Matematicko - fyzikální fakulta
Univerzity Karlovy, Praha

Disertační práce

Modern methods for solving linear systems

Erik Jurjen Duintjer Tebbens

Školitel:	doc. RNDr. Jan Zítko, CSc.
Konzultant:	doc. RNDr. Karel Najzar, CSc.
Obor:	Vědecko-technické výpočty

Abstract

In the present thesis we show that we can accelerate the convergence speed of restarted GMRES processes with the help of rank-one updated matrices of the form $\mathbf{A} - by^T$, where $\mathbf{A}$ is the system matrix, b is the right-hand side and y is a free parameter vector. Although some attempts to improve projection methods with rank-one updates of a different form have been undertaken (for example in Eirola, Nevanlinna [17] or in Weiss [76] and Wagner [71]), our approach, based on the Sherman-Morrison formula, is new. It allows to solve a parameter dependent auxiliary problem with the same right-hand side but a different system matrix. Regardless of the properties of $\mathbf{A}$ we can force any convergence speed of the second system when the initial guess is zero. Moreover, reasonable convergence speed of the second system is able to overcome stagnation of the original problem. This has been tested on different kinds of problems from practice. The computation of the parameter vector $y \in \mathbb{R}^n$ as well as computations with the rank-one updated matrix add only little costs to the restarted method.

When the initial guess is nonzero (for example at the end of restart cycles), we minimize residual norms over all possible parameter vectors. Stepwise minimization does not seem lucrative, but theoretical investigation of global minimization shows that we can project implicitly on subspaces of a dimension twice as large as the iteration number. In addition, we combine stepwise minimization with a preconditioning technique with the help of updated matrices of the form $\mathbf{A} - \mathbf{A}dy^T$ for some $d \in \mathbb{R}^n$. In numerical experiments it proved to be able to overcome stagnation of restarted GMRES.

We have also worked out results about the spectrum of the rank-one updated matrix. In theory, one can create any spectrum of $\mathbf{A} - by^T$ by the choice of the parameter vector $y \in \mathbb{R}^n$. In practice it is only feasible to prescribe Ritz values of the auxiliary matrix. But when we assume we have a nearly normal matrix also modification of selected eigenvalues can be achieved. Based on these ideas, we have constructed algorithms for both normal and nonnormal matrices. In problems where spectral properties hampered convergence, these techniques could accelerate the GMRES process.

Contents

Notations

In this thesis we will use the following abbreviations:

$\mathbf{A}^T$:= the transposed matrix $\mathbf{A}$

$\mathbf{A}^H$:= the transposed and complex conjugated matrix $\mathbf{A}$

$\mathbf{A}^{-2}$:= $\mathbf{A}^{-1} \cdot \mathbf{A}^{-1}$

$\mathrm{diag}(\mathbf{A})$:= the diagonal matrix with the diagonal elements of $\mathbf{A}$

$\mathrm{diag}(\lambda_1, \ldots, \lambda_n)$:= the diagonal matrix with elements $\lambda_1, \ldots, \lambda_n$

x^T := the transposed vector x

$x^T y$:= the Euclidean inner product of two real vectors

$\|x\|$:= the square root $\sqrt{x^T x}$

$\|\mathbf{A}\|$:= the matrix norm associated with the above vector norm

$\|\mathbf{A}\|_F$:= the Frobenius norm of $\mathbf{A}$

$\mathbf{I}_n$:= the identity matrix of dimension n, $\mathbf{I}_0$ being the empty matrix

e_i := the ith column of the identity matrix of the involved space

$\dim(\mathcal{W})$:= the dimension of the subspace $\mathcal{W}$

$\mathrm{rank}(\mathbf{A})$:= the rank of the matrix $\mathbf{A}$

$a_{i,j}$:= the element of the matrix $\mathbf{A}$ in its ith row and in its jth column

$\{x_1, \ldots, x_m\}$:= the sequence of vectors $x_1, \ldots, x_m$

$(x_1, \ldots, x_m)$:= the matrix whose columns are the vectors $x_1, \ldots, x_m$

$\sigma(\mathbf{A})$:= the spectrum of the matrix $\mathbf{A}$

$\bar{\lambda}$:= the complex conjugated value $\lambda \in \mathbb{C}$

$\bar{x}$:= the complex conjugated vector $x \in \mathbb{C}^n$

$\mathrm{Re}(x)$:= the real part of a complex value or vector

$\mathrm{Im}(x)$:= the imaginary part of a complex value or vector

Preface

In modern scientific computation solvers of linear and nonlinear systems of equations are tools of primary importance. For both problems a large scale of methods has been proposed, but many theoretical and practical questions remain unanswered. In this work we consider the solution of a system of linear equations characterized by a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, a right-hand side $b \in \mathbb{R}^n$ and an equation

$$\mathbf{A}x = b \tag{1}$$

with unknown $x \in \mathbb{R}^n$. It focusses on applications where $\mathbf{A}$ is a general nonsingular matrix, without assuming symmetry or other special properties. Leaving aside direct solvers and preconditioning techniques, we concentrate on iterative methods, especially on projection methods based on Krylov subspaces. We try to improve methods that belong to this class and that in addition cannot be implemented with short recurrences. They are more robust than methods allowing short recurrences, but in modern problems they have to be restarted to avoid too large computational and storage costs, thus loosing convergence properties. The present thesis specializes on techniques to accelerate convergence speed of restarted Krylov subspace methods and proposes new approaches to achieve this. The new approaches were inspired by a series of papers by Arioli, Greenbaum, Pták and Strakoš ([2], [30] and [31]). It proves the existence of matrices and right-hand sides yielding a prescribed convergence curve when the GMRES method is applied. In addition, it shows how to construct such linear systems and it is possible to prescribe the spectrum of the matrix too. In this thesis we show that we can define a small rank update of $\mathbf{A}$ which belongs to the class of linear systems with a given convergence speed. Alternatively we can choose the updated matrix such that it has arbitrary spectrum. It is possible to exploit the updated system for solving the original system (1) with the help of the Sherman-Morrison formula for inverting rank-m updates, $m \leq n$. Favorable properties of the updated system appear to accelerate the first system too. In the present work we restrict ourselves wittingly to a very precise exploitation of the Sherman-Morrison formula. It offers many acceleration options and gives raise to theoretical questions. Other exploitations have not been investigated although they might be as interesting.

In the first chapter we give a survey of the most popular projection methods. We point out advantages of projection compared with other strategies in the context of solving the problem (1) and we list some important properties of projection methods in general. Among others, we turn our attention to the choice of the Petrov-Galerkin orthogonality condition and to the choice of the space to project onto. It turns out Krylov subspaces are in some sense optimal when solving linear systems with the help of projections. The remaining of the chapter describes well-known projection methods, ranging from the classical conjugate gradient method that dates from back in the early fifties to recently proposed techniques to improve the restarted GMRES method. We apply some of the methods in numerical examples in order to compare them with our new approaches. All methods are treated explicitly from the projectional point of view without addressing implementational details such as the generation of bases, solution of least-squares problems or Hessenberg systems (algorithms connected with bases of the individual methods are displayed in the last chapter). We feel that, although the role of projective processes in Krylov subspace methods has been studied by others (e.g. in Eiermann, Ernst, Schneider [16], [18]), these results stay on a relatively abstract level, especially when the involved projection has a complicated character. On the other hand, the projection that underlies a method is the only key to convergence behavior if no information about structural, spectral or other properties of the matrix is a priori available. We have tried to give for every method a detailed description of the projector that characterizes it. We also treat truncated, restarted and accelerated restarted methods in this manner and even methods that were originally not conceived as projection methods. A more profound description of some of these methods which includes comparing numerical experiments can be found in our papers [13] and [65] published together with J. Zítko (the second one is written in Czech). The first chapter does not contain any original results apart from the generalization of a residual based version of the GMRES method (Walker, Zhou [74]) for other methods and a small supplement to a deflation technique. This part of the thesis consists mainly of elaboration of the theory of projective processes for iterative methods.

The remaining chapters propose new techniques to overcome slow convergence or stagnation of restarted projection methods. In the second chapter we present the main tool all these techniques are based on: The Sherman-Morrison theorem. With its help we can transform the original problem to an auxiliary problem with the same right-hand side. The matrix of the auxiliary system is a rank-one update of $\mathbf{A}$. We prove that when the initial guess $x_0 \in \mathbb{R}^n$ of a GMRES process is zero, then it is possible to define this rank-one update such that the process has *any* prescribed convergence speed. After the presentation of numerically stable implementations to compute such an update, we address backtransformation of the auxiliary system approximations to obtain iterates for the original system (1). We discuss possibilities to improve convergence of the first system by backtransformation from a predefined, fast converging second system. It turns out this makes sense only when we apply the *restarted* GMRES method, but in that case the proposed procedure seems reasonable and its effectiveness is confirmed by numerical experiments of various types. In all cases the procedure manages to overcome stagnation of restarted GMRES. With the help of a technical proposition we demonstrate the relation between the quality of the backtransformation and the prescribed convergence speed of the auxiliary system. This proposition enables us to influence the quality of the backtransformation during the iterative process.

The idea of switching to a system with arbitrary convergence speed is possible only when the initial guess is zero. We discuss GMRES processes with nonzero initial guesses in the third chapter. At first, we show how to minimize the residual norms of the second system at every single iteration by the choice of the rank-one update of $\mathbf{A}$. An example with an auxiliary system that is minimal in this sense is presented, but in general this locally minimizing technique is not all too effective. It seems more profitable to search for rank-one updates that yield after say k iterations a residual norm that is globally minimal, i.e. minimal regardless of the size of residual norms of previous iterations. We demonstrate how to compute such an update and prove that (under weak conditions) the rank-one updated system implicitly projects during the kth iteration onto a subspace of dimension $2k$. The last part of this chapter concerns a different technique for processes with $x_0 \neq 0$. It does minimize residual norms with the help of the Sherman-Morrison formula, but the involved rank-one update has a different structure. It is in fact a right preconditioning technique. This has the advantage that residual norms of auxiliary and backtransformed iterations are equal. Some of the ideas of the second and third chapter will be published in the paper [14].

Although the preceding accelerations of the restarted GMRES method are based on direct reducing of residual norms rather than on spectral deflation, practical problems exist in which modification of eigenvalues has a positive influence on convergence speed. This has been our motivation to address spectral properties of the rank-one updated matrix in the fourth chapter. Among others we prove that we can force any spectrum by the choice of the update. As an immediate consequence of this result it is possible to define arbitrary Ritz values of the auxiliary matrix at the beginning of a GMRES process and we can exploit this fact for deflation purposes. On the other hand, direct modification of the eigenvalues of $\mathbf{A}$ itself by rank-one update seems hard to handle in practice. But for normal or close to normal matrices we propose a deflative procedure that modifies precisely located eigenvalues and it has shown to be effective in numerical examples.

As mentioned before, we have placed concrete algorithms at the end of the thesis. The last chapter displays algorithms to generate bases of well-known methods. Theoretical properties, as far as they are not trivial, accompany the algorithms. For example, a detailed description of the properties of the look-ahead Lanczos process is given. In addition, we have constructed algorithms for the techniques proposed in chapters two, three and four. These new algorithms are followed by a brief discussion of computational and storage costs. Finally, we apply all new methods to a sample numerical experiment. Other numerical examples illustrate the new procedures in the previous chapters, immediately after their theoretical description.

Prague, March 2004.

Acknowledgments

In the first place I want to thank my supervisor, Jan Zítko, for his devoted and guiding supervision of my doctoral studies. He awoke slumbering mathematical ambitions with his encouraging and never resting enthusiasm. I very much appreciate the large interest he took in my work and am grateful for numerous remarks, corrections and comments. I am also pleased that he allowed at consultations other items than mathematics alone.

I would like to thank Zdeněk Strakoš and Petr Tichý for discussions that enhanced my insight in the themes of the present work. The former as well as Prof. Křížek enabled me to participate at various mathematical activities. I was also stimulated by many remarks of Thomas Korn, a hint of Miroslav Rozložník and lectures of Prof. Reusken. I am indebted to Jan Brandts for the critical reading and useful remarks concerning the paper [14]. I thank Radboud Duintjer Tebbens for corrections concerning the English and the formulation of this thesis.

I wish to thank my parents. Their assistance has been far from restricted to material matters. I cannot think of any situation where they did not support me. My stay in Czechia as a foreigner so far was agreeable thanks to mainly Michal Synek, Pavel Pisk, Valtrauda Fotterová, the Prague Singers, Veronika Hyksová and Jiřína Marešová, who certainly stimulated my inventiveness. I want to thank Egon Mars, Thomas Korn, Iona Hoogenberk and my brother for their friendship during the past years. Finally, I cannot help mentioning a Lady that never failed to comfort me and to remind me of the limitations of cerebral activity. Her name is Music.

Chapter 1

Projection methods

One of today's most popular classes of methods to solve linear systems are projection methods. In this chapter we point out some of the advantages of projection compared with other strategies and we will search for projections that are optimal in the context of solving linear equations. This will lead in a natural way to projection with Krylov subspaces. The chapter provides a brief survey of these projection methods, ranging from classical Krylov subspace methods to modern techniques that try to improve the classical ones. In addition we treat some methods that are not based on Krylov subspaces.

In order to compare projection techniques with different procedures to solve linear systems, let us start at the very beginning.

1.1 Introduction

We consider a system of linear equations characterized by a nonsingular matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, a right-hand side $b \in \mathbb{R}^n$ and an equation

$$\mathbf{A}x = b \tag{1.1}$$

with unknown $x \in \mathbb{R}^n$. In this thesis we will consider real problems, but generalization of the results for the complex case is straightforward. The linear system (1.1) can be preconditioned from the left by replacing it by the system

$$\mathbf{M}_l \mathbf{A} x = \mathbf{M}_l b,$$

where $\mathbf{M}_l \in \mathbb{R}^{n \times n}$ is a nonsingular matrix that yields an easier to solve linear problem. Alternatively, we can precondition from the right by solving

$$\mathbf{A}\mathbf{M}_r y = b,$$

where $\mathbf{M}_r \in \mathbb{R}^{n \times n}$, $y \in \mathbb{R}^n$, and where we compute x from $x = \mathbf{M}_r y$, or we can combine left and right preconditioning. For the moment, however, we assume system (1.1) is already written in preconditioned form. The exact solution, denoted by $x^* \in \mathbb{R}^n$, can be found either with a direct method or by applying an iterative method. Direct solvers calculate the inverse of $\mathbf{A}$ by decomposing $\mathbf{A}$ in easily invertible factors, such as triangular or orthogonal matrices. An iterative method finds approximate solutions of (1.1) by successively defining iterates. Starting with a first approximation, the initial guess $x_0 \in \mathbb{R}^n$, iterates can be written in the form

$$x_k := f_k(x_0, \ldots, x_{k-1}), \quad k = 1, 2, \ldots$$

where f_k is some function with domain $\mathbb{R}^{n\times k}$ and range $\mathbb{R}^n$. The two approaches can be combined when we use an approximation to $\mathbf{A}^{-1}$ given by the incomplete factorization of a direct solver as preconditioner and we apply an iterative method to the preconditioned system.

The computational costs of direct solvers can grow unacceptably high if the dimension of the system is large. For example, the number of operations needed to find x^* with Gauss elimination or LU-decomposition is of order n^3. An advantage of iterative methods is that they exploit possible sparsity of $\mathbf{A}$, whereas direct solvers have more difficulties to do so. Large matrices arising in modern computations often have an advantageous structure that is easily lost by direct solvers due to the fill-in they produce. In addition, when using a direct method the solution is found as late as at the very end of the computation and no approximate solution is available at an earlier stage. In practice we are often satisfied with an approximation of x^* because the original problem (a mathematical model of some scientific problem) has already been approximated and discretized to obtain the system (1.1). This approximation can best be obtained by an iterative process that seeks to reduce the distance of the iterates to x^*.

In the remaining of the thesis we restrict ourselves to iterative methods. Two frequently used vectors to express the quality of the iterative vectors during the process are the error and the residual vector.

Definition 1.1.1 *The k-th residual vector of an iterative method for (1.1) is the vector*

$$r_k := b - \mathbf{A}x_k.$$

The k-th error vector is the difference between the exact solution and the k-th iterate

$$d_k := x^* - x_k.$$

Thus we have the following relationship between residual and error vector:

$$r_k = \mathbf{A}d_k. \tag{1.2}$$

Note that when the exact solution is not known, the error vector d_k cannot be computed whereas the residual vector is always available. For this reason it is in practice the norm of the residual vector that is the indicator for convergence speed, although the norm of the error can be significantly larger when $\|\mathbf{A}\|$ is small in comparison with $\|r_k\|$. When convergence is achieved, however, the discrepancy vanishes. Moreover, many methods are implicitly based on values related to the residual vector rather than to the error vector. A commonly used procedure, for example, consists of adding to approximations a correction vector that depends on the associated residual vector. If the corresponding error vector were available we could use it as correction vector and immediately obtain x^*, but instead we must be satisfied with recursions of the form

$$x_k := \hat{x}_{k-1} + g_k(\hat{r}_{k-1}), \quad k = 1, 2, \ldots \tag{1.3}$$

where $\hat{x}_{k-1} \in \{x_0, \ldots, x_{k-1}\}$, g_k is some function with range in $\mathbb{R}^n$ and

$$\hat{r}_{k-1} = b - \mathbf{A}\hat{x}_{k-1}. \tag{1.4}$$

The action of g_k on $\hat{r}_{k-1}$ can usually be represented by matrix multiplication. For residual vectors this yields the recursion

$$r_k = \hat{r}_{k-1} - \mathbf{A}\mathbf{G}_k\hat{r}_{k-1} = (\mathbf{I}_n - \mathbf{A}\mathbf{G}_k)\hat{r}_{k-1},$$

with $\mathbf{G}_k$ being a matrix representation of the operator g_k. To stimulate convergence, it is useful to have operators $\mathbf{G}_k$ satisfying

$$\|\mathbf{I}_n - \mathbf{A}\mathbf{G}_k\|_N < 1 \tag{1.5}$$

in some norm $\|\cdot\|_N$. The matrix $\mathbf{G}_k$ can be an approximation in some sense of $\mathbf{A}^{-1}$ and is possibly independent from k (the Jacobi method uses $\mathbf{G}_k := (\mathrm{diag}(\mathbf{A}))^{-1}$) or it can be successively updated to yield better approximations of $\mathbf{A}^{-1}$ (see for example Eirola and Nevanlinna [17]).

But finding accurate approximations of $\mathbf{A}^{-1}$ is inefficient when approximations of only $\mathbf{A}^{-1}b = x^*$ can be found more easily. An alternative way to diminish the length of residual vectors consists of extracting them from an affine subspace of comparatively small dimension and imposing on them some optimality condition with respect to the subspace. Convergence can then be stimulated by letting the subspace dimension grow with every iteration or by advantageous choice of the subspace or by a combination of these two. A large class of currently used methods based on this idea is the class of so-called projection methods.

1.2 General remarks about projection methods

In analogy with (1.3), projection methods can be defined as follows.

Definition 1.2.1 *A projection method is an iterative method with successive residual vectors that satisfy*

$$r_k = \hat{r}_{k-1} - \wp_k(\hat{r}_{k-1}), \quad k = 1, 2, \ldots,$$

where $\hat{r}_{k-1} \in \{r_0, \ldots, r_{k-1}\}$ and where the operator $\wp_k$ is a projector.

We define a projector $\wp$ as an operator satisfying $\wp \circ \wp = \wp$. The projector $\wp_k$ does not need to be an orthogonal projector onto the involved subspace. In the remaining of this chapter we will denote matrix representations of $\wp_k$ by $\mathbf{P}_k$, the m-dimensional subspace, $m \leq n$, that $\wp_k$ projects onto, the *projection space*, by $\mathcal{W}_m$ and if the projection process is orthogonal to a space different from $\mathcal{W}_m$, that space, the *test space*, will be denoted by $\mathcal{V}_m$. In the literature a projection orthogonal to $\mathcal{V}_m$ is also called a projection along $\mathcal{V}_m^\perp$ and it is understood as a projection where the difference between the vector to project and its projection is orthogonal to $\mathcal{V}_m$. If the columns of $\mathbf{B}_m \in \mathbb{R}^{n\times m}$, $m \leq n$, form a basis of the test space $\mathcal{V}_m$ and those of $\mathbf{W}_m \in \mathbb{R}^{n\times m}$ a basis of the projection space $\mathcal{W}_m$, the kth residual vector of a projection method has the form

$$r_k = \hat{r}_{k-1} - \mathbf{W}_m z_m, \quad z_m \in \mathbb{R}^m,$$

and must satisfy

$$\mathbf{B}_m^T(\hat{r}_{k-1} - \mathbf{W}_m z_m) = 0. \tag{1.6}$$

The projection exists as long as $\det(\mathbf{B}_m^T\mathbf{W}_m) \neq 0$ and in that case

$$z_m = (\mathbf{B}_m^T\mathbf{W}_m)^{-1}\mathbf{B}_m^T\hat{r}_{k-1}.$$

The assumption $\det(\mathbf{B}_m^T\mathbf{W}_m) \neq 0$ holds if and only if no nonzero vector from $\mathcal{W}_m$ is orthogonal to the test space $\mathcal{V}_m$. Otherwise, r_k is not defined. If $\mathcal{V}_m = \mathcal{W}_m$, the projection is orthogonal and exists if $\mathcal{W}_m$ has full dimension. In that case, in mathematical methods of different kinds, condition (1.6) is referred to as the *Galerkin orthogonality condition.* When $\mathcal{V}_m \neq \mathcal{W}_m$ we will speak of the *Petrov-Galerkin orthogonality condition.* Because of

$$r_k = \hat{r}_{k-1} - \mathbf{P}_k \hat{r}_{k-1}$$

the matrix representation of $\wp_k$ with respect to $\mathbf{B}_m$ and $\mathbf{W}_m$ is

$$\mathbf{P}_k = \mathbf{W}_m (\mathbf{B}_m^T \mathbf{W}_m)^{-1} \mathbf{B}_m^T. \tag{1.7}$$

Trivially, matrix representations of a projector depend on the chosen bases for projection and test space.

In projection methods the error vector

$$\hat{d}_{k-1} := x^* - \hat{x}_{k-1}, \quad \hat{x}_{k-1} \in \{x_0, \ldots, x_{k-1}\},$$

is seen to be projected onto $\mathbf{A}^{-1}\mathcal{W}_m$ and the projector is orthogonal to $\mathbf{A}^T\mathcal{V}_m$ because of (1.2). Indeed, the kth error vector satisfies

$$d_k = \mathbf{A}^{-1}(\hat{r}_{k-1} - \mathbf{P}_k(\mathbf{A}\hat{d}_{k-1})) = \hat{d}_{k-1} - \mathbf{A}^{-1}\mathbf{P}_k\mathbf{A}\hat{d}_{k-1}. \tag{1.8}$$

The operator $\mathbf{A}^{-1}\mathbf{P}_k\mathbf{A}$ is in general an oblique projector, even when $\mathbf{P}_k$ is orthogonal because the matrix $\mathbf{A}^{-1}\mathbf{P}_k\mathbf{A}$ is not symmetric when $\mathbf{A}$ is not orthogonal.

When we consider convergence speed according to (1.5) we obtain a very pessimistic bound, namely

$$\|r_k\|_N = \|(\mathbf{I}_n - \mathbf{P}_k)\hat{r}_{k-1}\|_N \leq \|\mathbf{I}_n - \mathbf{P}_k\|_N \cdot \|\hat{r}_{k-1}\|_N.$$

With $\mathbf{I}_n - \mathbf{P}_k$ being a projector too, we have

$$\|\mathbf{I}_n - \mathbf{P}_k\|_N \geq 1$$

for projection methods and thus convergence is not guaranteed. The situation looks a little better if the projections are orthogonal, because in that case, with the Euclidean norm,

$$\|\mathbf{I}_n - \mathbf{P}_k\| = 1.$$

But of course, methods that project orthogonally can be characterized by the minimization property

$$\|r_k\| = \min_{w \in \mathcal{W}_m} \|\hat{r}_{k-1} - w\|. \tag{1.9}$$

Thus growth of the subspace dimension, m, will prevent residual norms from increasing. A similar norm minimizing property for oblique projections cannot be formulated, but the residuals produced by oblique projections are closely related to those of their orthogonal counterparts, as we will see later on (Theorem 1.3.2). The way of computing iterates with oblique processes can be cheaper than by orthogonal projection and that makes them attractive enough for practical use.

1.2.1 The projection space

Let us take a closer look at the projection space and try to gain some insight in how to choose it best in the context of solving linear systems. An essential concern of residual-based methods is to ensure inexpensive computation of iterates from residuals, though it might in some cases be possible to postpone this computation to the very end of the process when the residual norm is as small as desired. In projection methods residual and iterate are connected by

$$r_k = \hat{r}_{k-1} - \mathbf{W}_m z_m = b - \mathbf{A}x_k = b - \mathbf{A}(\hat{x}_{k-1} + \mathbf{A}^{-1}\mathbf{W}_m z_m)$$

because of (1.4) and we obtain for the corresponding iterate

$$x_k = \hat{x}_{k-1} + \mathbf{A}^{-1}\mathbf{W}_m z_m \tag{1.10}$$

because $\mathbf{A}$ is nonsingular. Unless a basis of $\mathbf{A}^{-1}\mathcal{W}_m$ is available, computing the approximations of projection methods from their residuals thus asks for full invertion of $\mathbf{A}$, which is precisely what we wish to avoid ! In other words, the elements of the subspace $\mathcal{W}_m$ must contain at least one multiplication with $\mathbf{A}$ and this is a first requirement for projection spaces. To demonstrate this, let us fulfil the requirement in a trivial way and choose the projection space to be spanned by the columns of $\mathbf{A}$, that is

$$\mathbf{W}_m = \mathbf{A}_k := \mathbf{A}(e_1, \ldots, e_k), \qquad m = k \leq n,$$

where during the kth iteration we project on a subspace of dimension k. If $\mathbf{B}_k$ is a basis of a k-dimensional test space $\mathcal{V}_k$ satisfying $\det(\mathbf{B}_k^T\mathbf{A}_k) \neq 0$, then with (1.7) the corresponding projector can be represented by

$$\mathbf{P}_k = \mathbf{A}_k(\mathbf{B}_k^T\mathbf{A}_k)^{-1}\mathbf{B}_k^T,$$

and with for example $\hat{r}_{k-1} := r_0$ relations (1.6) and (1.10) yield

$$x_k = x_0 + (e_1, \ldots, e_k)z_k = x_0 + (e_1, \ldots, e_k)(\mathbf{B}_k^T\mathbf{A}_k)^{-1}\mathbf{B}_k^T r_0.$$

A possible choice is $\mathbf{B}_k = (e_1, \ldots, e_k)$, which leads to successive invertion of the upper left $k \times k$ blocks of $\mathbf{A}$ if they are invertible. If we choose to project orthogonally, then the test space must be equal to the projection space, i.e. $\mathbf{B}_k := \mathbf{W}_m = \mathbf{A}_k$. Let us do so and in addition facilitate the inversion of the involved $k \times k$ matrix by successively orthogonalizing the columns of $\mathbf{A}$, e.g. by computing at the kth step the QR decomposition of dimension k,

$$\mathbf{A}_k = \mathbf{Q}_k\mathbf{R}_k, \quad \mathbf{Q}_k^T\mathbf{Q}_k = \mathbf{I}_k,$$

where $\mathbf{R}_k \in \mathbb{R}^{k\times k}$ is upper triangular. Then we have with (1.7)

$$\mathbf{P}_k = \mathbf{Q}_k\mathbf{R}_k(\mathbf{A}_k^T\mathbf{A}_k)^{-1}\mathbf{R}_k^T\mathbf{Q}_k^T = \mathbf{Q}_k\mathbf{Q}_k^T, \qquad \text{if} \quad \det(\mathbf{A}_k^T\mathbf{A}_k) \neq 0,$$

and

$$x_k = x_0 + \begin{pmatrix} \mathbf{R}_k^{-1} \\ \mathbf{0} \end{pmatrix} \mathbf{Q}_k^T r_0, \qquad \text{if} \quad \det(\mathbf{R}_k) \neq 0.$$

This procedure can be seen as a formulation of the direct method

$$x^* := (\mathbf{R}_n)^{-1}\mathbf{Q}_n^T b$$

as a projection method with approximations available at the k-th iteration when $\det(\mathbf{R}_k) \neq 0$ and because we project orthogonally residual norms do not increase.

In a similar way it is possible to regard other direct solvers as projection methods. But whether they converge reasonably, that is whether residual norms decrease rapidly enough to obtain a satisfying approximation after less than n steps depends on the evolution of the distance between r_0 and the projection spaces $\text{span}\{\mathbf{A}e_1, \ldots, \mathbf{A}e_k\}$. Fast decline of this distance is clearly merely a question of having good luck with the choice of r_0, that is of x_0 ! To have at least some connection with the projection space, we require that the projection space be somehow related to the residual we project, that is with $\hat{r}_{k-1}$. Combined with our first requirement that elements of the projection space must contain a multiplication with $\mathbf{A}$, this leads in a natural way to Krylov subspaces. They were originally introduced by Krylov [42] in the context of eigenvalue computations and have the following structure.

Definition 1.2.2 *The m-th Krylov Subspace $\mathcal{K}_m(\mathbf{C}, z)$ generated by $z \in \mathbb{R}^n$ and $\mathbf{C} \in \mathbb{R}^{n\times n}$ is defined through*

$$\mathcal{K}_m(\mathbf{C}, z) := \text{span}\{z, \mathbf{C}z, \mathbf{C}^2 z, \ldots, \mathbf{C}^{m-1}z\}.$$

Thus the subspace $\mathbf{A}\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$ exactly satisfies our two demands for projection spaces and in addition it contains information about multiple application of the operator represented by $\mathbf{A}$. Therefore, dominant properties of $\mathbf{A}$ become apparent at an early stage and under favorable circumstances we can expect the corresponding iterate to be relatively accurate. Moreover, one can for example extract spectral information on $\mathbf{A}$ from these spaces. Due to the connection of residual vector with system matrix in Krylov projection spaces we also obtain

Lemma 1.2.3 *Let a projection method project $\hat{r}_{k-1}$ onto $\mathbf{A}\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$ and let $k < n$ be the smallest integer for which $\dim(\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})) = k - 1$. Then $x_k = x^*$.*

Proof: The vectors $\{\hat{r}_{k-1}, \mathbf{A}\hat{r}_{k-1}, \ldots, \mathbf{A}^{k-1}\hat{r}_{k-1}\}$ are linearly dependent, hence we can write

$$\sum_{i=0}^{k-1} \alpha_i \mathbf{A}^i \hat{r}_{k-1} = 0$$

for some $\alpha_i \in \mathbb{R}$ with at least one nonzero α_i. If α_0 were zero, then

$$\mathbf{A}(\sum_{i=1}^{k-1} \alpha_i \mathbf{A}^{i-1} \hat{r}_{k-1}) = 0,$$

which contradicts $\dim(\mathcal{K}_{k-1}(\mathbf{A}, \hat{r}_{k-1})) = k - 1$ because $\mathbf{A}$ is nonsingular. Hence we can write

$$\hat{r}_{k-1} = \sum_{i=1}^{k-1} \frac{-\alpha_i}{\alpha_0} \mathbf{A}^i \hat{r}_{k-1} = \mathbf{A}(\sum_{i=0}^{k-2} \frac{-\alpha_{i+1}}{\alpha_0} \mathbf{A}^i \hat{r}_{k-1}) \in \mathbf{A}\mathcal{K}_{k-1}(\mathbf{A}, \hat{r}_{k-1}).$$

But by assumption $\mathbf{A}\mathcal{K}_{k-1}(\mathbf{A}, \hat{r}_{k-1}) = \mathbf{A}\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$. Thus $\mathbf{P}_k\hat{r}_{k-1} = \hat{r}_{k-1}$ and $r_k = 0$. □

A further advantage of Krylov projection spaces is that their bases can be updated by only one matrix vector multiplication per iteration, which is especially advantageous if the structure of $\mathbf{A}$ admits inexpensive multiplication. Finally, Krylov spaces are closely connected with polynomial spaces and this connection allows to exploit results from polynomial approximation theory for the theoretical investigation of methods based on Krylov subspaces.

Turning our attention to the bases of projection spaces, we will restrict ourselves to the projection spaces $\mathbf{A}\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$, these spaces being most suitable in the sense described above. Different choices of bases can result in mathematically equivalent projections with very different numerical properties. An obvious basis of a Krylov subspace is $\{\hat{r}_{k-1}, \mathbf{A}\hat{r}_{k-1}, \ldots, \mathbf{A}^{k-1}\hat{r}_{k-1}\}$, this has been proposed by Ipsen

[36], [37], but this basis is numerically very unstable. Instead, most implementations modify the obvious basis with the help of a combination of normalizing and orthogonalizing. Such procedures can be applied to two sorts of bases, resulting in two classes of implementation strategies. The first and most current one is iterate oriented in the sense that bases for the subspaces connected with the iterates are being computed. If a projection method projects $\hat{r}_{k-1}$ onto $\mathbf{A}\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$, then the corresponding iterate x_k is an element of the affine subspace $\hat{x}_{k-1} + \mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$ by (1.10). Iterate-based implementations compute bases of $\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$. Alternatively, one can work with the bases of the spaces that the residuals are projected onto, that is of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$. For the following considerations we assume that we project on a subspace of dimension k during the kth iteration, hence $m = k$ in (1.7). In addition, we assume $\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$ has dimension k for all $k \leq n$.

The first option consists of computing bases $\{v_1, \dots, v_k\}$ and $\{c_1, \dots, c_k\}$ of $\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$. Easy updating is obtained when we exploit *ascending* bases.

Definition 1.2.4 *A basis $\{v_1, \dots, v_k\}$ of $\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$ is ascending when*

$$\operatorname{span}\{v_1, \dots, v_i\} = \mathcal{K}_i(\mathbf{A}, \hat{r}_{k-1}), \quad i \leq k. \tag{1.11}$$

If both bases $\{c_1, \dots, c_k\}$ and $\{v_1, \dots, v_k\}$ are ascending and $\mathbf{V}_{k+1} = (v_1, \dots, v_{k+1})$ and $\mathbf{C}_k = (c_1, \dots, c_k)$, then $c_1, v_1 \in \operatorname{span}\{\hat{r}_{k-1}\}$ and there exists a decomposition of the form

$$\mathbf{A}\mathbf{C}_k = \mathbf{V}_{k+1}\tilde{\mathbf{H}}_k, \tag{1.12}$$

where $\tilde{\mathbf{H}}_k \in \mathbb{R}^{(k+1)\times k}$ is an upper Hessenberg matrix that has rank k due to the non-singularity of $\mathbf{A}$. In the following we define

$$v_1 := \hat{r}_{k-1}/\|\hat{r}_{k-1}\|.$$

The iterate-based implementation writes the k-th iterate in the form

$$x_k = \hat{x}_{k-1} + \mathbf{C}_k y_k, \qquad y_k \in \mathbb{R}^k, \tag{1.13}$$

which leads to the Petrov-Galerkin orthogonality condition

$$\mathbf{B}_k^T r_k = \mathbf{B}_k^T(\hat{r}_{k-1} - \mathbf{A}\mathbf{C}_k y_k) = \mathbf{B}_k^T \hat{r}_{k-1} - \mathbf{B}_k^T \mathbf{V}_{k+1}\tilde{\mathbf{H}}_k y_k = 0, \tag{1.14}$$

and y_k equals the solution of the linear system

$$\mathbf{B}_k^T \mathbf{V}_{k+1}\tilde{\mathbf{H}}_k y = \mathbf{B}_k^T \hat{r}_{k-1}, \tag{1.15}$$

as long as $\mathbf{B}_k^T \mathbf{V}_{k+1}\tilde{\mathbf{H}}_k$ is not singular.
The second option is to compute bases of the projection spaces, i.e. of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$. In analogy with the foregoing we consider ascending bases $\{c_1, \dots, c_k\}$ and $\{w_1, \dots, w_k\}$ of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$ with $\mathbf{W}_k = (w_1, \dots, w_k)$ and $\mathbf{C}_{k-1} = (c_1, \dots, c_{k-1})$ and with a decomposition

$$\mathbf{A}\mathbf{C}_{k-1} = \mathbf{W}_k\tilde{\mathbf{G}}_{k-1}, \tag{1.16}$$

where $\tilde{\mathbf{G}}_{k-1} \in \mathbb{R}^{k\times(k-1)}$ is an upper Hessenberg matrix of full rank and

$$w_1 := \mathbf{A}\hat{r}_{k-1}/\|\mathbf{A}\hat{r}_{k-1}\|.$$

To obtain the iterate corresponding to the kth residual, we need to change this decomposition to

$$\mathbf{A}(v_1, \mathbf{C}_{k-1}) = \mathbf{W}_k\mathbf{R}_k, \tag{1.17}$$

where $v_1 = \hat{r}_{k-1}/\|\hat{r}_{k-1}\|$ and

$$\mathbf{R}_k := \left(\frac{\|\mathbf{A}\hat{r}_{k-1}\|}{\|\hat{r}_{k-1}\|} e_1, \tilde{\mathbf{G}}_{k-1} \right) \in \mathbb{R}^{k\times k}$$

is upper triangular and nonsingular. Then $\operatorname{span}\{v_1, c_1, \ldots, c_{k-1}\} = \mathcal{K}_k(\mathbf{A}, \hat{r}_{k-1})$ and the kth iterate can be written in the form

$$x_k = \hat{x}_{k-1} + (v_1, \mathbf{C}_{k-1}) z_k, \qquad z_k \in \mathbb{R}^k. \tag{1.18}$$

We obtain z_k observing with (1.7), (1.17) and (1.18) that

$$r_k = \hat{r}_{k-1} - \mathbf{W}_k(\mathbf{B}_k^T \mathbf{W}_k)^{-1}\mathbf{B}_k^T \hat{r}_{k-1} =$$

$$b - \mathbf{A}x_k = \hat{r}_{k-1} - \mathbf{A}(v_1, \mathbf{C}_{k-1}) z_k = \hat{r}_{k-1} - \mathbf{W}_k \mathbf{R}_k z_k, \tag{1.19}$$

by solving $z =: z_k$ from

$$\mathbf{R}_k z = (\mathbf{B}_k^T \mathbf{W}_k)^{-1} \mathbf{B}_k^T \hat{r}_{k-1} \tag{1.20}$$

for nonsingular $\mathbf{B}_k^T \mathbf{W}_k$. Under that condition, this option solves an upper triangular system instead of the more complicated (1.15). Both variants have ,,projected" the original problem of dimension n to a smaller problem (1.15) or (1.20) of dimension k. Note that both variants ask for the same number of matrix vector multiplications. However, as is discussed in Liesen, Rozložník, Strakoš [46], the conditioning of the bases of the two approaches can be fairly different. In most methods we avoid the computation of the basis $\{c_1, \ldots, c_k\}$ of (1.12) and (1.16) because it is feasible to choose the basis to be equal to $\{v_1, \ldots, v_k\}$ or $\{w_1, \ldots, w_k\}$, respectively. Moreover, the systems (1.15) and (1.20) can often be simplified by the choice of $\mathbf{B}_k$, which makes them solvable in a numerically more stable way.

