *that uses only the diagonal of* A_0*. The results are shown in Figure 6.12. This examples emphasizes the point made earlier that the convergence of* MINRES *for problems of this kind is not fully analyzed. For* MINRES*, only a very poor convergence can be observed.*

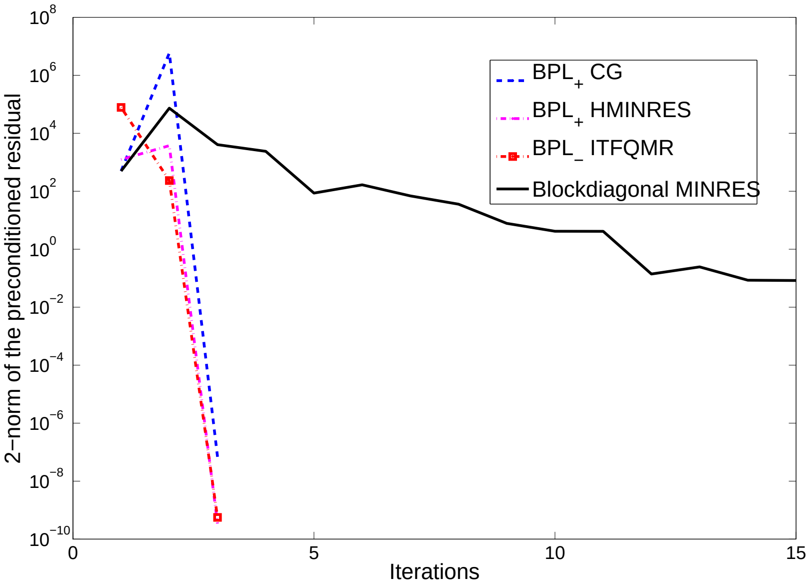

Figure 6.12: Indefinite A and semi-definite C for *CONT050*

The results given in this section reflect the flexibility of the Bramble-Pasciak-like method. In the setup where A is definite and C is semi-definite where the Bramble-Pasciak-like setup it not guaranteed to work in the $\mathcal{P}_{-}$ case and CG we still get reasonable results and if one is not willing to invest in the scaling of the preconditioners the results given by ITFQMR are competitive compared to the ones obtained from the classical Bramble-Pasciak-setup, which is tailored for problems of this type. The setup where A is indefinite and C is definite shows a performance similar to the method of Forsgren *et. al.* whereas more flexibility is given. The final setup where A is indefinite and C is semi-definite is illustrating the fact that our setup outperforms MINRES with block-diagonal preconditioning by a large margin.

6.3 Scattering amplitude

6.3.1 Solving the linear system

In this Section, we want to show numerical experiments for the methods introduced in Chapter 5 that solve the linear system associated with the computation of the scattering amplitude or the primal linear output.

Example 6.12. *In the first example, we apply the* QMR *and the* GLSQR *methods to a randomly perturbed sparse identity of dimension* 100*, e.g.* `A=sprandn(n,n,0.2)+speye(n);` *in Matlab notation. The maximal iteration number for both methods is* 200 *and it can be observed in Figure 6.13 that* GLSQR *outperforms* QMR *for this example.*

Example 6.13. *The second example is a matrix from the Matrix Market collection*[1]*, the matrix ORSIRR1 which represents a linear system used in oil reservoir modelling. The matrix size is* 1030*. The results without preconditioning are shown in Figure 6.14. Results using the Incomplete LU (ILU) factorization with zero fill-in as a preconditioner for* GLSQR *and* QMR *are*

[1] http://math.nist.gov/MatrixMarket/

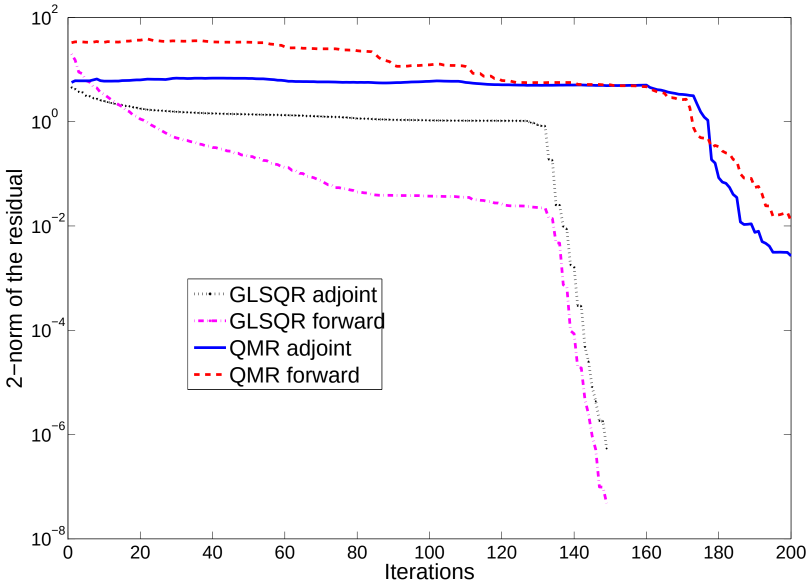

Figure 6.13: QMR and GLSQR for a matrix of dimension 100

given in Figure 6.15. Clearly, QMR *outperforms* GLSQR *in both cases. The choice of using ILU as a preconditioner is motivated mainly by the fact that we are not aware of more sophisticated implementations of Incomplete Orthogonal factorizations or Incomplete Modified Gram-Schmidt decompositions that can be used under MATLAB. Our tests with the basic implementations of cIGO and IMGS [96] did not yield better numerical results than the ILU preconditioner, and we have therefore omitted presenting these results. Nevertheless, we feel that further research in the possible use of Incomplete Orthogonal factorizations might result in useful preconditioners for* GLSQR.