Usually the involved bases will be as close to orthonormal as possible for the sake of numerical stability. For example, they can be $\mathbf{A}^T\mathbf{A}$-orthogonal or only orthogonal. In case the basis $\{v_1, \ldots, v_k\}$ from decomposition (1.12) is orthonormal and $\mathbf{C}_k = \mathbf{V}_k$, this decomposition is called an *Arnoldi decomposition* and the columns of $\mathbf{V}_k$ are named *Arnoldi vectors*. The same holds for (1.16). The initial procedure to compute Arnoldi decompositions is indeed due to Arnoldi [1]. Commonly used algorithms to calculate these decompositions include the Modified Gram-Schmidt process (see Algorithm 5.1.1) and the Householder reflection option (Algorithm 5.1.2). They break down if and only if $\operatorname{rank}(\mathbf{V}_{k+1}) = k$ and in that case we obtain the exact solution x^* with Lemma 1.2.3. The former algorithm is a modification of the Gram-Schmidt orthogonalization process applied to $\mathbf{A}\mathbf{V}_k$. The idea behind Householder reflections is to reflect with respect to the canonical basis $\{e_1, \ldots, e_k\}$ and can be described as follows:

Given the unit vector $v_1 = r_0/\|r_0\|$, the initial reflection $\mathbf{R}_1$ (not to be confounded with the matrices from (1.20)) satisfies $\mathbf{R}_1 v_1 = e_1$. In order to obtain equality in the first column of (1.12) (with $\mathbf{C} = \mathbf{V}$) we define a reflection $\mathbf{R}_2$ that leaves e_1 unchanged and such that

$$\mathbf{R}_2(\mathbf{R}_1 \mathbf{A} v_1) = h_{1,1} e_1 + h_{2,1} e_2,$$

for some $h_{1,1}, h_{2,1} \in \mathbb{R}$. Therefore,

$$\mathbf{A}v_1 = h_{1,1}\mathbf{R}_1\mathbf{R}_2 e_1 + h_{2,1}\mathbf{R}_1\mathbf{R}_2 e_2 = h_{1,1} v_1 + h_{2,1}\mathbf{R}_1\mathbf{R}_2 e_2,$$

and the desired equality follows by putting $v_2 := \mathbf{R}_1\mathbf{R}_2e_2$. In general, the jth reflection is chosen such that $\mathbf{R}_je_i = e_i$, $i < j$ and $\mathbf{R}_j(\mathbf{R}_{j-1}\dots\mathbf{R}_1\mathbf{A}v_{j-1}) = \sum_{i=1}^{j} h_{i,j-1}e_i$ for some $h_{i,j-1}$. Hence $v_j := \mathbf{R}_1\dots\mathbf{R}_je_j$ yields the wanted relation

$$\mathbf{A}v_{j-1} = \sum_{i=1}^{j} h_{i,j-1}v_i$$

and the vectors v_i are orthogonal because Householder reflections preserve the orthogonality of $\{e_1,\dots,e_j\}$. Descriptions of this alternative can be found in detail in Walker [72], [73]. It yields numerically more stable orthonormalization than the Gram-Schmidt orthogonalization procedure but it is also more expensive.

1.2.2 The test space

No specific restrictions exist for the test spaces $\mathcal{V}_k$ related with the Petrov-Galerkin condition. Many Krylov subspace-based projection methods seek to reduce the amount of work by choosing $\mathcal{V}_k$ to be a Krylov subspace whose basis can be inexpensively obtained from the computation of the basis for the projection space. Obvious choices are $\mathcal{V}_k := \mathbf{A}\mathcal{K}_k(\mathbf{A},\hat{r}_{k-1})$ or $\mathcal{V}_k := \mathcal{K}_k(\mathbf{A},\hat{r}_{k-1})$ but other choices can yield even lower computational costs. Computation and storage demands are significantly reduced when it is possible to define iterates and residuals with short recurrences. Often short recurrences for basis vectors enable short recurrences for iterates and in Weiss [75] (in Chapter 3) sufficient and necessary conditions for the existence of short iterate recurrences are formulated. Here we will only need the following proposition. It shows that an appropriate choice of the test space can induce short recurrences.

Proposition 1.2.5 *Let the basis of the test space of a projector $\wp_k$ be given by the columns of $\mathbf{B}_k$ and let us have a projection space with a pair of bases as in decomposition (1.16) (or (1.12)). If*

$$\mathbf{T}_k := \mathbf{B}_k^T\mathbf{A}(v_1, \mathbf{C}_{k-1})$$

(or $\mathbf{T}_k := \mathbf{B}_k^T\mathbf{A}\mathbf{C}_k$) is upper Hessenberg with bandwidth of upper elements m, i.e. $t_{i,j} = 0$ for $j-i > m$, and $\det(\mathbf{T}_j) \neq 0$ *for all $1 \le j \le k$, then iterates and residuals of $\wp_k$ can be defined with $(m+1)$-term recurrences.*

Proof: We will prove the case $\mathbf{T}_k := \mathbf{B}_k^T\mathbf{A}(v_1, \mathbf{C}_{k-1})$, the proof for $\mathbf{T}_k := \mathbf{B}_k^T\mathbf{A}\mathbf{C}_k$ is essentially identical. Because $\det(\mathbf{T}_j) \neq 0$ for all $1 \le j \le k$, the LDU decomposition of $\mathbf{T}_k$ exists. Let it have the form $\mathbf{T}_k = \mathbf{L}_k\mathbf{D}_k\mathbf{U}_k$, where $\mathbf{D}_k$ =diag$(d_1,\dots,d_k)$, $\mathbf{L}_k$ is unit lower bidiagonal with the elements $l_2,\dots,l_k$ on the subdiagonal and $\mathbf{U}_k$ is unit banded upper triangular with $u_{i,j} = 0$ for $j-i > m$. We initialize by putting $\mathbf{L}_1 = \mathbf{U}_1 := \mathbf{I}_1$, $\mathbf{D}_1 := t_{1,1}$ and with $\mathbf{T}_k$ being known, the LDU decomposition can easily be updated from $\mathbf{T}_{k-1} = \mathbf{L}_{k-1}\mathbf{D}_{k-1}\mathbf{U}_{k-1}$ by putting first

$$l_k = t_{k,k-1}/d_{k-1}, \quad k > 1,$$

and then

$$u_{k-m,k} = t_{k-m,k}/d_{k-m}, \quad k > m, \qquad \text{and for} \quad k \le m : \qquad u_{1,k} = t_{1,k}/d_1.$$

The remaining elements in the last column of $\mathbf{U}_k$ follow with

$$u_{i,k} = \frac{t_{i,k} - l_id_{i-1}u_{i-1,k}}{d_i}$$

for $k-m+1 \le i \le k-1$ if $k > m$ and else for $2 \le i \le k-1$. Finally,

$$d_k = t_{k,k} - l_k d_{k-1} u_{k-1,k}.$$

Because of decomposition (1.17) we can write $\mathbf{T}_k = \mathbf{B}_k^T \mathbf{W}_k \mathbf{R}_k$. Then we have with (1.18) and (1.20)

$$x_k = \hat{x}_{k-1} + (v_1, \mathbf{C}_{k-1}) \mathbf{T}_k^{-1} \mathbf{B}_k^T \hat{r}_{k-1},$$

and with the abbreviation $\mathbf{S}_k := (v_1, \mathbf{C}_{k-1}) \mathbf{U}_k^{-1}$ this changes to

$$x_k = \hat{x}_{k-1} + \mathbf{S}_k \mathbf{D}_k^{-1} \mathbf{L}_k^{-1} \mathbf{B}_k^T \hat{r}_{k-1}.$$

Now the last column s_k of $\mathbf{S}_k$ can be updated with the $(m+1)$-term recurrence

$$s_k = c_{k-1} - \sum_{i=1}^{m} u_{k-i,k} s_{k-i}, \qquad c_0 := v_1, \quad u_{j,k} := 0, \quad j \le 0.$$

If $\mathbf{D}_k^{-1} \mathbf{L}_k^{-1} \mathbf{B}_k^T \hat{r}_{k-1} = \begin{pmatrix} p_{k-1} \\ \pi_{k-1} \end{pmatrix}$, we can write

$$x_k = \hat{x}_{k-1} + \mathbf{S}_k \mathbf{D}_k^{-1} \mathbf{L}_k^{-1} \mathbf{B}_k^T \hat{r}_{k-1} = \hat{x}_{k-1} + \mathbf{S}_{k-1} p_{k-1} + \pi_{k-1} s_k = x_{k-1} + \pi_{k-1} s_k,$$

where $\pi_{k-1} = (b_k^T \hat{r}_{k-1} - \pi_{k-2} d_{k-1} l_k)/d_k$ because of the last row of $\mathbf{B}_k^T \hat{r}_{k-1} = \mathbf{L}_k \mathbf{D}_k \begin{pmatrix} p_{k-1} \\ \pi_{k-1} \end{pmatrix}$. For the residual we obtain

$$r_k = r_{k-1} - \pi_{k-1} \mathbf{A} s_k.$$

□

We have worked out an application of this proposition to the BCG method in Lemma 5.1.8. A similar proposition based on Gauss elimination with pivoting instead of LDU decomposition can also be formulated. Weiss [76] proposed a method that explicitly computes projections enabling m-term recurrences by means of rank-m updating of projectors (see also Wagner [71]).

We now proceed to the description of concrete projection methods.

1.3 Full projection methods

When the projection space of a projection method reaches dimension n the corresponding residual vector vanishes and the exact solution can be computed. In *full* methods we put $\hat{r}_{k-1} := r_0$ and test space and projection space always have dimension k during the kth iteration. Thus in (1.7) we have $m := k$ and the solution x^* is found at the latest at the nth iteration. Though it might be possible to formulate direct methods as full projection methods, as we have done for the QR-decomposition, we will restrict ourselves for the moment to methods traditionally known as projection methods. Their projection spaces are all based on Krylov subspaces. Moreover, the classical full methods we are going to describe *all* project, during the kth iteration, onto the same, in the sense of preceding considerations optimal projection space $\mathcal{W}_k$, namely

$$\mathcal{W}_k := \mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0).$$

The test space $\mathcal{V}_k$ is either chosen such that the projection becomes orthogonal, i.e. equal to the projection space, or otherwise it equals

$$\mathcal{V}_k := \mathcal{K}_k(\mathbf{A}, r_0).$$

The Petrov-Galerkin orthogonality condition

$$r_k \perp \mathcal{V}_k,$$

however, does not necessarily concern Euclidean orthogonality. Some methods project with respect to the energy norm and others with respect to iteration dependent inner products (see Definition (1.3.3)). Differences between the methods we will describe thus result from the involved orthogonality and from special properties of the system matrix $\mathbf{A}$. In the latter case, the method does project onto the Krylov subspace above, but the projection space might be simplified to a different space. We will always assume $\mathcal{K}_{k+1}(\mathbf{A}, r_0)$ has dimension $k+1$.

1.3.1 Euclidean orthogonality

Euclidean orthogonal projections

Let us start with the most natural case in which the orthogonality condition is Euclidean and the projector is orthogonal. Thus both projection and test space equal

$$\mathcal{W}_k = \mathcal{V}_k = \mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$$

and with (1.9), residual norms are non-increasing. The main representant of such a projection is the GMRES method, the MINRES method is a less expensive implementation for symmetric matrices.

- **The generalized minimal residual method (GMRES)**:
  In this method r_0 is orthogonally projected onto

  $$\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0), \quad k = 1, 2, \dots ,$$

  and thus residual vectors satisfy

  $$\|r_k\| = \min_{y \in \mathbb{R}^k} \|r_0 - \mathbf{W}_k y\|,$$

  if the columns of $\mathbf{W}_k$ span $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$. That last property gave the GMRES method its name. It can be seen as a generalization of the MINRES method for matrices that are not symmetric. In residual-based implementations of the GMRES method with an Arnoldi decomposition (1.16), the matrix $\mathbf{W}_k$ is orthonormal and with (1.7) a matrix representation of the orthogonal projector $\wp_k$ is

  $$\mathbf{P}_k = \mathbf{W}_k \mathbf{W}_k^T$$

  and thus

  $$r_k = r_0 - \mathbf{W}_k \mathbf{W}_k^T r_0.$$

  If the kth iterate is written as in (1.18) in the form

  $$x_k = x_0 + (v_1, \mathbf{W}_{k-1}) z_k, \qquad z_k \in \mathbb{R}^k,$$

  then the system from which z_k is to be obtained, (1.20), changes to

  $$\mathbf{R}_k z = \mathbf{W}_k^T r_0.$$

  This way of calculating GMRES iterates has been proposed by Walker and Zhou [74]. The classical implementation of Saad and Schultz [56], however, is based on an orthonormal basis $\mathbf{V}_k$ for $\mathcal{K}_k(\mathbf{A}, r_0)$ with Arnoldi decomposition (1.12). With

$$x_k = x_0 + \mathbf{V}_k y_k, \qquad y_k \in \mathbb{R}^k,$$

we have

$$\|r_k\| = \|r_0 - \mathbf{A}\mathbf{V}_k y_k\| = \|\mathbf{V}_{k+1}(\|r_0\|e_1 - \tilde{\mathbf{H}}_k y_k)\| = \|\|r_0\|e_1 - \tilde{\mathbf{H}}_k y_k\|$$

and y_k equals the solution of the least squares problem of dimension $(k+1) \times k$

$$\min_{y \in \mathbb{R}^k} \left\| \|r_0\|e_1 - \tilde{\mathbf{H}}_k y \right\|. \tag{1.21}$$

As soon as GMRES iterates are needed, the latter version is numerically more stable due to the fact that the columns of $(v_1, \mathbf{W}_{k-1})$ are not an orthogonal basis of the space connected to iterates, $\mathcal{K}_k(\mathbf{A}, r_0)$, whereas $\mathbf{V}_k$ is orthogonal (see for example Rozložník [55]). On the other hand, information about the length of residuals might be more reliable when using orthonormal bases $\mathbf{W}_k$ instead of $\mathbf{A}\mathbf{V}_k$.

- **The minimal residual method (MINRES):**
  This method is based on the same projector as the GMRES method but it requires symmetric matrices. The method was introduced before the GMRES method by Paige and Saunders [53]. The difference with the GMRES method is that in this implementation short recurrences for iterates exist because the upper Hessenberg matrices involved in the orthogonalization process are in the symmetric case tridiagonal. To compute an Arnoldi decomposition (1.12) or (1.16) for symmetric matrices one commonly uses the symmetric Lanczos procedure (Algorithm 5.1.4) that contains a three-term recurrence to update bases. With Proposition 1.2.5 (where $\mathbf{B}_k := \mathbf{W}_k$) we obtain three-term recurrences for iterates and residuals.

Euclidean oblique projections

Euclidean oblique projection with the test space

$$\mathcal{V}_k = \mathcal{K}_k(\mathbf{A}, r_0)$$

and the projection space

$$\mathcal{W}_k = \mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$$

was historically the first Krylov subspace projection to be used in the context of iterative methods, namely for symmetric positive definite matrices. In that case it can be shown that the projection always exists. But in general, when elements of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ happen to be orthogonal to $\mathcal{K}_k(\mathbf{A}, r_0)$, the corresponding iterate can not be defined, which is a drawback of oblique projections compared with orthogonal projectors. As mentioned before, implementations of projection methods try to exploit bases that are as close to orthonormal as possible for the sake of numerical stability. In the Euclidean oblique case, at least an orthogonal basis of $\mathcal{K}_k(\mathbf{A}, r_0)$ is given by the sequence of residuals:

Lemma 1.3.1 *If the projection space of $\wp_k$ is $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ and the test space is $\mathcal{K}_k(\mathbf{A}, r_0)$, then $\{r_0, \ldots, r_k\}$ is an orthogonal basis of $\mathcal{K}_{k+1}(\mathbf{A}, r_0)$.*

Proof: Clearly, the assumption is valid for $k = 1$. If it is valid for some $j-1 < k$ too, we have $r_j \in r_0 + \mathbf{A}\mathcal{K}_j(\mathbf{A}, r_0) \subset \mathcal{K}_{j+1}(\mathbf{A}, r_0)$. The vector r_j is linearly independent from $\{r_0, \ldots, r_{j-1}\}$ because it is orthogonal to this sequence spanning the test space of the projector $\wp_j$.
□

When an oblique method exploits an Arnoldi decomposition (1.12), that is an orthonormal basis $\mathbf{V}_{k+1}$, then it follows from the preceding lemma that the kth residual vector must be a multiple of the last basis vector v_{k+1}. The following methods are all based on the oblique projector of this lemma but they differ in the assumptions that are put on the system matrix and in the choice of bases for projection and test space.

- **The full orthogonalization method (FOM)**:
Without assumptions for $\mathbf{A}$ other than its non-singularity, the above oblique projection yields the FOM method which can be implemented as follows. Let the columns of $\mathbf{W}_k$ be an orthonormal basis of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ with Arnoldi decomposition (1.16) and let $\mathbf{V}_k$ have columns spanning $\mathcal{K}_k(\mathbf{A}, r_0)$ with first column $v_1 = r_0/\|r_0\|$. With $\mathbf{B}_k := \mathbf{V}_k$ in (1.7) the projector involved in the FOM method can be expressed through

$$\mathbf{P}_k = \mathbf{W}_k(\mathbf{V}_k^T\mathbf{W}_k)^{-1}\mathbf{V}_k^T, \quad \det(\mathbf{V}_k^T\mathbf{W}_k) \neq 0.$$

With the columns of $(v_1, \mathbf{W}_{k-1})$ being a basis of $\mathcal{K}_k(\mathbf{A}, r_0)$ too, the operator can be rewritten when we replace $\mathbf{V}_k$ by $(v_1, \mathbf{W}_{k-1})$ and exploit the equation

$$\begin{pmatrix} v_1^T\mathbf{W}_{k-1} & v_1^T w_k \\ \mathbf{I}_{k-1} & 0 \end{pmatrix}^{-1} = \frac{1}{v_1^T w_k}\begin{pmatrix} 0 & (v_1^T w_k)\mathbf{I}_{k-1} \\ 1 & -v_1^T\mathbf{W}_{k-1} \end{pmatrix},$$

to obtain the expression

$$\mathbf{P}_k = \mathbf{W}_k \begin{pmatrix} \mathbf{W}_{k-1}^T \\ (v_1^T - v_1^T\mathbf{W}_{k-1}\mathbf{W}_{k-1}^T)/v_1^T w_k \end{pmatrix}, \quad v_1^T w_k \neq 0.$$

In case $v_1^T w_k = 0$ the kth FOM approximation does not exist, but this need not break down the process since $v_1^T w_{k+1}$ can very well be nonzero again. FOM iterates of the form

$$x_k = x_0 + (v_1, \mathbf{W}_{k-1})z_k, \qquad z_k \in \mathbb{R}^k,$$

can be computed with (1.20) by solving $z =: z_k$ from

$$\mathbf{R}_k z = \begin{pmatrix} \mathbf{W}_{k-1}^T \\ (v_1^T - v_1^T\mathbf{W}_{k-1}\mathbf{W}_{k-1}^T)/v_1^T w_k \end{pmatrix} r_0.$$

This new formulation of FOM is the parallel of Walker and Zhou's GMRES version ([74]).

Originally the iterate-based variant was used and the bases of the corresponding Krylov subspaces were fully orthonormalized (hence the name of the method). Thus let us assume that we have an orthonormal basis $\{v_1, \ldots, v_{k+1}\}$ of $\mathcal{K}_{k+1}(\mathbf{A}, r_0)$ with Arnoldi decomposition (1.12), where $\mathbf{V}_k = (v_1, \ldots, v_k)$ and $v_1 := r_0/\|r_0\|$. In that case FOM iterates

$$x_k = x_0 + \mathbf{V}_k y_k, \qquad y_k \in \mathbb{R}^k,$$

yield a linear system of the form (1.15), where

$$\mathbf{B}_k^T \mathbf{V}_{k+1} \tilde{\mathbf{H}}_k = \mathbf{H}_k$$

is the Hessenberg matrix $\tilde{\mathbf{H}}_k$ without its last row. Then y_k equals the solution of the linear system

$$\mathbf{H}_k y = \|r_0\| e_1, \tag{1.22}$$

as long as $\mathbf{H}_k$ is not singular.

- **The conjugate gradient method (CG)**:
Let us apply the same oblique projection to a matrix $\mathbf{A}$ that is symmetric positive definite. Hence the projection space of $\wp_k$ is $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$, the test space is $\mathcal{K}_k(\mathbf{A}, r_0)$ and the iterates satisfy

$$x_k \in x_0 + \mathcal{K}_k(\mathbf{A}, r_0).$$

In contrast with the FOM method we will not compute an orthonormal basis of $\mathcal{K}_{k+1}(\mathbf{A}, r_0)$ with an Arnoldi decomposition (1.12), but use the sequence $\{r_0, r_1, \dots, r_k\}$ that is already an orthogonal basis due to Lemma 1.3.1. In addition we introduce the sequence $\{p_0, \dots, p_k\}$ defined by the recurrence

$$p_0 = r_0, \qquad p_k = r_k - \frac{r_k^T \mathbf{A} p_{k-1}}{p_{k-1}^T \mathbf{A} p_{k-1}} p_{k-1}, \quad k > 1,$$

which is possible because $\mathbf{A}$ is positive definite. It is clear that also $\{p_0, p_1, \dots, p_k\}$ generates $\mathcal{K}_{k+1}(\mathbf{A}, r_0)$. This basis, however, is not orthogonal anymore, but it is $\mathbf{A}$-orthogonal with respect to the energy inner product $(x, y)_A := x^T \mathbf{A} y$:

$$p_j^T \mathbf{A} p_k = 0, \quad j \neq k.$$

Indeed,

$$p_{k-1}^T \mathbf{A} p_k = p_{k-1}^T (\mathbf{A} r_k - \frac{r_k^T \mathbf{A} p_{k-1}}{p_{k-1}^T \mathbf{A} p_{k-1}} \mathbf{A} p_{k-1}) = 0$$

and

$$p_{k-2}^T \mathbf{A} p_k = r_k^T \mathbf{A} p_{k-2} - \frac{r_k^T \mathbf{A} p_{k-1}}{p_{k-1}^T \mathbf{A} p_{k-1}} p_{k-2}^T \mathbf{A} p_{k-1} = r_k^T (\sum_{i=0}^{k-1} \alpha_i r_i) = 0,$$

for some $\alpha_i \in \mathbb{R}$ because $\mathbf{A} p_{k-2} \in \operatorname{span}\{\mathbf{A} r_{k-2}, \mathbf{A} p_{k-3}\} \subset \operatorname{span}\{r_0, \dots, r_{k-1}\}$ and by inductive assumption $p_{k-2}^T \mathbf{A} p_{k-1} = 0$. Similarly $p_{k-3}^T \mathbf{A} p_k = 0$, because of $r_k^T \mathbf{A} p_{k-3} = 0$ and $p_{k-3}^T \mathbf{A} p_{k-1} = 0$. One can successively continue until showing $p_0^T \mathbf{A} p_k = 0$.
Because of this $\mathbf{A}$-orthogonality, the vectors p_j are called conjugate gradients and in the CG method we use the sequence of these vectors to obtain bases for test and projection space: We put $\mathbf{B}_k := (p_0, p_1, \dots, p_{k-1})$ and $\mathbf{W}_k := \mathbf{A}(p_0, p_1, \dots, p_{k-1})$ in (1.7). Then the matrix representation of the projector $\mathbf{P}_k$ (not to be confounded with the sequence $\{p_0, \dots, p_k\}$) is given by

$$\mathbf{P}_k = \mathbf{A}(p_0, \dots, p_{k-1}) \operatorname{diag}\left(\frac{1}{p_0^T \mathbf{A} p_0}, \dots, \frac{1}{p_{k-1}^T \mathbf{A} p_{k-1}} \right) (p_0, \dots, p_{k-1})^T, \tag{1.23}$$

and we can write residual vectors in the form

$$r_k = r_0 - \mathbf{P}_k r_0 = r_0 - \frac{p_0^T r_0}{p_0^T \mathbf{A} p_0} \mathbf{A} p_0 \dots - \frac{p_{k-1}^T r_0}{p_{k-1}^T \mathbf{A} p_{k-1}} \mathbf{A} p_{k-1}$$

$$= r_{k-1} - \frac{p_{k-1}^T r_0}{p_{k-1}^T \mathbf{A} p_{k-1}} \mathbf{A} p_{k-1}, \tag{1.24}$$

yielding for iterates

$$x_k = x_{k-1} + \frac{p_{k-1}^T r_{k-1}}{p_{k-1}^T \mathbf{A} p_{k-1}} p_{k-1},$$

when we note that $p_j^T r_0 = p_j^T r_j$ for all $j \leq k-1$. This follows with (1.24) and with the Petrov-Galerkin condition of the projector per induction from

$$r_0^T p_j = r_0^T (r_j - \frac{r_j^T \mathbf{A} p_{j-1}}{p_{j-1}^T \mathbf{A} p_{j-1}} p_{j-1}) = r_{j-1}^T (r_j - \frac{r_j^T \mathbf{A} p_{j-1}}{p_{j-1}^T \mathbf{A} p_{j-1}} p_{j-1})$$

$$= (r_j + \frac{p_{j-1}^T r_0}{p_{j-1}^T \mathbf{A} p_{j-1}} \mathbf{A} p_{j-1})^T p_j = r_j^T p_j.$$

This method for symmetric positive definite matrices was first proposed by Hestenes and Stiefel [34]. The original projector $\wp_k$ on a subspace of dimension k can with the help of the sequence $\{p_0, \ldots, p_{k-1}\}$ be simplified to a projection of $\hat{r}_{k-1} := r_{k-1}$ with matrix representation

$$\mathbf{P}_k := \frac{\mathbf{A} p_{k-1} p_{k-1}^T}{p_{k-1}^T \mathbf{A} p_{k-1}},$$

being an oblique projection onto span$\{\mathbf{A} p_{k-1}\}$, orthogonal to span$\{p_{k-1}\}$. Alternatively, two-term recurrences for residuals can also be obtained from the Lanczos process (Algorithm 5.1.4) with $v_1 := r_0$. The corresponding decomposition has a tridiagonal Hessenberg matrix due to Lemma 5.1.5 and Proposition 1.2.5 states that we can define a sequence $\{p_0, \ldots, p_k\}$, and with its help iterates and residuals, with two-term recurrences. This procedure is called Lanczos iterative solution method, but it is mathematically equivalent with the CG method. The fact that $\mathbf{A}$ is symmetric positive definite also entails this method is error minimizing in the energy norm induced by the energy inner product. This can be seen as follows: Combining (1.23) with (1.8) yields a projector connected to the error of the form

$$(p_0, p_1, \ldots, p_{k-1}) \text{diag} \left(\frac{1}{p_0^T \mathbf{A} p_0}, \ldots, \frac{1}{p_{k-1}^T \mathbf{A} p_{k-1}} \right) \left(\mathbf{A}(p_0, p_1, \ldots, p_{k-1}) \right)^T.$$

Because the sequence $\{p_0, \ldots, p_{k-1}\}$ spans $\mathcal{K}_k(\mathbf{A}, r_0)$, the projection of the error is onto and $\mathbf{A}$-orthogonal to $\mathcal{K}_k(\mathbf{A}, r_0)$. In several applications (especially from physics and quantum chemistry) the $\mathbf{A}$-norm is related to the original problem in a natural way and usable procedures to estimate the $\mathbf{A}$-norm of CG error vectors exist (see Tichý [66]).

- **The orthogonal residuals method (ORTHORES)**:
  The same oblique projection has also been implemented as follows. The set of the k first residual vectors forms an orthogonal basis of $\mathcal{K}_k(\mathbf{A}, r_0)$ by Lemma 1.3.1. The ORTHORES method uses $\mathbf{B}_k := (r_0, \ldots, r_{k-1})$ in (1.7) and the following basis of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$:

$$\{r_1 - r_0,\, r_2 - r_0,\, \ldots,\, r_{k-1} - r_0, \mathbf{A} r_{k-1}\}.$$

Clearly, $r_1 - r_0 \in \operatorname{span}\{\mathbf{A}r_0\}$. Furthermore, $r_{j+1} - r_0 \in \mathbf{A}\mathcal{K}_{j+1}(\mathbf{A}, r_0)$ for $j > 1$ and if $r_{j+1} - r_0$ would be linearly dependent on the foregoing basis vectors, then

$$r_{j+1} - r_0 = \sum_{i=1}^{j} \beta_i (r_i - r_0),$$

where at least one $\beta_l \neq 0$ and where we have assumed that $\{r_1 - r_0, \ldots, r_j - r_0\}$ spans $\mathbf{A}\mathcal{K}_j(\mathbf{A}, r_0)$. But then

$$r_l^T (r_{j+1} - r_0) = \beta_l r_l^T r_l \neq 0,$$

contradicting the orthogonality condition of the ORTHORES projector.

Similarly, from $r_{k-1} - r_0 \in \mathbf{A}\mathcal{K}_{k-1}(\mathbf{A}, r_0)$ we obtain $\mathbf{A}r_{k-1} \in \mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ through multiplication with $\mathbf{A}$. If we assume $\mathbf{A}r_{k-1} \in \mathbf{A}\mathcal{K}_{k-1}(\mathbf{A}, r_0)$ this implies $r_{k-1} \in \mathcal{K}_{k-1}(\mathbf{A}, r_0)$, which contradicts the fact that $\{r_0, \ldots, r_{j-1}\}$ spans $\mathcal{K}_j(\mathbf{A}, r_0)$ for all $j \leq k$ (see Lemma 1.3.1).

With $\mathbf{W}_k := (r_1 - r_0, r_2 - r_0, \ldots, r_{k-1} - r_0, \mathbf{A}r_{k-1})$ in (1.7) a matrix representation of the ORTHORES projector is given by

$$\mathbf{P}_k = (r_1 - r_0, \ldots, r_{k-1} - r_0, \mathbf{A}r_{k-1}) \cdot$$
$$\left((r_0, \ldots, r_{k-1})^T (r_1 - r_0, \ldots, r_{k-1} - r_0, \mathbf{A}r_{k-1})\right)^{-1} (r_0, \ldots, r_{k-1})^T.$$

The expression to be inverted equals

$$(r_0, \ldots, r_{k-1})^T (r_1 - r_0, \ldots, r_{k-1} - r_0, \mathbf{A}r_{k-1}) = \begin{pmatrix} -\|r_0\|^2 & \ldots & -\|r_0\|^2 & r_0^T \mathbf{A}r_{k-1} \\ \|r_1\|^2 & 0 & 0 & r_1^T \mathbf{A}r_{k-1} \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & \|r_{k-1}\|^2 & r_{k-1}^T \mathbf{A}r_{k-1} \end{pmatrix},$$

which changes to

$$\operatorname{diag}(\|r_0\|^2, \ldots, \|r_{k-1}\|^2) \begin{pmatrix} -1 & \ldots & -1 & -h_{0,k} \\ 1 & 0 & 0 & -h_{1,k} \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & 1 & -h_{k-1,k} \end{pmatrix}$$

when we introduce the notation

$$h_{i,k} = -\frac{r_i^T \mathbf{A}r_{k-1}}{\|r_i\|^2}, \quad 0 \leq i \leq k-1.$$

Hence we obtain

$$\det\left((r_0, \ldots, r_{k-1})^T (r_1 - r_0, \ldots, r_{k-1} - r_0, \mathbf{A}r_{k-1})\right) = \left(\prod_{i=0}^{k-1} \|r_i\|^2\right) \sum_{i=0}^{k-1} h_{i,k}$$

and when the determinant vanishes the projection is not defined. Otherwise, with

$$h_{k,k} = \left(\sum_{i=0}^{k-1} h_{i,k}\right)^{-1},$$

we have

$$\left((r_0, \ldots, r_{k-1})^T (r_1 - r_0, \ldots, r_{k-1} - r_0, \mathbf{A}r_{k-1})\right)^{-1} =$$

$$h_{k,k}\begin{pmatrix} -h_{1,k} & \sum_{i\neq 1} h_{i,k} & \dots & -h_{1,k} \\ \vdots & \ddots & \ddots & \vdots \\ -h_{k-1,k} & \dots & -h_{k-1,k} & \sum_{i\neq k-1} h_{i,k} \\ -1 & \dots & -1 & -1 \end{pmatrix} \mathrm{diag}(1/\|r_0\|^2, \dots, 1/\|r_{k-1}\|^2),$$

and the projection of r_0 becomes

$$\mathbf{P}_k r_0 = h_{k,k}(r_1 - r_0, \dots, r_{k-1} - r_0, \mathbf{A}r_{k-1}) \begin{pmatrix} -h_{1,k} \\ \vdots \\ -h_{k-1,k} \\ -1 \end{pmatrix}.$$

The kth residual can be written as

$$r_k = r_0 - \mathbf{P}_k r_0 = r_0 - h_{k,k}(r_1 - r_0, \dots, r_{k-1} - r_0, \mathbf{A}r_{k-1}) \begin{pmatrix} -h_{1,k} \\ \vdots \\ -h_{k-1,k} \\ -1 \end{pmatrix} =$$

$$r_0 + h_{k,k}\left(\mathbf{A}r_{k-1} + \sum_{i=1}^{k-1} h_{i,k}(r_i - r_0)\right) = r_0 + h_{k,k}\left(\mathbf{A}r_{k-1} + \sum_{i=1}^{k-1} h_{i,k} r_i\right) - h_{k,k}\sum_{i=0}^{k-1} h_{i,k} r_0$$

and thus the residuals are given by

$$r_k = h_{k,k}(\mathbf{A}r_{k-1} + \sum_{i=0}^{k-1} h_{i,k} r_i)$$

and the iterates by

$$x_k = h_{k,k}(r_{k-1} - \sum_{i=0}^{k-1} h_{i,k} x_i).$$

This procedure, originally designed for positive real matrices, was proposed by Young [77]. Of course, it can also be applied to matrices that are not positive real. Implementations of the GMRES method can be based on this procedure too (see for example Weiss [75]).

Relations between orthogonal and oblique projection methods

As the test spaces of an orthogonal method and its oblique parallel are very similar, a close relationship of convergence properties between the two exists. This has been intensively studied, for example in Eiermann, Ernst [18]. We cite here only some relations we will need later on.

Theorem 1.3.2 *Let a method project onto $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ and orthogonally to $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$, $k \geq 1$, with residual vectors denoted by r_k^M and let its oblique parallel project onto $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ and orthogonally to $\mathcal{K}_k(\mathbf{A}, r_0)$, $k \geq 1$, with residual vectors denoted by r_k^O. Then*

$$r_k^M = s_k^2 r_{k-1}^M + c_k^2 r_k^O,$$

$$\|r_k^M\| = s_k \|r_{k-1}^M\|,$$

$$\|r_k^M\| = s_1 s_2 \dots s_k \|r_0\|,$$

$$\|r_k^M\| = c_k \|r_k^O\|,$$

$$\|r_k^O\| = s_1 s_2 \ldots s_k \|r_0\|/c_k,$$

where $s_k = \sin \angle(r_{k-1}^M, \mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0))$, $c_k = \cos \angle(r_{k-1}^M, \mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0))$ *and* $\angle(r_{k-1}^M, S) := \inf_{0 \neq s \in S} \angle(r_{k-1}^M, s)$ *for subspaces* S *of* $\mathbb{R}^n$.

Proof: See Eiermann, Ernst [18], page 15 and 16. □
From these relations it is clear that oblique projections are more susceptible to irregular convergence behaviour than their orthogonal counterparts. If after the kth iteration s_k is close to 1, the residual norm of the orthogonal method is about as large as the previous residual norm. The corresponding oblique residual norm, in contrast, increases dramatically due to division by $c_k \approx 0$. This phenomenon is known as the peak-plane relation of convergence curves of oblique and orthogonal methods. In exact arithmetics, both projection strategies find the exact solution at the same step, but in practice the unstability of the oblique version can prevent residuals from vanishing.

1.3.2 $\mathcal{V}$-orthogonality

Unless favorable properties of the system matrix $\mathbf{A}$ are assumed (e.g. symmetry), Euclidean projection with orthonormal bases asks for orthogonalization of new basis vectors against *all* previous ones. When the iteration number k grows large this implies high computational and storage costs. A way to overcome this disadvantage is to give up orthogonality of bases and, in exchange, use bases that can be computed with short recurrences. But non-orthonormal bases, besides from being numerically less stable than orthonormal bases, are much harder to work with from the projection theoretical point of view. For example, if the columns of $\mathbf{V}_k$ form a non-orthonormal basis, terms of the form $(\mathbf{V}_k^T \mathbf{V}_k)^{-1}$ in (1.7) cannot be simplified to identity matrices anymore. To facilitate computations with non-orthonormal bases, one usually exploits projectors whose Petrov-Galerkin condition depends on the chosen basis and on the iteration number. The involved orthogonality is induced by the following inner product.

Definition 1.3.3 *If* $\mathbf{V} = (v_1, \ldots, v_m) \in \mathbb{R}^{n \times m}$ *is a matrix with full rank, then the* $\mathcal{V}$*-inner product of two vectors* $a, b \in \operatorname{span}\{v_1, \ldots, v_m\}$ *with respect to* $\mathbf{V}$ *is given by*

$$(a, b)_{\mathcal{V}} = (\mathbf{V}c, \mathbf{V}d)_{\mathcal{V}} := c^T d, \qquad \text{where} \qquad a = \mathbf{V}c,\ b = \mathbf{V}d,$$

and the $\mathcal{V}$*-norm is induced by the* $\mathcal{V}$*-inner product.*

With these notations it follows that

$$\mathbf{V}^T \mathbf{V} c = \mathbf{V}^T a, \quad c = (\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T a,$$

and d can be determined similarly, yielding for arbitrary elements
$a, b \in \operatorname{span}\{v_1, \ldots, v_m\}$

$$(a, b)_{\mathcal{V}} = ((\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T a)^T (\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T b = a^T \mathbf{V} (\mathbf{V}^T \mathbf{V})^{-2} \mathbf{V}^T b. \tag{1.25}$$

In $\mathcal{V}$-orthogonal projection methods the matrix $\mathbf{V}$ of the above definition is provided by the basis-generating algorithm. If the algorithm computes bases $\{v_1, \ldots, v_k\}$ and $\{c_1, \ldots, c_k\}$ of $\mathcal{K}_k(\mathbf{A}, r_0)$ with a decomposition (1.12) and $\mathbf{V}_{k+1} = (v_1, \ldots, v_{k+1})$, then $\mathcal{V}$-orthogonality is to be understood with respect to $\mathbf{V} := \mathbf{V}_{k+1}$ in Definition 1.3.3.

Oblique projection with respect to the $\mathcal{V}$-inner product

Oblique projections based on $\mathcal{V}$-orthogonality put the following Petrov-Galerkin condition on the residual vector:

$$r_k \perp_{\mathcal{V}} \mathcal{K}_k(\mathbf{A}, r_0).$$

From (1.25) and because r_k lies in $\mathcal{K}_{k+1}(\mathbf{A}, r_0)$, we see that the test space for these projectors, expressed in Euclidean orthogonality, is spanned by the columns of

$$\mathbf{V}_{k+1}(\mathbf{V}_{k+1}^T\mathbf{V}_{k+1})^{-2}\mathbf{V}_{k+1}^T\mathbf{V}_k = \mathbf{V}_{k+1}(\mathbf{V}_{k+1}^T\mathbf{V}_{k+1})^{-1}\begin{pmatrix} & \mathbf{I}_k & \\ 0 & \dots & 0 \end{pmatrix}, \qquad (1.26)$$

where we have exploited

$$\mathbf{V}_k = \mathbf{V}_{k+1}\begin{pmatrix} & \mathbf{I}_k & \\ 0 & \dots & 0 \end{pmatrix}. \qquad (1.27)$$

With the basis $\mathbf{A}\{c_1, \dots, c_k\}$ of the projection space $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ the projector is given by

$$\mathbf{AC}_k \left((\mathbf{V}_{k+1}(\mathbf{V}_{k+1}^T\mathbf{V}_{k+1})^{-2}\mathbf{V}_{k+1}^T\mathbf{V}_k)^T\mathbf{AC}_k\right)^{-1}\left(\mathbf{V}_{k+1}(\mathbf{V}_{k+1}^T\mathbf{V}_{k+1})^{-2}\mathbf{V}_{k+1}^T\mathbf{V}_k\right)^T,$$

which changes with (1.12) and (1.27) to

$$\mathbf{P}_k = \mathbf{V}_{k+1}\begin{pmatrix} & \vdots \\ \mathbf{I}_k & 0 \\ & \vdots \\ \tilde{h}_{k+1,k}e_k^T\mathbf{H}_k^{-1} & 0 \end{pmatrix}(\mathbf{V}_{k+1}^T\mathbf{V}_{k+1})^{-1}\mathbf{V}_{k+1}^T, \qquad (1.28)$$

where $\mathbf{H}_k \in \mathbb{R}^{k\times k}$ is the upper Hessenberg matrix $\tilde{\mathbf{H}}_k$ from (1.12) without its last row and we assume $\det(\mathbf{H}_k) \neq 0$. If $\mathbf{H}_k$ is singular the kth projection does not exist.

The reason for using the rather complicated $\mathcal{V}$-orthogonality is that with its help we can derive a procedure to compute iterates that leads to the same kind of linear system as in the FOM method. In the FOM case, the derivation of the linear system (1.22) exploited orthonormality of the involved bases. With $\mathcal{V}$-orthogonality, the Petrov-Galerkin condition can, with iterates of the form (1.13), be expressed by the equation

$$\left(\mathbf{V}_{k+1}(\mathbf{V}_{k+1}^T\mathbf{V}_{k+1})^{-1}\begin{pmatrix} & \mathbf{I}_k & \\ 0 & \dots & 0 \end{pmatrix}\right)^T(r_0 - \mathbf{AC}_k y_k) =$$

$$= \begin{pmatrix} & \vdots \\ \mathbf{I}_k & 0 \\ & \vdots \end{pmatrix}(\mathbf{V}_{k+1}^T\mathbf{V}_{k+1})^{-1}\mathbf{V}_{k+1}^T\mathbf{V}_{k+1}(\|r_0\|e_1 - \tilde{\mathbf{H}}_k y_k) = 0.$$

Hence y_k equals the solution of the linear system

$$\mathbf{H}_k y = \|r_0\|e_1, \qquad (1.29)$$

which coincides with the system (1.22). If $\det(\mathbf{H}_k) \neq 0$ this yields for residuals the expression

$$r_k = r_0 - \mathbf{AC}_k y_k = \mathbf{V}_{k+1}\left(\|r_0\|e_1 - \tilde{\mathbf{H}}_k\mathbf{H}_k^{-1}\|r_0\|e_1\right) =$$

$$\mathbf{V}_{k+1}\left(\|r_0\|e_1-\begin{pmatrix}\mathbf{I}_k\\ \tilde{h}_{k+1,k}e_k^T\mathbf{H}_k^{-1}\end{pmatrix}\|r_0\|e_1\right)=\mathbf{V}_{k+1}\begin{pmatrix}0\\ \vdots\\ 0\\ -\tilde{h}_{k+1,k}\|r_0\|e_k^T\mathbf{H}_k^{-1}e_1\end{pmatrix}$$

$$= -v_{k+1}\tilde{h}_{k+1,k}\|r_0\|e_k^T\mathbf{H}_k^{-1}e_1. \tag{1.30}$$

Thus the kth residual vector is a multiple of the last basis vector v_{k+1}, as was the case with Euclidean orthogonality.

In general $\mathcal{V}$-orthogonal projectors pay for their inexpensive iterations with a loss of stability, especially the algorithms for bases can break down. The following full methods work with such kind of projector. They differ only in the choice of the involved bases $\{v_1,\dots,v_{k+1}\}$ and, of course, they exploit different algorithms to generate their bases.

- **The bi-conjugate gradient method (BCG)**:
  The first method to generate nonorthogonal bases with short recurrences in the non-symmetric non-positive definite case, was the BCG method. This method works with an arbitrary shadow vector $\tilde{v}_1$ and generates a *pair* of bases, one corresponding to r_0 and one to the shadow vector, that is bi-orthogonal. More precisely, if the columns of $\mathbf{V}_k \in \mathbb{R}^{n\times k}$ span $\mathcal{K}_k(\mathbf{A}, r_0)$ (with $v_1 := r_0/\|r_0\|$) and those of $\tilde{\mathbf{V}}_k \in \mathbb{R}^{n\times k}$ span $\mathcal{K}_k(\mathbf{A}^T, \tilde{v}_1)$, then one variant of defining bi-orthogonality is the condition

  $$\tilde{\mathbf{V}}_k^T\mathbf{V}_k = \mathbf{I}_k \tag{1.31}$$

  (for a detailed description of other variants see Weiss [75], Chapter 4). An algorithm producing such bases is Algorithm 5.1.6, called the bi-orthogonal Lanczos process. It yields a decomposition (1.12) with $\mathbf{C}_k = \mathbf{V}_k$ due to Lemma 5.1.7.

  With the matrix $\mathbf{V}_{k+1}$ that results from the bi-orthogonal Lanczos process, the considerations based on $\mathcal{V}$-orthogonality that precede this method yield a test space that is spanned by the columns of (1.26), a projector represented by (1.28) and coordinate vectors y_k that solve a system of the form (1.29). If $\mathbf{H}_k$ is singular, the kth iterate does not exist.

  Interestingly, the BCG method can also be characterized by a projection orthogonal to a different test space, namely $\mathcal{K}_k(\mathbf{A}^T, \tilde{v}_1)$, where the orthogonality condition is Euclidean. Indeed, if

  $$x_k = x_0 + \mathbf{V}_k y_k, \qquad y_k \in \mathbb{R}^k,$$

  then (1.15) reads

  $$\mathbf{H}_k y = \tilde{\mathbf{V}}_k^T r_0$$

  , where $\mathbf{H}_k$ is the Hessenberg matrix $\tilde{\mathbf{H}}_k$ without its last row. This is the same system as (1.29) because $v_1^T\tilde{v}_1 = 1$ and thus this characterization yields identical iterates.

  A new, residual-based implementation results if we assume the columns of some matrix $\mathbf{W}_k \in \mathbb{R}^{n\times k}$ span $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ (with $w_1 := \mathbf{A}r_0/\|\mathbf{A}r_0\|$), those of $\tilde{\mathbf{V}}_k \in \mathbb{R}^{n\times k}$ span $\mathcal{K}_k(\mathbf{A}^T, \tilde{v}_1)$, and the bi-orthogonality condition consists of

  $$\tilde{\mathbf{V}}_k^T\mathbf{W}_k = \mathbf{I}_k.$$

This could be achieved by applying Algorithm 5.1.6 to a starting vector $\mathbf{A}r_0/\|\mathbf{A}r_0\|$. We can represent this projector with $\mathbf{B}_k := \tilde{\mathbf{V}}_k$ in (1.7) by

$$\mathbf{P}_k = \mathbf{W}_k\tilde{\mathbf{V}}_k^T. \tag{1.32}$$

Due to Lemma 5.1.7 the basis $\mathbf{W}_k$ possesses a decomposition (1.16) , where $\mathbf{C}_{k-1} = \mathbf{W}_{k-1}$ and $\tilde{\mathbf{G}}_{k-1} \in \mathbb{R}^{k\times(k-1)}$ is tridiagonal. Thus we can define an upper triangular matrix $\mathbf{R}_k$ according to (1.17) and if

$$x_k = x_0 + (v_1, \mathbf{W}_{k-1})z_k, \qquad z_k \in \mathbb{R}^k,$$

then (1.20) translates in

$$\mathbf{R}_k z = \tilde{\mathbf{V}}_k^T r_0.$$

When $\mathbf{R}_k$ is nonsingular, we solve a linear system of dimension k with an upper triangular tridiagonal system matrix.

Note that the projector of the BCG method depends on the choice of the shadow vector. For different shadow vectors the projection is orthogonal to different spaces. This item is treated for example in Tichý [66]. Also note that the BCG procedure significantly reduces computational costs in comparison with the oblique methods mentioned before, because bases can be defined with three-term recurrences and with Lemma 5.1.7 and Proposition 1.2.5 also iterates and residuals can. This has been worked out in Lemma 5.1.8. The method was first described by Lanczos [44].

An important drawback of this method is the risk of breakdowns. In analogy with algorithms for Arnoldi decompositions, the bi-orthogonal Lanczos process terminates prematurely when the dimension of $\mathcal{K}_k(\mathbf{A}, r_0)$ or $\mathcal{K}_k(\mathbf{A}^T, \tilde{v}_1)$ is maximal for some $k < n$. Then the process has found an $\mathbf{A}$- respectively $\mathbf{A}^T$-invariant subspace and in the first case we have found the exact solution because of Lemma 1.2.3. This kind of termination is called ,,happy" breakdown and occurs when the vector v_{k+1} or w_{k+1} vanishes in Algorithm 5.1.6. If neither of these vectors is zero, but still $v_{k+1}^T w_{k+1} = 0$, the algorithm breaks down too. This breakdown has been referred to as ,,serious". Assu-ming full dimension of all Krylov subspaces, it is readily seen that the matrix $\mathbf{H}_k$ from (1.29) is singular if and only if the bi-orthogonalization procedure connected to (1.32) breaks down in this manner. Thus a serious breakdown of the residual-based implementation occurs exactly when the Petrov-Galerkin condition cannot be satisfied, that is when the oblique projection does not exist. The iterate-based version, on the other hand, is able to skip an iterate if the projection does not exist (,,curable breakdown"), but it breaks down for a different reason: Orthogonality of a vector from $\mathcal{K}_{k+1}(\mathbf{A}, r_0)$ to the space $\mathcal{K}_{k+1}(\mathbf{A}^T, \tilde{v}_1)$. A well-known implementation of the iterate based version that does not separate the computation of bases from the computation of iterates is presented in Proposition 5.1.8. Although it is numerically more stable than the strategy described above (see Gutknecht, Strakoš [32]), there are three reasons why it can break down: Happy or serious breakdown of the underlying bi-orthogonalization process and nonexistence of the oblique projection.

So-called ,,look-ahead" variants of Algorithm 5.1.6 try to reduce the risk of breakdown by defining ,,nearly" bi-orthogonal bases (see for example Parlett, Taylor and Liu [48] or Freund, Gutknecht and Nachtigal [27]). They avoid computation of basis vectors if they are suspicious of being orthogonal to each other by skipping them, as long as it does not concern a ,,happy" breakdown. The look-ahead strategy can also overcome other kinds of threatening instability and it only exceptionally breaks down. An example is given in Algorithm 5.1.10. The bi-orthogonality condition (1.31) is in this case weakened to the block bi-orthogonality condition (5.6). The one drawback of the look-ahead process in comparison with bi-orthogonalization is that the three-term recurrences of the latter get lost. But a block generalization of Proposition 1.2.5 is possible and thus iterates can be defined at least by three-block recurrences because of Lemma 5.1.11. In general the dimension of the blocks $\mathbf{D}_l$ from (5.6) is, however, very small (i.e. less than five in most applications) and if their dimension equals one for all l the process coincides with bi-orthogonalization. Otherwise, a method based on a look-ahead algorithm yields a different projector than classical bi-orthogonalization and is therefore slightly different from the original BCG method.

- **The conjugate gradient squared method (CGS)**:
In the BCG method we generate a basis for the subspace $\mathcal{K}_k(\mathbf{A}^T, \tilde{v}_1)$ (see Lemma 5.1.7 or 5.1.11). This subspace is related to the linear system

$$\mathbf{A}^T x = b. \tag{1.33}$$

When we desire to solve this dual system simultaneously with system (1.1) we have to choose the shadow vector $\tilde{v}_1 = r_0^* / \|r_0^*\|$, where $r_0^* = b - \mathbf{A}^T x_0^*$ for some initial guess x_0^* of the dual system. In the common case, however, the dual system does not play any role and the bases of $\mathcal{K}_k(\mathbf{A}^T, \tilde{v}_1)$ merely serve as shadowing sequences that enable short recurrence formulaes for the bases of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$. A current choice is $\tilde{v}_1 := v_1$. Unfortunately, one needs two matrix vector multiplications (one with $\mathbf{A}$ and one with $\mathbf{A}^T$) to compute only one new basis vector for $\mathbf{A}\mathcal{K}_{k+1}(\mathbf{A}, r_0)$. In order to make the computation of bases more profitable, in the CGS method we modify Algorithm 5.1.6, the bi-orthogonal Lanczos process, to obtain only bases for $\mathcal{K}_{2k}(\mathbf{A}, r_0)$ that can be defined with short recurrences. This is achieved by squaring the residual polynomials generated by the BCG method: When the jth BCG residual has the form

$$\overset{\text{BCG}}{r_j} = \rho_j(\mathbf{A}) r_0, \tag{1.34}$$

for some polynomial ρ_j of degree j with $\rho_j(0) = 1$, then the CGS algorithm computes a sequence of basis vectors $q_0, q_1, \ldots$ whose elements with even index satisfy

$$q_{2j} = (\rho_j(\mathbf{A}))^2 r_0.$$

The sequence generated by Algorithm 5.1.12 with $q_0 := r_0$ has this property. This is proved in Lemma 5.1.14. The idea of squaring comes from rewriting BCG scalars $(\mathbf{A}^T r_0^*)^T \mathbf{A} r_0$ in the form $(r_0^*)^T \mathbf{A}^2 r_0$, which is a way to avoid multiplications with the transposed matrix $\mathbf{A}^T$. Moreover, this algorithm, when $q_0 := r_0$, yields bases for the desired Krylov subspaces (see Lemma 5.1.13) and only one matrix vector multiplication per basis vector is needed. But the generated sequences are still dependent on the choice of the shadow vector and as for bi-orthogonalization the process can break down.

We will now show that Algorithm 5.1.12 with $q_0 := r_0$ computes a decomposition of the form (1.12) and moreover, that the generated basis vectors are at the same time the residual vectors resulting from projection with respect to the corresponding $\mathcal{V}$-inner product. For the quantities used in the following

considerations we refer to Algorithm 5.1.12 and we assume that in the algorithm $\alpha_i \neq 0$ for all $i > 0$.
If $\mathbf{C}_k := (c_0, c_1, \dots, c_k)$ we obtain, when we omit the distinction between odd and even indexes for the moment, that

$$q_{i+1} = q_i - \alpha_i \mathbf{A} c_i, \quad i < k$$

and due to this recurrence the vector q_k can be written as

$$q_k = q_0 - \mathbf{A}\mathbf{C}_{k-1} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{k-1} \end{pmatrix}.$$

With the first equation we have

$$\mathbf{A} c_i = \frac{1}{\alpha_i}(q_i - q_{i+1}), \quad i < k,$$

and this yields the decomposition

$$\mathbf{A}\mathbf{C}_{k-1} = (q_0, \dots, q_k)\tilde{\mathbf{G}}_k, \tag{1.35}$$

where $\tilde{\mathbf{G}}_k \in \mathbb{R}^{(k+1)\times k}$ has the form

$$\tilde{\mathbf{G}}_k = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ -1 & 1 & & & \vdots \\ 0 & -1 & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & -1 & 1 \\ 0 & \dots & & 0 & -1 \end{pmatrix} \operatorname{diag}(1/\alpha_0, \dots, 1/\alpha_{k-1}).$$

Hence

$$q_k = q_0 - \mathbf{A}\mathbf{C}_{k-1} \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{k-1} \end{pmatrix} = (q_0, \dots, q_k) \left(e_1 - \tilde{\mathbf{G}}_k \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{k-1} \end{pmatrix} \right) =$$
$$(q_0/\|q_0\|, \dots, q_k/\|q_k\|) \left(\|q_0\| e_1 - \operatorname{diag}(\|q_0\|, \dots, \|q_k\|) \tilde{\mathbf{G}}_k \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{k-1} \end{pmatrix} \right).$$

With the notations

$$\mathbf{V}_{k+1} := (q_0/\|q_0\|, \dots, q_k/\|q_k\|), \quad \tilde{\mathbf{H}}_k := \operatorname{diag}(\|q_0\|, \dots, \|q_k\|) \tilde{\mathbf{G}}_k, \tag{1.36}$$

we obtain

$$q_k = q_0 - \mathbf{A}\mathbf{C}_{k-1} \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{k-1} \end{pmatrix} = \mathbf{V}_{k+1} \left(\|q_0\| e_1 - \tilde{\mathbf{H}}_k \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{k-1} \end{pmatrix} \right). \tag{1.37}$$

In this equation we recognize that the transpose free algorithm yields a decomposition of the form (1.12) , where in contrast with preceding methods $\mathbf{C}_k \neq \mathbf{V}_k$. If $q_0 = r_0$, then because of Lemma 5.1.13, the algorithm generates a basis $\mathbf{A}\{c_0, c_1, \dots, c_{k-1}\}$ of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ and a basis $\{v_0, v_1, \dots, v_{k-1}\}$, columns from $\mathbf{V}_{k+1}$ in (1.36), of $\mathcal{K}_k(\mathbf{A}, r_0)$. Thus the oblique projector can be represented by (1.28). In theory, iterates can be computed by solving the Hessenberg system (1.29), but in the CGS case it is more efficient to extract them from the residuals. This is due to the fact that the residuals are already avai-lable in Algorithm 5.1.12, as we will now show. With (1.30), residual vectors have the form

$$r_k = -v_k \tilde{h}_{k+1,k} \|r_0\| e_k^T \mathbf{H}_k^{-1} e_1 = v_k \frac{\|q_k\|}{\alpha_{k-1}} \|r_0\| e_k^T \mathbf{H}_k^{-1} e_1. \tag{1.38}$$

With (1.36) $\mathbf{H}_k$ can be written as

$$\mathbf{H}_k = \mathrm{diag}(\|q_0\|, \ldots, \|q_{k-1}\|)\mathbf{G}_k,$$

where $\mathbf{G}_k \in \mathbb{R}^{k \times k}$ is the upper Hessenberg matrix $\tilde{\mathbf{G}}_k$ from (1.35) without its last row. Its inverse equals

$$\mathbf{H}_k^{-1} = \mathrm{diag}(\alpha_0, \ldots, \alpha_{k-1}) \begin{pmatrix} 1 & 0 & \ldots & \ldots & 0 \\ 1 & 1 & & & \vdots \\ 1 & 1 & 1 & \ldots & \\ \vdots & & \ddots & \ddots & 0 \\ 1 & & & 1 & 1 \end{pmatrix} \mathrm{diag}(1/\|q_0\|, \ldots, 1/\|q_{k-1}\|).$$

Thus $e_k^T \mathbf{H}_k^{-1} e_1 = \frac{\alpha_{k-1}}{\|q_0\|}$ and this means that with (1.36) and (1.38) residual vectors in the CGS method satisfy

$$r_k = v_k \|q_k\| = q_k = q_0 - \mathbf{A}\mathbf{C}_{k-1} \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{k-1} \end{pmatrix}$$

if we choose $q_0 = r_0$. Thus $r_i \equiv q_i$, $i \leq k$ and the residuals are nothing but the basis vectors for the Krylov subspaces.

Sonneveld [62] introduced this method and his version defines only even iterates

$$x_{2k} = x_0 + \mathbf{C}_{2k-1} \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{2k-1} \end{pmatrix} = x_{2k-2} + \alpha_{2k-2}(c_{2k-2} + c_{2k-1}).$$

because $\alpha_{2k-1} = \alpha_{2k-2}$ in Algorithm 5.1.12. Even iterates are due to the original derivation of the method by squaring BCG residual polynomials. This squaring has the following consequences concerning convergence behaviour: The CGS method finds the exact solution at the same iteration number as the BCG method and a breakdown of the former occurs precisely when the latter breaks down. Fast convergence of BCG residual norms will be even accelerated by CGS residuals, but also convergence oscillations are emphasized by the latter. This leads to high peaks in the convergence curve and a loss of stability in general.

- **The bi-conjugate gradient stabilized method (BCGSTAB)**:
The main idea behind the BCGSTAB method is to combine smoothing of the irregular convergence behaviour of the BCG method with a transpose free procedure to generate bases. In such a procedure stabilizing parameters are inserted. An example is Algorithm 5.1.15 with parameters ω_{2j} and with short

recurrences for basis vectors. It can be seen from this algorithm that the smoothing parameters ω_{2j} are defined by a standard line search:

$$\|q_{2j+2}\| = \|q_{2j+1} - \omega_{2j}c_{2j+1}\| = \min_{\omega\in\mathbb{R}} \|q_{2j+1} - \omega c_{2j+1}\|. \tag{1.39}$$

Furthermore, Algorithm 5.1.15 with $q_0 := r_0$ generates a sequence $\{q_0, \dots, q_{2j}\}$ with the following property: Instead of squaring the BCG polynomial as in the CGS case, BCGSTAB basis vectors are of the form

$$q_{2j} = \left(\prod_{i=0}^{j-1}(\mathbf{I}_n - \omega_{2i}\mathbf{A})\right)\rho_j(\mathbf{A})r_0,$$

where $\rho_j(\mathbf{A})r_0 = r_j^{\text{BCG}}$. Lemma 5.1.17 proves this fact.

We will now proceed along the same lines as we have done in the CGS method for Algorithm 5.1.12, i.e. we demonstrate how the basis $\mathbf{V}_{k+1}$ of the BCGSTAB $\mathcal{V}$-inner product can be derived from the Algorithm 5.1.15. In contrast with the preceding method we consider even indexes from the very start. The stabilizing algorithm, when it does not break down, yields vectors

$$q_{2k} = q_{2k-2} - \alpha_{2k-2}c_{2k-2} - \omega_{2k-2}c_{2k-1}$$

or, with $\mathbf{C}_{2k-1} := (c_0, c_1, \dots, c_{2k-1})$,

$$q_{2k} = q_0 - \mathbf{C}_{2k-1}\begin{pmatrix}\alpha_0\\ \omega_0\\ \vdots\\ \alpha_{2k-2}\\ \omega_{2k-2}\end{pmatrix}.$$

From Algorithm 5.1.15 we also obtain the decomposition

$$\mathbf{C}_{2k-1} = (q_0, q_1, \dots, q_{2k})\tilde{\mathbf{G}}_{2k}, \tag{1.40}$$

where $\tilde{\mathbf{G}}_{2k} \in \mathbb{R}^{(2k+1)\times(2k)}$ has the form

$$\tilde{\mathbf{G}}_{2k} = \begin{pmatrix}1/\alpha_0 & 0 & \dots & \dots & 0\\ -1/\alpha_0 & 1/\omega_0 & & & \vdots\\ 0 & -1/\omega_0 & 1/\alpha_2 & & \\ \vdots & & \ddots & \ddots & \vdots\\ \vdots & & & -1/\alpha_{2k-2} & 1/\omega_{2k-2}\\ 0 & \dots & & & -1/\omega_{2k-2}\end{pmatrix}.$$

The vector q_{2k} can be written as

$$q_{2k} = (q_0/\|q_0\|, \dots, q_{2k}/\|q_{2k}\|)\left(\|q_0\|e_1 - \operatorname{diag}(\|q_0\|, \dots, \|q_{2k}\|)\tilde{\mathbf{G}}_{2k}\begin{pmatrix}\alpha_0\\ \omega_0\\ \vdots\\ \alpha_{2k-2}\\ \omega_{2k-2}\end{pmatrix}\right).$$

With the notations

$$\mathbf{V}_{2k+1} := (q_0/\|q_0\|, \dots, q_{2k}/\|q_{2k}\|), \quad \tilde{\mathbf{H}}_{2k} := \operatorname{diag}(\|q_0\|, \dots, \|q_{2k}\|)\tilde{\mathbf{G}}_{2k},$$

we obtain

$$q_{2k} = q_0 - \mathbf{C}_{2k-1} \begin{pmatrix} \alpha_0 \\ \omega_0 \\ \vdots \\ \alpha_{2k-2} \\ \omega_{2k-2} \end{pmatrix} = \mathbf{V}_{2k+1} \left(\|q_0\| e_1 - \tilde{\mathbf{H}}_{2k} \begin{pmatrix} \alpha_0 \\ \omega_0 \\ \vdots \\ \alpha_{2k-2} \\ \omega_{2k-2} \end{pmatrix} \right). \tag{1.41}$$

When $q_0 := r_0$, then Lemma 5.1.16 states that $\{c_0, c_1, \dots, c_{2k-1}\}$ is a basis of $\mathbf{A}\mathcal{K}_{2k}(\mathbf{A}, r_0)$ and $\{v_1, \dots, v_{2k}\}$ spans $\mathcal{K}_{2k}(\mathbf{A}, r_0)$. The matrix representation of $\wp_{2k}$ in these bases is given by (1.28) when we replace indexes k by $2k$. Because the projection space is spanned by $\{c_0, c_1, \dots, c_{2k-1}\}$ and with $q_0 := r_0$, residual vectors with even indexes have the form

$$r_{2k} = r_0 - \mathbf{C}_{2k-1} y_{2k} = \mathbf{V}_{2k+1}(\|r_0\| e_1 - \tilde{\mathbf{H}}_{2k} y_{2k}), \tag{1.42}$$

for some $y_{2k} \in \mathbb{R}^{2k}$ because of (1.41). According to (1.29), the Petrov-Galerkin condition yields a linear system

$$\mathbf{H}_{2k} y = \|r_0\| e_1,$$

where $\mathbf{H}_{2k} \in \mathbb{R}^{2k \times 2k}$ is the upper Hessenberg matrix $\tilde{\mathbf{H}}_{2k}$ without its last row. This matrix can be written as

$$\mathbf{H}_{2k} = \operatorname{diag}(\|q_0\|, \dots, \|q_{2k-1}\|) \mathbf{G}_{2k},$$

where $\mathbf{G}_{2k} \in \mathbb{R}^{2k \times 2k}$ is the upper Hessenberg matrix $\tilde{\mathbf{G}}_{2k}$ from (1.40) without its last row. Its inverse equals

$$\mathbf{H}_{2k}^{-1} = \begin{pmatrix} \alpha_0 & 0 & \dots & \dots & 0 \\ \omega_0 & \omega_0 & & & \vdots \\ \alpha_2 & \alpha_2 & \alpha_2 & 0 & \\ \vdots & & \ddots & \ddots & \vdots \\ \omega_{2k-2} & & & \omega_{2k-2} & \omega_{2k-2} \end{pmatrix} \operatorname{diag}(1/\|q_0\|, \dots, 1/\|q_{2k-1}\|).$$

In analogy with (1.30) we have

$$r_{2k} = -v_{2k+1} \tilde{h}_{2k+1,2k} \|r_0\| e_{2k}^T \mathbf{H}_{2k}^{-1} e_1 = v_{2k+1} \frac{\|q_{2k}\|}{\omega_{2k-2}} \|r_0\| e_{2k}^T \mathbf{H}_{2k}^{-1} e_1$$

and hence

$$r_{2k} = r_0 - \mathbf{C}_{2k-1} y_{2k} = \mathbf{V}_{2k+1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{\|q_{2k}\| \cdot \|r_0\|}{\omega_{2k-2}} e_{2k}^T \mathbf{H}_{2k}^{-1} e_1 \end{pmatrix} = \mathbf{V}_{2k+1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \|q_{2k}\| \end{pmatrix} = q_{2k}.$$

Thus $r_{2j} = q_{2j}$ for all j if we choose $q_0 = r_0$ and the stabilizing parameters in (1.39) appear to minimize the residual vectors. In addition, we obtain $y_{2k} = (\alpha_0, \omega_0, \dots, \omega_{2k-2})^T$. With Algorithm 5.1.15 we see

$$\begin{aligned} r_{2k} = b - \mathbf{A} x_{2k} &= r_{2k-2} - \alpha_{2k-2} c_{2k-2} - \omega_{2k-2} c_{2k-1} \\ &= b - \mathbf{A} x_{2k-2} - \alpha_{2k-2} \mathbf{A} p_{2k-2} - \omega_{2k-2} \mathbf{A} r_{2k-1}, \end{aligned}$$

hence

$$x_{2k} = x_{2k-2} + \alpha_{2k-2} p_{2k-2} + \omega_{2k-2} r_{2k-1}.$$

The method was proposed by Van der Vorst [67]. Although breakdowns of a new origin have been created, namely vanishing stabilization parameters ω_{2j}, it appears to yield smoother and faster convergence than its predecessors, the BCG and CGS methods.

Orthogonal projection with respect to the $\mathcal{V}$-inner product

In analogy with oblique projections we can define orthogonal projections with respect to iteration dependent inner products. If the involved algorithm generates bases $\{v_1, \ldots, v_k\}$ and $\{c_1, \ldots, c_k\}$ of $\mathcal{K}_k(\mathbf{A}, r_0)$ that are not necessarily orthonormal and if they posses a decomposition (1.12), then the projection is asked to be $\mathcal{V}$-orthogonal with $\mathbf{V} := \mathbf{V}_{k+1}$ in Definition 1.3.3. This significantly facilitates the computation of iterates with non-orthonormal bases, because when we put

$$x_k = x_0 + \mathbf{C}_k y_k, \quad y_k \in \mathbb{R}^k, \tag{1.43}$$

for some initial guess x_0, then

$$r_k = r_0 - \mathbf{A}\mathbf{C}_k y_k = \mathbf{V}_{k+1}(\|r_0\| e_1 - \tilde{\mathbf{H}}_k y_k).$$

The $\mathcal{V}$-orthogonality condition

$$\|r_k\| = \min_{y \in \mathbb{R}^k} \|r_0 - \mathbf{A}\mathbf{C}_k y\|_{\mathcal{V}},$$

results in the following property of y_k (see Definition 1.3.3):

$$\left\| \|r_0\| e_1 - \tilde{\mathbf{H}}_k y_k \right\| = \min_{y \in \mathbb{R}^k} \left\| \|r_0\| e_1 - \tilde{\mathbf{H}}_k y \right\|. \tag{1.44}$$

Thus the norm of r_k is being „quasi-minimized" and we have reduced the least-squares minimization problem of dimension $n \times k$ to a problem of dimension $(k+1) \times k$, as was the case in the GMRES method. In terms of classical orthogonality, the test spaces are spanned by the columns of

$$\mathbf{B}_k := \mathbf{V}_{k+1}(\mathbf{V}_{k+1}^T \mathbf{V}_{k+1})^{-2} \mathbf{V}_{k+1}^T \mathbf{A}\mathbf{C}_k.$$

With the basis $\mathbf{A}\{c_1, \ldots, c_k\}$ of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ the matrix representation of $\wp_k$ becomes with (1.7)

$$\mathbf{P}_k = \mathbf{A}\mathbf{C}_k \left((\mathbf{V}_{k+1}(\mathbf{V}_{k+1}^T \mathbf{V}_{k+1})^{-2} \mathbf{V}_{k+1}^T \mathbf{A}\mathbf{C}_k)^T \mathbf{A}\mathbf{C}_k \right)^{-1} \left(\mathbf{V}_{k+1}(\mathbf{V}_{k+1}^T \mathbf{V}_{k+1})^{-2} \mathbf{V}_{k+1}^T \mathbf{A}\mathbf{C}_k \right)^T$$

or, with (1.12),

$$\mathbf{P}_k = \mathbf{V}_{k+1} \tilde{\mathbf{H}}_k (\tilde{\mathbf{H}}_k^T \tilde{\mathbf{H}}_k)^{-1} \tilde{\mathbf{H}}_k^T (\mathbf{V}_{k+1}^T \mathbf{V}_{k+1})^{-1} \mathbf{V}_{k+1}^T. \tag{1.45}$$

In theory, the minimization property (1.9) for residuals is lost when we project $\mathcal{V}$-orthogonally and non-increasing of residual norms can not be guaranteed anymore. But compared with their oblique counterparts $\mathcal{V}$-orthogonal projections appear to stabilize irregular convergence behaviour. Indeed, the so-called TFQMR method smoothes the oscillations of the residual norms of its oblique parallel, and the QMR method does so for the BCG method. The orthogonal version of the BCGSTAB method is the QMRCGSTAB method. We will treat here briefly the QMR and the TFQMR method.

- **The quasi-minimal residual method (QMR)**:
  The QMR method is the orthogonal counterpart of the oblique BCG method. Therefore, bases are generated by the bi-orthogonal Lanczos proces (Algorithm 5.1.6), or eventually by a look-ahead version such as Algorithm 5.1.10 and they satisfy (1.31), respectively (5.6). Let us use the notations of the BCG method,

where we have a decomposition (1.12) with $\mathbf{C}_k = \mathbf{V}_k$. We assume the basis-generating algorithm does not break down and if we consider a look-ahead algorithm, let us assume in (5.5) that $m = k_{i+1} - 1$, i.e. $\mathbf{D}_m$ is nonsingular. Then, in accordance with the preceding considerations, if we replace $\mathbf{C}_k$ by $\mathbf{V}_k$, QMR iterates of the form (1.43) can be characterized by the property (1.44) and the projector is given by (1.45).

In the original version of Freund and Nachtigal [24] the underlying algorithm is a look-ahead version and in addition the least squares problem (1.44) is scaled by a weight matrix. As for the BCG method, the orthogonality condition of a QMR projection depends upon the shadow vector involved in the algorithm and in this sense the notion QMR method covers a whole class of projectors. In case we use Algorithm 5.1.6 it is possible to define iterates with three term recurrences (although their derivation differs from the one of Proposition 1.2.5). For the look-ahead version these short recurrences change to block recurrences, see Freund and Nachtigal [24]. In this paper, some relations between QMR and BCG residual norms have been pointed out. They are analogue to those of Theorem 1.3.2 and one example of the smoothing potential of the QMR method compared with the BCG method is presented later on (Example 3 in Chapter 2). We used the QMR method also in experiments where improvements of the GMRES method are discussed, because it is the method with short recurrences that is closest to GMRES from the projectional point of view.

- **The transpose free QMR method (TFQMR)**:
  The TFQMR method is the orthogonal parallel of the oblique CGS method and hence is based on the same algorithm as the CGS method. With the notations of the CGS method described above, we have a sequence $c_0, \dots, c_{k-1}$ and a sequence $v_0, \dots, v_{k-1}$ that both generate $\mathcal{K}_k(\mathbf{A}, r_0)$ if we choose q_0 to be the initial residual vector r_0 in Algorithm 5.1.12 (see Lemma 5.1.13). We have shown these bases can be decomposed according to (1.37). If we replace indexes k by $k-1$, TFQMR iterates of the form (1.43) can be characterized by the property (1.44) and the projector is given by (1.45). The method was introduced by Freund [25]. Two-term recurrences of iterates and residuals are possible for the following reason. The matrix $\mathbf{T}_k$ from Proposition 1.2.5 with $\mathbf{B}_k := \mathbf{V}_k(\mathbf{V}_k^T\mathbf{V}_k)^{-2}\mathbf{V}_k^T\mathbf{A}\mathbf{C}_{k-1}$ has the form $\tilde{\mathbf{H}}_{k-1}^T\tilde{\mathbf{H}}_{k-1}$ because of (1.37). Now $\tilde{\mathbf{H}}_j^T\tilde{\mathbf{H}}_j$ has rank j for all $j \leq k-1$ and is tridiagonal because its factors are bidiagonal. Hence we can apply Proposition 1.2.5.

Other full projection methods we did not describe above, be it oblique or orthogonal ones, include the CGNE and CGNR methods for normal equations, GCR (Elman [19]), ORTHODIR (Jea, Young [38]), ORTHOMIN (Vinsome, [70]), hybrid BCG methods (Sleijpen, Van der Vorst, Fokkema [23]) and row projection methods. Some of them might be mathematically equivalent to previously treated methods but then they are implemented in a different way. For their description we refer to the indicated literature.

1.4 Truncated and restarted projection methods

From the preceding full methods, methods that exploit orthonormal bases are the most robust. But unless additional properties of the system matrix exist, it is for them not possible to define iterates or residuals by short recurrences. If we wish to use orthonormal bases but to avoid recurrences of more than l terms for some integer l, two options immediately offer themselves:

- Truncation: Only l orthogonality conditions are kept, the remaining conditions are dropped.
- Restarting: After having reached projection onto a subspace of dimension l we restart the same or a similar process with the new initial residual vector being equal to r_l.

Of course, the two approaches can be combined (see, for example, De Sturler [11]). These techniques pay for computations of limited costs with the loss of convergence in at most n steps. In theory it is possible to apply them to all projectors seen in the previous section, but there is no need to do so for full methods with short term recurrences.

1.4.1 Truncation

Several definitions of truncated methods are used in the literature. Truncation can for example be applied to the orthogonalization process for bases. In that case, a new basis vector is orthogonalized only against the l previous ones. Adapted versions of Algorithm 5.1.1 or 5.1.2 then compute decompositions with a banded Hessenberg matrix. The GMRES method with an adapted version of Algorithm 5.1.1 can be formulated with l-term recurrences by Proposition 1.2.5 and it is possible to show that also the truncated FOM method can. These truncated versions project, as for the full versions, onto $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$. The matrix $\mathbf{V}_k$ from Arnoldi decomposition (1.12), however, (or $\mathbf{V} := \mathbf{W}_k$ from (1.16) when using the residual-based approach) is not orthonormal anymore, although the coordinate vector of iterates is extracted from the same linear equations as for GMRES or FOM (from (1.44) or (1.29), respectively). Thus the Petrov-Galerkin conditions must be understood with respect to $\mathcal{V}$-orthogonality, where $\mathbf{V} := \mathbf{V}_k$ in Definition 1.3.3.

We will here use a definition where truncation concerns the orthogonality conditions of the projector.

Definition 1.4.1 *An l-truncated projection method is an iterative method with successive residual vectors that satisfy*

$$r_k = \hat{r}_{k-1} - \wp_k(\hat{r}_{k-1}), \quad k = 1, 2, \dots,$$

where $\hat{r}_{k-1} = r_0$ if $k \leq l$ and else $\hat{r}_{k-1} = r_{k-l}$ and where the operator $\wp_k$ is a projector whose matrix representations have rank $\min(l, k)$.

When we truncate a full method, the rank-l projector is obtained by considering l-dimensional projection and test spaces that result from discarding the first $k - l$ dimensions of the spaces from the full method. Several truncated methods are listed below. Some of them do not arise from truncation of full methods. We also describe iterative methods that were not originally conceived as projection methods but appear to belong to them. For these methods, this classification may only have theoretical interest as implementation based on projection is here computationally more expensive and more complicated than the original process.

- **The steepest descent method**:
  The steepest descent method can be regarded as an l-truncated version of the CG method for symmetric positive definite matrices with $l = 1$. In the

steepest descent method we drop all but the last basis vector of the test space of the CG projector. This space, $\mathcal{K}_k(\mathbf{A}, r_0)$, is spanned by the sequence of CG residuals because of Lemma 1.3.1 and, as will be clear from the following, also the sequence of steepest descent residuals spans this space. We thus have in (1.7) $\mathbf{B}_m := r_{k-1}$ (we put $m \equiv l$ for l-truncated methods), which yields the Petrov-Galerkin condition

$$r_{k-1}^T r_k = 0.$$

Even so, the projection space is reduced to a last basis vector for the CG projection space $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$, namely $\mathbf{A}r_{k-1}$. Then residual vectors are given by

$$r_k = r_{k-1} - \alpha_{k-1}\mathbf{A}r_{k-1}$$

and with (1.7) the corresponding rank-one matrix representation of the projector is

$$\mathbf{P}_k = \frac{\mathbf{A}r_{k-1}r_{k-1}^T}{r_{k-1}^T\mathbf{A}r_{k-1}},$$

where $r_{k-1}^T\mathbf{A}r_{k-1} \neq 0$ because $\mathbf{A}$ is positive definite. Hence

$$\alpha_{k-1} = \frac{r_{k-1}^T r_{k-1}}{r_{k-1}^T\mathbf{A}r_{k-1}}$$

and iterates have the form

$$x_k = x_{k-1} + \alpha_{k-1}r_{k-1}.$$

- **Richardson iteration**:
A very simple example of an iterative method with two-term recurrence is Richardson iteration (for some applications see for example Fischer, Reichel [22]). Iterates are characterized by

$$x_k = x_{k-1} + \delta_k r_{k-1}, \quad \delta_k \in \mathbb{R},$$

and therefore residuals by

$$r_k = r_{k-1} - \delta_k\mathbf{A}r_{k-1}, \tag{1.46}$$

yielding

$$r_k = \prod_{i=1}^{k}(\mathbf{I}_n - \delta_i\mathbf{A})r_0.$$

This method can be seen as a 1-truncated projection method with projection spaces span$\{\mathbf{A}r_{k-1}\}$. If s_k is given by

$$s_k := r_{k-1} - \frac{r_k^T r_{k-1}}{\|r_k\|^2}\, r_k = r_{k-1} - \frac{r_{k-1}^T(r_{k-1} - \delta_k\mathbf{A}r_{k-1})}{\|r_{k-1} - \delta_k\mathbf{A}r_{k-1}\|^2}\left(r_{k-1} - \delta_k\mathbf{A}r_{k-1}\right),$$

then the residual vector r_k is orthogonal to s_k by construction. With (1.7) the rank-one matrix representing $\wp_k$ becomes

$$\mathbf{P}_k = \frac{\delta_k\mathbf{A}r_{k-1}s_k^T}{\delta_k s_k^T\mathbf{A}r_{k-1}} = \frac{\delta_k\mathbf{A}r_{k-1}s_k^T}{s_k^T r_{k-1}}.$$

The last equality follows from (1.46) and the orthogonality of r_k and s_k. The projector is well defined because of

$$r_{k-1}^T s_k = \|r_{k-1}\|^2 - \frac{(r_k^T r_{k-1})^2}{\|r_k\|^2}.$$

This expression vanishes only in case $\alpha r_k = r_{k-1}$ for some $\alpha \in \mathbb{R}$. But that would imply the projection space is span$\{\mathbf{A}r_{k-1}\}$=span$\{r_{k-1}\}$ and the $(k-1)$st residual would already have vanished.
For this method executing projections does not make sense in practice since the computation of the projector asks for the vector s_k and hence for the wanted residual itself.