Example 6.14. *The next example is motivated by [81] where Nachtigal et al. introduce examples that show how different solvers for nonsymmetric systems can outperform others by a large factor. The original example in their paper*

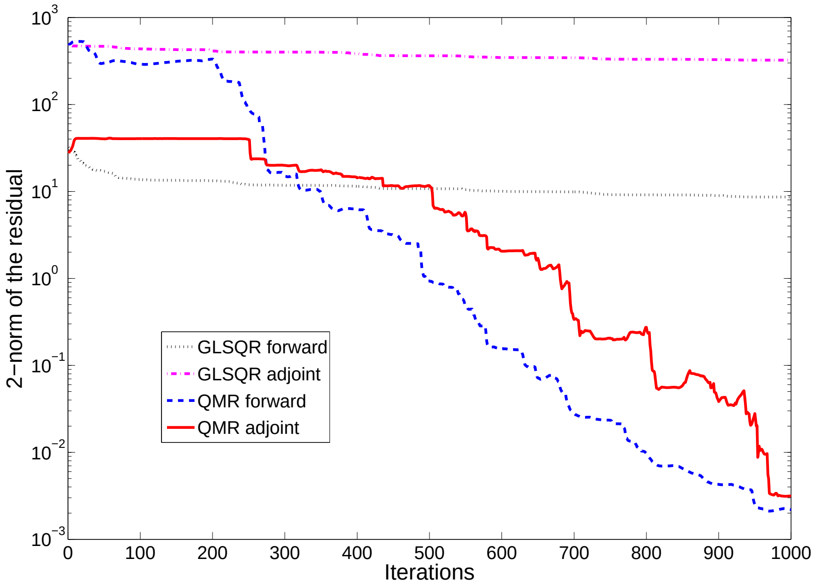

Figure 6.14: GLSQR and QMR for the matrix: orsirr_1.mtx

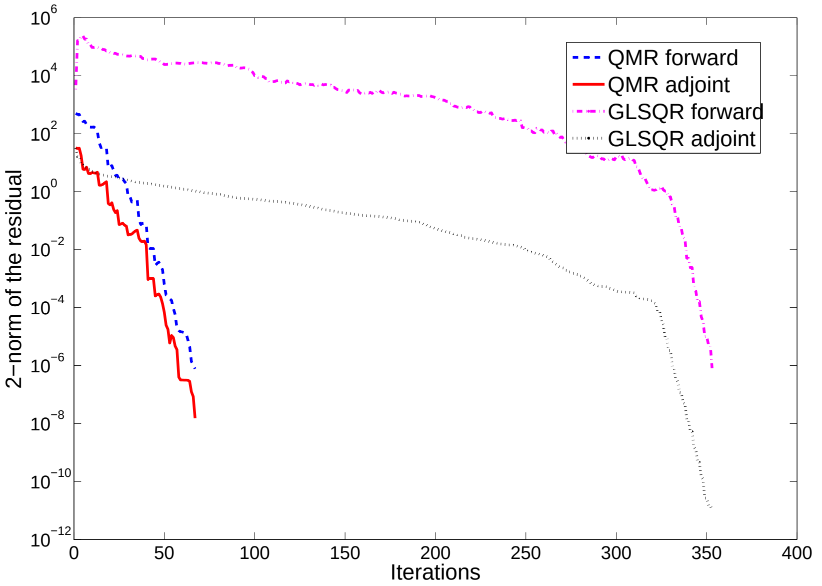

Figure 6.15: ILU preconditioned GLSQR and QMR for the matrix: orsirr_1.mtx

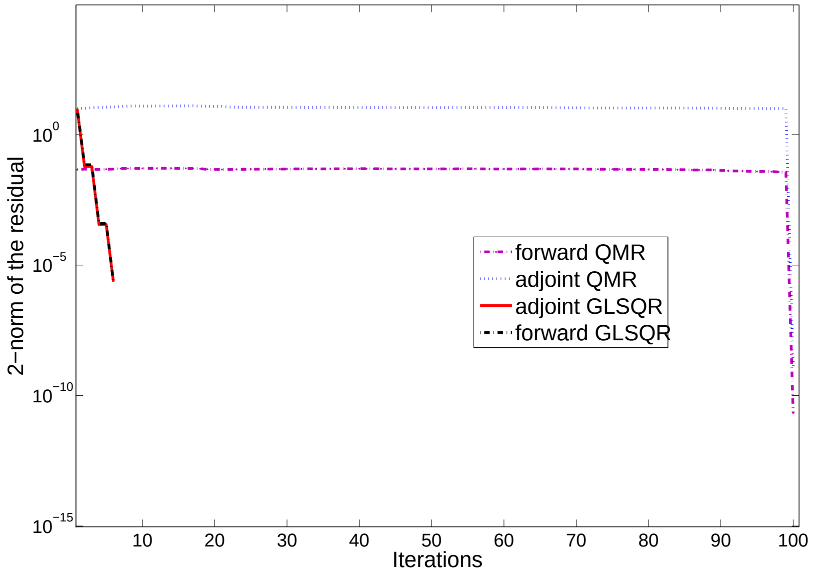

Figure 6.16: Perturbed circulant shift matrix (Example 6.14)

is given by the matrix

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

The results shown in Figure 6.16 are for a sparse perturbation of the matrix J, i.e. in Matlab notation `A=1e-3*sprandn(n,n,0.2)+J;`*. It is seen that* QMR *convergence for both forward and adjoint systems is slow, whereas* GLSQR *convergence is essentially identical for the forward and adjoint systems and is rapid.*

The convergence of GLSQR has not yet been analyzed, but we feel that using the connection to the Block-Lanczos process for $\mathcal{A}^T\mathcal{A}$ we can try to look for similarities to the convergence of CG for the normal equations (CGNE). It is well known [81] that the convergence of CGNE is governed by the singular values of the matrix $\mathcal{A}$. We therefore illustrate in the next example how the convergence of GLSQR is influenced by the distribution of the singular values of $\mathcal{A}$. This should not be seen as a concise description of the convergence behavior but rather as a starting point for further research.