- **Matrix splitting methods**:
Let us write the system matrix $\mathbf{A}$ in the form

$$\mathbf{A} = \mathbf{M} - \mathbf{N},$$

where $\mathbf{M}$ is nonsingular. Then a matrix splitting method is an iterative method defined by

$$\mathbf{M}x_k = \mathbf{N}x_{k-1} + b, \quad k \geq 1.$$

Thus we have

$$x_k = \mathbf{M}^{-1}(\mathbf{M} - \mathbf{A})x_{k-1} + \mathbf{M}^{-1}b = (\mathbf{I} - \mathbf{M}^{-1}\mathbf{A})x_{k-1} + \mathbf{M}^{-1}b$$

and the residuals satisfy

$$r_k = r_{k-1} - \mathbf{A}\mathbf{M}^{-1}r_{k-1}. \tag{1.47}$$

They are orthogonal to the vector

$$s_k \equiv r_{k-1} - \frac{r_k^T r_{k-1}}{\|r_k\|^2} r_k = r_{k-1} - \frac{r_{k-1}^T(r_{k-1} - \mathbf{A}\mathbf{M}^{-1}r_{k-1})}{\|r_{k-1} - \mathbf{A}\mathbf{M}^{-1}r_{k-1}\|^2}\left(r_{k-1} - \mathbf{A}\mathbf{M}^{-1}r_{k-1}\right).$$

As for Richardson iteration matrix splitting methods can theoretically be seen as 1-truncated projection methods with projectors

$$\mathbf{P}_k = \frac{\mathbf{A}\mathbf{M}^{-1}r_{k-1}s_k^T}{s_k^T\mathbf{A}\mathbf{M}^{-1}r_{k-1}} = \frac{\mathbf{A}\mathbf{M}^{-1}r_{k-1}s_k^T}{s_k^T r_{k-1}}.$$

The projection is onto span$\{\mathbf{A}\mathbf{M}^{-1}r_{k-1}\}$ and orthogonal to span$\{s_k\}$ and is well defined for the same reasons as for Richardson iteration. The nonsingular matrix $\mathbf{M}$ is best chosen such, that it is close to $\mathbf{A}$ in some sense, but more easy to invert than $\mathbf{A}$ itself. Many methods are based on the splitting

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U},$$

where $\mathbf{D}$ =diag($\mathbf{A}$), $\mathbf{L}$ is strictly lower triangular and $\mathbf{U}$ strictly upper triangular. With these notations the following matrix splitting methods are characterized by their choice of $\mathbf{M}$:

  - **The Jacobi method**: $\mathbf{M} := \mathbf{D} = \mathrm{diag}(\mathbf{A})$.
  - **The Gauss-Seidel method**: $\mathbf{M} := \mathbf{D} - \mathbf{L}$.

- **The successive over-relaxating method (SOR)**: $\mathbf{M} := \frac{1}{\omega}\mathbf{D} - \mathbf{L}$ with relaxation parameter $1/\omega$. In order to enable convergence ω should be chosen to be larger than one (i.e. over-relaxing).
- **The symmetric successive over-relaxating method (SSOR)**: When we alternate SOR iterations with SOR iterative steps in which the ordering of unknowns is reversed we obtain the SSOR method. This yields

$$\mathbf{M} = \frac{1}{\omega(2-\omega)}(\mathbf{I}_n - \omega\mathbf{D}^{-1}\mathbf{L})(\mathbf{I}_n - \omega\mathbf{D}^{-1}\mathbf{U}).$$

1.4.2 Restarting

Restarted methods can be defined as follows.

Definition 1.4.2 *An m-restarted projection method is an iterative method with successive residual vectors that satisfy*

$$r_k = \hat{r}_{k-1} - \wp_k(\hat{r}_{k-1}), \quad k = 1, 2, \dots,$$

where $\hat{r}_{k-1} = r_{k-1-(k-1) \bmod m}$ *and where the operator* $\wp_k$ *is a projector whose matrix representations have rank* $1 + (k-1) \bmod m$.

When we restart a full projection method based on Krylov subspaces, we put the projection space of the k-th iteration to be equal to

$$\mathbf{A}\mathcal{K}_{1+(k-1) \bmod m}(\mathbf{A}, r_{k-1-(k-1) \bmod m})$$

and define the test space analogously.

The classical method to restart is the GMRES method because firstly the full version asks for full, expensive orthonormalization and secondly, though we lose by restarting the fact convergence occurs in at most n iterations, at least GMRES generated residual norms do not increase. We will denote the GMRES method restarted after m iterations by GMRES(m). If $\mathbf{A}$ is positive definite a well-known result (Eisenstat, Elman, Schultz [18]) guarantees that residual norms even decrease as long as all projection spaces contain the initial residual. Without assuming $\mathbf{A}$ is positive definite, however, solving a linear system by restarting the GMRES method is sometimes a laborious task. In the worst case, convergence of the method restarted after m iterations is not guaranteed at all for $m < n$ (for some global convergence criteria we refer to Saad [59] and [61], for criteria in dependency of the restart parameter m of GMRES(m) see for example Zítko [78], the worst case for normal matrices is discussed in Liesen, Tichý [47]). If restarted GMRES does converge, decreasing of residual norms can still be very slow. If the process becomes unsuitable for practical use, we shall call this phenomenon *stagnation*.

One expects convergence to be faster when the dimension of the projection space grows, i.e. the restart parameter m is larger. This is not entirely true because the distance of $\hat{r}_{k-1}$ to the projection space $\mathbf{A}\mathcal{K}_m(\mathbf{A}, \hat{r}_{k-1})$ is also influenced by the choice of $\hat{r}_{k-1}$. Examples exist in which convergence is faster for smaller restart parameters than for larger ones, because the resulting vectors $\hat{r}_{k-1}$ generate closer Krylov subspaces (see Eiermann, Ernst, Schneider [16]). To accelerate the convergence speed of stagnating restarted projection methods, especially the GMRES method, various techniques have been proposed in the literature. Most of them are based on deflation of eigenvalues.

Deflation techniques

This strategy applies to cases where the eigenvalue distribution of the system matrix $\mathbf{A}$ is known, a priori, to have a negative influence on the convergence speed of the method. In the GMRES case, for example, we can derive the bound

$$\frac{\|r_{m+1}\|}{\|r_0\|} \leq \|\mathbf{Z}\|\|\mathbf{Z}^{-1}\|\varepsilon^{(m)},$$

where m is the restart parameter, the columns of $\mathbf{Z}$ are eigenvectors of $\mathbf{A}$ and

$$\varepsilon^{(m)} = \min_{\rho\in\Pi_m^0} \max_{\lambda\in\sigma(\mathbf{A})} |\rho(\lambda)|, \tag{1.48}$$

(see Saad, Schultz [56]). Π_m^0 denotes the polynomials ρ of degree m with $\rho(0) = 1$. This bound however, assumes that $\mathbf{A}$ be diagonalizable, i.e. the existence of a decomposition $\mathbf{A} = \mathbf{Z}\Lambda\mathbf{Z}^{-1}$, where Λ is a diagonal matrix. For matrices that are far from normal the bound might not be indicative because the conditioning of the eigenvectors given by the columns of $\mathbf{Z}$ can be very poor. But on the other hand, in several examples a convergence behaviour that is related to the spectrum has been observed (for example in Van der Vorst, Vuik [68]). To improve the convergence of restarted methods in such cases one can *deflate* the spectrum of $\mathbf{A}$, that is eliminate eigenvalues from the spectrum yielding a tighter bound due to (1.48). In particular, it is assumed and seen in various experiments that the eigenvalues closest to the origin hamper convergence most.

Techniques achieving deflation of the spectrum exploit the fact that the projection process of the corresponding full method implicitly provides approximations to eigenvalues. In the case of Krylov projection spaces, approximate eigenvalues are given by the eigenvalues (called *Ritz values*) of the matrix $\tilde{\mathbf{H}}_k$ of (1.12) without its last row because this matrix represents the restriction of $\mathbf{A}$ onto the kth Krylov subspace. Eigenvectors of this matrix yield approximate eigenvectors of $\mathbf{A}$ by multiplication with the matrix $\mathbf{V}_k$ from (1.12) (they are called *Ritz vectors*). Information about the quality of the approximate eigenvalue-eigenvector pairs can be easily obtained from decomposition (1.12), this will be seen later on in Chapter 4, (4.4). Thus spectral information gained during one restart cycle is exploited in the next one. Deflation has also been studied for some methods that do not require restarting, such as the CG method for symmetric positive definite matrices (e.g. Nicolaides [52]), because in that case the connection between convergence and eigenvalue distribution is more or less obvious.

In the context of restarting, Morgan [50] proposes augmentation of the projection space with approximate eigenvectors. The influence of the approximate eigenvalues they correspond to essentially vanishes. A different approach is the construction of a left preconditioner whose product with $\mathbf{A}$ yields a deflated matrix (proposed by Baglama et al.[3]) or, similarly, use deflating right preconditioning (Erhel et al.[7]). In Eiermann et al.[16] it is shown that Morgan's method eventually has better convergence properties than the methods based on preconditioning. A further deflating preconditioning technique is presented in Kharchenko, Yeremin [41]. Finally, block Krylov subspace methods (described in Chan, Wan [8], Farhat, Crivelli, Roux [10], Fischer [21], Prasad, Keyes, Kane [40]) solving multiple right-hand sides can be used for deflation purposes by choosing right-hand sides to be approximate eigenvectors (Saad, Chapman [9],[60]). To illustrate at least one deflation technique, we will here describe Erhel's method in more detail. We have used this deflation technique to compare it in numerical experiments with the new convergence accelerating techniques presented in the following chapters.

The construction of a right preconditioner $\mathbf{M}$ such that $\mathbf{AM}^{-1}$ has a deflated spectrum is based on the following proposition due to Erhel et al. [7]. We assume $\mathbf{A}$ is diagonalizable with eigenvalues satisfying

$$|\lambda_1| \leq |\lambda_2| \leq \ldots \leq |\lambda_n|$$

and we assume the columns of an orthonormal matrix $\mathbf{U}$ span the $\mathbf{A}$-invariant subspace corresponding to the r eigenvalues of smallest modulus, i.e. corresponding to $\lambda_1, \ldots, \lambda_r$. Furthermore, let the matrix $\mathbf{T}$ represent the restriction of $\mathbf{A}$ onto span($\mathbf{U}$), that is $\mathbf{T} = \mathbf{U}^T\mathbf{AU}$.

Proposition 1.4.3 *The matrix*

$$\mathbf{M} := \mathbf{I}_n + \mathbf{U}\left(\frac{1}{|\lambda_n|}\mathbf{T} - \mathbf{I}_r\right)\mathbf{U}^T$$

is nonsingular with

$$\mathbf{M}^{-1} = \mathbf{I}_n + \mathbf{U}(|\lambda_n|\mathbf{T}^{-1} - \mathbf{I}_r)\mathbf{U}^T \tag{1.49}$$

and the eigenvalues of $\mathbf{AM}^{-1}$ *are* $\lambda_{r+1}, \ldots, \lambda_n, |\lambda_n|, \ldots, |\lambda_n|$ *, where* $|\lambda_n|$ *has multiplicity at least* r*.*

Proof: Let $\mathbf{W}$ be the orthogonal complement of $\mathbf{U}$ in $\mathbb{R}^n$. Then

$$\begin{pmatrix} \mathbf{U}^T \\ \mathbf{W}^T \end{pmatrix} \mathbf{A}(\mathbf{U}, \mathbf{W}) = \begin{pmatrix} \mathbf{T} & \mathbf{A}_{1,2} \\ 0 & \mathbf{A}_{2,2} \end{pmatrix},$$

where $\mathbf{A}_{1,2} = \mathbf{U}^T\mathbf{AW}$, $\mathbf{A}_{2,2} = \mathbf{W}^T\mathbf{AW}$.

We can write $\mathbf{M}$ in the form $\mathbf{M} = \frac{1}{|\lambda_n|}\mathbf{UTU}^T + (\mathbf{I}_n - \mathbf{UU}^T) = \frac{1}{|\lambda_n|}\mathbf{UTU}^T + \mathbf{WW}^T$, hence

$$\mathbf{M} = (\mathbf{U}, \mathbf{W}) \begin{pmatrix} \frac{1}{|\lambda_n|}\mathbf{T} & 0 \\ 0 & \mathbf{I}_{n-r} \end{pmatrix} \begin{pmatrix} \mathbf{U}^T \\ \mathbf{W}^T \end{pmatrix}$$

and the inverse of $\mathbf{M}$ equals

$$\mathbf{M}^{-1} = (\mathbf{U}, \mathbf{W}) \begin{pmatrix} |\lambda_n|\mathbf{T}^{-1} & 0 \\ 0 & \mathbf{I}_{n-r} \end{pmatrix} \begin{pmatrix} \mathbf{U}^T \\ \mathbf{W}^T \end{pmatrix}.$$

Therefore, the preconditioned matrix $\mathbf{AM}^{-1}$ takes the form

$$\mathbf{AM}^{-1} = (\mathbf{U}, \mathbf{W}) \begin{pmatrix} |\lambda_n|\mathbf{I}_r & \mathbf{A}_{1,2} \\ 0 & \mathbf{A}_{2,2} \end{pmatrix} \begin{pmatrix} \mathbf{U}^T \\ \mathbf{W}^T \end{pmatrix},$$

hence its eigenvalues are $|\lambda_n|$ and the eigenvalues of $\mathbf{A}_{2,2}$. □

During the projective process of the projection method, however, exact invariant subspaces $\mathbf{U}$ to construct the above preconditioner are in general not available. Instead, we can compute at the end of every restart cycle Ritz vectors corresponding to approximate eigenvalues of smallest modulus. Also, the value $|\lambda_n|$ must be approximated by the largest Ritz value $|\tilde{\lambda}|$. With these approximations we can modify the proof of the above proposition and we obtain a perturbed preconditioned matrix $\mathbf{A}\tilde{\mathbf{M}}^{-1}$ that is similar to a matrix of the form

$$\begin{pmatrix} |\tilde{\lambda}_n|\mathbf{I}_r & \tilde{\mathbf{A}}_{1,2} \\ |\tilde{\lambda}_n|\tilde{\mathbf{A}}_{2,1}\tilde{\mathbf{T}}^{-1} & \tilde{\mathbf{A}}_{2,2} \end{pmatrix},$$

where $\tilde{\mathbf{A}}_{2,1} = \mathbf{W}^T\mathbf{A}\tilde{\mathbf{U}}$, $\tilde{\mathbf{U}}$ is the approximate invariant subspace and $\tilde{\mathbf{T}} = \tilde{\mathbf{U}}^T\mathbf{A}\tilde{\mathbf{U}}$. The quality of the preconditioner depends upon the size of the block $|\tilde{\lambda}_n|\tilde{\mathbf{A}}_{2,1}\tilde{\mathbf{T}}^{-1}$, which in turn depends on the distance of the Ritz vectors to the corresponding eigenvectors of $\mathbf{A}$. Thus the motivation for using this technique is based on somewhat heuristic assumptions: The system matrix must be diagonalizable and Ritz values must converge to eigenvalues, preferably beginning with the smallest values. This last assumption has to our knowledge never been fully proved (attempts have been made for example in Sorensen [63]), though it has been frequently observed (as in Van der Vorst, Vuik [68]). Indeed, Erhel's method does very well in applications where eigenvalues of small modulus slow down convergence. The overall process can be roughly described as follows:

1. Execute m GMRES projections for the problem $\mathbf{A}x = b$ with some initial guess x_0
2. Compute from the Hessenberg matrix involved in the projections the r Ritz values of smallest modulus and their corresponding Ritz vectors
3. Construct an orthonormal matrix $\mathbf{U}$ whose columns span the space spanned by the Ritz vectors and define $\mathbf{M}^{-1}$ according to (1.49)
4. Restart the process applied to the preconditioned system $\mathbf{AM}^{-1}y = b$, $x = \mathbf{M}^{-1}y$.

The method of Baglama et al.[3] differs from the above process in that preconditioning is applied from the left and a more sophisticated technique to approximate invariant subspaces, the implicitly restarted Arnoldi process (Sorensen [63]), is exploited. But the construction of the preconditioner is based on the same principle. It is possible to successively deflate all eigenvalues if we adapt the third step of the process:

3. Orthogonalize newly computed Ritz vectors against the orthonormal basis of the space spanned by all previously computed Ritz vectors, obtain a new orthonormal matrix $\mathbf{U}$ whose columns span the space spanned by old and new Ritz vectors and define $\mathbf{M}^{-1}$ according to (1.49)

Then, under the assumption that all Ritz vectors are exact eigenvectors, the resulting preconditioner deflates eigenvalues corresponding to newly computed Ritz values and previously computed Ritz values as well. This has not been worked out in the paper of Erhel [7] and we demonstrate it below per induction:

Having executed i restart cycles, we assume we have an $\mathbf{A}$-invariant subspace that is spanned by the orthonormal columns of $\mathbf{U}$ and that belongs to the r_i smallest eigenvalues of $\mathbf{A}$. Let $\mathbf{M}_i^{-1}$ be the corresponding preconditioner, defined according to (1.49). With Proposition 1.4.3 the matrix $\mathbf{AM}_i^{-1}$ has eigenvalues $\lambda_{r_i+1}, \ldots, \lambda_n, |\lambda_n|, \ldots, |\lambda_n|$. The proof of Proposition 1.4.3 also shows that

$$\mathbf{AM}_i^{-1}\mathbf{U} = |\lambda_n|\mathbf{U},$$

hence $\mathbf{U}$ is not only $\mathbf{A}$-invariant but also $\mathbf{AM}_i^{-1}$-invariant. We put $\mathbf{T}_i := \mathbf{U}^T\mathbf{AU}$. Then we execute a next cycle applied to $\mathbf{AM}_i^{-1}$ and assume we find an $\mathbf{AM}_i^{-1}$-invariant subspace $\mathbf{U}'$ that belongs to the $r_{i+1} - r_i$ smallest eigenvalues of $\mathbf{AM}_i^{-1}$, that is to $\lambda_{r_i+1}, \ldots, \lambda_{r_{i+1}}$. If we orthonormalize $\mathbf{U}'$ against $\mathbf{U}$, i.e. we construct an orthonormal basis $(\mathbf{U}, \mathbf{V})$ of $\mathbf{U} \oplus \mathbf{U}'$, then $(\mathbf{U}, \mathbf{V})$ is $\mathbf{AM}_i^{-1}$-invariant. Moreover, if $\mathbf{W}$ is the orthogonal complement in $\mathbb{R}^n$ of $(\mathbf{U}, \mathbf{V})$, we can write

$$\mathbf{AM}_i^{-1} = (\mathbf{U}, \mathbf{V}, \mathbf{W}) \begin{pmatrix} |\lambda_n|\mathbf{I}_{r_i} & \mathbf{A}_{1,2} & \mathbf{A}_{1,3} \\ 0 & \mathbf{T}_{i+1} & \mathbf{A}_{2,3} \\ 0 & 0 & \mathbf{A}_{3,3} \end{pmatrix} \begin{pmatrix} \mathbf{U}^T \\ \mathbf{V}^T \\ \mathbf{W}^T \end{pmatrix},$$

where $\mathbf{T}_{i+1} := \mathbf{V}^T\mathbf{AM}_i^{-1}\mathbf{V}$, $\mathbf{A}_{1,2} := \mathbf{U}^T\mathbf{AM}_i^{-1}\mathbf{V}$, $\mathbf{A}_{1,3} := \mathbf{U}^T\mathbf{AM}_i^{-1}\mathbf{W}$, $\mathbf{A}_{2,3} := \mathbf{V}^T\mathbf{AM}_i^{-1}\mathbf{W}$ and $\mathbf{A}_{3,3} := \mathbf{W}^T\mathbf{AM}_i^{-1}\mathbf{W}$. The matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = \mathbf{AM}_i^{-1}\mathbf{M}_i = (\mathbf{U}, \mathbf{V}, \mathbf{W}) \begin{pmatrix} |\lambda_n|\mathbf{I}_{r_i} & \mathbf{A}_{1,2} & \mathbf{A}_{1,3} \\ 0 & \mathbf{T}_{i+1} & \mathbf{A}_{2,3} \\ 0 & 0 & \mathbf{A}_{3,3} \end{pmatrix} \begin{pmatrix} \mathbf{U}^T \\ \mathbf{V}^T \\ \mathbf{W}^T \end{pmatrix} \left(\mathbf{I}_n + \mathbf{U}(\mathbf{T}_i/|\lambda_n| - \mathbf{I}_{r_i})\mathbf{U}^T\right) =$$

$$(\mathbf{U},\mathbf{V},\mathbf{W})\begin{pmatrix}|\lambda_n|\mathbf{I}_{r_i} & \mathbf{A}_{1,2} & \mathbf{A}_{1,3}\\ 0 & \mathbf{T}_{i+1} & \mathbf{A}_{2,3}\\ 0 & 0 & \mathbf{A}_{3,3}\end{pmatrix}\left(\begin{pmatrix}\mathbf{U}^T\\ \mathbf{V}^T\\ \mathbf{W}^T\end{pmatrix}+\begin{pmatrix}\mathbf{I}_{r_i}\\ 0\\ 0\end{pmatrix}(\mathbf{T}_i/|\lambda_n|-\mathbf{I}_{r_i})\mathbf{U}^T\right)=$$

$$(\mathbf{U},\mathbf{V},\mathbf{W})\begin{pmatrix}|\lambda_n|\mathbf{I}_{r_i} & \mathbf{A}_{1,2} & \mathbf{A}_{1,3}\\ 0 & \mathbf{T}_{i+1} & \mathbf{A}_{2,3}\\ 0 & 0 & \mathbf{A}_{3,3}\end{pmatrix}\left(\begin{pmatrix}\mathbf{U}^T\\ \mathbf{V}^T\\ \mathbf{W}^T\end{pmatrix}+\begin{pmatrix}\mathbf{T}_i/|\lambda_n|-\mathbf{I}_{r_i}\\ 0\\ 0\end{pmatrix}\mathbf{U}^T\right)=$$

$$(\mathbf{U},\mathbf{V},\mathbf{W})\begin{pmatrix}|\lambda_n|\mathbf{I}_{r_i} & \mathbf{A}_{1,2} & \mathbf{A}_{1,3}\\ 0 & \mathbf{T}_{i+1} & \mathbf{A}_{2,3}\\ 0 & 0 & \mathbf{A}_{3,3}\end{pmatrix}\begin{pmatrix}\mathbf{T}_i/|\lambda_n| & 0 & 0\\ 0 & \mathbf{I}_{r_{i+1}-r_i} & 0\\ 0 & 0 & \mathbf{I}_{n-r_{i+1}}\end{pmatrix}\begin{pmatrix}\mathbf{U}^T\\ \mathbf{V}^T\\ \mathbf{W}^T\end{pmatrix},$$

hence

$$\mathbf{A}(\mathbf{U},\mathbf{V},\mathbf{W})=(\mathbf{U},\mathbf{V},\mathbf{W})\begin{pmatrix}\mathbf{T}_i & \mathbf{A}_{1,2} & \mathbf{A}_{1,3}\\ 0 & \mathbf{T}_{i+1} & \mathbf{A}_{2,3}\\ 0 & 0 & \mathbf{A}_{3,3}\end{pmatrix}$$

and thus $(\mathbf{U},\mathbf{V})$ is not only $\mathbf{AM}_i^{-1}$-invariant but also $\mathbf{A}$-invariant. Moreover, the subspace spanned by $(\mathbf{U},\mathbf{V})$ belongs to the eigenvalues $\lambda_1,\ldots,\lambda_{r_{i+1}}$ of $\mathbf{A}$. Therefore we can apply Proposition 1.4.3 to the $\mathbf{A}$-invariant subspace that is spanned by the columns of $(\mathbf{U},\mathbf{V})$ and obtain a matrix $\mathbf{AM}_{i+1}^{-1}$ that has a spectrum consisting of the eigenvalues $\lambda_{r_{i+1}+1},\ldots,\lambda_n,|\lambda_n|,\ldots,|\lambda_n|$.

Other approaches

The method of Erhel et al. is a restarted method where the projection space of each cycle depends upon the updated preconditioner. At the end of the i-th restart it equals $\mathbf{AM}_i^{-1}\mathcal{K}_m(\mathbf{AM}_i^{-1},r_{im})$. Morgan's method, on the other hand, seeks during the process approximate invariant subspaces and augments the projection space $\mathbf{A}\mathcal{K}_m(\mathbf{A},r_{im})$ with it. Even larger flexibility of projection spaces is enabled in a method such as flexible GMRES (Saad [58]). This is not a method that seeks to deflate. Decompositions of the form (1.12) are generated, but $\mathbf{V}_k$ and $\mathbf{C}_k$ are not asked to be ascending bases of $\mathcal{K}_k(\mathbf{A},\hat{r}_{k-1})$. Instead, two vectors v_j and c_j are connected by the relation

$$c_j=\mathbf{M}_j^{-1}v_j, \tag{1.50}$$

where $\mathbf{M}_j$ is some nonsingular and iteration dependent preconditioner that approximates $\mathbf{A}$. The next basis vector v_{j+1} results from orthonormalization of $\mathbf{A}c_j$ against the previous vectors v_i, $i\leq j$, and $\mathbf{V}_k$ remains orthonormal. Therefore, the computation of iterates of the form (1.13) can be reduced to solving a $(k{+}1)\times k$-dimensional least squares problem of the form (1.21) as in full GMRES. Of course, the columns of $\mathbf{C}_k$ and $\mathbf{V}_k$ do not span Krylov subspaces anymore. The computation of c_j according to (1.50) is referred to as *inner iteration.* Indeed, because the preconditioner $\mathbf{M}_j$ is close to $\mathbf{A}$ is some sense, c_j approximately satisfies $\mathbf{A}c_j=v_j$ and can be found by executing several iterations of any iterative method, especially of the outer method itself, applied to the same matrix but with the right hand side v_j. Inner iterations can enhance convergence and this has been attributed to the fact that when (1.50) is solved exactly for some j, that is $\mathbf{A}c_j=v_j$, then the $(j+1)$-st row of the corresponding Hessenberg matrix $\tilde{\mathbf{H}}_j$ from (1.12) has only zero's. Thus (1.21) can be solved exactly when $\mathbf{H}_j$ is nonsingular and in that case the residual vector vanishes. If (1.50) is solved only approximatively, at least higher convergence speed can be expected.

A different method that uses a double loop to keep a part of the Krylov subspace was presented in Van der Vorst and Vuik [69]. A last option proposed to accelerate the convergence of restarted methods that we would only like to mention, apart from the new techniques proposed in the next chapter, is polynomial preconditioning (Joubert [39]).

Chapter 2

A rank-one update for the initial cycle

In the preceding chapter we considered a large scale of methods to solve problem (1.1). If we want to use a projection method based on Krylov subspaces we have the choice between relatively inexpensive methods with short recurrences or more robust methods without short recurrences. Although smoothness of convergence of the former methods can be enhanced by exploiting quasi-minimalization as in the QMR method or by introduction of stabilization parameters (BCGSTAB), the latter class has naturally the most reliable convergence properties. We have seen in Theorem 1.3.2 that oblique representants of this class, for example the FOM method, are more susceptible to irregular behaviour than their orthogonal counterparts. If the system matrix is nonsymmetric, the GMRES method seems most appropriate in cases where robustness has high priority. Unfortunately, the absence of short recurrences entails that we must restart the method in practice and this can seriously slow down convergence speed. Therefore, an important part of today's research in the field of projection methods concern convergence analysis and improvement of the restarted GMRES method.

We listed some of the best known techniques to accelerate restarted GMRES at the end of the first chapter. It is interesting to notice that most of these techniques assume convergence of restarted GMRES is related to the system matrix only, without taking in account the given right-hand side. For example, the techniques try to modify the spectrum of $\mathbf{A}$ or they search for invariant subspaces of $\mathbf{A}$. But we know from chapter one that the convergence behaviour of GMRES is given by the evolution of the distance between projection space and the residual we project. This residual is, of course, related to the right-hand side and also the projection space is, because it is the Krylov subspace generated by the residual and the system matrix.

In this chapter, we will find for the given right-hand side of (1.1) modifications of $\mathbf{A}$ that yield any desired convergence curve, regardless of the properties of $\mathbf{A}$. We exploit the modified matrix to accelerate the GMRES process for the original matrix. The connection between the proposed procedure and the right-hand side is particularly narrow because we update $\mathbf{A}$ with a matrix of small rank that is immediately constructed from b. Let us begin with this construction.

2.1 The Sherman-Morrison-Woodbury theorem

One of the most powerful techniques to improve poor properties of a linear system with regard to a certain method that is applied to it, consists of passing to a preconditioned system. Preconditioning is based on the simple idea of multiplying the matrix with another specially constructed matrix. In case of left preconditioning the solution of the preconditioned system is the same as the solution of the original system, when using right preconditioning one more matrix vector product is needed to obtain the wanted solution. On the other hand, residuals of right preconditioned systems are equal to the unpreconditioned residuals whereas residual norms can increase with left preconditioning.

The situation becomes slightly more complicated when we subtract a specially constructed matrix from the system matrix instead of multiplying with it. The solution of the auxiliary system is in general not equal to the one of the first system and, as we will see later, finding the wanted solution can lead to division by zero but it is feasible to circumvent such cases. The formula that enables us to change the linear system by matrix subtraction is the Sherman-Morrison formula for inversion of rank-m updated matrices.

Theorem 2.1.1 *(Sherman-Morrison-Woodbury):*
Let the matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ be nonsingular and $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n\times m}$ be rank m matrices with $m \leq n$. Then the rank-m updated matrix $\mathbf{A} + \mathbf{U}\mathbf{V}^T$ is nonsingular if and only if the m-dimensional matrix $\mathbf{I}_m + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}$ is nonsingular and its inverse equals

$$(\mathbf{A} + \mathbf{U}\mathbf{V}^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I}_m + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{A}^{-1}. \tag{2.1}$$

Proof: Under the assumption that $\mathbf{I}_m + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}$ is nonsingular, we have

$$(\mathbf{A} + \mathbf{U}\mathbf{V}^T)(\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I}_m + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{A}^{-1}) =$$

$$\mathbf{I}_n - \mathbf{U}(\mathbf{I}_m + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{A}^{-1} + \mathbf{U}\left(\mathbf{I}_m - \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}(\mathbf{I}_m + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\right)\mathbf{V}^T\mathbf{A}^{-1}.$$

Straightforward computation shows that

$$\left(\mathbf{I}_m - \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}(\mathbf{I}_m + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\right)(\mathbf{I}_m + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}) = \mathbf{I}_m.$$

Hence $(\mathbf{A} + \mathbf{U}\mathbf{V}^T)^{-1}$ exists and (2.1) holds.
On the other hand, assuming non-singularity of $\mathbf{A}+\mathbf{U}\mathbf{V}^T$ also implies non-singularity of $\mathbf{I} + \mathbf{A}^{-1}\mathbf{U}\mathbf{V}^T$. Let $\mathbf{U}_1$ denote the $n \times n$ matrix whose first m columns coincide with the columns of $\mathbf{U}$ and the remaining columns are zero vectors and let us define $\mathbf{V}_1$ analogously. Then

$$\mathbf{A}^{-1}\mathbf{U}\mathbf{V}^T = \mathbf{A}^{-1}\mathbf{U}_1\mathbf{V}_1^T = \mathbf{A}^{-1}\mathbf{U}_1\mathbf{A}^{-1}\mathbf{A}\mathbf{V}_1^T.$$

When we put $\mathbf{A}^{-1}\mathbf{U}_1 =: \tilde{\mathbf{V}}^T$ and $\mathbf{A}\mathbf{V}_1^T =: \tilde{\mathbf{U}}$, then the matrix $\mathbf{I}_n + \tilde{\mathbf{V}}^T\mathbf{A}^{-1}\tilde{\mathbf{U}}$ is nonsingular and by the first half of the proof also $\mathbf{A} + \tilde{\mathbf{U}}\tilde{\mathbf{V}}^T$ is. But

$$\mathbf{A} + \tilde{\mathbf{U}}\tilde{\mathbf{V}}^T = \mathbf{A} + \mathbf{A}\mathbf{V}_1^T\mathbf{A}^{-1}\mathbf{U}_1 = \mathbf{A}(\mathbf{I}_n + \mathbf{V}_1^T\mathbf{A}^{-1}\mathbf{U}_1).$$

In the last expression the matrix between brackets has the form

$$\mathbf{I}_n + \mathbf{V}_1^T\mathbf{A}^{-1}\mathbf{U}_1 = \begin{pmatrix} \mathbf{I}_m + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-m} \end{pmatrix}$$

and $\mathbf{I}_n + \mathbf{V}_1^T\mathbf{A}^{-1}\mathbf{U}_1$ is nonsingular. Therefore also $\mathbf{I}_m + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}$ is nonsingular and (2.1) holds with the first half of the proof. □

In order to switch to an auxiliary system we change the original matrix $\mathbf{A}$ of (1.1) to

$$\hat{\mathbf{A}} := \mathbf{A} - \mathbf{U}\mathbf{V}^T,$$

for some $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n\times m}$. If $\mathbf{A} - \mathbf{U}\mathbf{V}^T$ is nonsingular the solution of (1.1) can be written as

$$\mathbf{A}^{-1}b = (\hat{\mathbf{A}} + \mathbf{U}\mathbf{V}^T)^{-1}b = \left(\hat{\mathbf{A}}^{-1} - \hat{\mathbf{A}}^{-1}\mathbf{U}(\mathbf{I}_m + \mathbf{V}^T\hat{\mathbf{A}}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\hat{\mathbf{A}}^{-1}\right)b. \qquad (2.2)$$

For $m = 1$, $\mathbf{U} := b$ and $\mathbf{V} := y \in \mathbb{R}^n$, this equation changes to

$$\mathbf{A}^{-1}b = (\hat{\mathbf{A}} + by^T)^{-1}b = \hat{\mathbf{A}}^{-1}b - \hat{\mathbf{A}}^{-1}b(1 + y^T\hat{\mathbf{A}}^{-1}b)^{-1}y^T\hat{\mathbf{A}}^{-1}b. \qquad (2.3)$$

With the auxiliary system being defined by

$$\hat{\mathbf{A}}x = b, \qquad (2.4)$$

where

$$\hat{\mathbf{A}} := \mathbf{A} - by^T, \qquad \det(\mathbf{A} - by^T) \neq 0,$$

it is clear that we can find the solution of the original system (1.1) by solving the auxiliary system (2.4) and substituting $\hat{\mathbf{A}}^{-1}b$ in (2.3).

One could also choose $m > 1$ and thus in (2.2) $\mathbf{V}$ would consist of several columns v_i, $i \leq m$. On one side such choice augments the number of free parameter vectors v_i, but on the other hand solving $\mathbf{A}^{-1}b$ implies the computation of an expression of the form $\hat{\mathbf{A}}^{-1}\mathbf{U}$ in (2.2), that is of more than one linear system. We restrict ourselves here to the case $m = 1$. The one parameter vector $y \equiv v_1$ gives enough possibilities to construct an auxiliary system (2.4) having better convergence properties than has the original system when GMRES is applied to it. In the remainder of this chapter we always assume $y \in \mathbb{R}^n$ is such that $\det(\mathbf{A} - by^T) \neq 0$.

2.2 Convergence of the updated system

An interesting result in the context of arbitrary convergence speed can be found in a series of papers by Arioli, Greenbaum, Pták and Strakoš ([2], [30] and [31]). From these papers it follows that given a non-increasing positive sequence $f_0 \geq f_1 \geq \ldots \geq f_{n-1} > f_n = 0$ and a right-hand side b with $\|b\| = f_0$, the residual vectors $\hat{r}_k$ at each step of the GMRES method applied to $\hat{\mathbf{A}}x = b$ with zero initial guess satisfy

$$\|\hat{r}_k\| = f_k, \quad 0 \leq k \leq n-1,$$

if and only if $\hat{\mathbf{A}}$ is of the form

$$\hat{\mathbf{A}} = \mathbf{W}\mathbf{R}\tilde{\mathbf{H}}\mathbf{W}^T, \qquad (2.5)$$

where $\mathbf{R}$ is an arbitrary nonsingular upper triangular matrix, $\mathbf{W} = (w_1, \ldots, w_n)$ is an orthonormal matrix such that

$$\mathbf{W}^T b = \begin{pmatrix} \pm\sqrt{f_0^2 - f_1^2} \\ \vdots \\ \pm\sqrt{f_{n-2}^2 - f_{n-1}^2} \\ \pm\sqrt{f_{n-1}^2} \end{pmatrix}, \qquad (2.6)$$

and $\tilde{\mathbf{H}}$ is given by

$$\tilde{\mathbf{H}} = \begin{pmatrix} 0 & \dots & 0 & 1/(b^T w_n) \\ 1 & & 0 & -(b^T w_1)/(b^T w_n) \\ & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & -(b^T w_{n-1})/(b^T w_n) \end{pmatrix}. \tag{2.7}$$

We are particularly interested in the question whether, given the non-increasing positive sequence $f_0 \geq f_1 \geq \dots \geq f_{n-1} > f_n = 0$ with $f_0 = \|b\|$, our rank-one updated matrix $\mathbf{A} - by^T$ belongs to the class of matrices (2.5) for some $y \in \mathbb{R}^n$. The answer is yes. We first give a straightforward proof based on projection properties. After that we propose a feasible procedure to construct the parameter vector $y \in \mathbb{R}^n$ and finally, for the sake of completeness, we address the exact form that $\mathbf{R}$ and $\mathbf{W}$ take in (2.5) in our case of rank-one updating. This yields a second way to construct the parameter vector $y \in \mathbb{R}^n$.

2.2.1 Any convergence curve is possible for $\mathbf{A} - by^T$ with $x_0 = 0$

The convergence speed of a GMRES process is determined by the evolution of the distance from r_0 to the projection spaces. Let us first take a look at the projection spaces generated by the auxiliary system. They have the form

$$\hat{\mathbf{A}}\mathcal{K}_k(\hat{\mathbf{A}}, r_0) = (\mathbf{A} - by^T)\mathcal{K}_k(\mathbf{A} - by^T, r_0).$$

The first residual norm, for example, is the distance from r_0 to $\mathbf{A}r_0 - b(y^T r_0)$. Clearly, we create optimal opportunities to minimize this distance when $r_0 = b$, because all projection spaces have a component in the direction of b. Moreover, when we apply the Arnoldi process to $\mathbf{A} - by^T$ with zero initial guess, multiples of the first Arnoldi vector, $b/\|b\|$, are added to the Arnoldi vectors we would have generated for $\mathbf{A}$. Krylov subspaces therefore remain the same and the influence of y is simple to control. This is demonstrated in the following proposition.

Proposition 2.2.1 *The Arnoldi algorithm applied to the matrix $\hat{\mathbf{A}} := \mathbf{A} - by^T$, $y \in \mathbb{R}^n$, and first Arnoldi vector $v_1 := b/\|b\|$ generates Arnoldi vectors v_k, $k \geq 2$, that are independent from the choice of y. Moreover, if $\tilde{\mathbf{H}}_k$ is the upper Hessenberg matrix of the Arnoldi decomposition (1.12) associated with $\mathbf{A}$ and $\mathbf{V}_k = (v_1, \dots, v_k)$, then*

$$\tilde{\mathbf{H}}_k - \|b\| e_1 (\mathbf{V}_k^T y)^T$$

is the Hessenberg matrix for the Arnoldi decomposition associated with $\hat{\mathbf{A}}$.

Proof: Application of the Arnoldi process to $\mathbf{A}$ with starting vector $v_1 = b/\|b\|$ gives the relation

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\tilde{\mathbf{H}}_k,$$

where $\mathbf{V}_k$ is orthogonal and $\tilde{\mathbf{H}}_k \in \mathbb{R}^{(k+1)\times k}$ is a Hessenberg matrix. Hence

$$\hat{\mathbf{A}}\mathbf{V}_k = (\mathbf{A} - by^T)\mathbf{V}_k =$$

$$\mathbf{V}_{k+1}\tilde{\mathbf{H}}_k - \|b\|\mathbf{V}_{k+1}e_1 y^T \mathbf{V}_k = \mathbf{V}_{k+1}\left(\tilde{\mathbf{H}}_k - \|b\| e_1 (\mathbf{V}_k^T y)^T\right). \tag{2.8}$$

The matrix $\tilde{\mathbf{H}}_k - \|b\| e_1 (\mathbf{V}_k^T y)^T$ differs from $\tilde{\mathbf{H}}_k$ only in its first row. Thus it is upper Hessenberg and (2.8) is the Arnoldi decomposition for $\hat{\mathbf{A}}$ that starts with $v_1 = b/\|b\|$. The involved orthogonal matrix is the same as for the decomposition of $\mathbf{A}$. □

The proposition implies $\mathcal{K}_k(\mathbf{A}, b) \equiv \mathcal{K}_k(\hat{\mathbf{A}}, b)$ and as a consequence one can turn the planes $\hat{\mathbf{A}}\mathcal{K}_k(\hat{\mathbf{A}}, b) \equiv (\mathbf{A} - by^T)\mathcal{K}_k(\mathbf{A}, b)$ by changing y. The distance between $r_0 \equiv b$ and these planes can be made as small as wanted and *any* convergence speed can be forced.

Theorem 2.2.2 *If the GMRES method applied to (1.1) with $x_0 = 0$ terminates at step n, then any convergence curve terminating at step n for the GMRES method applied to (2.4) with $x_0 = 0$ can be forced by the choice of $y \in \mathbb{R}^n$.*

Proof: With the relations of Theorem 1.3.2 we have

$$\|\hat{r}_1\| = \sin\angle(b, \hat{\mathbf{A}}b)\|b\| = \sin\angle(b, \mathbf{A}b - \alpha_0 b)\|b\|,$$

where $\alpha_0 := y^T b$. Hence

$$\|\hat{r}_1\|^2 = \left(1 - \frac{(b^T(\mathbf{A}b - \alpha_0 b))^2}{\|b\|^2\|\mathbf{A}b - \alpha_0 b\|^2}\right)\|b\|^2 =$$

$$\frac{\|b\|^2(\|\mathbf{A}b\|^2 - 2\alpha_0 b^T\mathbf{A}b + \alpha_0^2\|b\|^2) - (b^T\mathbf{A}b - \alpha_0\|b\|^2)^2}{\|\mathbf{A}b\|^2 - 2\alpha_0 b^T\mathbf{A}b + \alpha_0^2\|b\|^2} = \frac{\|b\|^2\|\mathbf{A}b\|^2 - (b^T\mathbf{A}b)^2}{\|\mathbf{A}b\|^2 - 2\alpha_0 b^T\mathbf{A}b + \alpha_0^2\|b\|^2}.$$

By choosing $\alpha_0 = \frac{b^T\mathbf{A}b}{\|b\|^2}$ the first residual norm stays as large as the initial one. On the other hand, large enough $|\alpha_0|$ will force $\|\hat{r}_1\|$ to be as small as wanted.
Now let us assume we have fixed the values $\alpha_0, \ldots, \alpha_{k-1}$, $\alpha_i := y^T\mathbf{A}^i b$. Then the $(k+1)$st residual is the difference between the initial residual and its projection on $\hat{\mathbf{A}}\mathcal{K}_{k+1}(\mathbf{A}, b)$, hence

$$\|\hat{r}_{k+1}\| = \sin\angle(b, \hat{\mathbf{A}}\mathcal{K}_{k+1}(\mathbf{A}, b))\|b\|.$$

The Krylov subspace $\hat{\mathbf{A}}\mathcal{K}_{k+1}(\mathbf{A}, b)$ is given by

$$(\mathbf{A} - by^T)\,\mathrm{span}\{b, \mathbf{A}b, \ldots, \mathbf{A}^k b\} = \mathrm{span}\{\mathbf{A}b - \alpha_0 b, \mathbf{A}^2 b - \alpha_1 b, \ldots, \mathbf{A}^{k+1}b - \alpha_k b\},$$

where $\alpha_k = y^T\mathbf{A}^k b$. The angle $\angle(b, \hat{\mathbf{A}}\mathcal{K}_{k+1}(\mathbf{A}, b))$ is the minimum angle $\angle(b, c)$ over all $c \in \hat{\mathbf{A}}\mathcal{K}_{k+1}(\mathbf{A}, b)$. By defining $\angle(b, \mathbf{A}^{k+1}b - \alpha_k b)$ smaller than $\angle(b, \hat{\mathbf{A}}\mathcal{K}_k(\mathbf{A}, b))$ we can force $\|\hat{r}_{k+1}\|$ to take the wanted value. The angle $\angle(b, \mathbf{A}^{k+1}b - \alpha_k b)$ can be made as small as possible by choice of α_k for the same reason as above for $\angle(b, \mathbf{A}b - \alpha_0 b)$. In total we obtain $n-1$ conditions for $y \in \mathbb{R}^n$, namely $y^T\mathbf{A}^i b = \alpha_i$, $0 \le i \le n-2$, the last residual norm vanishes automatically. At least one y satisfying these conditions exists.
□

In case of applying the FOM method to (2.4) we can prove an analogue result as follows. This proof constructs the wanted $y \in \mathbb{R}^n$ from the iterates of FOM applied to (1.1).

Proposition 2.2.3 *Let $f_0 \ge f_1 \ge f_2 \ldots > f_n = 0$ be a non-increasing sequence of real values and let the system of linear equations*

$$\hat{\mathbf{A}}x = b \tag{2.9}$$

with $\|b\| = f_0$ be given by

$$\hat{\mathbf{A}} := \mathbf{A} - by^T, \quad y \in \mathbb{R}^n,$$

and let us choose the initial approximation $x_0 = 0$. Let for $y = 0$ all iterates x_k, $1 \leq k \leq n$, be defined, be linearly independent and $\|b - \mathbf{A}x_k\| > 0$ for $k < n$. Then there exists at least one $y \in \mathbb{R}^n$ such that the residual vectors $\hat{r}_k$ obtained by application of the FOM method to the system (2.9) satisfy

$$\|\hat{r_k}\| = f_k, \quad 0 \leq k \leq n.$$

Proof: If $x_0 = 0$, FOM iterates of the original system $\mathbf{A}x = b$ are given by $x_k = \|b\|\mathbf{V}_k\mathbf{H}_k^{-1}e_1$, where $\mathbf{H}_k$ is the Hessenberg matrix from (1.12) without its last row (see also (1.22)). If we put $\hat{\mathbf{H}}_k = \mathbf{H}_k - \|b\|e_1y^T\mathbf{V}_k$, we have

$$\mathbf{H}_k u = \|b\|e_1 \Leftrightarrow \hat{\mathbf{H}}_k u = \|b\|(e_1 - e_1y^T\mathbf{V}_k u) \Leftrightarrow \hat{\mathbf{H}}_k \frac{u}{1 - y^T\mathbf{V}_k u} = \|b\|e_1.$$

Hence FOM iterates $\hat{x}_k$ of the second system satisfy

$$\hat{x}_k = \mathbf{V}_k \frac{\mathbf{H}_k^{-1}\|b\|e_1}{1 - y^T\mathbf{V}_k\mathbf{H}_k^{-1}\|b\|e_1} = \frac{1}{1 - y^Tx_k}\|b\|\mathbf{V}_k\mathbf{H}_k^{-1}e_1 = \frac{x_k}{1 - y^Tx_k},$$

having made use of Proposition 2.2.1. If we denote by r_k the residual vectors for the first system, then

$$\hat{r}_k = b - \hat{\mathbf{A}}\hat{x}_k = b - \hat{\mathbf{A}}\frac{x_k}{1 - y^Tx_k} = \frac{1}{1 - y^Tx_k}(b - \mathbf{A}x_k) = \frac{1}{1 - y^Tx_k}r_k,$$

and

$$\|\hat{r}_k\| = \frac{1}{|1 - y^Tx_k|} \cdot \|r_k\|, \qquad y^Tx_k \neq 1.$$

The last equation shows that we can, by choice of the inner product y^Tx_k, size $\|\hat{r}_k\|$. By solving, for example, the underdetermined linear system

$$\begin{pmatrix} x_1^T \\ \vdots \\ x_{n-1}^T \end{pmatrix} y = \begin{pmatrix} 1 - \|r_1\|/f_1 \\ \vdots \\ 1 - \|r_{n-1}\|/f_{n-1} \end{pmatrix}$$

we obtain the wanted $y \in \mathbb{R}^n$. □

2.2.2 Iterate based implementation

Next we present a technique to compute the parameter vector y that forces a given convergence curve of GMRES when the implemented is based on (1.13). Due to the close connection between FOM and GMRES method, one expects a result similar to the last one, with construction of y from iterates and residuals of the first system, to be easily derivable in the GMRES case. For example, the relations of Theorem 1.3.2 could be exploited. Unfortunately, the involved angles are dependent on the choice of the parameter vector y, which makes working with them complicated. Moreover, construction of y from iterates and residuals of the first system is not interesting for practice. One would need to apply a GMRES process to the original system to be able to define the auxiliary system. The angles of Theorem 1.3.2, used in a different way, give the key to successive definition of the auxiliary system *during* the GMRES process, as we will prove now. The proof works with the Givens rotations that are involved in computing the approximations of the GMRES method. The parameters of these rotations have an immediate influence on the convergence speed of the method.

Definition 2.2.4 *A Givens rotation for the ith row, $i > 1$, of a vector is a rotation represented by left multiplication with an orthogonal matrix of the form*

$$\begin{pmatrix} \mathbf{I}_{i-2} & 0 & \dots & 0 \\ 0 & c_{i-1} & s_{i-1} & \vdots \\ \vdots & -s_{i-1} & c_{i-1} & \\ 0 & \dots & 0 & \mathbf{I}_{n-i} \end{pmatrix}.$$

Consequently, $c_{i-1}^2 + s_{i-1}^2 = 1$, and c_{i-1} is called Givens cosine and s_{i-1} Givens sine.

Lemma 2.2.5 *The action of k consecutive Givens rotations, from a rotation for the 2nd row till a rotation for the $(k+1)$st row, on a given vector $g = (g_1, \dots, g_{k+1})^T$ yields a vector $(g'_1, \dots, g'_k, g^*_{k+1})^T$, where*

$$g'_i = s_i g_{i+1} + c_i \sum_{j=1}^{i} c_{j-1} g_j \prod_{l=j}^{i-1} (-s_l), \quad i \le k, \tag{2.10}$$

$$\text{and} \qquad g^*_{k+1} = \sum_{j=1}^{k+1} c_{j-1} g_j \prod_{l=j}^{k} (-s_l),$$

with $c_0 := 1$.

Proof: At first, note that a Givens rotation for the ith row changes only rows $(i-1)$ and i. After application of the Givens rotation for the second row to g, the first element has the form

$$g'_1 := c_1 g_1 + s_1 g_2,$$

and the second element will change to

$$g^*_2 := -s_1 g_1 + c_1 g_2.$$

Now, let us assume that the rotation for the ith row has been executed and that we have

$$g^*_i = \sum_{j=1}^{i} c_{j-1} g_j \prod_{l=j}^{i-1} (-s_l).$$

The rotation for the $(i+1)$st changes g^*_i to

$$c_i g^*_i + s_i g_{i+1} = s_i g_{i+1} + c_i \sum_{j=1}^{i} c_{j-1} g_j \prod_{l=j}^{i-1} (-s_l) = g'_i$$

and g_{i+1} to

$$-s_i g^*_i + c_i g_{i+1} = -s_i \sum_{j=1}^{i} c_{j-1} g_j \prod_{l=j}^{i-1} (-s_l) + c_i g_{i+1} = \sum_{j=1}^{i+1} c_{j-1} g_j \prod_{l=j}^{i} (-s_l) = g^*_{i+1}.$$

□

Givens rotations for the ith row can be used to zero out the ith element of a vector $v = (v_1, \dots, v_n)^T$, $v_i \neq 0$. To do so, we will in the present work use the choices

$$c_{i-1} := \frac{v_{i-1}}{\sqrt{v_{i-1}^2 + v_i^2}}, \quad s_{i-1} := \frac{v_i}{\sqrt{v_{i-1}^2 + v_i^2}}. \tag{2.11}$$

The Givens cosines and sines involved in the computation of a GMRES approximation can be expressed in values that were calculated during the Arnoldi orthogonalization process and by previously executed Givens rotations:

Corollary 2.2.6 *After the kth step of the GMRES method applied to a linear system (1.1), let the Arnoldi process have generated an orthonormal basis $v_1, \ldots, v_k$ of the associated k-dimensional Krylov subspace and an upper Hessenberg matrix $\tilde{\mathbf{H}}_k \in \mathbb{R}^{(k+1)\times k}$ with coefficients*

$$h_{j,k} := v_j^T \mathbf{A} v_k, \quad 1 \le j \le k, \quad h_{k+1,k} := \|\mathbf{A} v_k - \sum_{j=1}^{k} h_{j,k} v_j\|.$$

If this Hessenberg matrix is brought to upper triangular form with the help of Givens rotations and the rotation for the $(i+1)$st row is given by c_i and s_i, $i \le k-1$, then the rotation for the $(k+1)$st row has Givens sine s_k satisfying

$$s_k^2 = \frac{h_{k+1,k}^2}{h_{k+1,k}^2 + (\sum_{j=1}^{k} c_{j-1} h_{j,k} \prod_{i=j}^{k-1}(-s_i))^2},$$

with $c_0 := 1$.

Proof: The Hessenberg matrix $\tilde{\mathbf{H}}_k \in \mathbb{R}^{(k+1)\times k}$ can be brought to upper triangular form by means of Givens rotations that consecutively zero out all lower subdiagonal elements. According to Lemma 2.2.5, after $(k-1)$ rotations the kth element of the last column has the form

$$h_{k,k}^* := \sum_{j=1}^{k} c_{j-1} h_{j,k} \prod_{i=j}^{k-1}(-s_i).$$

To zero out the last entry $h_{k+1,k}$ of this column we define a Givens rotation for the $(k+1)$st row satisfying with (2.11)

$$s_k^2 := \frac{h_{k+1,k}^2}{h_{k+1,k}^2 + (h_{k,k}^*)^2}.$$

□

By introducing the parameter-dependent matrix $\mathbf{A} - by^T$, $y \in \mathbb{R}^n$, we create the opportunity to modify the values $h_{i,j}$ and hence the sines s_k according to our own needs. Proposition 2.2.1 stated that only the first row of the Hessenberg matrix involved in the GMRES process is dependent from the parameter vector y. Its influence on the Givens sines is given by Corollary 2.2.6. We thus have to our disposal an easy to handle tool for manipulating the associated Givens rotations. It is a well known fact, that in the GMRES method the residual norm of an iteration can be expressed as the product of all previously executed Givens rotation sines and the initial residual norm. In fact, Givens sines coincide with the sines of the angles between initial residuals and projection spaces from Theorem 1.3.2. For our parameter-dependent matrix, these sines appear to be sensible to changes of the parameter. The next theorem states exactly the same as Theorem 2.2.2, but it explicitly demonstrates how to construct, during the GMRES computations, the wanted parameter vector y.

Theorem 2.2.7 *Let $f_0 \geq f_1 \geq f_2 \ldots > f_n = 0$ be a non-increasing sequence of real values and let the system of linear equations*

$$\hat{\mathbf{A}}x = b \tag{2.12}$$

with $\|b\| = f_0$ be given by

$$\hat{\mathbf{A}} := \mathbf{A} - by^T, \quad y \in \mathbb{R}^n,$$

and let us choose the initial approximation $x_0 = 0$. If the Arnoldi algorithm applied to $\mathbf{A}$ does not break down before the nth step, then there exists at least one $y \in \mathbb{R}^n$ such, that the residual vectors $\hat{r}_k$ obtained by application of the GMRES method to the system (2.12) satisfy

$$\|\hat{r_k}\| = f_k, \quad 0 \leq k \leq n.$$

Proof: In the GMRES method we have the following recurrence formula for the residual norms (see for example Saad [59], p. 167):

$$\|\hat{r}_k\| = |\hat{s}_k \cdot \ldots \cdot \hat{s}_1|\beta,$$

where $\beta := \|\hat{r}_0\| = \|b\|$. As a consequence,

$$\frac{\|\hat{r}_k\|}{\|\hat{r}_{k-1}\|} = |\hat{s}_k|.$$

The theorem is proved if we show that we can find a vector $y \in \mathbb{R}^n$ such that all values $|\hat{s}_k|$ satisfy

$$|\hat{s}_k| = \frac{f_k}{f_{k-1}}, \quad 1 \leq k \leq n-1.$$

The last residuum vanishes automatically.
For $k = 1$, the Arnoldi process yields values $\hat{h}_{1,1}$ and $\hat{h}_{2,1}$ and the Givens sine that zeroes out (together with the Givens cosine $\hat{c}_1$) $\hat{h}_{2,1}$ satisfies

$$\hat{s}_1^2 = \frac{\hat{h}_{2,1}^2}{\hat{h}_{2,1}^2 + \hat{h}_{1,1}^2}.$$

Let us denote by α_1 the value $y^T v_1$. Note that $\hat{h}_{2,1}$ is independent from α_1 because of Proposition 2.2.1. As for $\hat{h}_{1,1}$, we have

$$\hat{h}_{1,1}^2 = (v_1^T(\mathbf{A} - by^T)v_1)^2 = (v_1^T\mathbf{A}v_1 - \beta\alpha_1)^2.$$

Hence $\hat{h}_{1,1}^2$ can have whatever nonnegative value if we choose α_1 accordingly. In other words, $|\hat{s}_1|$ can assume, in dependency from α_1, every positive value smaller than or equal to 1. In particular it can take the wanted value f_1/f_0 if we solve

$$(\frac{f_1}{f_0})^2 = \frac{\hat{h}_{2,1}^2}{\hat{h}_{2,1}^2 + (v_1^T\mathbf{A}v_1 - \beta\alpha_1)^2},$$

that is

$$\alpha_1 = \frac{\pm\sqrt{\frac{1-(f_1/f_0)^2}{(f_1/f_0)^2}}\hat{h}_{2,1} - v_1^T\mathbf{A}v_1}{-\beta}.$$

This puts a first condition on the vector y, namely it fixes for the chosen α_1 the value of $y^T v_1$.

Now, let k be greater than 1 and let us assume we have chosen all values $\alpha_i := y^T v_i$ such, that $|\hat{s}_i| = f_i/f_{i-1}$ for $i < k$. According to Corollary 2.2.6,

$$\hat{s}_k^2 = \frac{\hat{h}_{k+1,k}^2}{\hat{h}_{k+1,k}^2 + (\sum_{j=1}^k \hat{c}_{j-1}\hat{h}_{j,k}\prod_{i=j}^{k-1}(-\hat{s}_i))^2}.$$

In this expression, only $\hat{h}_{1,k} = v_1^T\mathbf{A}v_1 - \beta\alpha_k = h_{1,k} - \beta\alpha_k$ is dependent from the choice of $\alpha_k := y^T v_k$ because of Proposition 2.2.1, hence $\hat{h}_{i,k} = h_{i,k} = v_i^T\mathbf{A}v_k$, $i \geq 2$. The second term of the denominator, including $\hat{h}_{1,k}$, can be written as

$$\left(\sum_{j=1}^{k} \hat{c}_{j-1}\hat{h}_{j,k}\prod_{i=j}^{k-1}(-\hat{s}_i)\right)^2 = \left(\hat{h}_{1,k}\prod_{i=1}^{k-1}(-\hat{s}_i) + \sum_{j=2}^{k}\hat{c}_{j-1}h_{j,k}\prod_{i=j}^{k-1}(-\hat{s}_i)\right)^2 = \tag{2.13}$$

$$\left((-\beta\alpha_k + h_{1,k})\prod_{i=1}^{k-1}(-\hat{s}_i) + \sum_{j=2}^{k}\hat{c}_{j-1}h_{j,k}\prod_{i=j}^{k-1}(-\hat{s}_i)\right)^2. \tag{2.14}$$

Again, this is a nonnegative expression that is dependent from α_k and it can possibly assume the value 0. Therefore, an appropriate choice of α_k yields the desired equality $|\hat{s}_k| = f_k/f_{k-1}$, $0 < \frac{f_k}{f_{k-1}} \leq 1$. Possible choices are given by

$$\alpha_k = \frac{\pm\sqrt{\frac{1-(f_k/f_{k-1})^2}{(f_k/f_{k-1})^2}}h_{k+1,k} - \sum_{j=1}^k \hat{c}_{j-1}h_{j,k}\prod_{i=j}^{k-1}(-\hat{s}_i)}{-\beta\prod_{i=1}^{k-1}(-\hat{s}_i)}. \tag{2.15}$$

After $n-1$ steps we have defined all Arnoldi vectors $v_1, \ldots, v_n$ and have been putting $n-1$ conditions on the vector y:

$$y^T(v_1, \ldots, v_{n-1}) = (\alpha_1, \ldots, \alpha_{n-1}). \tag{2.16}$$

There exists at least one $y \in \mathbb{R}^n$ solving this underdetermined linear system. □

It it this proof that we have used to implement accelerations of restarted GMRES exploiting an auxiliary system with arbitrary convergence speed, see Algorithm 5.2.1. In Algorithm 5.2.1 we always used the positive root of (2.15).

2.2.3 Residual based implementation

As we have seen, it is possible to find a vector $y \in \mathbb{R}^n$ such that $\hat{\mathbf{A}} = \mathbf{A} - by^T$ belongs to the class of matrices given by (2.5). In other words, $\hat{\mathbf{A}}$ can assume the form $\hat{\mathbf{A}} = \mathbf{W}\mathbf{R}\tilde{\mathbf{H}}\mathbf{W}^T$ for some upper triangular matrix $\mathbf{R}$ and with $\mathbf{W}$ and $\tilde{\mathbf{H}}$ satisfying the equations (2.6) and (2.7). To illustrate this, we will describe the exact form of $\mathbf{R}$ and $\mathbf{W}$ when $\hat{\mathbf{A}} = \mathbf{A} - by^T$. This yields an alternative to the preceding section for constructing $y \in \mathbb{R}^n$ during the GMRES process. It is connected with GMRES implementations that are based on decomposition (1.17), as for example Walker [74] proposes, and, besides being from theoretical interest, is useful if we want to apply Theorem 2.2.7 to the latter, residual based implementation strategy. For this reason we explicitly display in the following proof the conditions on $y \in \mathbb{R}^n$ necessary to force a given convergence curve.

First, we introduce the notations

$$g(i) := \sqrt{f_{i-1}^2 - f_i^2}, \qquad 1 \leq i \leq n,$$

and it will be useful to have the following small lemma.

Lemma 2.2.8 *The subdiagonal elements* $h_{k+1,k} \equiv \|\mathbf{A}v_k - \sum_{j=1}^{k} h_{j,k}v_j\|$, $k \geq 1$, *with* $h_{j,k} \equiv v_j^T \mathbf{A}v_k$, $j \leq k$, *of Hessenberg matrices involved in the Arnoldi process satisfy the equation*

$$h_{k+1,k}^2 = \|\mathbf{A}v_k\|^2 - \sum_{j=1}^{k} h_{j,k}^2.$$

Proof: We have

$$h_{k+1,k}^2 = \|\mathbf{A}v_k - \sum_{j=1}^{k} h_{j,k}v_j\|^2 = \|\mathbf{A}v_k\|^2 - 2(\mathbf{A}v_k, \sum_{j=1}^{k} h_{j,k}v_j) + \|\sum_{j=1}^{k} h_{j,k}v_j\|^2 =$$

$$\|\mathbf{A}v_k\|^2 - 2\sum_{j=1}^{k} h_{j,k}^2 + \sum_{j=1}^{k} h_{j,k}^2 = \|\mathbf{A}v_k\|^2 - \sum_{j=1}^{k} h_{j,k}^2,$$

because of the orthonormality of the Arnoldi vectors v_i, $i \geq 1$. □

Proposition 2.2.9 *Let* $f_0 \geq f_1 \geq f_2 \ldots > f_n = 0$ *be a non-increasing sequence of real values. If the Arnoldi algorithm applied to* $\mathbf{A}$ *with* $x_0 = 0$ *does not break down before the nth step, then there exists at least one* $y \in \mathbb{R}^n$ *such that (2.5) holds for* $\hat{\mathbf{A}} = \mathbf{A} - by^T$ *with* $\mathbf{W}$ *satisfying (2.6) and* $\tilde{\mathbf{H}}$ *satisfying (2.7).*

Proof: We will search for an appropriate $y \in \mathbb{R}^n$ by successively imposing conditions on the vector y during the computation of a suitable orthonormal matrix $\mathbf{W}$. This matrix will be computed by applying the Arnoldi process to $\hat{\mathbf{A}}$ with $w_1 := \hat{\mathbf{A}}b/\|\hat{\mathbf{A}}b\|$. The process cannot break down because by assumption $\dim \mathcal{K}_i(\mathbf{A}, b) = i$ for all $i \leq n$ and also $\dim(\text{span}\{b, \mathbf{A}b + \gamma_1 b, \ldots, \mathbf{A}^{i-1}b + \gamma_{i-1}b\}) = i$ for arbitrary real scalars γ_j. In particular,

$$\dim \hat{\mathbf{A}}\mathcal{K}_{i-1}(\hat{\mathbf{A}}, b) = \dim \hat{\mathbf{A}}\mathcal{K}_{i-1}(\mathbf{A}, b) =$$

$$\dim(\text{span}\{\mathbf{A}b + (y^T b)b, \ldots, \mathbf{A}^{i-1}b + (y^T\mathbf{A}^{i-1}b)b\}) = i - 1.$$

Hence, for all possible choices of y the Arnoldi vectors w_i, $i \leq n$, are non-vanishing.

Let the first condition on y be that the Arnoldi vector w_1 satisfies $b^T w_1 = g(1)$. This condition can be fulfilled by fixing the value $y^T b$, which we denote by α_0. We have

$$b^T w_1 = \frac{b^T(\mathbf{A} - by^T)b}{\|\mathbf{A}b - \alpha_0 b\|} = \frac{b^T\mathbf{A}b - \alpha_0\|b\|^2}{\sqrt{\|\mathbf{A}b\|^2 - 2\alpha_0 b^T\mathbf{A}b + \alpha_0^2\|b\|^2}}. \qquad (2.17)$$

The right-hand side of this equation is well defined because w_1 never vanishes, it is continuously dependent from α_0 and takes values between $-\|b\|$ and $\|b\|$, among others it can take the value $g(1) < \|b\|$ if we choose α_0 accordingly. Squaring both sides of the above equation yields the roots

$$\alpha_{0\pm} = \frac{-b^T\mathbf{A}b(\|b\|^2 - g(1)^2)}{-\|b\|^2(\|b\|^2 - g(1)^2)} \quad \pm$$

$$\frac{\sqrt{(b^T\mathbf{A}b(\|b\|^2 - g(1)^2))^2 + \|b\|^2(\|b\|^2 - g(1)^2)(g(1)^2\|\mathbf{A}b\|^2 - (b^T\mathbf{A}b)^2)}}{-\|b\|^2(\|b\|^2 - g(1)^2)}.$$

Because the term $\|b\|^2 - g(1)^2 = f_1^2$ is positive, one sees that only the smallest of the two roots (α_{0+}) can give a positive right side in (2.17). The value $b^T w_1$ is positive by definition, thus the smaller root is the unique value for α_0 that solves (2.17).

Now let us assume we have defined orthonormal $w_1, \ldots, w_i$ for some i and, by the choices of the values $\alpha_{j-1} := y^T w_{j-1}$, forced these w_j to satisfy $b^T w_j = g(j)$ for $2 \leq j \leq i$. The next step of the Arnoldi orthogonalization process, that is orthogonalization of $\hat{\mathbf{A}} w_i$ against $w_1, \ldots, w_i$, yields a vector

$$\tilde{w}_{i+1} = \hat{\mathbf{A}} w_i - \sum_{j=1}^{i} (w_j^T \hat{\mathbf{A}} w_i) w_j.$$

The dependency of this vector from y is restricted to the value $y^T w_i$ which we will denote by α_i. The Arnoldi vector w_{i+1} results from normalizing $\tilde{w}_{i+1}$ and it must satisfy

$$b^T w_{i+1} = g(i+1)$$

because of (2.6). The inner product $b^T w_{i+1}$ equals

$$\frac{b^T(\hat{\mathbf{A}} w_i - \sum_{j=1}^{i} (w_j^T \hat{\mathbf{A}} w_i) w_j)}{\|\tilde{w}_{i+1}\|} = \frac{b^T \hat{\mathbf{A}} w_i - \sum_{j=1}^{i} (w_j^T \hat{\mathbf{A}} w_i) g(j)}{\sqrt{\|\hat{\mathbf{A}} w_i\|^2 - \sum_{j=1}^{i} (w_j^T \hat{\mathbf{A}} w_i)^2}} =$$

$$\frac{b^T \mathbf{A} w_i - \alpha_i \|b\|^2 - \sum_{j=1}^{i} g(j)(w_j^T \mathbf{A} w_i - g(j)\alpha_i)}{\sqrt{\|\mathbf{A} w_i\|^2 + \|b\|^2 \alpha_i^2 - 2\alpha_i b^T \mathbf{A} w_i - \sum_{j=1}^{i} (w_j^T \mathbf{A} w_i - g(j)\alpha_i)^2}} =$$

$$\frac{b^T \mathbf{A} w_i - \sum_{j=1}^{i} (w_j^T \mathbf{A} w_i) g(j) - \alpha_i f_i^2}{\sqrt{\|\mathbf{A} w_i\|^2 - \sum_{j=1}^{i} (w_j^T \mathbf{A} w_i)^2 + \alpha_i^2 f_i^2 - 2\alpha_i (b^T \mathbf{A} w_i - \sum_{j=1}^{i} g(j) w_j^T \mathbf{A} w_i)}},$$

where we have made use of Lemma 2.2.8. This is an expression that is continuously dependent from α_i, it is defined on the entire real axis because $\|\tilde{w}_{i+1}\|$ never vanishes, for $\alpha_i \to -\infty$ it tends to f_i and for $\alpha_i \to \infty$ to $-f_i$. Thus, due to

$$f_i > \sqrt{f_i^2 - f_{i+1}^2} = g(i+1),$$

the value $g(i+1)$ is assumed for some $\alpha_i \neq \infty$. More precisely, we can apply the same computation as for the first step and with the abbreviation $\gamma = b^T \mathbf{A} w_i - \sum_{j=1}^{i} g(j)(w_j^T \mathbf{A} w_i)$, we obtain the following value for α_i :

$$\alpha_i = \frac{-\gamma(f_i^2 - g(i+1)^2)}{-f_i^2(f_i^2 - g(i+1)^2)} \quad +$$

$$\frac{\sqrt{\gamma^2(f_i^2 - g(i+1)^2) + f_i^2(f_i^2 - g(i+1)^2)(g(i+1)^2(\|\mathbf{A} w_i\|^2 - \sum_{j=1}^{i} (w_j^T \mathbf{A} w_i)^2) - \gamma^2)}}{-f_i^2(f_i^2 - g(i+1)^2)}.$$

For reasons analogue to the case above, this solution is unique.
After $(n-2)$ Arnoldi steps we have defined $w_1, \ldots, w_{n-1}$ and we have obtained $n-1$ conditions for y, namely $y^T w_i = \alpha_i$ for $i \leq n-2$ and the condition $y^T b = \alpha_0$. Because of $f_0 \geq f_1 \geq f_2 \ldots > f_n = 0$ the sequence $\{b, w_1, \ldots, w_{n-2}\}$ is linearly independent and thus at least one $y \in \mathbb{R}^n$ satisfying these conditions exists. For the same reason, $\{b, w_1, \ldots, w_{n-1}\}$ is a basis of $\mathbb{R}^n$. Let us put

$$w_n = \frac{\mathbf{A} w_{n-1} - \sum_{j=1}^{n-1} (w_j^T \mathbf{A} w_{n-1}) w_j}{\|\mathbf{A} w_{n-1} - \sum_{j=1}^{n-1} (w_j^T \mathbf{A} w_{n-1}) w_j\|}.$$

Then w_n is one of the two unit vectors that are orthogonal to all $w_1, \ldots, w_{n-1}$. In the basis $\{b, w_1, \ldots, w_{n-1}\}$ it has the form

$$w_n = \frac{\pm 1}{\|b - \sum_{i=1}^{n-1} (b^T w_i) w_i\|} (b - \sum_{i=1}^{n-1} (b^T w_i) w_i)$$

and

$$b^T w_n = \pm \frac{b^T b - \sum_{i=1}^{n-1} (b^T w_i)^2}{\|b - \sum_{i=1}^{n-1} (b^T w_i) w_i\|} = \pm \frac{b^T b - \sum_{i=1}^{n-1} g(i)^2}{\sqrt{b^T b - \sum_{i=1}^{n-1} (b^T w_i)^2}} = \pm f_{n-1}.$$

If $b^T w_n$ happens to be negative, we can change the sign of w_n without loss of generality. With $\mathbf{W}$ being the orthonormal matrix whose columns consist of $w_1, \ldots, w_n$, we thus can define an y so that (2.6) is satisfied. To complete the proof it remains to choose elements of $\mathbf{R}$ for whom (2.5) holds. To achieve this, define the upper triangular elements of $\mathbf{R}$ as follows

$$r_{1,1} := \|\hat{\mathbf{A}} b\|, \qquad r_{i,j} := w_i^T \hat{\mathbf{A}} w_{j-1}, \quad i < j, \qquad r_{j,j} := \|\tilde{w}_j\|, \quad j \geq 2.$$

From the Arnoldi orthonormalization procedure that we have applied, we obtain

$$\hat{\mathbf{A}}(w_1, \ldots, w_{n-1}) = \mathbf{W} \begin{pmatrix} r_{1,2} & \cdots & r_{1,n} \\ r_{2,2} & & r_{2,n} \\ & \ddots & \vdots \\ 0 & \cdots & r_{n,n} \end{pmatrix} = \mathbf{WR} \begin{pmatrix} 0 & \cdots & 0 \\ 1 & & 0 \\ & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}. \tag{2.18}$$

The missing product in (2.18), $\hat{\mathbf{A}} w_n$, equals

$$\hat{\mathbf{A}} w_n = \hat{\mathbf{A}} \frac{1}{b^T w_n} (b - \sum_{i=1}^{n-1} (b^T w_i) w_i) = \frac{1}{b^T w_n} (\hat{\mathbf{A}} b - \sum_{i=1}^{n-1} (b^T w_i) \hat{\mathbf{A}} w_i) =$$
$$\frac{1}{b^T w_n} \left(\hat{\mathbf{A}} b - \sum_{i=1}^{n-1} b^T w_i \left(\|\tilde{w}_{i+1}\| w_{i+1} + \sum_{j=1}^{i} (w_j^T \hat{\mathbf{A}} w_i) w_j \right) \right).$$

This vector is equal to the last column of $\mathbf{W}\mathbf{R}\tilde{\mathbf{H}}$, where $\tilde{\mathbf{H}}$ is given by (2.7):

$$\mathbf{W}\mathbf{R}\tilde{\mathbf{H}} e_n = \frac{1}{b^T w_n} \mathbf{WR} \begin{pmatrix} 1 \\ -b^T w_1 \\ \vdots \\ -b^T w_{n-1} \end{pmatrix} = \frac{1}{b^T w_n} \left(\|\hat{\mathbf{A}} b\| - \sum_{i=1}^{n-1} (b^T w_i)(w_1^T \hat{\mathbf{A}} w_i) \right) w_1 +$$

$$\frac{1}{b^T w_n} \left(-(b^T w_1) \|\tilde{w}_2\| - \sum_{i=2}^{n-1} (b^T w_i)(w_2^T \hat{\mathbf{A}} w_i) \right) w_2 + \ldots + \frac{1}{b^T w_n} \left(-(b^T w_{n-1}) \|\tilde{w}_n\| \right) w_n.$$

Hence, (2.18) augmented with the vector $\hat{\mathbf{A}} w_n$ is easily transformed to equation (2.5). □

2.3 The backtransformed approximation

We now turn our attention to options of substituting the solution of the auxiliary system in the Sherman-Morrison formula (2.3) to obtain the solution of the original system. In case the solution of the second system is exact, Sherman-Morrison also yields the exact solution of the first system. A first way to improve the GMRES method with the help of the Sherman-Morrison formula that comes to mind, is to try to define an auxiliary system that finds its exact solution earlier than the original system does. This, unfortunately, is not possible. As soon as the second system has found the exact solution, the corresponding Krylov subspace has maximal dimension. But at the same iteration number the Krylov subspace of the original system would have reached maximal dimension, because the Krylov subspaces of both systems are identical (see Proposition 2.2.1).

As a second option, one could substitute only approximate solutions of the second system in the Sherman-Morrison formula. As we were able in the previous section to construct auxiliary systems with arbitrary convergence speed, finding a ,,good" approximate solution $\hat{x}_k$, $k < n$, for such system should not be too difficult. Having done so, we can back-transform with the Sherman-Morrison formula as follows:

$$\begin{aligned}\mathbf{A}^{-1}b = (\hat{\mathbf{A}} + by^T)^{-1}b = \hat{\mathbf{A}}^{-1}b - \hat{\mathbf{A}}^{-1}b(1 + y^T\hat{\mathbf{A}}^{-1}b)^{-1}y^T\hat{\mathbf{A}}^{-1}b \approx \\ \hat{x}_k - (1 + y^T\hat{x}_k)^{-1}\hat{x}_k y^T\hat{x}_k = \left(1 - \frac{y^T\hat{x}_k}{1 + y^T\hat{x}_k}\right)\hat{x}_k = \frac{\hat{x}_k}{1 + y^T\hat{x}_k}.\end{aligned}$$

As far as $y^T\hat{x}_k \neq -1$, we can use

$$\bar{x}_k \equiv \frac{\hat{x}_k}{1 + y^T\hat{x}_k} \tag{2.19}$$

as an approximation for the original system. Due to the fact that $\bar{x}_k$ is only a scalar multiple of $\hat{x}_k$, we have $\bar{x}_k \in \mathcal{K}_k(\hat{\mathbf{A}}, b)$. But again the equality $\mathcal{K}_k(\hat{\mathbf{A}}, b) = \mathcal{K}_k(\mathbf{A}, b)$ prevents any improvement with regards to the classical GMRES method, because the classical GMRES iterate x_k for the original system already minimizes

$$\|b - \mathbf{A}x\| \quad \text{over all} \quad x \in \mathcal{K}_k(\mathbf{A}, b) = \mathcal{K}_k(\hat{\mathbf{A}}, b).$$

Summarizing, application of the Sherman-Morrison formula to the full GMRES method in the above proposed manner will not improve its convergence and this is essentially due to the identity of the involved Krylov subspaces. But initially we were interested in overcoming stagnation of the *restarted* GMRES method. When we use the restarted version, initial guesses at the beginning of every restart are nonzero and therefore Proposition 2.2.1 does not hold anymore and one can expect some improvement. In fact, we could use Theorem 2.2.7 to construct an auxiliary system whose say k first iterations do not stagnate and apply GMRES(m) for $m \geq k$, to that specific auxiliary system. The philosophy behind this way of doing is that non-stagnation of the k very first steps might cause non-stagnation during the k first steps of restarts that follow too. Having found an approximation $\hat{x}_k$ of the second system, back-transformation with Sherman-Morrison according to (2.19) yields an approximation to the original system that is perhaps not as accurate as $\hat{x}_k$ is for the auxiliary system, but it is not unreasonable to expect $\bar{x}_k$ to be a better approximation than the approximations of the original, possibly stagnating GMRES(m). An algorithm based on this idea is Algorithm 5.2.1. Let us demonstrate the procedure with some examples.

Example 1. PDE stiffness matrix of dimension 400.

We consider a linear system that arises from the discretization of the differential equation

$$-e^{-xy} \triangle u + (10 + ye^{-xy})u_x + (10 + xe^{-xy})u_y - 60u = 1 \tag{2.20}$$

on the unit square with Dirichlet boundary condition $u = 0$ on $\partial([0, 1])^2$. Finite difference approximation on a 20×20 grid yields the stiffness matrix $\mathbf{A} \in \mathbb{R}^{400\times400}$

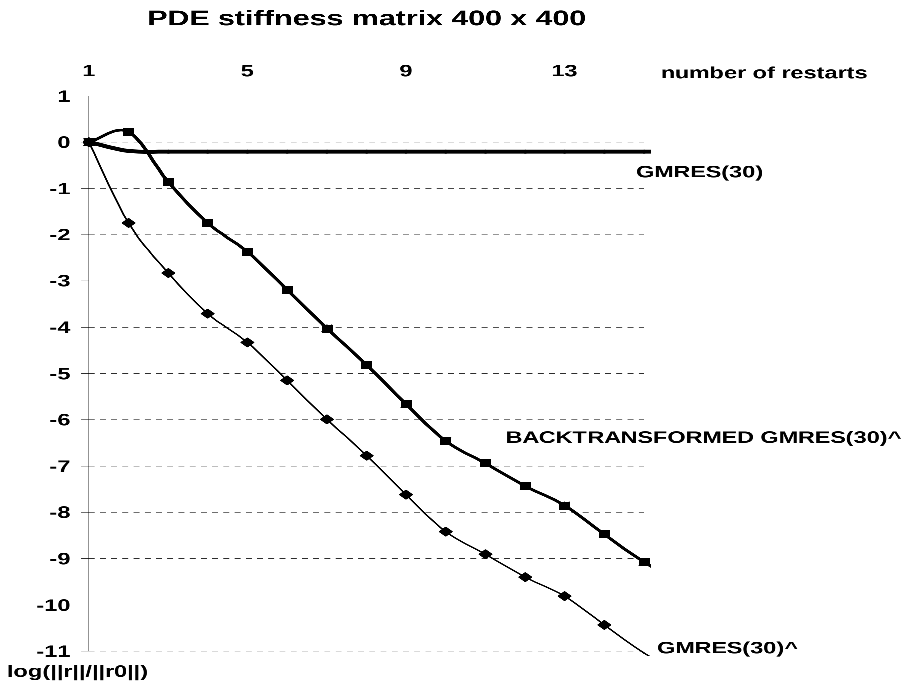

Figure 2.1: Auxiliary system and back-transformed residuals

and our right-hand side b has only elements which have the value 0.5 . Thus $||b|| = 10$ and we choose $x_0 = 0$.