Example 6.15. *In this example, we create a diagonal matrix* $\Sigma = \operatorname{diag}(D_1, D_2)$ *with*

$$D_1 = \begin{bmatrix} 1000 & & \\ & \ddots & \\ & & 1000 \end{bmatrix} \in \mathbb{R}^{p,p} \text{ and } D_2 = \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & \ddots & \\ & & & q \end{bmatrix} \in \mathbb{R}^{q,q}$$

with $p + q = n$. *We then create* $\mathcal{A} = U\Sigma V^T$ *where* U *and* V *are orthogonal matrices. For* $n = 100$ *the results of* GLSQR *for* $D_1 \in \mathbb{R}^{90,90}$, $D_1 \in \mathbb{R}^{10,10}$ *and* $D_1 \in \mathbb{R}^{50,50}$ *are given in Figure 6.17. There is better convergence when there are fewer distinct singular values. Figure 6.18 shows the comparison of* QMR *and* GLSQR *without preconditioning on an example with* $n = 1000$ *and* D_1 *of dimension* 600*; clearly* GLSQR *is superior in this example.*

It can be seen in this Section that QMR outperforms GLSQR on many of the given examples. Nevertheless, QMR is not always guaranteed to work because it is based on the non-symmetric Lanczos process and can hence break down (see Section 2.2.2). The results also emphasize that better preconditioning strategies should be developed to make GLSQR more competitive, and we see this as an interesting area for further research.

6.3.2 Approximating the functional

In this section, we want to present results for the methods that approximate the scattering amplitude directly, and hence avoiding the computation of approximate solutions for the linear systems with $\mathcal{A}$ and $\mathcal{A}^T$.

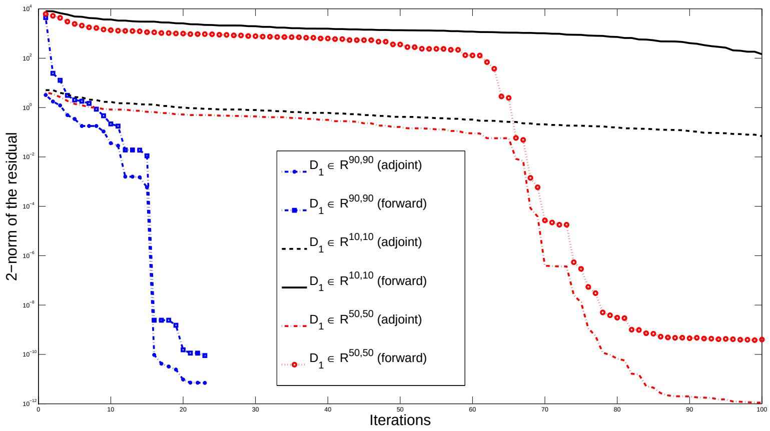

Figure 6.17: GLSQR for different D_1

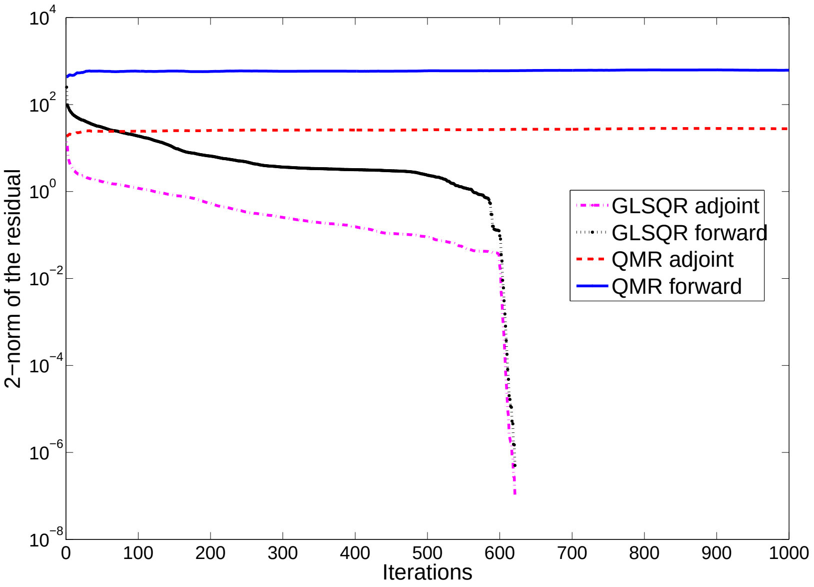

Figure 6.18: GLSQR and QMR for matrix of dimension 1000

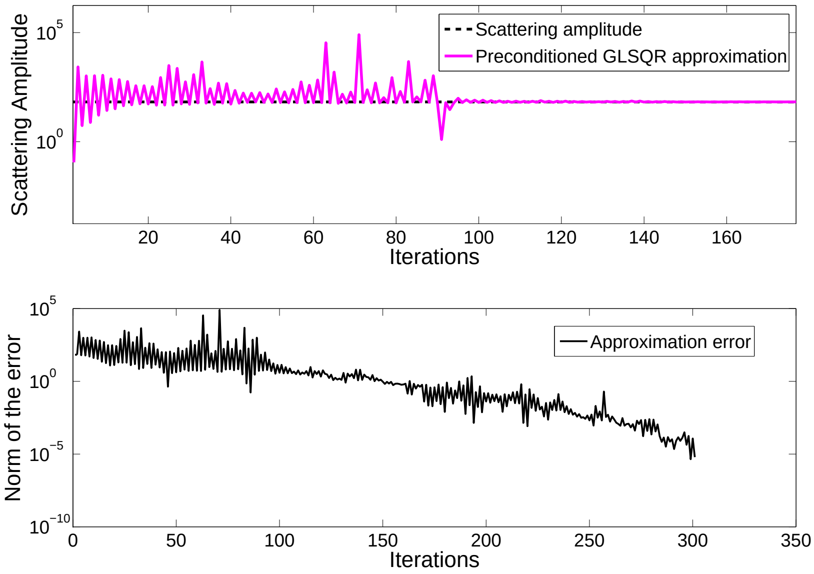

Figure 6.19: Approximations to the scattering amplitude and error (GLSQR)

Example 6.16. *In this example, we compute the scattering amplitude using the preconditioned* GLSQR *approach for the oil reservoir example ORSIRR1. The matrix size is* 1030*. We use the Incomplete LU (ILU) factorization as a preconditioner. The absolute values of the approximation from* GLSQR *are shown in the top part of Figure 6.19 and the bottom part shows the norm of the error against the number of iterations. Note that the non-monotonicity of the remainder term can be observed for the application of* GLSQR *.*