When we apply the GMRES method restarted after 30 steps to this system, the process stagnates. With Theorem 2.2.7 one can construct an auxiliary system with non-stagnating initial iterations. For example, let us ask the first 10 residual norms to fulfill

$$\|\hat{r}_0\| = 10, \quad \|\hat{r}_1\| = 9, \quad \|\hat{r}_2\| = 8, \quad \|\hat{r}_3\| = 7, \quad \|\hat{r}_4\| = 6, \quad \|\hat{r}_5\| = 5,$$

$$\|\hat{r}_6\| = 4, \quad \|\hat{r}_7\| = 3, \quad \|\hat{r}_8\| = 2, \quad \|\hat{r}_9\| = 1.5, \quad \|\hat{r}_{10}\| = 1.$$

Having defined a vector y forcing such residual norms, we can apply GMRES(30) to the resulting system with the same right-hand side and matrix $\mathbf{A} - by^T$. This process does not stagnate anymore. Moreover, when we use the approximations of this system to back-transform according to (2.19), the approximations to the original system do not stagnate either. This is shown in Figure 2.1. Graph GMRES(30) displays restarted GMRES(30) applied to the first system and graph GMRES(30)^ concerns the auxiliary system.

Example 2. PDE stiffness matrix of dimension 102400.

Similar behaviour is observed when we proceed to larger dimensions. When we discretize differential equation (2.20) on a 320×320 grid, we obtain a stiffness matrix of dimension 102400×102400. It has 510720 nonzero elements and the right-hand side belonging to this problem is $b = (9.7 \cdot 10^{-6}, \ldots, 9.7 \cdot 10^{-6})^T$ with $\|b\| = 0.003104$. We choose, in order to apply Theorem 2.2.7, the initial guess zero. The system is so large that even with restart parameter 50, GMRES does not converge at all. A relatively stable projection method with short recurrences, the QMR method, meets with similar problems.

When we apply Algorithm 5.2.1 restarted after 50 iterations and with an auxiliary system with 4 prescribed residual norms, $\|\hat{r}_1\| := 0.003$, $\|\hat{r}_2\| := 0.002$,

$\|\hat{r}_3\| := 0.001$ and $\|\hat{r}_4\| := 0.0005$, then the process converges (both auxiliary and original system after back-transformation). During initial restart cycles residual norms temporarily increase due to denominators $1 + y^T \hat{x}_{50}$ in (2.19) being comparatively small.

We address this problem in the next section. The resulting curve (after back-transformation) is SHERMOR(50,4) in Figure 2.2.

Example 3. Convection-diffusion matrix of dimension 4720.

This is an example that is hard to accelerate for preconditioning techniques as well as with our strategy. We consider the convection-diffusion equation

$$-\varepsilon\triangle u + \mathbf{b}\cdot\nabla u + cu = f \qquad \text{in} \quad \Omega, \qquad u = u_b \quad \text{on} \quad \partial\Omega,$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain with a polygonal boundary $\partial\Omega$, $\varepsilon \in (0,1)$ is constant, $\mathbf{b} \in W^{1,\infty}(\Omega)^2$, $c \in L^\infty(\Omega)$, $f \in L^2(\Omega)$, and $u_b \in H^{3/2}(\partial\Omega)$. With $\varepsilon = 10^{-8}$ we create a very convection dominated problem, for details see Knobloch, Tobiska [43]. In this paper a streamlined diffusion term with control parameter δ_K is introduced for discretization purposes. It is shown that the resulting discrete problem has a unique solution when δ_K is chosen small enough. We wittingly took the large control parameter $\delta_K = 50$ to build a corresponding stiffness matrix with bad convergence properties. On a grid with 21 inner points we obtained a matrix of dimension 4720 with 23284 nonzero elements. In addition, we applied restarted GMRES to the linear system with the small restart parameter 16. Figure 2.3 illustrates the troubles projection methods have with this kind of problem. GMRES(16) converges very slowly, it needs 30000 matrix vector products to reduce the initial residual norm with a factor 10^{-5}. The QMR method makes a promising start but stagnates afterwards. Especially uncontrolled is the behaviour of the BCG method. This example demonstrates very well the smoothing influence QMR has on its oblique parallel.

Concerning our acceleration technique, we could not achieve any improvement by forcing arbitrarily chosen non-stagnating residual norms for the auxiliary system. It was necessarily to select the prescribed values very carefully. We applied the following strategy: We used the first 5 residual norms that classical GMRES(16) generates and modified them very slightly in order to stimulate convergence but to avoid too large norm decreasing. In this case too large norm decreasing occurs very easily. With the prescribed values

$$\|\hat{r}_1\| = 3.552793222,\ \|\hat{r}_2\| = 3.323801355,\ \|\hat{r}_3\| = 3.106983257,\ \|\hat{r}_4\| = 2.91$$

and $\|\hat{r}_5\| = 2.4497933359$, we obtained the curve SHERMOR(16,5), which reaches a residual norm reduction of 10^{-5} after about 3 times less iterations than GMRES(16).

In the sample experiment of Chapter 5 a last application of this strategy is presented.

2.4 The gap between original and auxiliary system

In the preceding section we have proposed a strategy to exploit fast converging auxiliary systems to accelerate original systems. The philosophy consisted of forcing non-stagnation for one system and expecting that this non-stagnation is transferred

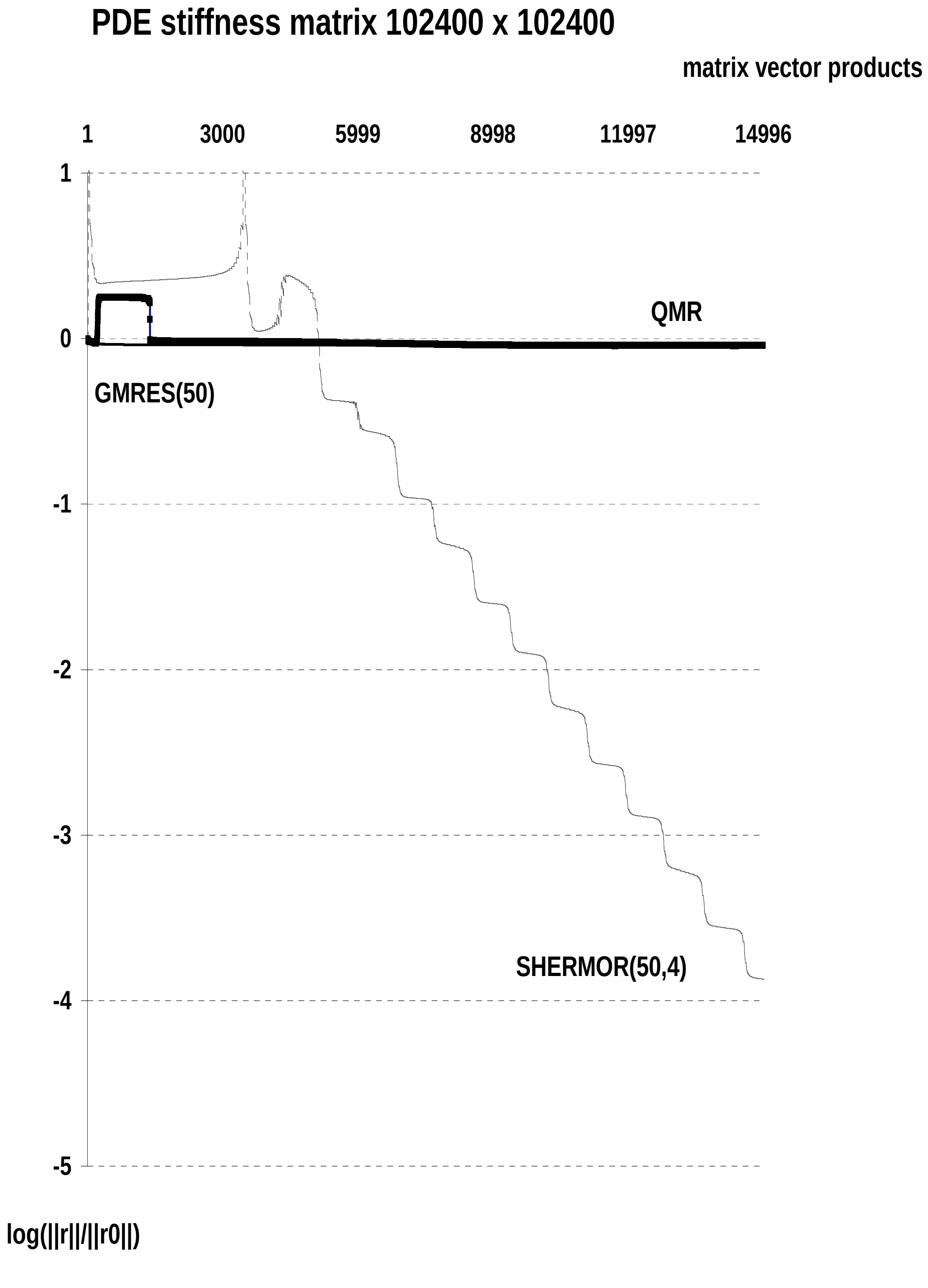

Figure 2.2: QMR, GMRES(50) and GMRES(50) with prescribed auxiliary system

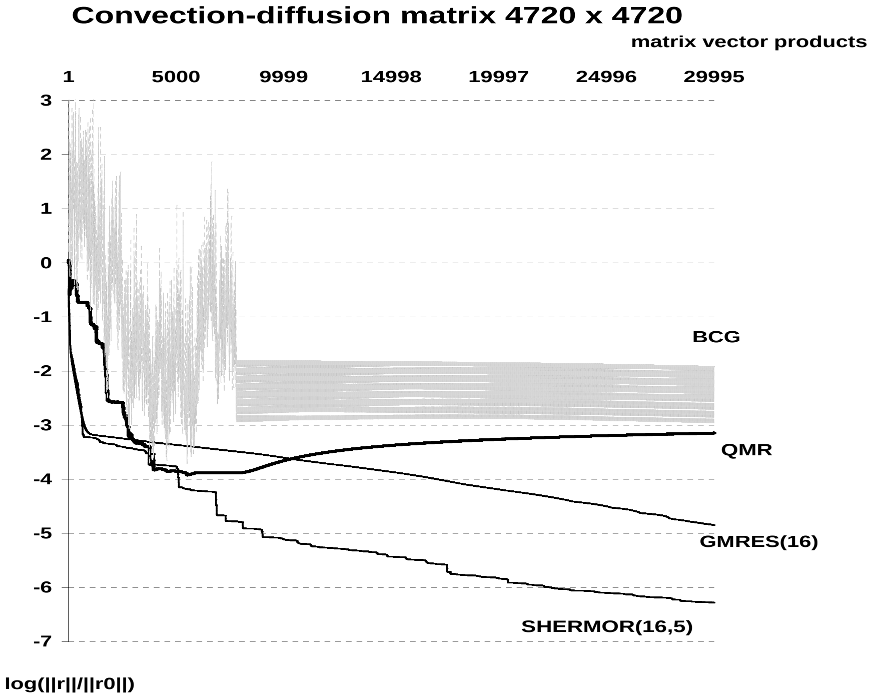

Figure 2.3: QMR, BCG, GMRES and SHERMOR for a convection dominated problem

to the other system. But it is logical to expect that we lose a part of the convergence speed during the transfer, especially when the original system has very poor convergence properties. In Figure 2.1 the loss of quality is expressed by the gap between GMRES30^ and the back-transformed curve. In the other examples we did not display the curve of the auxiliary system anymore. The gap between the curve for the first system after back-transformation and the curve for the second system can be explained by the following equation:

$$b - \mathbf{A}\bar{x} = b - \mathbf{A}\left(\frac{\hat{x}}{1 + y^T\hat{x}}\right) = \frac{b + by^T\hat{x} - \mathbf{A}\hat{x}}{1 + y^T\hat{x}} = \frac{b - (\mathbf{A} - by^T)\hat{x}}{1 + y^T\hat{x}} = \frac{b - \hat{\mathbf{A}}\hat{x}}{1 + y^T\hat{x}},$$

hence

$$||b - \mathbf{A}\bar{x}|| = \frac{||b - \hat{\mathbf{A}}\hat{x}||}{|1 + y^T\hat{x}|}.$$

The loss of quality after back-transformation grows particularly large when the scalar $1 + y^T\hat{x}$ tends to 0. Due to Theorem 2.1.1 and the non-singularity of $\hat{\mathbf{A}} + by^T = \mathbf{A}$, the expression $1 + y^T\hat{\mathbf{A}}^{-1}b$ cannot vanish when $\hat{\mathbf{A}}$ is nonsingular. In other words, when $1 + y^T\hat{x}$ is very small and $\hat{x}$ is a good approximation of $\hat{\mathbf{A}}^{-1}b$, then $\hat{\mathbf{A}}$ must be close to singular. Non-singularity of $\hat{\mathbf{A}}$ appears to be in danger whenever we force the auxiliary system to have too fast convergence speed. This is expressed by the following proposition. The result is not surprising since when we ask for fast convergence, our parameter vector y must have comparatively large elements (see equation (2.15)). The matrix $\mathbf{A} - by^T$ can become merely a perturbation of by^T, a singular rank-one matrix.

Proposition 2.4.1 *With the assumptions and notations of Theorem 2.2.7, let $\hat{x}_k$ denote the kth approximation calculated by the GMRES method applied to the system*

$$\hat{\mathbf{A}}x = b, \tag{2.21}$$

and let $k < n$. Let $\alpha_i := y^T v_i$, $i \leq k$, and $\hat{\mathbf{R}}_k$ denote the upper triangular matrix obtained by elimination of the lower subdiagonal elements of the kth Hessenberg matrix with the help of the Givens parameters $\hat{c}_i$, $\hat{s}_i$. Let further $\mathbf{r_k}$ denote the last column without last element of the matrix obtained by applying the first $k-1$ Givens rotations associated with (2.21) to the kth Hessenberg matrix associated with the original system (1.1). Then

$$1 + y^T \hat{x}_k = \left(1 + y^T \hat{x}_{k-1}\right) \left(\hat{s}_k^2 + \frac{\hat{s}_k \hat{c}_k (\sum_{j=1}^{k} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{k-1} (-\hat{s}_l))}{h_{k+1,k}} \right) \tag{2.22}$$

$$- \quad \beta \frac{\hat{s}_k \hat{c}_k}{h_{k+1,k}} \prod_{l=1}^{k-1} (-\hat{s}_l) (\alpha_1, \ldots, \alpha_{k-1}) \hat{\mathbf{R}}_{k-1}^{-1} \mathbf{r_k}.$$

Proof: After the kth GMRES iteration applied to (2.21), let $\mathbf{V}_k$ denote the matrix whose columns are the first k Arnoldi vectors, $\tilde{\mathbf{H}}_k$ denote the upper Hessenberg matrix of dimension $(k+1) \times k$ obtained together with the computation of $\mathbf{V}_k$ and let $\beta = \|b\|$. In the kth step we have k conditions for y:

$$y^T \mathbf{V}_k = (\alpha_1, \ldots, \alpha_k). \tag{2.23}$$

With $a := (\alpha_1, \ldots, \alpha_k)^T$, solutions for y all satisfy

$$y \in \mathbf{V}_k a + \mathbf{U},$$

where $\mathbf{U}$ is the orthogonal complement in $\mathbb{R}^n$ to span$\{v_1, \ldots, v_k\}$. The kth approximation $\hat{x}_k$ has because of the zero initial guess the form $\mathbf{V}_k w_k$ with $w_k \in \mathbb{R}^k$ minimizing $\|\beta e_1 - \tilde{\mathbf{H}}_k w\|$. Hence,

$$y^T \hat{x}_k = (\mathbf{V}_k a + \mathbf{U})^T (\mathbf{V}_k w_k) = a^T w_k. \tag{2.24}$$

The vector w_k can be obtained by computing the QR-decomposition of $\tilde{\mathbf{H}}_k$, $\tilde{\mathbf{H}}_k = \mathbf{Q}_k \tilde{\mathbf{R}}_k$. With $\hat{\mathbf{R}}_k$ being the upper triangular matrix given by $\tilde{\mathbf{R}}_k$ without its last row and g being $\mathbf{Q}_k^T \beta e_1$ without last element, we have

$$w_k = \hat{\mathbf{R}}_k^{-1} g.$$

When we denote the last column of $\hat{\mathbf{R}}_k$ without its last element by $\hat{\mathbf{r}}_{\mathbf{k}}$, then according to Lemma 2.2.5

$$e_i^T \hat{\mathbf{r}}_{\mathbf{k}} = \hat{s}_i \hat{h}_{i+1,k} + \hat{c}_i \sum_{j=1}^{i} \hat{c}_{j-1} \hat{h}_{j,k} \prod_{l=j}^{i-1} (-\hat{s}_l), \quad i \leq k-1,$$

and the element of $\hat{\mathbf{R}}_k$ on position $k \times k$ equals

$$\hat{r}_{k,k} = \hat{s}_k \hat{h}_{k+1,k} + \hat{c}_k \sum_{j=1}^{k} \hat{c}_{j-1} \hat{h}_{j,k} \prod_{l=j}^{k-1} (-\hat{s}_l).$$

Similarly, g has elements

$$g_i = \hat{c}_i \beta \prod_{l=1}^{i-1}(-\hat{s}_l), \quad i \leq k. \tag{2.25}$$

For the upper triangular matrix the following recurrence formula holds:

$$\hat{\mathbf{R}}_k^{-1} = \begin{pmatrix} \hat{\mathbf{R}}_{k-1}^{-1} & -\hat{\mathbf{R}}_{k-1}^{-1} \cdot \hat{\mathbf{r}}_{\mathbf{k}}/\hat{r}_{k,k} \\ 0 & 1/\hat{r}_{k,k} \end{pmatrix},$$

with $\hat{\mathbf{R}}_{k-1}$ being the left upper $(k-1)\times(k-1)$ part of $\hat{\mathbf{R}}_k$. If $g = (g^*, g_k)^T$, then

$$w_k = \hat{\mathbf{R}}_k^{-1} g = \begin{pmatrix} \hat{\mathbf{R}}_{k-1}^{-1} & -\hat{\mathbf{R}}_{k-1}^{-1} \cdot \hat{\mathbf{r}}_{\mathbf{k}}/\hat{r}_{k,k} \\ 0 & 1/\hat{r}_{k,k} \end{pmatrix} \begin{pmatrix} g^* \\ g_k \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{R}}_{k-1}^{-1}(g^* - g_k \hat{\mathbf{r}}_{\mathbf{k}}/\hat{r}_{k,k}) \\ g_k/\hat{r}_{k,k} \end{pmatrix},$$

and hence

$$a^T w_k = (\alpha_1, \ldots, \alpha_{k-1}) \hat{\mathbf{R}}_{k-1}^{-1}(g^* - g_k \hat{\mathbf{r}}_{\mathbf{k}}/\hat{r}_{k,k}) + \alpha_k g_k/\hat{r}_{k,k}. \tag{2.26}$$

Combined with (2.24) this yields

$$1 + y^T \hat{x}_k = 1 + y^T \hat{x}_{k-1} - (\alpha_1, \ldots, \alpha_{k-1}) \hat{\mathbf{R}}_{k-1}^{-1} g_k \hat{\mathbf{r}}_{\mathbf{k}}/\hat{r}_{k,k} + \alpha_k g_k/\hat{r}_{k,k}. \tag{2.27}$$

The last term of this expression equals

$$\frac{\alpha_k g_k}{\hat{r}_{k,k}} = \frac{\alpha_k \hat{c}_k \beta \prod_{l=1}^{k-1}(-\hat{s}_l)}{\hat{s}_k \hat{h}_{k+1,k} + \hat{c}_k \sum_{j=1}^{k} \hat{c}_{j-1} \hat{h}_{j,k} \prod_{l=j}^{k-1}(-\hat{s}_l)}.$$

Because $\hat{s}_k$ and $\hat{c}_k$ must zero out $\hat{h}_{k+1,k}$ we choose them according to (2.11) as

$$\hat{s}_k = \frac{\hat{h}_{k+1,k}}{\sqrt{\hat{h}_{k+1,k}^2 + (\sum_{j=1}^{k} \hat{c}_{j-1} \hat{h}_{j,k} \prod_{l=j}^{k-1}(-\hat{s}_l))^2}}, \tag{2.28}$$

$$\hat{c}_k = \frac{\sum_{j=1}^{k} \hat{c}_{j-1} \hat{h}_{j,k} \prod_{l=j}^{k-1}(-\hat{s}_l)}{\sqrt{\hat{h}_{k+1,k}^2 + (\sum_{j=1}^{k} \hat{c}_{j-1} \hat{h}_{j,k} \prod_{l=j}^{k-1}(-\hat{s}_l))^2}}. \tag{2.29}$$

Hence

$$\begin{aligned} \frac{\alpha_k g_k}{\hat{r}_{k,k}} &= \frac{\alpha_k \beta \prod_{l=1}^{k-1}(-\hat{s}_l)(\sum_{j=1}^{k} \hat{c}_{j-1} \hat{h}_{j,k} \prod_{l=j}^{k-1}(-\hat{s}_l))}{\hat{h}_{k+1,k}^2 + (\sum_{j=1}^{k} \hat{c}_{j-1} \hat{h}_{j,k} \prod_{l=j}^{k-1}(-\hat{s}_l))^2} \\ &= \frac{\hat{c}_k^2}{\sum_{j=1}^{k} \hat{c}_{j-1} \hat{h}_{j,k} \prod_{l=j}^{k-1}(-\hat{s}_l)} \alpha_k \beta \prod_{l=1}^{k-1}(-\hat{s}_l). \end{aligned} \tag{2.30}$$

Only $\hat{h}_{1,k} = h_{1,k} - \beta\alpha_k$ depends upon y, otherwise $\hat{h}_{j,k} = h_{j,k} = v_j^T \mathbf{A} v_k$. Values for α_k that yield a Givens sine of the form (2.28) are given by (2.15) and straightforward computation shows that in (2.15) only the positive root gives the Givens cosine of (2.29). Thus

$$\alpha_k = \frac{\frac{\hat{c}_k}{\hat{s}_k} h_{k+1,k} - \sum_{j=1}^{k} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{k-1}(-\hat{s}_l)}{-\beta \prod_{l=1}^{k-1}(-\hat{s}_l)} \tag{2.31}$$

and exploiting twice this equation we obtain

$$\frac{\alpha_k g_k}{\hat{r}_{k,k}} = \hat{c}_k^2 \frac{\frac{\hat{c}_k}{\hat{s}_k} h_{k+1,k} - \sum_{j=1}^{k} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{k-1}(-\hat{s}_l)}{\alpha_k \beta \prod_{l=1}^{k-1}(-\hat{s}_l) - h_{1,k} \prod_{l=1}^{k-1}(-\hat{s}_l) - \sum_{j=2}^{k} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{k-1}(-\hat{s}_l)} =$$

$$\hat{c}_k^2 \frac{\frac{\hat{c}_k}{\hat{s}_k} h_{k+1,k} - \sum_{j=1}^{k} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{k-1} (-\hat{s}_l)}{-\frac{\hat{c}_k}{\hat{s}_k} h_{k+1,k}} = -\hat{c}_k^2 + \frac{\hat{c}_k \hat{s}_k \sum_{j=1}^{k} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{k-1} (-\hat{s}_l)}{h_{k+1,k}}.$$

Thus (2.27) changes to

$$1+y^T\hat{x}_k = 1+y^T\hat{x}_{k-1} - (\alpha_1,\dots,\alpha_{k-1})\hat{\mathbf{R}}_{k-1}^{-1} g_k \hat{\mathbf{r}}_{\mathbf{k}}/\hat{r}_{k,k} - \hat{c}_k^2 + \frac{\hat{c}_k \hat{s}_k \sum_{j=1}^{k} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{k-1} (-\hat{s}_l)}{h_{k+1,k}} =$$

$$\hat{s}_k^2 + \frac{\hat{c}_k \hat{s}_k \sum_{j=1}^{k} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{k-1} (-\hat{s}_l)}{h_{k+1,k}} + y^T\hat{x}_{k-1} - (\alpha_1,\dots,\alpha_{k-1})\hat{\mathbf{R}}_{k-1}^{-1} g_k \hat{\mathbf{r}}_{\mathbf{k}}/\hat{r}_{k,k}. \quad (2.32)$$

Concerning the last term of this expression, we have

$$\frac{g_k}{\hat{r}_{k,k}} e_i^T \hat{\mathbf{r}}_{\mathbf{k}} = \hat{c}_k \beta \prod_{l=1}^{k-1} (-\hat{s}_l) \frac{\hat{s}_i h_{i+1,k} + \hat{c}_i \sum_{j=1}^{i} \hat{c}_{j-1} \hat{h}_{j,k} \prod_{l=j}^{i-1} (-\hat{s}_l)}{\hat{s}_k h_{k+1,k} + \hat{c}_k \sum_{j=1}^{k} \hat{c}_{j-1} \hat{h}_{j,k} \prod_{l=j}^{k-1} (-\hat{s}_l)} =$$

$$\frac{\hat{c}_k \beta \prod_{l=1}^{k-1} (-\hat{s}_l)}{\hat{s}_k h_{k+1,k} + \hat{c}_k \frac{\hat{c}_k}{\hat{s}_k} h_{k+1,k}} \left(\hat{s}_i h_{i+1,k} + \hat{c}_i \left((h_{1,k} - \beta\alpha_k) \prod_{l=1}^{i-1} (-\hat{s}_l) \right) + \hat{c}_i \sum_{j=2}^{i} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{i-1} (-\hat{s}_l) \right),$$

having used (2.31). Again with (2.31) this expression equals

$$\frac{\hat{c}_k \hat{s}_k \beta \prod_{l=1}^{k-1} (-\hat{s}_l)}{h_{k+1,k}} \quad \times$$

$$\left(\hat{s}_i h_{i+1,k} + \hat{c}_i \prod_{l=i}^{k-1} (\frac{-1}{\hat{s}_l}) \left(\frac{\hat{c}_k}{\hat{s}_k} h_{k+1,k} - \sum_{j=1}^{k} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{k-1} (-\hat{s}_l) \right) + \hat{c}_i \sum_{j=1}^{i} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{i-1} (-\hat{s}_l) \right).$$

Thus

$$g_k \hat{\mathbf{r}}_{\mathbf{k}}/\hat{r}_{k,k} = \beta \frac{\hat{c}_k \hat{s}_k \prod_{l=1}^{k-1} (-\hat{s}_l)}{h_{k+1,k}} \mathbf{r}_{\mathbf{k}} \quad +$$

$$\left(\hat{c}_k^2 - \frac{\hat{c}_k \hat{s}_k \sum_{j=1}^{k} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{k-1} (-\hat{s}_l)}{h_{k+1,k}} \right) \begin{pmatrix} \beta\hat{c}_1 \prod_{l=1}^{0} (-\hat{s}_l) \\ \vdots \\ \beta\hat{c}_{k-1} \prod_{l=1}^{k-2} (-\hat{s}_l) \end{pmatrix}.$$

The claim follows by substitution in (2.32) and when we realize that

$$\begin{pmatrix} \beta\hat{c}_1 \prod_{l=1}^{0} (-\hat{s}_l) \\ \vdots \\ \beta\hat{c}_{k-1} \prod_{l=1}^{k-2} (-\hat{s}_l) \end{pmatrix} = g^*.$$

□

Proposition 2.4.1 shows that for too fast prescribed convergence, that is for $\hat{s}_k \approx 0$, the value $1 + y^T\hat{x}_k$ vanishes. Thus attempts to overcome stagnation of the restarted GMRES method become ineffective when we ask for too dramatic residual norm decreasing in the auxiliary system. On the other hand, a reasonably defined, non-stagnating curve of the second system and back-transformation with Sherman-Morrison can successfully overcome stagnating, as was seen in the above examples.

2.5 Open questions

The procedure we proposed in this chapter needs some more investigation concerning the following points. First of all, the algorithm we constructed from it (Algorithm 5.2.1), is not a classical algorithm in the sense that it needs prescribed auxiliary system residual norms as input. One can easily implement default auxiliary system residual norms but we did not yet find out what prescribed iterations yield best convergence in general. We do know from the preceding that too fast decreasing norms spoil the back-transformation, on the other hand we must at least define a non-stagnating second system. In some cases this is a very delicate problem. Let us give an extreme example.

Example 4. PDE matrix of dimension 400.

Let us consider the first example of Section 2.3, where we prescribed the first 10 residual norms of the auxiliary system. Here we will describe only 4 residual norms of GMRES applied to $\mathbf{A} - by^T$, namely

$$\|\hat{r}_1\| := 9, \quad \|\hat{r}_2\| := 8, \quad \|\hat{r}_3\| := 7, \quad \|\hat{r}_4\| := 6.$$

The resulting auxiliary system perfectly converges, the back-transformed curve is seen in Figure 2.4 and denoted by pmpm6. But when we force the initial residual norms to equal

$$\|\hat{r}_1\| := 9, \quad \|\hat{r}_2\| := 8, \quad \|\hat{r}_3\| := 7, \quad \|\hat{r}_4\| := 5,$$

we obtain a stagnating auxiliary system, and consequently back-transformed iterations stagnate too. This is expressed by the curve pmpm5. Apparently, the distance between third and fourth residual norm was chosen too large.

A different matter of concern is the choice of the concrete parameter vector that forces certain residual norms. If we prescribe k norms, the proof of Theorem 2.2.7 finds 2^k possible sequences of k conditions to put on $y \in \mathbb{R}^n$ (see (2.15)). We are confronted with the choice of conditions and after that with the choice of y that satisfies the chosen conditions. We solved the latter problem by putting $y := \mathbf{V}_k(\alpha_1, \ldots, \alpha_k)^T$ when we have the conditions $y^T v_i = \alpha_i$, $i \leq k$. Thus the computation of $y \in \mathbb{R}^n$ is the least expensive option, but other choices yield of course other auxiliary systems. Concerning the choice of the conditions, we have used so far only the conditions with positive root in (2.15). An example of the influence of this choice is the following:

Example 5. PDE matrix of dimension 400.

We consider again the first example of Section 2.3, and describe as before 4 residual norms of GMRES applied to $\mathbf{A} - by^T$, namely

$$\|\hat{r}_1\| := 9, \quad \|\hat{r}_2\| := 8, \quad \|\hat{r}_3\| := 7, \quad \|\hat{r}_4\| := 6.$$

The curve pmpm6 that we have seen in the preceding example (in Figure 2.4) was obtained by choosing for the first condition on y in (2.15) the plus sign, for the second the minus sign, for the third plus and minus for the last (the same holds for pmpm5). If we inverse these choices, i.e. choose minus-plus-minus-plus, we obtain the curve mpmp6. The original nor the auxiliary system converges.

The only thing we can do to overcome these difficulties for the moment is adding, when a prescribed auxiliary system yields stagnation, a correcting condition

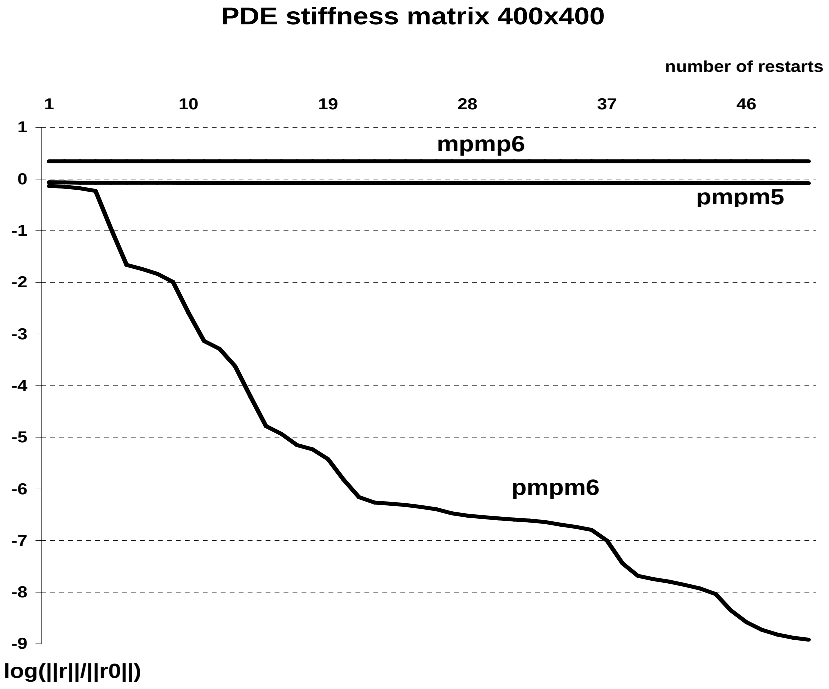

Figure 2.4: Sensibility of prescribed curve and parameter vector choice when applying SHERMOR(30,4)

for y in the following sense. We have seen in the preceding section that the quality of the back-transformation depends on the denominator $1+y^T\hat{x}_k$. With the information gained from Proposition 2.4.1 we can compute, when the $(k-1)$st prescribed residual norms yield stagnation, at iteration number k a denominator that is as far from zero as possible. In case the stagnation is indeed caused by a too small denominator $1+y^T\hat{x}_{k-1}$, it is reasonable to expect that the new denominator $1+y^T\hat{x}_k$ corrects at least for a while the problem of the back-transformation. And when the stagnation has a different cause it can still be stimulating to have an auxiliary matrix that is further away from singularity than at step $k-1$. The correcting condition for y is obtained as follows.

Let us consider a given value $\gamma_k \neq 0$ and determine the values of $\hat{s}_k$ in (2.22) that make the denominator equal γ_k. For this purpose we introduce the following notations:

$$T_k := \frac{\beta}{h_{k+1,k}} \prod_{l=1}^{k-1}(-\hat{s}_l)(\alpha_1,\ldots,\alpha_{k-1})\hat{\mathbf{R}}_{k-1}^{-1}\mathbf{r}_k, \quad S_k := \frac{\sum_{j=1}^{k} \hat{c}_{j-1} h_{j,k} \prod_{l=j}^{k-1}(-\hat{s}_l)}{h_{k+1,k}},$$

and $1+y^T\hat{x}_{k-1} =: \gamma_{k-1}$. Then

$$1+y^T\hat{x}_k = \gamma_k \quad \Longleftrightarrow \quad \gamma_{k-1}\hat{s}_k^2 + (\gamma_{k-1}S_k - T_k)\hat{s}_k\hat{c}_k - \gamma_k = 0$$

$$\Longleftrightarrow \quad \gamma_{k-1}\hat{s}_k^2 - \gamma_k = \pm(\gamma_{k-1}S_k - T_k)\hat{s}_k\sqrt{1-\hat{s}_k^2},$$

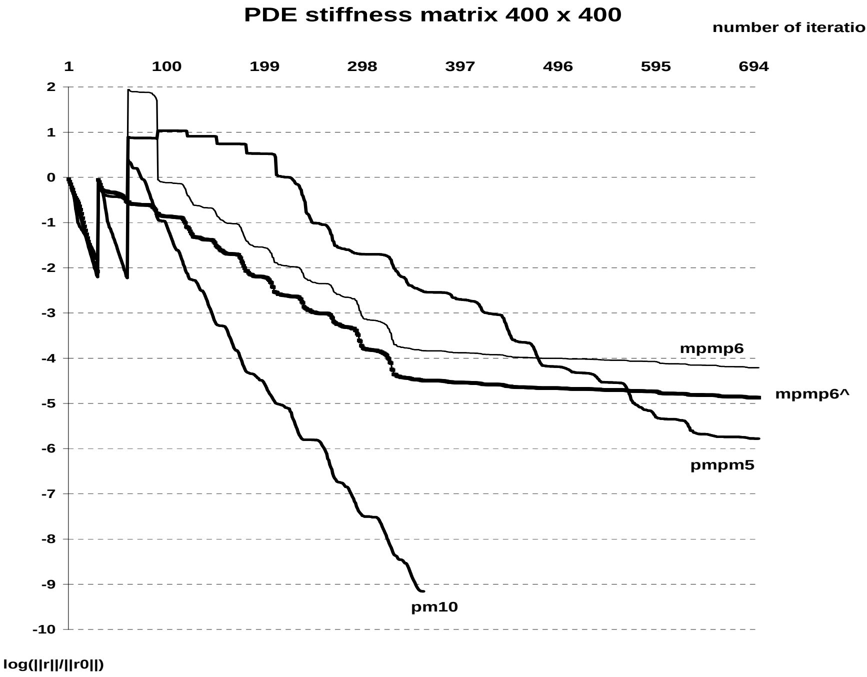

Figure 2.5: Intervention to enhance the quality of the back-transformation when applying SHERMOR(30,4) and SHERMOR(30,10)

yielding for $\hat{s}_k^2$ the roots

$$\hat{s}_k^2 = \frac{(\gamma_{k-1}S_k - T_k)^2 + 2\gamma_{k-1}\gamma_k \pm (\gamma_{k-1}S_k - T_k)\sqrt{(\gamma_{k-1}S_k - T_k)^2 + 4\gamma_k(\gamma_{k-1} - \gamma_k)}}{2(\gamma_{k-1}^2 + (\gamma_{k-1}S_k - T_k)^2)}. \tag{2.33}$$

From every root we can extract one value for $\hat{s}_k$ that forces the kth denominator to equal γ_k as long as the above square root is defined. Thus the term under the square root in (2.33), the determinant, can tell us what values the denominator assumes at all. To correct $k-1$ prescribed residual norms we choose a possibly large γ_k in the interval that is determined by the roots of the determinant in (2.33). From the corresponding values for $\hat{s}_k^2$ we obtain a correcting condition on y. To illustrate this, we will apply the procedure to the stagnating curves of examples 4 and 5.

In curve pmpm5 we have prescribed 4 auxiliary system residual norms. The roots of the determinant of (2.33) with $k = 5$ are -0.008333 and 0.175647. We choose a relatively large fifth denominator in this interval, namely

$$1 + y^T\hat{x}_5 := 0.17$$

and obtained two Givens sines forcing such denominator, $\hat{s}_k = 0.999265$ or $\hat{s}_k = 0.924693$. We have chosen to force the smallest one and we computed the resulting condition for y. Having defined the auxiliary system with these 5 conditions for y, the corresponding back-transformed curve pmpm5 in Figure 2.5 converges very well.

As for curve mpmp6 from Example 5, we applied exactly the same strategy: The roots of the determinant of (2.33) with $k = 5$ are -0.004602 and 0.457942. We choose in this interval $1 + y^T\hat{x}_5 := 0.4$ and obtained two Givens sines forcing such denominator, $\hat{s}_k = 0.965911$ or $\hat{s}_k = 0.895302$. Of course, we have chosen the smallest, the second one. Having defined the auxiliary system with these 5 conditions for y, the corresponding auxiliary system curve mpmp6$^{\wedge}$ in Figure 2.5 converges and the back-transformed curve mpmp6 does so too. We see the gap between the two systems stays, after the initial cycles, very constant. It even corresponds to a denominator of size about 0.3 but this is certainly coincidence because the gap changes heavily after the moment of intervention, the 5th iteration.

Finally, we showed the positive influence an extra condition can have on a system that converges already. We prescribed the 10 residual norms of Example 1 and choose the conditions with alternating plus and minus sign in (2.15). This system converges slower than the one displayed in Example 1. When we apply a correcting eleventh iteration in the same manner as before, we obtain the curve pm10 in Figure 2.5. It is even steeper than the one of Figure 2.1.

Of course, the motivation for using this last strategy is rather heuristic. But we believe that observation of the denominator is the key to optimal prescribed convergence curves. As we mentioned above, this item and the choice of the concrete parameter vector need further investigation.

Chapter 3

Rank-one updates after the initial cycle

In the preceding chapter we succeeded in prescribing the convergence speed of the GMRES method when applied to the rank-one updated matrix $\mathbf{A} - by^T$ by the choice of y. This was possible because we considered processes with zero initial guess. The power of exploiting $\mathbf{A} - by^T$ in a GMRES cycle with $x_0 = 0$ is mainly caused by the fact that adding the rank-one matrix $-by^T$, *where b equals the initial residual itself*, one can make the distance between this residual and the projection space as small as wanted. For $x_0 \neq 0$ this does not hold anymore. But we can apply the results of Chapter 2 to an original system with nonzero first guess in a simple way by considering Theorem 2.2.7 with

$$b := v_1 = (b' - \mathbf{A}x_0)/\|b' - \mathbf{A}x_0\|,$$

b' being the right-hand side of the original system. Then we obtain whatever convergence curve for

$$\hat{\mathbf{A}} = \mathbf{A} - v_1 y^T$$

with right-hand side v_1. With (2.3) one could find an approximate solution $\bar{x}$ of

$$\mathbf{A}x = v_1,$$

and hence

$$\mathbf{A}(\|b' - \mathbf{A}x_0\|\bar{x} + x_0) \approx b'.$$

We have seen convincing examples of the effectiveness of the technique from Chapter 2, but we also presented an example where stagnation could not be overcome. In principle it is possible to define, as soon as a process stagnates, an auxiliary system with the nonzero approximate from the preceding cycle as initial guess in the way just described. One must only realize that transition to a new system after *every* restart does not make sense, because an auxiliary system never improves the iterates of the original system as long as it is not restarted. We have pointed this out earlier in Section 2.3.