Example 6.17. *In this example, we compute the scattering amplitude using the preconditioned* BICG *approach for the oil reservoir example ORSIRR1. The matrix size is* 1030*. We once more use the Incomplete LU (ILU) factorization as a preconditioner. The absolute values of the approximation from* BICG *are shown in the top part of Figure 6.20 and the bottom part shows the norm of the error against the number of iterations.*

Example 6.18. *In this example, we compute the scattering amplitude by using the* LSQR *approach presented in Section 5.2.2. The test matrix is of*

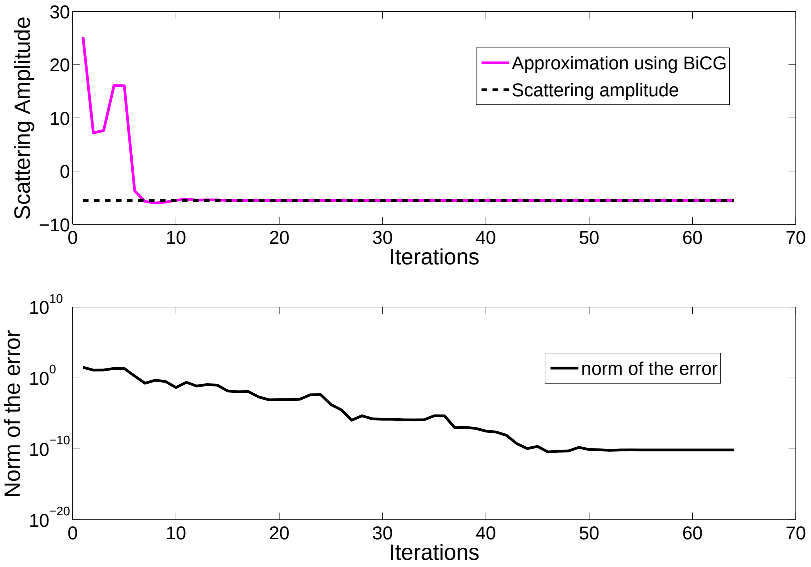

Figure 6.20: Approximations to the scattering amplitude and error (BICG)

size 187 × 187 *and represents a Navier-Stokes problem generated by the IFISS package [23]. The result is shown in Figure 6.21, again with approximations in the top part and the error in the bottom part.*

Again, for the examples computed in this section the method based on the non-symmetric Lanczos process – in this case BICG– was able to outperform the GLSQR based approximation of the scattering amplitude. Further research needs to be devoted to the task to find good preconditioners for GLSQR. Note that in [112] Strakoš and Tichý show more examples using GLSQR to approximate the scattering amplitude where its performance was competitive.

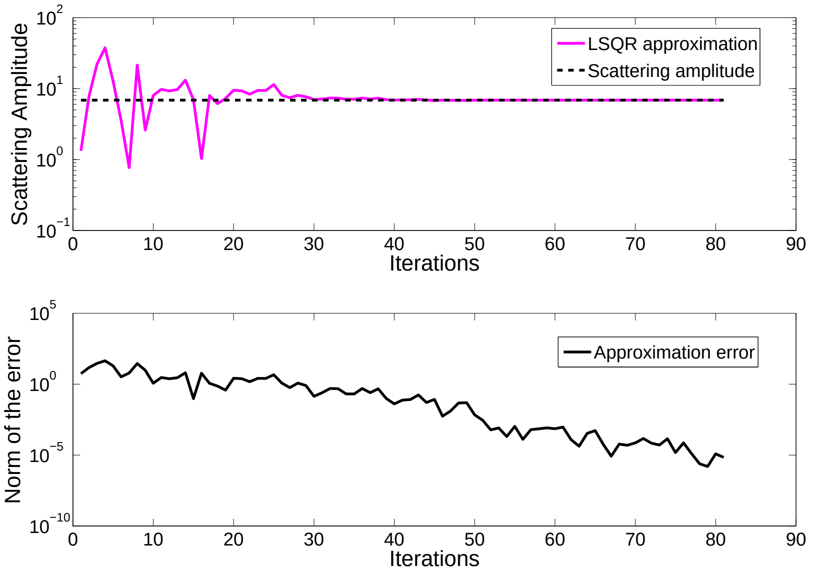

Figure 6.21: Approximations to the scattering amplitude and error (LSQR)

CHAPTER 7

CONCLUSIONS

We have explained the general concept of self-adjointness in non-standard inner products or symmetric bilinear forms, and in the specific case of saddle point problems have shown how a number of known examples fit into this paradigm. We have indicated how self-adjointness may be taken advantage of in the choice of iterative solution methods of Krylov subspace type. In general it is more desirable to be able to work with iterative methods for self-adjoint matrices compared to general non-symmetric matrices than non-symmetric iterative methods. This is because of the greater efficiency of symmetric iterative methods. The understanding of the convergence of symmetric iterative methods like CG is much more secure and descriptive than for non-symmetric methods.

The possibility of combination preconditioning by exploiting self-adjointness in different non-standard inner products or symmetric bilinear forms has been analyzed and examples given of how two methods can be combined to obtain a new preconditioner and a different symmetric bilinear form. The first example combines the new BP$^+$ method which we have introduced with the classical Bramble-Pasciak method. We demonstrate that a particular combination outperforms the widely used classical method; it requires fewer iterations while the work per iteration is the same. The second example is of more academic than practical value. The third example combines the BP method and a recently introduced method by Schöberl and Zulehner. The combination preconditioning method was able to outperform both the Bramble-Pasciak CG and Schöberl and Zulehner's CG method.

Our analysis may provide the basis for the discovery of further useful examples where self-adjointness may hold in non-standard inner products and also shows how preconditioning can usefully be employed to create rather than destroy symmetry of matrices.

Furthermore, we proposed a reformulation of the saddle point problem which represents a framework for many well known methods for solving saddle point problems. We employed this structure to introduce a Bramble-Pasciak-like method based on a recently introduced constrained preconditioning technique. This method gives competitive results when applied to problems coming from optimization whilst being less restrictive on the system matrix. This results in a greater flexibility of the method and we have seen that standard methods can be outperformed when applied to the same problem.

Moreover, we studied the possibility of using LSQR for the simultaneous solution of forward and adjoint problems which can be reformulated as problem in saddle point form. Due to the link between the starting vectors of the two sequences, this method did not show much potential for a practical solver. As a remedy, we proposed using the GLSQR method which we carefully analyzed, showing its relation to a Block-Lanczos method. Due to its special structure we are able to choose the two starting vectors independently and can therefore approximate the solutions for forward and adjoint system at the same time. Furthermore, we introduced preconditioning for the GLSQR method and proposed different preconditioners. We feel that more research has to be done to fully understand which preconditioners are well-suited for GLSQR, especially with regard to the experiments where different singular value distributions were used.

The approximation of the scattering amplitude without first computing solutions to the linear systems was introduced based on the Golub-Kahan bidiagonalization and its connection to Gauss quadrature. In addition, we showed how the interpretation of GLSQR as a Block-Lanczos procedure can be used to allow approximations of the scattering amplitude directly by using the connection to Block-Gauss quadrature.

We showed that for some examples the linear systems approach using GLSQR can outperform QMR which is based on the nonsymmetric Lanczos process and others where QMR performed better. We also showed how LSQR and GLSQR can be used to approximate the scattering amplitude on real world examples.

Future work based on this thesis should investigate the following points. The use of non-standard inner product solvers in PDE constrained optimization. This is particularly attractive since the eigenstructure of blocks of the saddle point matrix is well understood. In addition the solver presented by Schöberl and Zulehner is shown to be independent of the regularization parameter used in PDE constrained optimization. It should be investigated whether this can also be shown for the combination of this setup with the Bramble-Pasciak setup. Of course, further combinations should be sought after and analyzed when found. The performance of the Bramble-Pasciak-like method for indefinite left upper blocks in the saddle point system is particularly encouraging and this should be more carefully analyzed. The possibilities of using the general framework for the design of new methods should be investigated. Preconditioning for GLSQR is a major point that should be studied in the future since it will not only be beneficial for the approximation of the scattering amplitude but might also be used for general Schur-complement approximations.

APPENDIX A

EIGENVALUE ANALYSIS FOR BRAMBLE-PASCIAK-LIKE METHOD

We now want to show how a more general eigenvalue analysis can be made for the Bramble-Pasciak-like method presented in Chapter 4. The resulting bounds are not very practical and are hence not fully derived. We follow a technique used in Section 3.4 where the generalized eigenvalue problem $\mathcal{A}u = \lambda \mathcal{P}_{\pm} u$ is modified using the additional block-diagonal preconditioner

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ 0 & C_0 \end{bmatrix}.$$

Using $v = \mathcal{P}^{1/2}u$ we get $\mathcal{P}^{-1/2}\mathcal{A}\mathcal{P}^{-1/2}v = \lambda \mathcal{P}^{-1/2}\mathcal{P}^{\pm}\mathcal{P}^{-1/2}v$. This gives rise to a new generalized eigenvalue problem $\tilde{\mathcal{A}}v = \lambda \tilde{P}_{\pm}v$ with

$$\tilde{\mathcal{A}} = \begin{bmatrix} A_0^{-1/2}AA_0^{-1/2} & A_0^{-1/2}B^TC_0^{-1/2} \\ C_0^{-1/2}BA_0^{-1/2} & -C_0^{-1/2}CC_0^{-1/2} \end{bmatrix} = \begin{bmatrix} \tilde{A} & \tilde{B}^T \\ \tilde{B} & -\tilde{C} \end{bmatrix}$$

and

$$\tilde{\mathcal{P}}_{\pm} = \begin{bmatrix} I & -C_0^{-1/2} B^T A_0^{-1/2} \\ 0 & I \end{bmatrix} = \begin{bmatrix} I & -\tilde{B}^T \\ & \pm I \end{bmatrix}.$$

Hence, we get

$$\begin{aligned} \tilde{A}v_1 + \tilde{B}^T v_2 &= \lambda v_1 - \lambda \tilde{B}^T v_2 && \text{(A.1)} \\ \tilde{B}v_1 - \tilde{C}v_2 &= \pm \lambda v_2. && \text{(A.2)} \end{aligned}$$

Multiplying (A.1) on the left by v_1^*, where v_1^* denotes the conjugate transpose of v_1 gives,

$$v_1^* \tilde{A} v_1 + v_1^* \tilde{B}^T v_2 = \lambda v_1^* v_1 - \lambda v_1^* \tilde{B}^T v_2 \qquad \text{(A.3)}$$

and multiplying the conjugate transpose of (A.2) on the right by v_2 yields

$$v_1^* \tilde{B}^T v_2 - v_2^* \tilde{C} v_2 = \pm \bar{\lambda} v_2^* v_2. \qquad \text{(A.4)}$$

Finally, combining (A.3) and (A.4) results in

$$(1+\lambda)(v_2^* \tilde{C} v_2 \pm \bar{\lambda} v_2^* v_2) + v_1^* \tilde{A} v_1 - \lambda v_1^* v_1 = 0 \qquad \text{(A.5)}$$

which can be further rewritten using $\|v_1\|^2 = 1 - \|v_2\|^2$

$$v_2^* \tilde{C} v_2 + \lambda v_2^* \tilde{C} v_2 \pm \bar{\lambda} \|v_2\|^2 \pm |\lambda|^2 \|v_2\|^2 + v_1^* \tilde{A} v_1 - \lambda + \lambda \|v_2\|^2 = 0 \qquad \text{(A.6)}$$

From here on we can proceed in a similar way to Section 3.4 but feel that the resulting bounds on the eigenvalues are not of much practical use and are hence omitted.

BIBLIOGRAPHY

[1] M. Arioli, *A stopping criterion for the conjugate gradient algorithms in a finite element method framework*, Numer. Math, 97 (2004), pp. 1–24.