Thus we can translate the advantages of zero initial guesses to arbitrary guesses, but we also have to deal with the problems we formulated in the end of the preceding chapter. Unfavorably prescribed residual norms can spoil the convergence of the first cycle during later restarts and moreover, the auxiliary system can converge but back-transformation leads to division by nearly zero. It would be very useful to be able to improve a given matrix in a different way after the first GMRES cycle, that is when iterates start to converge and to be further away from the origin.

3.1 Local minimization

If $x_0 \neq 0$, Proposition 2.2.1 is not valid anymore. Instead of the first row elements, *all* the elements of the Hessenberg matrix generated by an Arnoldi process for $\mathbf{A} - by^T$ are dependent on y. The dependency from the lower subdiagonal elements is not even linear anymore. As a consequence, also the property $\mathcal{K}_k(\hat{\mathbf{A}}, r_0) = \mathcal{K}_k(\mathbf{A}, r_0)$ is lost. This has one advantage, namely that back-transformation from an auxiliary system can improve convergence during non-restarted, that is full GMRES processes too. On the other hand, the computation of a favorable parameter vector exploiting a modified Hessenberg matrix as we did in Section 3.2 is significantly more complicated. But modification of projection angles with the help of Givens sines is still a useful tool. With Corollary 2.2.6 Givens sines satisfy

$$\hat{s_k}^2 = \frac{\hat{h}^2_{k+1,k}}{\hat{h}^2_{k+1,k} + (\sum_{j=1}^k \hat{c}_{j-1}\hat{h}_{j,k}\prod_{i=j}^{k-1}(-\hat{s}_i))^2}, \qquad \hat{c}_0 := 1.$$

With $\hat{h}_{k+1,k}$ depending upon y it is clear we cannot prescribe arbitrary small sines anymore. In fact, all we can do is minimize them.

Before we proceed to minimization techniques we would like to remark the following. We are primarily interested in acceleration of stagnating systems. When a process stagnates we have very large Givens sines, or equivalently, nearly vanishing Givens cosines. Thus the term

$$\sum_{j=1}^{k} \hat{c}_{j-1}\hat{h}_{j,k}\prod_{i=j}^{k-1}(-\hat{s}_i)$$

approximately equals $\pm\hat{h}_{1,k}$! Only the first row element and the lower subdiagonal one do really matter if stagnation occurs and we have

$$\hat{s}_k^2 \approx \frac{\hat{h}^2_{k+1,k}}{\hat{h}^2_{k+1,k} + \hat{h}^2_{1,k}} = \frac{\|\hat{\mathbf{A}}\hat{v}_k\|^2 - \sum_{j=1}^k \hat{h}^2_{j,k}}{\|\hat{\mathbf{A}}\hat{v}_k\|^2 - \sum_{j=2}^k \hat{h}^2_{j,k}}$$

due to Lemma 2.2.8. Interestingly, this remark is the more valid the worse a system converges. As in the previous chapter, a relatively large choice of $\hat{h}_{1,k}$ will stimulate convergence. This fact has already been observed in the past in the context of GMRES processes before. Indeed, large numbers $h_{1,k} = v_1^T\mathbf{A}v_k$ imply that the normed residual v_1 is close to the kth dimension of the projection space $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$. Hence the kth residual is small.

Now let us investigate minimization of Givens sines. A new possibility with nonzero initial guesses is that we can intervene already from the initial residual norm on:

$$\|\hat{r}_0\|^2 = \|b - \hat{\mathbf{A}}x_0\|^2 = \|r_0 + b(y^Tx_0)\|^2 = \|r_0\|^2 + 2(y^Tx_0)(r_0^Tb) + (y^Tx_0)^2\|b\|^2.$$

The choice

$$y^Tx_0 = -\frac{r_0^Tb}{\|b\|^2}$$

minimizes the initial residual norm and the minimized residual equals

$$\hat{r}_0 = r_0 - \frac{r_0^Tb}{\|b\|^2}b. \tag{3.1}$$

Concerning the residual norms to follow, we have

$$\|\hat{r}_k\| = \|\hat{r}_0\||\hat{s}_1 \cdot \ldots \cdot \hat{s}_k|,$$

and again with Corollary 2.2.6

$$\hat{s}_k^2 = \frac{\hat{h}_{k+1,k}^2}{\hat{h}_{k+1,k}^2 + (\sum_{j=1}^k \hat{c}_{j-1}\hat{h}_{j,k}\prod_{i=j}^{k-1}(-\hat{s}_i))^2},$$

with $\hat{c}_0 := 1$. Having minimized $\|\hat{r}_1\|$ until $\|\hat{r}_{k-1}\|$, minimizing the kth residual norm amounts to maximizing

$$\left|\frac{\sum_{j=1}^k \hat{c}_{j-1}\hat{h}_{j,k}\prod_{i=j}^{k-1}(-\hat{s}_i)}{\hat{h}_{k+1,k}}\right|. \tag{3.2}$$

If we have minimized the initial residual norm as above and consecutively minimize all following norms, we obtain the following result.

Proposition 3.1.1 *Let us consider minimization of residual norms by the choice of $y \in \mathbb{R}^n$ when the GMRES method is applied to the system*

$$\hat{\mathbf{A}}x = b, \quad \hat{\mathbf{A}} = \mathbf{A} - by^T, \tag{3.3}$$

with nonzero initial guess and assume that the $k+1$st Krylov subspace $\mathcal{K}_{k+1}(\hat{\mathbf{A}}, \hat{r}_0)$, $k+1 \leq n$, has dimension $k+1$ and is spanned by $\{v_1, \ldots, v_{k+1}\}$. If we have minimized the initial residual norm by requiring

$$y^T x_0 = -\frac{r_0^T b}{\|b\|^2},$$

and consecutively minimize the first k residual norms, then the condition

$$y^T \hat{v}_{k+1} = \frac{b^T \mathbf{A}\hat{v}_{k+1}}{\|b\|^2}$$

minimizes the $(k+1)$st residual norm. Moreover, the Arnoldi process with these choices generates a Krylov subspace $\mathcal{K}_{k+1}(\hat{\mathbf{A}}, \hat{r}_0)$ that is orthogonal to b.

Proof: Per induction. $k = 1$:
Because of (3.1) we have

$$\hat{v}_1^T b = \frac{1}{\|\hat{r}_0\|} b^T (r_0 - \frac{r_0^T b}{\|b\|^2} b) = 0.$$

The first Hessenberg element therefore equals

$$\hat{h}_{1,1} = \hat{v}_1^T (\mathbf{A} - by^T)\hat{v}_1 = \hat{v}_1^T \mathbf{A}\hat{v}_1.$$

Minimizing the first Givens sine is in this case the same as minimizing $\hat{h}_{2,1}$ because of (3.2). We have with Lemma 2.2.8

$$\hat{h}_{2,1}^2 = \|\hat{\mathbf{A}}\hat{v}_1\|^2 - \hat{h}_{1,1}^2 = \|(\mathbf{A} - by^T)\hat{v}_1\|^2 - \hat{h}_{1,1}^2 =$$
$$\|\mathbf{A}\hat{v}_1\|^2 - 2y^T \hat{v}_1 b^T \mathbf{A}\hat{v}_1 + (y^T \hat{v}_1)^2 \|b\|^2 - \hat{h}_{1,1}^2$$

and $\hat{h}_{2,1}$ is minimized by the choice

$$y^T\hat{v}_1 = \frac{b^T\mathbf{A}\hat{v}_1}{\|b\|^2}.$$

$k \to k+1$:
Let us assume we have minimized the first k Givens sines with

$$y^T\hat{v}_i = \frac{b^T\mathbf{A}\hat{v}_i}{\|b\|^2}, \quad i \le k,$$

and have proved that $\hat{v}_i^T b = 0$, $i \le k$.
From these assumptions it follows that

$$\hat{v}_{k+1}^T b = \frac{1}{\hat{h}_{k+1,k}} b^T(\hat{\mathbf{A}}\hat{v}_k - \sum_{j=1}^{k} \hat{h}_{j,k}\hat{v}_j) = \frac{1}{\hat{h}_{k+1,k}} b^T(\mathbf{A}\hat{v}_k - (y^T\hat{v}_k)b)$$

$$= \frac{1}{\hat{h}_{k+1,k}} b^T(\mathbf{A}\hat{v}_k - \frac{b^T\mathbf{A}\hat{v}_k}{\|b\|^2} b) = 0.$$

Therefore all

$$\hat{h}_{i,k+1} = \hat{v}_i^T(\mathbf{A} - by^T)\hat{v}_{k+1} = \hat{v}_i^T\mathbf{A}\hat{v}_{k+1}, \quad i \le k+1,$$

are independent from further conditions on y. Minimizing the $(k+1)$st Givens sine is in this case the same as minimizing $\hat{h}_{k+2,k+1}$ because of (3.2). We have with Lemma 2.2.8

$$\hat{h}_{k+2,k+1}^2 = \|\hat{\mathbf{A}}\hat{v}_{k+1}\|^2 - \sum_{j=1}^{k+1} \hat{h}_{j,k+1}^2 = \|(\mathbf{A} - by^T)\hat{v}_{k+1}\|^2 - \sum_{j=1}^{k+1} \hat{h}_{j,k+1}^2 =$$

$$\|\mathbf{A}\hat{v}_{k+1}\|^2 - 2y^T\hat{v}_{k+1}b^T\mathbf{A}\hat{v}_{k+1} + (y^T\hat{v}_{k+1})^2\|b\|^2 - \sum_{j=1}^{k+1} \hat{h}_{j,k+1}^2$$

and hence $\hat{h}_{k+2,k+1}$ is minimized by the choice

$$y^T\hat{v}_{k+1} = \frac{b^T\mathbf{A}\hat{v}_{k+1}}{\|b\|^2}.$$

□

The dependency of the Hessenberg matrix from y, which was concentrated on its first row in the previous chapter, for nonzero initial guesses moves to the lower subdiagonal. With the foregoing proposition, one can easily find a vector $y \in \mathbb{R}^n$ such, that the GMRES method applied to the alternative system (3.3) with nonzero initial guess reduces the residual norm of every step maximally. The resulting algorithm, Algorithm 5.2.5, is amazingly simple. Let us give an example with this kind of minimization.

Example 1. Nonnormal test matrix.

This highly nonnormal matrix is taken from Erhel [7]. It has the form $\mathbf{A} = \mathbf{SDS}^{-1}$ with $\mathbf{A}, \mathbf{D}, \mathbf{S} \in \mathbb{R}^{100\times100}$. $\mathbf{D}$ is a diagonal matrix, $\mathbf{D}$=diag$(1,\dots,100)$ and $\mathbf{S}$ is bidiagonal with the element 1 on the diagonal and 1.1 on the upper subdiagonal. The resulting matrix has 5050 nonzero elements, we used the right-hand side $b = (1,\dots,1)^T$

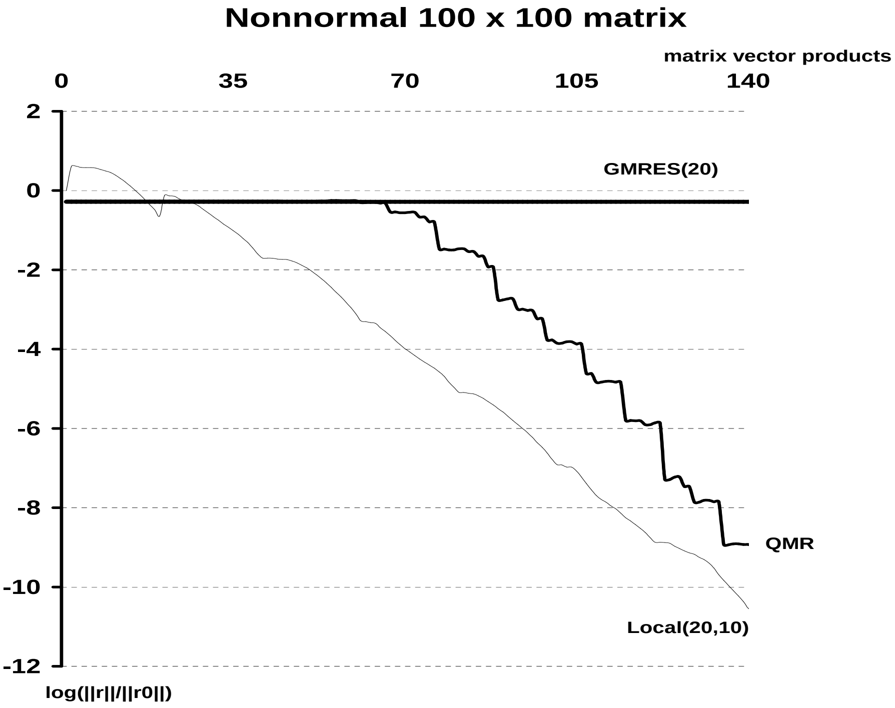

Figure 3.1: QMR, GMRES(20) and local minimization

and in this case a nonzero initial guess is needed. We chose $x_0 = (0.001, \dots, 0.001)$. Successive minimization of the auxiliary system during the first 10 iterations overcomes the stagnation of GMRES(20). The curve after back-transformation from the auxiliary system is denoted by LOCAL(20,10). The QMR method, however, gives about as satisfactory a convergence curve and is computational less expensive due to three term recurrences.

The minimization we achieve with Algorithm 5.2.5 must be understood in dependency from the iteration number. Moreover, maximal norm reducing of one step can prevent the next steps from being reasonably minimalizable and in the worst case a cycle without any minimization converges better than stepwise minimization. In the example at the end of Chapter 5 this is exactly what happens. GMRES(50) converges very slowly but it converges and our minimization stagnates without any convergence. This phenomenon reminds us on the paradox that we already mentioned in Section 1.4 and that is due to Eiermann, Ernst and Schneider [16]: Examples exist in which convergence is faster for smaller restart parameters than for larger ones, because the resulting vectors $\hat{r}_{k-1}$ generate closer Krylov subspaces.

3.2 Global minimization

The process of Proposition 3.1.1 minimizes residual norms at every single iteration in relation with the foregoing residual, that is only locally. One would expect better results when globally minimizing the residual norm after say k steps, regardless of the norms of previous residual vectors. This is what we will do next.

To begin with, let us assume we do not try to minimize the initial residual. Instead let $y^T x_0$ be zero, so that $\|r_0\| = \|\hat{r}_0\|$.
When GMRES is applied to the system (3.3), the residual norm of the kth iterate is the distance from the initial residual to the subspace

$$\mathrm{span}(\hat{\mathbf{A}} r_0, \ldots, \hat{\mathbf{A}}^k r_0).$$

Let us take a closer look at this subspace.

Lemma 3.2.1 *The vector* $\hat{\mathbf{A}}^k r_0$, $k \geq 1$, *has the form:*

$$\hat{\mathbf{A}}^k r_0 = \mathbf{A}^k r_0 + \sum_{j=0}^{k-1} \alpha_{k-j} \mathbf{A}^j b,$$

with

$$\alpha_1 = -y^T r_0 \quad \text{and} \quad \alpha_k = -y^T \mathbf{A}^{k-1} r_0 - \sum_{j=0}^{k-2} (y^T \mathbf{A}^j b) \alpha_{k-1-j}, \quad k > 1.$$

Proof: By induction. $k = 1$:

$$\hat{\mathbf{A}} r_0 = (\mathbf{A} - b y^T) r_0 = \mathbf{A} r_0 - b y^T r_0.$$

$k - 1 \to k$:

$$\hat{\mathbf{A}}^k r_0 = (\mathbf{A} - b y^T)(\sum_{j=0}^{k-2} \alpha_{k-1-j} \mathbf{A}^j b + \mathbf{A}^{k-1} r_0) =$$

$$\sum_{j=0}^{k-2} \alpha_{k-1-j} \mathbf{A}^{j+1} b + \mathbf{A}^k r_0 - b \left(\sum_{j=0}^{k-2} \alpha_{k-1-j} (y^T \mathbf{A}^j b) + y^T \mathbf{A}^{k-1} r_0 \right) = \sum_{j=0}^{k-1} \alpha_{k-j} \mathbf{A}^j b + \mathbf{A}^k r_0,$$

with

$$\alpha_k := -\left(\sum_{j=0}^{k-2} (y^T \mathbf{A}^j b) \alpha_{k-1-j} + y^T \mathbf{A}^{k-1} r_0 \right). \quad \square$$

As a consequence, we have

$$\hat{\mathbf{A}} \mathcal{K}_k(\hat{\mathbf{A}}, r_0) \subseteq \mathrm{span}\{\mathbf{A} r_0, \ldots, \mathbf{A}^k r_0, b, \ldots, \mathbf{A}^{k-1} b\}.$$

In order to investigate the behaviour of the subspace $\hat{\mathbf{A}} \mathcal{K}_k(\hat{\mathbf{A}}, r_0)$ in current stable GMRES implementations we need to express its elements in terms of orthonormal bases for $\mathbf{A} \mathcal{K}_k(\mathbf{A}, r_0)$ and $\mathcal{K}_k(\mathbf{A}, b)$.
Let $\{v_1, \ldots, v_k\}$ be an orthonormal basis of $\mathcal{K}_k(\mathbf{A}, b)$ with $v_1 := b/\|b\|$ and Arnoldi decomposition

$$\mathbf{A} \mathbf{V}_k = \mathbf{V}_{k+1} \tilde{\mathbf{H}}_k, \quad h_{i,j} = (\tilde{\mathbf{H}}_k)_{i,j},$$

and let $\{w_1, \ldots, w_k\}$ be an orthonormal basis of $\mathbf{A} \mathcal{K}_k(\mathbf{A}, r_0)$ with $w_1 := \mathbf{A} r_0 / \|\mathbf{A} r_0\|$ and Arnoldi decomposition

$$\mathbf{A} \mathbf{W}_k = \mathbf{W}_{k+1} \tilde{\mathbf{G}}_k, \quad g_{i,j} = (\tilde{\mathbf{G}}_k)_{i,j}.$$

We will propose a basis $\{\hat{w}_1, \ldots, \hat{w}_k\}$ for $\hat{\mathbf{A}} \mathcal{K}_k(\hat{\mathbf{A}}, r_0)$ that consists of linear combinations of $\{w_1, \ldots, w_k\}$ and $\{v_1, \ldots, v_k\}$.
For $k = 1$ we have

$$\hat{\mathbf{A}} r_0 = (\mathbf{A} - b y^T) r_0 = \|\mathbf{A} r_0\| w_1 - \|b\| y^T r_0 v_1.$$

We initialize by putting

$$\hat{w}_1 := w_1 + \alpha_{1,1} v_1, \qquad \alpha_{1,1} := -\frac{\|b\| y^T r_0}{\|\mathbf{A} r_0\|}. \tag{3.4}$$

Clearly, $\mathrm{span}\{\hat{w}_1, \hat{\mathbf{A}}\hat{w}_1\} = \hat{\mathbf{A}}\mathcal{K}_2(\hat{\mathbf{A}}, r_0)$. We have

$\hat{\mathbf{A}}\hat{w}_1 = \mathbf{A}w_1 + \alpha_{1,1}\mathbf{A}v_1 - b(y^T\hat{w}_1) = g_{1,1}w_1 + g_{2,1}w_2 + \alpha_{1,1}(h_{1,1}v_1 + h_{2,1}v_2) - \|b\|y^T\hat{w}_1 v_1.$

But $\hat{\mathbf{A}}\mathcal{K}_2(\hat{\mathbf{A}}, r_0)$ also equals $\mathrm{span}\{\hat{w}_1, (\hat{\mathbf{A}}\hat{w}_1 - g_{1,1}\hat{w}_1)/g_{2,1}\}$, giving a basis that is more easy to work with. The second basis vector $\hat{w}_2$ then equals

$\hat{w}_2 := (\hat{\mathbf{A}}\hat{w}_1 - g_{1,1}\hat{w}_1)/g_{2,1} = \big(g_{2,1}w_2 + (\alpha_{1,1}h_{1,1} - \alpha_{1,1}g_{1,1} - \|b\|y^T\hat{w}_1)v_1 + \alpha_{1,1}h_{2,1}v_2\big)/g_{2,1}.$

In general, we define

$$\hat{w}_{i+1} := (\hat{\mathbf{A}}\hat{w}_i - \sum_{j=1}^{i} g_{j,i}\hat{w}_j)/g_{i+1,i}. \tag{3.5}$$

Note that this is *not* the same as Gram-Schmidt orthonormalization.

Lemma 3.2.2 *The basis* $\{\hat{w}_1, \ldots, \hat{w}_{i+1}\}$ *for* $\hat{\mathbf{A}}\mathcal{K}_{i+1}(\hat{\mathbf{A}}, r_0)$ *described by (3.5) has* $(i+1)$*st basis vector of the form*

$$\hat{w}_{i+1} = w_{i+1} + \alpha_{1,i+1}v_1 + \ldots + \alpha_{i+1,i+1}v_{i+1}, \tag{3.6}$$

where

$$\alpha_{1,i+1} = (-\|b\|y^T\hat{w}_i + \sum_{k=1}^{i}\alpha_{k,i}h_{1,k} - \alpha_{1,k}g_{k,i})/g_{i+1,i}\,,$$

$$\alpha_{j,i+1} = (\sum_{k=j-1}^{i}\alpha_{k,i}h_{j,k} - \sum_{k=j}^{i}\alpha_{j,k}g_{k,i})/g_{i+1,i}\,, \quad 2 \le j \le i, \quad \text{and} \quad \alpha_{i+1,i+1} = \frac{\alpha_{i,i}h_{i+1,i}}{g_{i+1,i}}.$$

Proof: If $\hat{w}_i = w_i + \alpha_{1,i}v_1 + \ldots + \alpha_{i,i}v_i$, then

$$\hat{\mathbf{A}}\hat{w}_i = \sum_{j=1}^{i+1} g_{j,i}w_j + \sum_{j=1}^{i}\alpha_{j,i}\sum_{k=1}^{j+1}h_{k,j}v_k - by^T\hat{w}_i.$$

Hence, with the definition of (3.5),

$$g_{i+1,i}\hat{w}_{i+1} = \sum_{j=1}^{i+1}g_{j,i}w_j - \sum_{j=1}^{i}g_{j,i}\hat{w}_j + \sum_{j=2}^{i+1}v_j\sum_{k=j}^{i+1}\alpha_{k-1,i}h_{j,k-1} - \|b\|y^T\hat{w}_i v_1 + v_1\sum_{k=1}^{i}\alpha_{k,i}h_{1,k} =$$

$$g_{i+1,i}w_{i+1} - \sum_{j=1}^{i}g_{j,i}(\sum_{k=1}^{j}\alpha_{k,j}v_k) + \sum_{j=2}^{i+1}v_j\sum_{k=j-1}^{i}\alpha_{k,i}h_{j,k} - v_1(\|b\|y^T\hat{w}_i - \sum_{k=1}^{i}\alpha_{k,i}h_{1,k}) =$$

$$g_{i+1,i}w_{i+1} - \sum_{j=1}^{i}v_j\sum_{k=j}^{i}\alpha_{j,k}g_{k,i} + \sum_{j=2}^{i+1}v_j\sum_{k=j-1}^{i}\alpha_{k,i}h_{j,k} - v_1(\|b\|y^T\hat{w}_i - \sum_{k=1}^{i}\alpha_{k,i}h_{1,k}) =$$

$$g_{i+1,i}w_{i+1} - \sum_{j=2}^{i}v_j(\sum_{k=j}^{i}\alpha_{j,k}g_{k,i} - \sum_{k=j-1}^{i}\alpha_{k,i}h_{j,k}) + \alpha_{i,i}h_{i+1,i}v_{i+1}$$

$$-v_1(\|b\|y^T\hat{w}_i - \sum_{k=1}^{i}\alpha_{k,i}h_{1,k} + \sum_{k=1}^{i}\alpha_{1,k}g_{k,i}). \quad \Box$$

Thus elements of $\hat{\mathbf{A}}\mathcal{K}_k(\hat{\mathbf{A}}, r_0)$ can be expressed as linear combinations of orthonormal bases for $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ and $\mathcal{K}_k(\mathbf{A}, b)$.

A more interesting result is that by choice of the free parameter vector $y \in \mathbb{R}^n$, we can nearly make true the opposite: If we have given an element of the form

$$\sum_{i=1}^{k} \beta_i w_i + \sum_{i=1}^{k} \gamma_i v_i \in \text{span}(\mathbf{A}r_0, \dots, \mathbf{A}^k r_0, b, \dots, \mathbf{A}^{k-1} b)$$

with real coefficients β_i, γ_i. Then we consider

$$\sum_{i=1}^{k} \beta_i \hat{w}_i = \sum_{i=1}^{k} \beta_i (w_i + \sum_{j=1}^{i} \alpha_{j,i} v_j) = \sum_{i=1}^{k} \beta_i w_i + \sum_{i=1}^{k} v_i (\sum_{j=i}^{k} \beta_j \alpha_{i,j}),$$

because of (3.6). This expression equals $\sum_{i=1}^{k} \beta_i w_i + \sum_{i=1}^{k} \gamma_i v_i$ when the values $\alpha_{i,j}$ happen to be such that $\sum_{j=i}^{k} \beta_j \alpha_{i,j} = \gamma_i$ for all i. In matrix vector representation,

$$\begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & & \alpha_{1,k} \\ 0 & \alpha_{2,2} & & \alpha_{2,k} \\ & & \ddots & \\ 0 & & & \alpha_{k,k} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \\ \vdots \\ \beta_k \end{pmatrix} = \begin{pmatrix} \gamma_1 \\ \\ \vdots \\ \gamma_k \end{pmatrix}, \tag{3.7}$$

where the elements of the matrix are the unknowns ! But they are dependent the one on the other. For example, from $\alpha_{i+1,i+1} = \frac{\alpha_{i,i} h_{i+1,i}}{g_{i+1,i}}$ in Lemma 3.2.2 follows that

$$\alpha_{k,k} = \alpha_{1,1} \prod_{j=1}^{k-1} \frac{h_{j+1,j}}{g_{j+1,j}}$$

and similarly for the other diagonal elements. Fortunately, we can attribute $\alpha_{1,1}$ whatever value by choice of $y^T r_0$ due to (3.4), especially a value that makes $\alpha_{k,k}$ equal $\frac{\gamma_k}{\beta_k}$. The only case this does not work is when $\beta_k = 0$. In a similar way, all elements of other upper diagonals depend upon the first row element of the diagonal they belong to. We can solve our ,,linear system" (3.7) by diagonally defining the elements of the involved matrix: Having computed the value an element of the last column must take to solve the system, that is having put

$$\alpha_{k-i,k} = (\gamma_{k-i} - \sum_{j=k-i}^{k-1} \alpha_{k-i,j} \beta_j)/\beta_k, \quad i \geq 0,$$

we compute the values that follow for the other elements on that same diagonal, including the corresponding first row value. This last value is dependent on $y \in \mathbb{R}^n$ and thus yields a condition for y. From Lemma 3.2.2 we easily obtain the following recursion to diagonally define the unknowns of (3.7).

```
for $i = 0,\ k-2$
for $j = 0,\ k-i-2$
$\alpha_{k-i-j-1,k-j-1} = \alpha_{k-i-j,k-j} g_{k-j,k-j-1} + \sum_{m=k-i-j}^{k-j-1} \alpha_{k-i-j,m} g_{m,k-j-1} - \alpha_{m,k-j-1} h_{k-i-j,m}$
$\alpha_{k-i-j-1,k-j-1} = \frac{1}{h_{k-i-j,k-i-j-1}} \alpha_{k-i-j-1,k-j-1}$
enddo
enddo
```

With the $\alpha_{1,i}$ obtained by this recursion, the conditions to put on y are

$$y^T \hat{w}_i = \left(\sum_{k=1}^{i} (\alpha_{k,i} h_{1,k} - \alpha_{1,k} g_{k,i}) - \alpha_{1,i+1} g_{i+1,i} \right) / \|b\|, \qquad i \leq k-1, \tag{3.8}$$

$$y^T r_0 = \frac{-\alpha_{1,1} \|\mathbf{A} r_0\|}{\|b\|}, \qquad y^T x_0 = 0.$$

We have proved the following result.

Theorem 3.2.3 *Let $k \leq n-1$ be such that $\hat{\mathbf{A}}\mathcal{K}_{k-1}(\hat{\mathbf{A}}, r_0)$ has full dimension and r_0 as well as x_0 are excluded from this subspace and linearly independent from each other. Furthermore, let $\{v_1, \dots, v_k\}$ be an orthonormal basis of $\mathcal{K}_k(\mathbf{A}, b)$ and $\{w_1, \dots, w_k\}$ be an orthonormal basis of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$. We can choose $y \in \mathbb{R}$ such that an element of the form*

$$\sum_{i=1}^{k} \beta_i w_i + \sum_{i=1}^{k} \gamma_i v_i$$

with real coefficients β_i, γ_i and with $\beta_k \neq 0$, is also an element from $\hat{\mathbf{A}}\mathcal{K}_k(\hat{\mathbf{A}}, r_0)$.

The theorem states that when we minimize, by the choice of the parameter vector, residual norms of an auxiliary system with nonzero initial guess, we have the opportunity to create a system that finds during the kth GMRES iteration a residual norm that is equal to the distance from the initial residual to a subspace of maximal dimension $2k$, namely $\text{span}\{\mathbf{A}r_0, \dots, \mathbf{A}^k r_0, b, \dots, \mathbf{A}^{k-1} b\}$. In this sense, the theorem describes the possibilities as well as the restrictions of global minimization. It also partly explains what happens if we apply the strategies of the *preceding* chapter: During restarts of an auxiliary system with a number of prescribed residual norms, Krylov subspaces that are subspaces from $\text{span}\{\mathbf{A}r_0, \dots, \mathbf{A}^k r_0, b, \dots, \mathbf{A}^{k-1} b\}$ are being generated. The prescribed norms can, apart from elimination of convergence hampering properties of $\mathbf{A}$, produce favorable Krylov subspaces. They can spoil the convergence properties of projection spaces too, but the spaces $\mathcal{K}_k(\mathbf{A}, r_0)$ causing stagnation already, we do not expect the auxiliary spaces to be worse.

Concerning options to apply the last theorem to practice, we have postponed this item to the last section of the chapter.

3.3 Preconditioning with the Sherman-Morrison formula

All proposed applications of the Sherman-Morrison formula that we have seen so far work with the rank-one updated matrix $\mathbf{A} - by^T$. With this choice, the solution of the auxiliary system with right-hand side b suffices to compute the solution of the original problem, see (2.3). But this choice also implies troubles when the scalar $1 + y^T \hat{x}$ in (2.19) tends to 0. For that reason, we propose an alternative way to define the auxiliary system. The rank-one updated matrix is different, but the right-hand side remains. It avoids the singularity at $y^T \hat{x} = -1$ and moreover, the decreasing of residual norms of auxiliary and original system will go hand in hand.

In contrast with the preceding sections, let us define the auxiliary matrix $\hat{\mathbf{A}}$ as follows:

$$\hat{\mathbf{A}} := \mathbf{A} - \mathbf{A} d y^T, \quad y, d \in \mathbb{R}^n, \quad d \neq \mathbf{0}.$$

The first condition to put on y is

$$y^T d = \gamma, \tag{3.9}$$

for some scalar $\gamma \in \mathbb{R}$, $\gamma \neq 1$. Then we have

$$\hat{\mathbf{A}}d = (\mathbf{A} - \mathbf{A}dy^T)d = (1-\gamma)\mathbf{A}d.$$

Denoting by $\hat{x}$ the calculated approximation of the auxiliary system, the Sherman-Morrison formula yields

$$\mathbf{A}^{-1}b = (\hat{\mathbf{A}} + \mathbf{A}dy^T)^{-1}b = \hat{\mathbf{A}}^{-1}b - \hat{\mathbf{A}}^{-1}\mathbf{A}d(1 + y^T\hat{\mathbf{A}}^{-1}\mathbf{A}d)^{-1}y^T\hat{\mathbf{A}}^{-1}b \approx$$

$$\hat{x} - \frac{d}{1-\gamma}\left(1 + \frac{y^Td}{1-\gamma}\right)^{-1} y^T\hat{x} = \hat{x} - \frac{d}{1-\gamma}\left(\frac{1-\gamma+\gamma}{1-\gamma}\right)^{-1} y^T\hat{x} = \hat{x} - (y^T\hat{x})d =: \bar{x}. \tag{3.10}$$

The singularity at $y^Td = 1$ is excluded because of (3.9). Also, in comparison with the previous strategies, the error of the back-transformed solution is far less sensible to the error at the auxiliary system, because the expression $\hat{\mathbf{A}}^{-1}\mathbf{A}d$ is known to exactly equal $\frac{d}{1-\gamma}$. The residual vectors of original and auxiliary system even appear to be equal if we use (3.10) to compute the approximation of the first system:

$$b - \hat{\mathbf{A}}\hat{x} = b - (\mathbf{A} - \mathbf{A}dy^T)\hat{x} = b - \mathbf{A}\hat{x} + \mathbf{A}dy^T\hat{x} = b - \mathbf{A}(\hat{x} - (y^T\hat{x})d) = b - \mathbf{A}\bar{x}.$$

In fact, this application of the Sherman-Morrison formula can be seen as right preconditioning with the preconditioner

$$\mathbf{M} := \mathbf{I}_n - dy^T,$$

satisfying the restriction $y^Td \neq 1$. An approximate solution $\hat{z}$ of

$$\mathbf{AM}z = b$$

yields an approximation $\bar{x} = \mathbf{M}\hat{z} = \hat{z} - (y^T\hat{z})d$ of the unpreconditioned system with the same residual norm.

In analogy with the beginning of this chapter we formulate a process of minimization of residual norms by considering Givens sines. Again, the Krylov subspaces $\mathcal{K}_k(\mathbf{A}, r_0)$ and $\mathcal{K}_k(\hat{\mathbf{A}}, r_0)$ are in general not equal anymore and elements of the auxiliary Hessenberg matrix are not anymore independent from the parameter vector y. Instead, we have

$$\hat{h}_{j,k} = v_j^T\hat{\mathbf{A}}v_k = v_j^T(\mathbf{A} - \mathbf{A}dy^T)v_k = h_{j,k} - v_j^T\mathbf{A}d\alpha_k, \tag{3.11}$$

where $h_{j,k} := v_j^T\mathbf{A}v_k$ represents elements of the Hessenberg matrix of the original system and $\alpha_k := y^Tv_k$. The subdiagonal elements change to

$$\hat{h}_{k+1,k} = \|\hat{\mathbf{A}}v_k - \sum_{j=1}^{k}\hat{h}_{j,k}v_j\| = \|\mathbf{A}v_k - \sum_{j=1}^{k}h_{j,k}v_j + \left(\sum_{j=1}^{k}(v_j^T\mathbf{A}d)v_j - \mathbf{A}d\right)\alpha_k\|.$$

Minimizing the Givens sines from Corollary 2.2.6 amounts to maximizing

$$\frac{(\sum_{j=1}^{k}\hat{c}_{j-1}(\prod_{i=j}^{k-1}(-\hat{s}_i))\hat{h}_{j,k})^2}{\hat{h}_{k+1,k}^2}$$

With $w := \sum_{j=1}^{k}(v_j^T\mathbf{A}d)v_j - \mathbf{A}d$ and because of $(w, \mathbf{A}v_k - \sum_{j=1}^{k}h_{j,k}v_j) = (w, \mathbf{A}v_k)$, the denominator equals

$$\hat{h}_{k+1,k}^2 = \|\mathbf{A}v_k - \sum_{j=1}^{k}h_{j,k}v_j + \alpha_kw\|^2 = h_{k+1,k}^2 + 2\alpha_k(w, \mathbf{A}v_k) + \alpha_k^2\|w\|^2.$$

From (3.11) it can be seen that also the numerator consists of an expression that is quadratically dependent from the variable α_k. In other words, we have to maximize a value that can be expressed as

$$\frac{a_1\alpha_k^2 + a_2\alpha_k + a_3}{b_1\alpha_k^2 + b_2\alpha + b_3},$$

for some $a_i, b_i \in \mathbb{R}$, $1 \leq i \leq 3$. Straightforward computations show that

$$\frac{a_1\alpha_k^2 + a_2\alpha_k + a_3}{b_1\alpha_k^2 + b_2\alpha + b_3} = \frac{a_1}{b_1}\left(1 + \frac{(a_2b_1 - a_1b_2)\alpha_k + a_3b_1 - a_1b_3}{a_1b_1\alpha_k^2 + a_1b_2\alpha_k + a_1b_3}\right).$$

Extrema of this expression are the roots of the first derivative of the involved quotient.

$$\frac{\partial}{\partial\alpha_k} \quad \frac{(a_2b_1 - a_1b_2)\alpha_k + a_3b_1 - a_1b_3}{a_1b_1\alpha_k^2 + a_1b_2\alpha_k + a_1b_3} =$$

$$\frac{(a_2b_1 - a_1b_2)b_3 - (a_3b_1 - a_1b_3)b_2 - 2(a_3b_1 - a_1b_3)b_1\alpha_k - (a_2b_1 - a_1b_2)b_1\alpha_k^2}{a_1(b_1\alpha_k^2 + b_2\alpha_k + b_3)^2} = 0. \tag{3.12}$$

As far as $\hat{h}_{k+1,k}$ is dependent on α_k, it seems reasonable to minimize $\hat{s}_k$ and hence the corresponding residual norm by taking for α_k the smallest root of the equation above if $(a_2b_1 - b_2a_1)$ is negative and the largest root if $(a_2b_1 - a_1b_2)$ is positive. In the exceptional case that $\hat{h}_{k+1,k}$ is independent from α_k we can force whatever norm reducing (as in the preceding chapter).

In theory one could successively minimize all $\hat{s}_k$ with the help of (3.12) and define y by solving

$$y^T(v_1, \ldots, v_{n-1}, d) = (\alpha_1, \ldots, \alpha_{n-1}, \gamma),$$

which is merely a question of orthogonalizing d against $(v_1, \ldots, v_{n-1})$ and one matrix vector multiplication. The resulting convergence curve cannot be drawn arbitrarily, but it will at every step find steepest possible descent (in dependency of the chosen vector d). In practice, as we aim to avoid stagnation of the restarted GMRES(m) method, we will put only k, $k \leq m \ll n$, conditions on y. The corresponding algorithm could have the form of Algorithm 5.2.2, which we denote by PSHERMOR. In this version we have chosen $\gamma = 0$ and the auxiliary vector d to be equal to the actual approximation. In that way, with convergence, $\hat{\mathbf{A}}$ will tend to have the form $\mathbf{A} - by^T$ as was the case in the previous sections.

Example 2. Non-normal test matrix.