[2] D. Arnett, *Supernovae and Nucleosynthesis: An Investigation of the History of Matter, from the Big Bang to the Present*, Princeton University Press, 1996.

[3] W. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Q. Appl. Math, 9 (1951), pp. 17–29.

[4] Z.-Z. Bai, I. S. Duff, and A. J. Wathen, *A class of incomplete orthogonal factorization methods. I. Methods and theories*, BIT, 41 (2001), pp. 53–70.

[5] R. Becker and R. Rannacher, *A feed-back approach to error control in finite element methods: basic analysis and examples*, East West Journal of Numerical Mathematics, 4 (1996), pp. 237–264.

[6] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer, 14 (2005), pp. 1–137.

[7] M. Benzi and V. Simoncini, *On the eigenvalues of a class of saddle point matrices*, Numer. Math, 103 (2006), pp. 173–196.

[8] Å. Björck, *A bidiagonalization algorithm for solving ill-posed system of linear equations*, BIT, 28 (1988), pp. 659–670.

[9] Å. Björck and C. C. Paige, *Loss and recapture of orthogonality in the modified Gram-Schmidt algorithm*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 176–190.

[10] J. H. Bramble and J. E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp, 50 (1988), pp. 1–17.

[11] J. R. Bunch and B. N. Parlett, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal, 8 (1971), pp. 639–655.

[12] Z.-H. Cao, *A note on eigenvalues of matrices which are self-adjoint in symmetric bilinear forms*, SIAM J. Matrix Anal.Appl, accepted, (2008).

[13] S. H. Cheng and N. J. Higham, *A modified Cholesky algorithm based on a symmetric indefinite factorization*, SIAM J. Matrix Anal. Appl, 19 (1998), pp. 1097–1110.

[14] P. Concus and G. Golub, *Generalized conjugate gradient method for nonsymmetric systems of linear equations*, in 2. international symposium on computing methods in applied sciences and engineering, vol. 15, 1975.

[15] G. Dahlquist, S. C. Eisenstat, and G. H. Golub, *Bounds for the error of linear systems of equations using the theory of moments*, J. Math. Anal. Appl, 37 (1972), pp. 151–166.

[16] P. J. Davis and P. Rabinowitz, *Methods of numerical integration*, Computer Science and Applied Mathematics, Academic Press Inc, Orlando, FL, second ed., 1984.

[17] C. R. Dohrmann and R. B. Lehoucq, *A primal-based penalty preconditioner for elliptic saddle point systems*, SIAM J. Numer. Anal, 44 (2006), pp. 270–282.

[18] H. S. Dollar, N. I. M. Gould, M. Stoll, and A. Wathen, *A Bramble-Pasciak-like method with applications in optimization*, submitted to SIAM J. Sci. Computing, (2008).

[19] S. Dollar, *Iterative Linear Algebra for Constrained Optimization*, PhD thesis, University of Oxford, 2005. http://web.comlab.ox.ac.uk/oucl/research/na/thesis/thesisdollar.pdf.

[20] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct methods for sparse matrices*, Monographs on Numerical Analysis, The Clarendon Press Oxford University Press, New York, 1989.

[21] M. Eiermann and O. Ernst, *Geometric aspects of the theory of Krylov subspace methods*, Acta Numerica, 10 (2003), pp. 251–312.

[22] J. Elliott and J. Peraire, *Aerodynamic Optimization on Unstructured Meshes with Viscous Effects*, AIAA Paper, (1997), pp. 97–1849.

[23] H. Elman, A. Ramage, and D. Silvester, *IFISS: A Matlab Toolbox for Modelling Incompressible Flow.* www.manchester.ac.uk/ifiss.

[24] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.

[25] M. Embree, *The tortoise and the hare restart GMRES*, SIAM Rev, 45 (2003), pp. 259–266 (electronic).

[26] V. Faber, J. Liesen, and P. Tichý, *The Faber-Manteuffel theorem for linear operators*, SIAM J. Numer. Anal, 46 (2008), pp. 1323–1337.

[27] V. Faber and T. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal, 21 (1984), pp. 352–362.

[28] H. Fang and D. OLeary, *Modified Cholesky algorithms: a catalog with new approaches*, Mathematical Programming, (2008), pp. 1–31.

[29] B. Fischer, *Polynomial based iteration methods for symmetric linear systems*, Wiley-Teubner Series Advances in Numerical Mathematics, John Wiley & Sons Ltd, Chichester, 1996.

[30] B. Fischer, A. Ramage, D. J. Silvester, and A. J. Wathen, *Minimum residual methods for augmented systems*, BIT, 38 (1998), pp. 527–543.

[31] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical analysis (Proc 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975), Springer, Berlin, 1976, pp. 73–89. Lecture Notes in Math., Vol. 506.

[32] A. Forsgren, P. E. Gill, and J. D. Griffin, *Iterative solution of augmented systems arising in interior methods*, SIAM J. Optim, 18 (2007), pp. 666–690.

[33] R. Freund, *Transpose-free quasi-minimal residual methods for non-Hermitian linear systems*, Numerical analysis manuscript 92-97, 14 (1992), pp. 470 – 482. AT&T Bell Labs.

[34] R. Freund and H. Zha, *Simplifications of the nonsymmetric lanczos process and a new algorithm for Hermitian indefinite linear systems*, Numerical analysis manuscript, (1994). AT&T Bell Labs.

[35] R. W. Freund, *Transpose-free quasi-minimal residual methods for non-Hermitian linear systems*, in Recent advances in iterative methods, vol. 60 of IMA Vol. Math. Appl, Springer, New York, 1994, pp. 69–94.

[36] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput, 14 (1993), pp. 137–158.

[37] R. W. Freund and N. M. Nachtigal, *QMR: a quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math, 60 (1991), pp. 315–339.

[38] ——, *An implementation of the QMR method based on coupled two-term recurrences*, SIAM J. Sci. Comput, 15 (1994), pp. 313–337.

[39] ——, *Software for simplified Lanczos and QMR algorithms*, Appl. Numer. Math, 19 (1995), pp. 319–341.

[40] W. Gautschi, *Construction of Gauss-Christoffel quadrature formulas*, Math. Comp, 22 (1968), pp. 251–270.

[41] M. Giles and N. Pierce, *An Introduction to the Adjoint Approach to Design*, Flow, Turbulence and Combustion, 65 (2000), pp. 393–415.

[42] P. Gill, W. Murray, and M. Wright, *Practical optimization*, London: Academic Press, 1981, (1981).

[43] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM J. Matrix Anal. Appl, 13 (1992), pp. 292–311.

[44] I. Gohberg, P. Lancaster, and L. Rodman, *Indefinite linear algebra and applications*, Birkhäuser Verlag, Basel, 2005.

[45] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal, 2 (1965), pp. 205–224.

[46] G. Golub and D. O'Leary, *Some history of the conjugate gradient and Lanczos algorithms: 1948-1976*, SIAM Review, 31 (1989), pp. 50–102.

[47] G. H. Golub, *Some modified matrix eigenvalue problems*, SIAM Rev, 15 (1973), pp. 318–334.

[48] ——, *Bounds for matrix moments*, in Proceedings of the International Conference on Padé Approximants, Continued Fractions and Related Topics (Univ. Colorado, Boulder, Colo., 1972; dedicated to the memory of H. S. Wall), vol. 4, 1974, pp. 207–211.

[49] G. H. Golub and G. Meurant, *Matrices, moments and quadrature*, in Numerical analysis 1993 (Dundee, 1993), vol. 303 of Pitman Res. Notes Math. Ser., Longman Sci. Tech, Harlow, 1994, pp. 105–156.

[50] G. H. Golub and G. Meurant, *Matrices, moments and quadrature. II. How to compute the norm of the error in iterative methods*, BIT, 37 (1997), pp. 687–705.

[51] ——, *Matrices, moments and quadrature with applications.* Draft, 2007.

[52] G. H. Golub, M. Stoll, and A. Wathen, *Approximation of the scattering amplitude and linear systems*, ETNA, accepted, (2007).

[53] G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.

[54] G. H. Golub and J. H. Welsch, *Calculation of Gauss quadrature rules*, Math. Comp. 23 (1969), 221-230; addendum, ibid., 23 (1969), pp. A1–A10.

[55] N. Gould, D. Orban, and P. Toint, *CUTEr and SifDec: A constrained and unconstrained testing environment, revisited*, ACM Transactions on Mathematical Software (TOMS), 29 (2003), pp. 373–394.

[56] A. Greenbaum, *Iterative methods for solving linear systems*, vol. 17 of Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[57] W. Hackbusch, *Multigrid methods and applications*, vol. 4 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1985.

[58] L. A. Hageman and D. M. Young, *Applied iterative methods*, Dover Publications Inc., Mineola, NY, 2004. Unabridged republication of the 1981 original.

[59] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand, 49 (1952), pp. 409–436 (1953).

[60] N. J. Higham, *Functions of Matrices: Theory and Computation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

[61] I. Hnětynková and Z. Strakoš, *Lanczos tridiagonalization and core problems*, Linear Algebra Appl., 421 (2007), pp. 243–251.

[62] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge University Press, Cambridge, 1990.

[63] A. Jameson, *Aerodynamic design via control theory*, Journal of Scientific Computing, 3 (1988), pp. 233–260.

[64] C. Keller, N. Gould, and A. Wathen, *Constraint Preconditioning for Indefinite Linear Systems*, SIAM J. Matrix Anal. Appl, 21 (2000), pp. 1300–1317.

[65] A. Klawonn, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comput, 19 (1998), pp. 172–184. Special issue on iterative methods (Copper Mountain, CO, 1996).

[66] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Stand, 45 (1950), pp. 255–282.

[67] C. Lanczos, *Solution of systems of linear equations by minimized-iterations*, J. Research Nat. Bur. Standards, 49 (1952), pp. 33–53.

[68] L. Landau and E. Lifshitz, *Quantum mechanics*, Course of theoretical physics, (1965).

[69] J. Liesen, *A note on the eigenvalues of saddle point matrices*, Tech. Rep. 10-2006, TU Berlin, 2006.

[70] J. Liesen and B. N. Parlett, *On nonsymmetric saddle point matrices that allow conjugate gradient iterations*, Numer. Math, 108 (2008), pp. 605–624.

[71] J. Liesen and P. E. Saylor, *Orthogonal Hessenberg reduction and orthogonal Krylov subspace bases*, SIAM J. Numer. Anal, 42 (2005), pp. 2148–2158.

[72] J. Liesen and Z. Strakoš, *On optimal short recurrences for generating orthogonal Krylov subspace bases*, SIAM Rev, 50 (2008), pp. 485–503.

[73] J. Liesen and P. Tichý, *Convergence analysis of Krylov subspace methods*, GAMM-Mitt, 27 (2004), pp. 153–173.

[74] J. Lu and D. L. Darmofal, *A quasi-minimal residual method for simultaneous primal-dual solutions and superconvergent functional estimates*, SIAM J. Sci. Comput, 24 (2003), pp. 1693–1709.

[75] J. A. Meijerink and H. A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp, 31 (1977), pp. 148–162.

[76] G. Meurant, *Computer solution of large linear systems*, vol. 28 of Studies in Mathematics and its Applications, North-Holland Publishing Co, Amsterdam, 1999.

[77] G. Meurant and Z. Strakoš, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numer., 15 (2006), pp. 471–542.

[78] A. Meyer and T. Steidten, *Improvements and experiments on the Bramble-Pasciak Type CG for mixed problems in elasticity*, Tech. Rep. SFB393/01-13, TU Chemnitz, 2001.

[79] J. J. Moré and D. C. Sorensen, *On the use of directions of negative curvature in a modified Newton method*, Math. Programming, 16 (1979), pp. 1–20.

[80] N. Nachtigal, *A Look-ahead Variant of the Lanczos Algorithm and Its Application to the Quasi-minimal Residual Method for Non-Hermitian Linear Systems*, PhD thesis, Massachusetts Institute of Technology (MIT), 1991.

[81] N. M. Nachtigal, S. C. Reddy, and L. N. Trefethen, *How fast are nonsymmetric matrix iterations?*, SIAM J. Matrix Anal. Appl, 13 (1992), pp. 778–795.

[82] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, second ed., 2006.

[83] C. Paige and Z. Strakoš, *Correspondence between exact arithmetic and finite precision behaviour of krylov space methods.* The Householder Symposium on Numerical Linear Algebra(extended abstract), 1999.

[84] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal, 12 (1975), pp. 617–629.

[85] C. C. Paige and M. A. Saunders, *Algorithm 583; LSQR: sparse linear equations and least-squares problems*, ACM Trans. Math. Soft, 8 (1982), pp. 195–209.

[86] ——, *LSQR: an algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Soft, 8 (1982), pp. 43–71.

[87] A. T. Papadopoulos, I. S. Duff, and A. J. Wathen, *A class of incomplete orthogonal factorization methods. II. Implementation and results*, BIT, 45 (2005), pp. 159–179.

[88] B. N. Parlett, *The symmetric eigenvalue problem*, vol. 20 of Classics in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Corrected reprint of the 1980 original.

[89] B. N. Parlett, D. R. Taylor, and Z. S. Liu, *The look ahead Lánczos algorithm for large unsymmetric eigenproblems*, in Computing methods in applied sciences and engineering, VI (Versailles, 1983), North-Holland, Amsterdam, 1984, pp. 87–96.

[90] N. Pierce and M. Giles, *Adjoint Recovery of Superconvergent Functionals from PDE Approximations*, SIAM Rev, 42 (2000), pp. 247–266.

[91] L. Reichel and Q. Ye, *A generalized LSQR algorithm*, Numer. Linear Algebra Appl, in press, (2007).

[92] J. Reid, *The Use of Conjugate Gradients for Systems of Linear Equations Possessing Property A*, SIAM J. Numer. Anal, 9 (1972), p. 325.

[93] J. Reuther, A. Jameson, J. Alonso, M. Rimlinger, and D. Saunders, *Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 1*, Journal of Aircraft, 36 (1999).

[94] P. D. Robinson and A. J. Wathen, *Variational bounds on the entries of the inverse of a matrix*, IMA J. Numer. Anal, 12 (1992), pp. 463–486.

[95] M. Rozložník and V. Simoncini, *Krylov subspace methods for saddle point problems with indefinite preconditioning*, SIAM J. Matrix Anal.Appl, 24 (2002), pp. 368–391.

[96] Y. Saad, *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[97] Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput, 7 (1986), pp. 856–869.

[98] M. A. Saunders, *Cholesky-based methods for sparse least squares: the benefits of regularization*, in Linear and nonlinear conjugate gradient-related methods (Seattle, WA, 1995), SIAM, Philadelphia, PA, 1996, pp. 92–100.

[99] M. A. Saunders, H. D. Simon, and E. L. Yip, *Two conjugate-gradient-type methods for unsymmetric linear equations*, SIAM J. Numer. Anal, 25 (1988), pp. 927–940.

[100] P. E. Saylor and D. C. Smolarski, *Addendum to: "Why Gaussian quadrature in the complex plane?" [Numer. Algorithms* **26** *(2001), no. 3, 251–280; MR1832543 (2002a:65050)]*, Numer. Algorithms, 27 (2001), pp. 215–217.

[101] ——, *Why Gaussian quadrature in the complex plane?*, Numer. Algorithms, 26 (2001), pp. 251–280.

[102] J. Schöberl and W. Zulehner, *Symmetric indefinite preconditioners for saddle point problems with applications to pde-constrained optimization problems*, SIAM J. Matrix Anal. Appl, 29 (2007), pp. 752–773.

[103] J. Shewchuk, *An introduction to the conjugate gradient method without the agonizing pain*, unpublished paper.

[104] D. Silvester and A. Wathen, *Fast iterative solution of stabilised Stokes systems. II. Using general block preconditioners*, SIAM J. Numer. Anal, 31 (1994), pp. 1352–1367.

[105] V. Simoncini, *Block triangular preconditioners for symmetric saddle-point problems*, Appl. Numer. Math, 49 (2004), pp. 63–80.

[106] G. Sleijpen, H. van der Vorst, and J. Modersitzki, *Differences in the Effects of Rounding Errors in Krylov Solvers for Symmetric Indefinite Linear Systems*, SIAM J. Matrix Anal. Appl, 22 (2001), pp. 726–751.

[107] P. Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 36–52.

[108] D. C. Sorensen, *Numerical methods for large eigenvalue problems*, Acta Numer., 11 (2002), pp. 519–584.

[109] M. Stoll and A. Wathen, *Combination preconditioning and the Bramble–Pasciak$^+$ preconditioner*, SIAM J. Matrix Anal. Appl, 30 (2008), pp. 582–608.

[110] Z. Strakoš and P. Tichý, *On error estimation in the conjugate gradient method and why it works in finite precision computations*, Electron. Trans. Numer. Anal, 13 (2002), pp. 56–80.

[111] ——, *Error estimation in preconditioned conjugate gradients*, BIT, 45 (2005), pp. 789–817.

[112] ——, *Estimation of $c^*A^{-1}b$ via matching moments*, submitted to SIAM J. Sci. Comput, (2008).

[113] D. Venditti and D. Darmofal, *Adjoint Error Estimation and Grid Adaptation for Functional Outputs: Application to Quasi-One-Dimensional Flow*, J Comput Phys, 164 (2000), pp. 204–227.

[114] A. Wathen and D. Silvester, *Fast iterative solution of stabilised Stokes systems. I. Using simple diagonal preconditioners*, SIAM J. Numer. Anal, 30 (1993), pp. 630–649.

[115] P. Wesseling, *An introduction to multigrid methods*, Pure and Applied Mathematics (New York), John Wiley & Sons Ltd., Chichester, 1992.

[116] O. Widlund, *A Lanczos Method for a Class of Nonsymmetric Systems of Linear Equations*, SIAM J. Numer. Anal, 15 (1978), p. 801.

[117] W. Zulehner, *Analysis of iterative methods for saddle point problems: a unified approach*, Math. Comp, 71 (2002), pp. 479–505.