We used the same 100 × 100 matrix as in Example 1. Our initial guess is

$$x_0 = (3.44 \cdot 10^{-5}, \ldots, 3.44 \cdot 10^{-5}).$$

This guess minimizes $\|b - \rho\mathbf{A}(1, \ldots, 1)^T\|$ over all $\rho \in \mathbb{R}$, where $b = (1, \ldots, 1)^T$. Full GMRES converges quickly. GMRES(25) stagnates and stagnation can be overcome with PSHERMOR($25, k$), though minimization of the first Givens sine only is not sufficient to do so (PSHERMOR(m, k) denotes Algorithm 5.2.2 with restart parameter m and k sine minimizations at the beginning of every restart). We compare PSHERMOR with an other technique to accelerate restarted GMRES, with a deflation technique. The technique was proposed in Erhel [7]. At the ith restart of ERHEL($m, k_1, k_2, \ldots$) the system is preconditioned by right multiplication with a

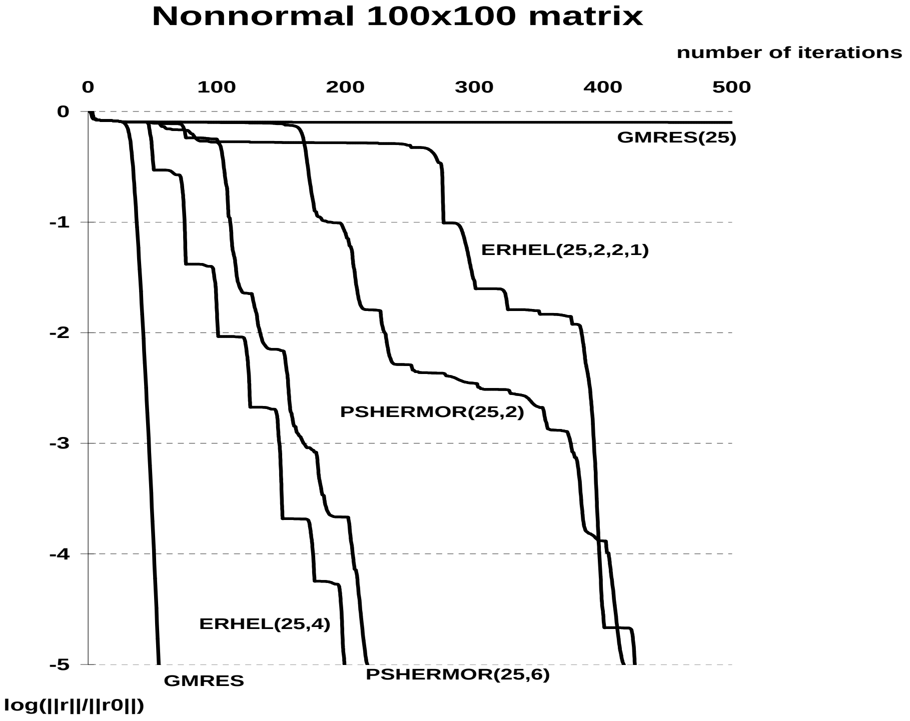

Figure 3.2: Comparison of GMRES, PSHERMOR and ERHEL for a non-normal test matrix.

matrix that tries to eliminate k_i eigenvalues from the spectrum of the original matrix. The construction of the preconditioner is based on addition of Ritz vectors (see also Chapter 1, Section 1.4). Erhel's method starts to converge when we add at least 4 Ritz vectors: ERHEL$(25, 4)$ converges a little faster than PSHERMOR$(25, 6)$. On the other hand, adding very few Ritz vectors at every restart seems to be less advantageous for this technique. ERHEL$(25, 2, 2, 1)$ has convergence speed comparable with PSHERMOR$(25, 2)$. The curves are shown in Figure 3.2.

Example 3. PDE stiffness matrix of dimension 10000.

This is the stiffness matrix resulting from discretization of (2.20) on a 100× 100 grid. It has 49600 nonzero elements, the right-hand side is $b = (9.803 \cdot 10^{-5}, \ldots, 9.803 \cdot 10^{-5})^T$ and we chose $x_0 = (0.01, \ldots, 0.01)^T$. The relatively large dimension, 10.000, seems to ask for large restart parameters before PSHERMOR becomes effective. When we restart after 60 steps the curves of Figure 3.3 show an interesting behaviour: During about 100 restarts the system appears to stagnate, but then all of a sudden the action of Givens sine minimization becomes visible. In this example we compare the influence of different Givens minimization numbers per restart: As is to be expected, only one Givens minimization at the beginning of every restart cycle yields slower convergence than 10 minimizations per cycle. Especially, the steep descent of the curve is postponed when using a smaller amount of minimizations. But this observation must be handled with care: Too many minimizations spoil the effect, as curve PSHERMOR(60,25) shows.

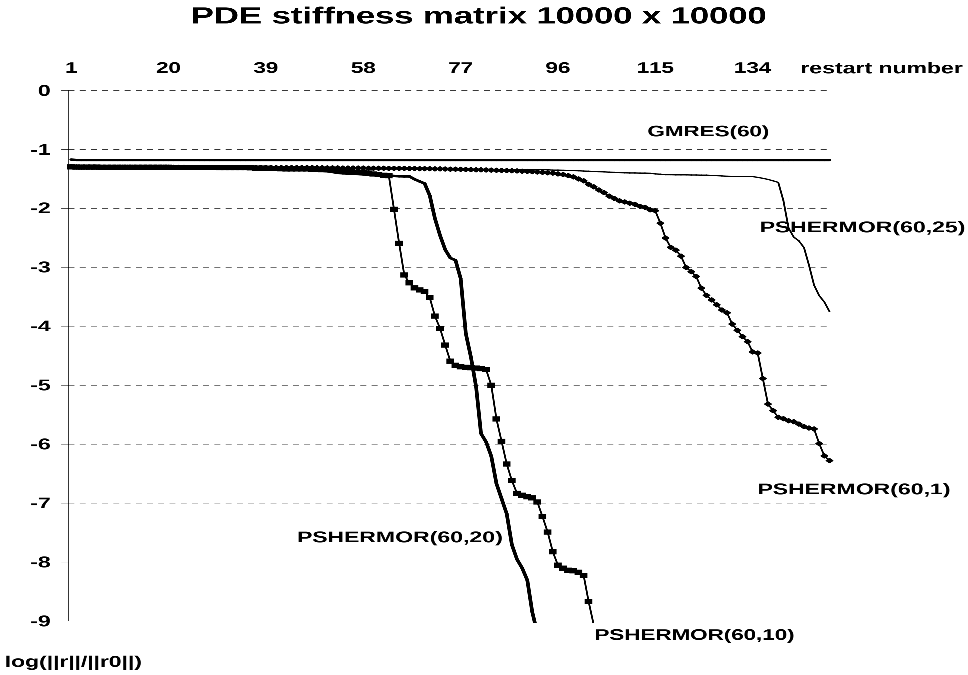

Figure 3.3: PSHERMOR for different numbers of minimizations

Example 4. Steam1 from Matrix Market.

This is an example where minimization does extremely well. It is more a curiosity than an indicative example. The matrix „Steam1", taken from the Matrix Market collection, has dimension 240 × 240, 3762 nonzero entries and its spectrum consists of 160 evenly distributed eigenvalues with norms ranging from 21711984 to 20918 and the last 80 eigenvalues lie evenly distributed between 19.565531 and -0.768583. With right-hand side $b = e_1$ and initial guess $x_0 = (0.01, \ldots, 0.01)^T$, the convergence curves of GMRES and PSHERMOR are displayed in Figure 3.4. For all experiments residual norm reduction of a factor 10^{-6} is not problematic at all (the ini-tial residual norm is relatively large, $\|r_0\| = 8407$), but further convergence appears to be a laborious task. In this example the eigenvalue distribution of Steam1 probably has a hampering influence on GMRES's behaviour. Stagnation of the restarted GMRES(m) starts to disappear for $m > 60$. But PSHERMOR with only one minimization reaches residual norm reduction of 10^{-8} about 5 times faster than full GMRES itself.

3.4 Open questions

An obvious item that should be investigated concerning the preceding procedure is the choice of the free vector $d \in \mathbb{R}^n$ in the update $\mathbf{A}dy^T$ and the scalar $\gamma \neq 1$ in (3.9). An inexpensive way to optimize these parameters could enhance the effectiveness of the preconditioning technique.

One has to realize that although this algorithm yields better results than Algorithm 5.2.5 from Section 3.1, the involved minimization is local in the sense we described in Section 3.1. We expect even better results when we manage to apply to

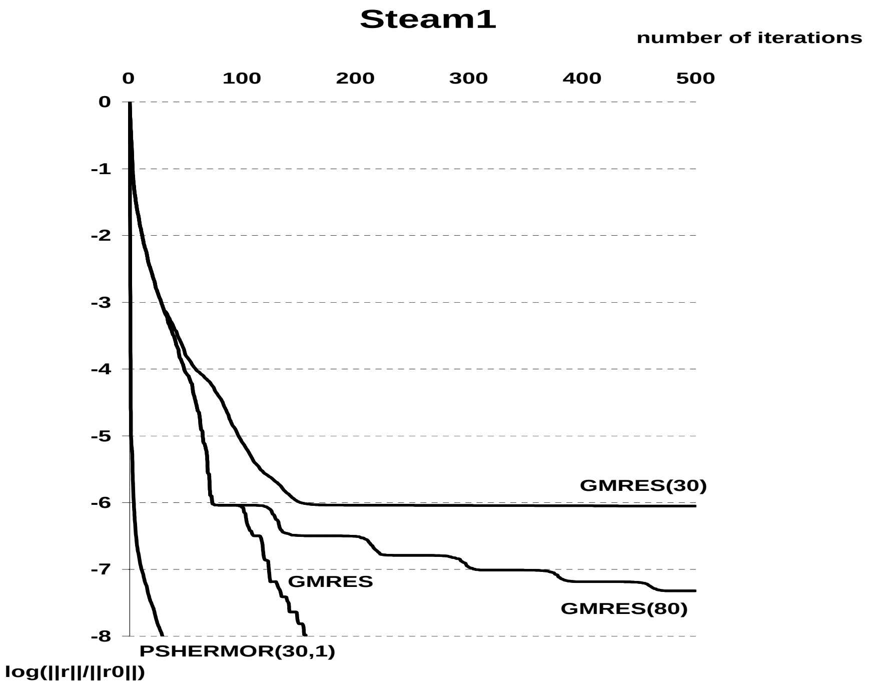

Figure 3.4: GMRES and PSHERMOR applied to the matrix ,,Steam1"

our preconditioning technique the theory of *global* minimization presented in Section 3.2. We outline an application of the theoretical results about global minimization in the remaining of this section.

Theorem 3.2.3 has shown we can minimize the distance from r_0 to $\hat{\mathbf{A}}\mathcal{K}_k(\hat{\mathbf{A}}, r_0)$ as follows: We calculate the projection of r_0 onto span$\{\mathbf{A}r_0, \dots, \mathbf{A}^k r_0, b, \dots, \mathbf{A}^{k-1}b\}$ and as long as the projection can be forced to belong to $\hat{\mathbf{A}}\mathcal{K}_k(\hat{\mathbf{A}}, r_0)$ by the choice of y, we calculate that y with the help of the conditions (3.8). Then GMRES applied to the auxiliary system with this special y finds at the kth step the projection we calculated and the kth residual norm has been implicitly minimized over a subspace of maximal dimension $2k$. In fact, it is not necessary to execute the first cycle of k auxiliary system iterations because we have computed the resulting residual before, during the projection on span$\{\mathbf{A}r_0, \dots, \mathbf{A}^k r_0, b, \dots, \mathbf{A}^{k-1}b\}$. In order to obtain the iterates that belong to this residual inexpensively, we can exploit the following lemma.

Lemma 3.4.1 *Let $\{v_1, \dots, v_k\}$ be an orthonormal basis of $\mathcal{K}_k(\mathbf{A}, b)$ with Arnoldi decomposition $\mathbf{AV}_k = \mathbf{V}_{k+1}\tilde{\mathbf{H}}_k$ and $\{w_1, \dots, w_k\}$ be an orthonormal basis of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ with Arnoldi decomposition $\mathbf{AW}_k = \mathbf{W}_{k+1}\tilde{\mathbf{G}}_k$. Any element of the form*

$$\sum_{j=1}^{k} \beta_j w_j + \sum_{j=1}^{k} \gamma_j v_j$$

for some real coefficients β_j and γ_j can be written as

$$\sum_{j=1}^{k} \beta_j w_j + \sum_{j=1}^{k} \gamma_j v_j = \hat{\mathbf{A}} \left(\mu_0 r_0 + \sum_{j=1}^{k-1} (\mu_j w_j + \nu_j v_j) \right) \tag{3.13}$$

if $y \in \mathbb{R}^n$ is chosen such that $y^T x_0 = 0$ and

$$y^T \left(\mu_0 r_0 + \sum_{j=1}^{k-1} (\mu_j w_j + \nu_j v_j) \right) = -\nu_0. \tag{3.14}$$

The coefficients μ_j and ν_j, $0 \leq j \leq k-1$, are recursively defined through

$$\mu_j = \frac{\beta_{j+1} - \sum_{i=j+1}^{k-1} \mu_i g_{j+1,i}}{g_{j+1,j}}, \qquad \nu_j = \frac{\gamma_{j+1} - \sum_{i=j+1}^{k-1} \nu_i h_{j+1,i}}{h_{j+1,j}},$$

where $g_{1,0} := \|\mathbf{A} r_0\|$ and $h_{1,0} := \|b\|$.

Proof: With the notations introduced above we have

$$\mu_0 \mathbf{A} r_0 + \hat{\mathbf{A}} \sum_{j=1}^{k-1} \mu_j w_j = \mu_0 g_{1,0} w_1 + \mathbf{A} \sum_{j=1}^{k-1} \mu_j w_j - b \sum_{j=1}^{k-1} \mu_j y^T w_j$$

$$= \mu_0 g_{1,0} w_1 + \sum_{j=1}^{k-1} \mu_j \sum_{i=1}^{j+1} g_{i,j} w_i - b \sum_{j=1}^{k-1} \mu_j y^T w_j = \sum_{i=1}^{k} w_i \sum_{j=i-1}^{k-1} \mu_j g_{i,j} - b \sum_{j=1}^{k-1} \mu_j y^T w_j$$

$$= \sum_{i=1}^{k} w_i (\mu_{i-1} g_{i,i-1} + \sum_{j=i}^{k-1} \mu_j g_{i,j}) - b \sum_{j=1}^{k-1} \mu_j y^T w_j = \sum_{i=1}^{k} \beta_i w_i - b \sum_{j=1}^{k-1} \mu_j y^T w_j.$$

Similarly,

$$\nu_0 b + \hat{\mathbf{A}} \sum_{j=1}^{k-1} \nu_j v_j = \sum_{i=1}^{k} \gamma_i v_i - b \sum_{j=1}^{k-1} \nu_j y^T v_j.$$

Hence

$$\hat{\mathbf{A}} \left(\mu_0 r_0 + \sum_{j=1}^{k-1} (\mu_j w_j + \nu_j v_j) \right) + \nu_0 b =$$

$$\sum_{j=1}^{k} \beta_j w_j - b \sum_{j=1}^{k-1} \mu_j y^T w_j + \sum_{j=1}^{k} \gamma_j v_j - b \sum_{j=1}^{k-1} \nu_j y^T v_j - \mu_0 y^T r_0 b.$$

For an $y \in \mathbb{R}^n$ satisfying (3.14) the scalars before b vanish. □

A sketch of the resulting globally minimizing algorithm is the following procedure:

1. Compute $(v_1, \ldots, v_k)$, an orthonormal basis of $\mathcal{K}_k(\mathbf{A}, b)$ with Arnoldi decomposition
$$\mathbf{A}(v_1, \ldots, v_k) = (v_1, \ldots, v_{k+1}) \tilde{\mathbf{H}}_k,$$
where $v_1 = b/\|b\|$.
Compute an orthonormal basis $(w_1, \ldots, w_k)$ of $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ with Arnoldi decomposition
$$\mathbf{A}(w_1, \ldots, w_k) = (w_1, \ldots, w_{k+1}) \tilde{\mathbf{G}}_k,$$
where $w_1 = \mathbf{A} r_0 / \|\mathbf{A} r_0\|$.

2. In order to minimize the kth residual norm, we project r_0 onto $\mathrm{span}(\mathbf{A}r_0, \dots, \mathbf{A}^k r_0, b, \dots, \mathbf{A}^{k-1}b)$: If the projection is given by $\mathbf{P}r_0 = \sum_{j=1}^k \gamma_j v_j + \sum_{j=1}^k \beta_j w_j$, then we obtain the conditions

$$v_i^T(r_0 - \sum_{j=1}^k \gamma_j v_j - \sum_{j=1}^k \beta_j w_j) = r_0^T v_i - \gamma_i - \sum_{j=1}^k \beta_j w_j^T v_i = 0, \quad 1 \le i \le k,$$

and

$$w_i^T(r_0 - \sum_{j=1}^k \gamma_j v_j - \sum_{j=1}^k \beta_j w_j) = r_0^T w_i - \beta_i - \sum_{j=1}^k \gamma_j w_i^T v_j = 0, \quad 1 \le i \le k.$$

With the abbreviations

$$\gamma := \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_k \end{pmatrix}, \quad \beta := \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}, \quad r_0^T v := \begin{pmatrix} r_0^T v_1 \\ \vdots \\ r_0^T v_k \end{pmatrix}, \quad r_0^T w := \begin{pmatrix} r_0^T w_1 \\ \vdots \\ r_0^T w_k \end{pmatrix},$$

and

$$(\mathbf{B})_{i,j} := v_i^T w_j, \quad 1 \le i, j \le k,$$

these conditions can be written as the system of linear equations

$$\begin{pmatrix} \mathbf{I}_k & \mathbf{B} \\ \mathbf{B}^T & \mathbf{I}_k \end{pmatrix} \begin{pmatrix} \gamma \\ \beta \end{pmatrix} = \begin{pmatrix} r_0^T v \\ r_0^T w \end{pmatrix}.$$

Equivalently,

$$\begin{pmatrix} 0 & \mathbf{B}^T\mathbf{B} - \mathbf{I}_k \\ \mathbf{B}^T & \mathbf{I}_k \end{pmatrix} \begin{pmatrix} \gamma \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{B}^T r_0^T v - r_0^T w \\ r_0^T w \end{pmatrix},$$

which means that we have to solve 2 systems of dimension k,

$$\mathbf{B}^T\mathbf{B}\beta - \beta = \mathbf{B}^T r_0^T v - r_0^T w, \quad \mathbf{B}^T \gamma = r_0^T w - \beta.$$

3. The orthogonal projection of r_0 onto $\mathrm{span}\{\mathbf{A}r_0, \dots, \mathbf{A}^k r_0, b, \dots, \mathbf{A}^{k-1}b\}$ is given by

$$\mathbf{P}r_0 = \sum_{j=1}^k \beta_j w_j + \sum_{j=1}^k \gamma_j v_j$$

As long as $\beta_k \neq 0$, it is possible to calculate with (3.8) an y such, that

$$\mathbf{P}r_0 \in \hat{\mathbf{A}}\mathcal{K}_k(\hat{\mathbf{A}}, r_0).$$

4. When we add to y condition (3.14), the calculated kth residual equals

$$\hat{r}_k = r_0 - \mathbf{P}r_0 = b - \mathbf{A}x_0 - \sum_{j=1}^k \beta_j w_j - \sum_{j=1}^k \gamma_j v_j = b - \hat{\mathbf{A}} \left(x_0 + \mu_0 r_0 + \sum_{j=1}^{k-1} (\mu_j w_j + \nu_j v_j) \right),$$

hence the kth iterate of the auxiliary system is given by

$$\hat{x}_k = x_0 + \mu_0 r_0 + \sum_{j=1}^{k-1} (\mu_j w_j + \nu_j v_j)$$

and the iterate of the original system by

$$x_k = \frac{\hat{x}_k}{1 + y^T \hat{x}_k}.$$

Thus the new matrix $\mathbf{A} - by^T$ builds a Krylov subspace that is, given the nonzero initial guess x_0, after k iterations as close to r_0 as possible. The distance results from projection on a space of maximal dimension $2k$. Of course, the expenses to achieve this are comparable with $2k$ classical GMRES iterations, but one expects that *restarting* with this auxiliary matrix is less susceptible to stagnation than with $\mathbf{A}$. Would stagnation occur nevertheless, then one can apply the same algorithm to $\mathbf{A} - by^T$ to define a second parameter vector y' and continue with $\mathbf{A} - b(y + y')^T$. Note that in that case the basis $\{v_1, \ldots, v_k\}$ needs not be computed again. On the other hand, the separate orthonormal bases for $\mathbf{A}\mathcal{K}_k(\mathbf{A}, r_0)$ and $\mathcal{K}_k(\mathbf{A}, b)$ are certainly numerically not as stable as a unit orthonormal basis for $\text{span}\{\mathbf{A}r_0, \ldots, \mathbf{A}^k r_0, b, \ldots, \mathbf{A}^{k-1}b\}$. The construction of a basis for the auxiliary Krylov subspace $\hat{\mathbf{A}}\mathcal{K}_k(\hat{\mathbf{A}}, r_0)$ presented in Section 3.2 (see (3.5)) was based on theoretical considerations and is also susceptible to become unstable in practice. Clearly, the above algorithm makes only sense if $x_0 \neq 0$.

Chapter 4

Spectral properties of the updated matrix

Most of the techniques to accelerate convergence of the restarted GMRES method that we considered in the first chapter are based on the idea that the eigenvalue distribution of the system matrix has an influence on the convergence of GMRES, as is the case for the Conjugate Gradient method for symmetric positive definite matrices. In particular, it is assumed that eigenvalues lying comparatively close to the origin hamper convergence and the mentioned techniques try, in some way or another, to eliminate unwanted eigenvalues from the spectrum. Though many well-known bounds for GMRES generated residual norms do in fact refer to the spectrum (see, for example, Saad [61]) and in many cases the eigenvalue distribution does have an influence on the residual norms, it is also known this is not generally true. In Greenbaum, Pták, Strakoš [31] it has even been proved one can construct for whatever spectrum and whatever convergence curve a linear system (1.1) such, that $\mathbf{A}$ has that specific spectrum and GMRES applied to (1.1) yields the prescribed convergence curve.

Nevertheless, nonsymmetric linear systems whose convergence behaviour is immediately connected with the spectrum of the involved matrix can arise in practical applications. This chapter applies to problems where we know a priori that the spectrum of the system has a crucial influence on convergence speed. For these cases it is worth investigating possibilities to achieve spectral deflation with the help of the Sherman-Morrison formula and rank-one update of $\mathbf{A}$. We design two new deflation techniques, test them on such cases and compare them with a known deflation technique. We computed eigenvalues and -vectors with the help of an implementation of the QZ method. Conversely, we try to gain information about the spectrum of a given updated matrix, for example an auxiliary matrix that is obtained by one of the techniques from the preceding chapters.

4.1 Prescription of spectra

As we have just mentioned, it has been proved that one can construct for whatever spectrum and whatever convergence curve a linear system (1.1) such, that $\mathbf{A}$ has that specific spectrum and GMRES applied to (1.1) yields the prescribed convergence curve. In the second chapter we showed that the rank-one update of our auxiliary system can be chosen such that the system belongs to the class of systems with a given convergence curve. We can also prove the spectral part of the specialization of the result from Greenbaum, Pták and Strakoš to our rank-one updated systems.

4.1.1 Any spectrum is possible for $\mathbf{A} - by^T$

Theorem 4.1.1 *Let the Krylov subspace $\mathcal{K}_n(\mathbf{A}, b)$ have dimension n and $\{\theta_1, \ldots, \theta_n\}$ be a set of complex values. Then there exists a vector $y \in \mathbb{R}^n$ such that $\mathbf{A} - by^T$ has the eigenvalues $\theta_1, \ldots, \theta_n$.*

Proof: If

$$\prod_{i=1}^{n}(\lambda - \theta_i) = \lambda^n + \alpha_{n-1}\lambda^{n-1} + \ldots + \alpha_1\lambda + \alpha_0,$$

for scalars $\alpha_i \in \mathbb{R}$, then the theorem is proved when we show that $\mathbf{A} - by^T$ is similar to

$$\begin{pmatrix} 0 & \ldots & 0 & -\alpha_0 \\ 1 & 0 & \ldots & -\alpha_1 \\ \vdots & \ddots & & \vdots \\ 0 & \ldots & 1 & -\alpha_{n-1} \end{pmatrix}$$

for some $y \in \mathbb{R}^n$. We will construct a similarity transformation represented by the matrix $\mathbf{X} = (x_1, \ldots, x_m)$. Let the first column x_1 of $\mathbf{X}$ be the vector b. Because $\mathbf{X}$ must satisfy

$$(\mathbf{A} - by^T)\mathbf{X} = \mathbf{X}\begin{pmatrix} 0 & \ldots & 0 & -\alpha_0 \\ 1 & 0 & \ldots & -\alpha_1 \\ \vdots & \ddots & & \vdots \\ 0 & \ldots & 1 & -\alpha_{n-1} \end{pmatrix}, \tag{4.1}$$

we have to define x_2 as $(\mathbf{A} - by^T)x_1$. Thus $x_2 = \mathbf{A}b - b\gamma_1$ when we denote (y^Tb) with γ_1. Similarly, $x_3 = (\mathbf{A} - by^T)^2x_1 = \mathbf{A}^2b - \mathbf{A}b\gamma - b\gamma_2$ with $\gamma_2 := y^Tx_2$ and we can in this manner continue until we have

$$x_n = (\mathbf{A} - by^T)^{n-1}b = \mathbf{A}^{n-1}b - \mathbf{A}^{n-2}b\gamma_1 - \ldots - \mathbf{A}b\gamma_{n-2} - b\gamma_{n-1}, \quad \gamma_i = y^Tx_i.$$

Note that for all $y \in \mathbb{R}^n$ the space $\mathcal{K}_n(\mathbf{A} - by^T, b)$ has dimension n because by assumption $\mathcal{K}_n(\mathbf{A}, b)$ has. Thus the vectors $x_1, \ldots, x_n$ span $\mathbb{R}^n$ and $\mathbf{X}$ is a nonsingular matrix and (4.1) holds except for the last column. The last column of the right side is given by

$$\alpha_0x_1 - \alpha_1x_2 - \ldots - \alpha_{n-1}x_n =$$

$$-\alpha_0b - \alpha_1\left(\mathbf{A}b - b\gamma_1\right) - \alpha_{n-1}\left(\mathbf{A}^{n-1}b - \mathbf{A}^{n-2}b\gamma_1 - \ldots - \mathbf{A}b\gamma_{n-2} - b\gamma_{n-1}\right).$$

On the left side we have

$$(\mathbf{A} - by^T)x_n = \mathbf{A}^nb - \mathbf{A}^{n-1}x_1\gamma_1 \ldots - \mathbf{A}^2b\gamma_{n-2} - \mathbf{A}b\gamma_{n-1} - b\gamma_n) =$$

$$\sum_{i=0}^{n-1}\left(\beta_i - \gamma_{n-i}\right)\mathbf{A}^ib,$$

when we write $\mathbf{A}^nb$ in the basis $\{b, \ldots, \mathbf{A}^{n-1}b\}$ as $\sum_{i=0}^{n-1}\beta_i\mathbf{A}^ib$. Hence equality of the two sides follows if we can find values $\gamma_1, \ldots, \gamma_n$ with

$$\begin{pmatrix} 1 & \alpha_{n-1} & \ldots & \alpha_1 \\ 0 & 1 & \ddots & \\ & & \ddots & \alpha_{n-1} \\ 0 & \ldots & & 1 \end{pmatrix}\begin{pmatrix} \gamma_n \\ \vdots \\ \gamma_2 \\ \gamma_1 \end{pmatrix} = \begin{pmatrix} \alpha_0 + \beta_0 \\ \vdots \\ \alpha_{n-2} + \beta_{n-2} \\ \alpha_{n-1} + \beta_{n-1} \end{pmatrix}.$$

Trivially, this system has exactly one solution and the obtained values $\gamma_1, \dots, \gamma_n$ yield a matrix $\mathbf{X}$ that satisfies (4.1). Due to the non-singularity of $\mathbf{X}$, we can put

$$y := \mathbf{X}^{-T} \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{n-1} \\ \gamma_n \end{pmatrix}.$$

□

Clearly, application of this result to the system matrix $\mathbf{A}$ of (1.1) is not feasible for large n. But we can ,,project" our matrix to its restriction on a Krylov subspace of small dimension and consider the Hessenberg matrix that resulted from the Arnoldi (or Householder) process. Fortunately, this Hessenberg matrix is a rank-one updated matrix too when the process has zero initial guess. By Proposition 2.2.1, the Hessenberg matrix of the auxiliary system equals

$$\hat{\mathbf{H}}_k = \mathbf{H}_k - e_1 z^T, \quad z = \|b\| \mathbf{V}_k^T y, \tag{4.2}$$

where the columns of $\mathbf{V}_k$ are the generated Arnoldi vectors. We know from the preceding theorem that any spectrum can be forced by the choice of z when $\mathcal{K}_k(\mathbf{H}_k, e_1)$ has full dimension. Methods that try to improve the spectrum during GMRES computations (Erhel [7], Calvetti [3]), have shown that the eigenvalues of the Hessenberg matrix $\mathbf{H}_k$ from the related Arnoldi process can provide useable approximations to the eigenvalues of $\mathbf{A}$. Moreover, elimination of the smallest eigenvalues of $\mathbf{H}_k$ can successfully remove the smallest eigenvalues of $\mathbf{A}$. Thus it is worth to investigating options to modify certain convergence hampering eigenvalues of $\mathbf{H}_k$ by exploiting $\hat{\mathbf{H}}_k$. In principle one can even replace the total spectrum of $\mathbf{H}_k$, but we restrict ourselves to elimination of the smallest eigenvalue or when this value is complex, the smallest complex eigenvalue pair.

4.1.2 Initial cycle deflation

Let the eigenvalue-eigenvector pairs of $\mathbf{H}_k$ be given by

$$(\theta_j, c_j), \quad 1 \leq j \leq k, \tag{4.3}$$

and be ordered such that

$$|\theta_j| \geq |\theta_{j+1}|, \quad 1 \leq j \leq k-1.$$

Then the Ritz vectors $\mathbf{V}_k c_j$, $1 \leq j \leq k$, approximate the eigenvectors of $\mathbf{A}$. This can be seen from the Arnoldi decomposition

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_k \mathbf{H}_k + \tilde{v}_{k+1} e_k^T,$$

where $\tilde{v}_{k+1}$ is the unscaled $(k+1)$st Arnoldi vector. Hence

$$\mathbf{A}\mathbf{V}_k c_j = (\mathbf{V}_k \mathbf{H}_k + \tilde{v}_{k+1} e_k^T) c_j = \theta_j \mathbf{V}_k c_j + \tilde{v}_{k+1} e_k^T c_j \tag{4.4}$$

and the quality of the Ritz vector depends upon $\|\tilde{v}_{k+1}\| |e_k^T c_j|$. Even so does the quality of the Ritz value θ_j as eigenvalue for $\mathbf{A}$ depend on this value.

If we assume θ_k is real, we can easily force all eigenvalues of $\hat{\mathbf{H}}_k$ but the smallest one, θ_k, to be equal to the those of $\mathbf{H}_k$ by imposing the conditions

$$y^T \mathbf{V}_k c_j = 0, \quad 1 \leq j \leq k-1.$$

The spectrum $(\hat{\theta}_1, \ldots, \hat{\theta}_k)$ of $\hat{\mathbf{H}}_k$ satisfies the trace equation

$$\sum_{j=1}^{k} \hat{\theta}_j = \sum_{j=1}^{k} \hat{h}_{j,j},$$

and because all eigenvalues are fixed except for the last one, this last one follows from the only diagonal element that is dependent from y, the value $\hat{h}_{1,1}$, and can be chosen arbitrarily. To force a prescribed Ritz value $\hat{\theta}_k$, $\hat{h}_{1,1}$ must satisfy the condition

$$\hat{h}_{1,1} = h_{1,1} - \|b\| y^T v_1 = \sum_{j=1}^{k-1} \theta_j + \hat{\theta}_k - \sum_{j=2}^{k} h_{j,j},$$

which is achieved by appropriate choice of $y^T v_1$. Not only are the ,,good" eigenvalues of $\mathbf{H}_k$ preserved in $\hat{\mathbf{H}}_k$, they also remain practically unchanged with respect to $\mathbf{A} - by^T$: The approximate eigenvector of $\mathbf{A}$, the Ritz vector corresponding to θ_j, is $\mathbf{V}_k c_j$. For $j \leq k-1$ this is a Ritz vector of $\mathbf{A} - by^T$ too, because

$$(\mathbf{A} - by^T)\mathbf{V}_k c_j \approx \theta_j \mathbf{V}_k c_j - b(y^T \mathbf{V}_k c_j) = \theta_j \mathbf{V}_k c_j.$$

In case θ_k is complex the situation is a little more complicated. We can only modify the real part of θ_k by the procedure that we just described. For full prescription of the value of $\hat{\theta}_k$ we need Corollary 4.1.2 and we change $\bar{\theta}_k$ simultaneously. The corollary is nothing else but a specialization of Theorem 4.1.1 for Hessenberg matrices.

Corollary 4.1.2 *Let $\mathbf{H} \in \mathbb{R}^{k \times k}$ be a nonsingular Hessenberg matrix with nonzero subdiagonal elements and $\{\theta_1, \ldots, \theta_k\}$ be a set of complex values. Then a vector $z \in \mathbb{R}^k$ exists such that the rank-one updated Hessenberg matrix $\mathbf{H} - e_1 z^T$ has eigenvalues $\theta_1, \ldots, \theta_k$.*

Proof: Let us denote the matrix $\mathbf{H} - e_1 z^T$ by $\hat{\mathbf{H}}$ and the elements that differ from those of $\mathbf{H}$ by $\hat{h}_{1,j}$. If

$$\prod_{i=1}^{k} (\lambda - \theta_i) = \lambda^k + \alpha_{k-1}\lambda^{k-1} + \ldots + \alpha_1 \lambda + \alpha_0,$$

for scalars $\alpha_i \in \mathbb{R}$, then the corollary is proved when we show that $\hat{\mathbf{H}}$ is similar to

$$\begin{pmatrix} 0 & \ldots & 0 & -\alpha_0 \\ 1 & 0 & \ldots & -\alpha_1 \\ \vdots & \ddots & & \vdots \\ 0 & \ldots & 1 & -\alpha_{k-1} \end{pmatrix}$$

for some $z \in \mathbb{R}^k$. We will construct a similarity transformation represented by the matrix $\mathbf{X} = (x_1, \ldots, x_k)$. Let the first column x_1 of $\mathbf{X}$ be the first unit vector e_1. Because $\mathbf{X}$ must satisfy

$$\mathbf{HX} = \mathbf{X} \begin{pmatrix} 0 & \ldots & 0 & -\alpha_0 \\ 1 & 0 & \ldots & -\alpha_1 \\ \vdots & \ddots & & \vdots \\ 0 & \ldots & 1 & -\alpha_{k-1} \end{pmatrix}, \tag{4.5}$$

we have to define x_2 as $\mathbf{H}x_1$. Thus

$$x_2 = \begin{pmatrix} \hat{h}_{1,1} \\ h_{2,1} \\ 0 \\ \vdots \end{pmatrix}$$

and similarly

$$x_3 = \mathbf{H}x_2 = \begin{pmatrix} \hat{h}_{1,1}^2 + \hat{h}_{1,2}h_{2,1} \\ h_{2,1}\hat{h}_{1,1} + h_{2,2}h_{2,1} \\ h_{3,2}h_{2,1} \\ 0 \\ \vdots \end{pmatrix}.$$

We facilitate the expression of x_j, $j \geq 4$, by introducing the auxiliary values

$$g_j := \prod_{i=1}^{j-2} h_{i+1,i}(\sum_{i=2}^{j-1} h_{i,i}), \quad j \leq k$$

and

$$f_j := \prod_{i=1}^{j-3} h_{i+1,i} \left(\sum_{i=2}^{j-2} h_{i,i}^2 + \sum_{i,l=2,\, i<l}^{j-2} h_{i,i}h_{l,l} + \sum_{i=2}^{j-2} h_{i,i+1}h_{i+1,i} \right), \quad j \leq k.$$

Note none of these values depends upon $z \in \mathbb{R}^k$. For $j \geq 4$ straightforward computation shows we can write

$$x_j = \begin{pmatrix} * \\ \vdots \\ * \\ (\hat{h}_{1,1}^2 + \hat{h}_{1,2}h_{2,1} + \hat{h}_{1,1}\sum_{i=2}^{j-2} h_{i,i}) \prod_{i=1}^{j-3} h_{i+1,i} + f_j \\ \hat{h}_{1,1}\prod_{i=1}^{j-2} h_{i+1,i} + g_j \\ \prod_{i=1}^{j-1} h_{i+1,i} \\ 0 \\ \vdots \end{pmatrix},$$

where $*$ represents elements whose values do not matter. Having constructed x_1 until x_k, the matrix $\mathbf{X}$ is nonsingular due to the non-vanishing lower subdiagonal elements and equation (4.5) is satisfied except for the last column. By choice of $z \in \mathbb{R}^k$ we can force the last columns to coincide too: The element on the right-hand side of (4.5) on position (k,k) is independent from $z \in \mathbb{R}^k$, but its left side parallel equals

$$e_k^T \hat{\mathbf{H}} x_k = h_{k,k-1}(\hat{h}_{1,1} \prod_{i=1}^{k-2} h_{i+1,i} + g_k) + h_{k,k} \prod_{i=1}^{k-1} h_{i+1,i} \tag{4.6}$$

and can be modified by the choice of $\hat{h}_{1,1} = h_{1,1} - z_1$ because the scalar before $\hat{h}_{1,1}$ does not vanish. With the choice that gives the element on position (k,k) the wanted value, the element on position $(k-1,k)$ depends only on $\hat{h}_{1,2} = h_{1,2} - z_2$ and equals

$$e_{k-1}^T \hat{\mathbf{H}} x_k = h_{k-1,k-2} \left((\hat{h}_{1,1}^2 + \hat{h}_{1,2}h_{2,1} + \hat{h}_{1,1} \sum_{i=2}^{k-2} h_{i,i}) \prod_{i=1}^{k-3} h_{i+1,i} + f_k \right) +$$

$$h_{k-1,k-1}(\hat{h}_{1,1}\prod_{i=1}^{k-2} h_{i+1,i} + g_k) + h_{k-1,k}\prod_{i=1}^{k-1} h_{i+1,i}. \tag{4.7}$$

Again this value can be made to coincide with the corresponding element on the left side of (4.5) and one can continue until modifying the first row element. It is readily seen that the element on position $(k-i+1,k)$ depends on $\hat{h}_{1,1},\ldots,\hat{h}_{1,i}$ and that the dependency on $\hat{h}_{1,i}$ is linear with a non-vanishing scalar before the element due to the nonzero subdiagonal elements of $\hat{\mathbf{H}}$. □

The proof of the corollary has been formulated in such a way that we can easily derive from it an implementation that modifies the two smallest eigenvalues of the Hessenberg matrix $\mathbf{H}_k$ from (4.2). If we prescribe a complex eigenvalue pair θ_k, $\bar{\theta}_k$ by their real and complex part and leave the remaining eigenvalues unchanged, the characteristic polynomial of $\hat{\mathbf{H}}_k$ is

$$(\lambda^2 - 2\mathrm{Re}(\hat{\theta}_k)\lambda + \mathrm{Re}(\hat{\theta}_k)^2 + \mathrm{Im}(\hat{\theta}_k)^2)(\lambda^{k-2} + \alpha_{k-3}\lambda^{k-3} + \ldots + \alpha_0)$$

for some coefficients $\alpha_i \in \mathbb{R}$. As a consequence, the scalar before λ^{k-1} is given by $\alpha_{k-3} - 2\mathrm{Re}(\hat{\theta}_k) = -\sum_{i=1}^{k-2}\theta_i - 2\mathrm{Re}(\hat{\theta}_k)$, as straightforward computation of α_{k-3} shows. Similarly, we have

$$\alpha_{k-4} := \sum_{i,l=1,\, i<l}^{k-2} \theta_i\theta_l.$$

The scalar before λ^{k-2} then equals

$$\mathrm{Re}(\hat{\theta}_k)^2 + \mathrm{Im}(\hat{\theta}_k)^2 + 2\mathrm{Re}(\hat{\theta}_k)\sum_{i=1}^{k-2}\theta_i + \alpha_{k-4}.$$

We now obtain the value of $\hat{h}_{1,1}$ by forcing (4.6) to equal

$$\left(-2\mathrm{Re}(\hat{\theta}_k) - \sum_{i=1}^{k-2}\theta_i\right)\prod_{i=1}^{k-1} h_{i+1,i}.$$

It is not difficult to see that we achieve this by assigning $y^T v_1$ the value

$$y^T v_1 = \frac{\sum_{i=1}^{k} h_{i,i} - 2\mathrm{Re}(\hat{\theta}_k) - \sum_{i=1}^{k-2}\theta_i}{\beta} = \frac{2(\mathrm{Re}(\theta_k) - \mathrm{Re}(\hat{\theta}_k))}{\beta},$$

which is not surprising when we compare with the case $\theta_k \in \mathbb{R}$. With this value we can compute (4.7) and similarly obtain $\hat{h}_{1,2}$ by forcing (4.7) to equal

$$\left(\mathrm{Re}(\hat{\theta}_k)^2 + \mathrm{Im}(\hat{\theta}_k)^2 + 2\mathrm{Re}(\hat{\theta}_k)\sum_{i=1}^{k-2}\theta_i + \alpha_{k-4}\right)\prod_{i=1}^{k-2} h_{i+1,i} +$$

$$\left(-2\mathrm{Re}(\hat{\theta}_k) - \sum_{i=1}^{k-2}\theta_i\right)\left(\hat{h}_{1,1}\prod_{i=1}^{k-2} h_{i+1,i} + g_k\right).$$

This yields

$$y^T v_2 = \frac{1}{h_{2,1}\beta}\left(\alpha_{k-4} - (\hat{h}_{1,1}+1)\sum_{j=2}^{k} h_{j,j} - \sum_{i,l=2,\, i<l}^{k-1} h_{i,i}h_{l,l} + \sum_{i=1}^{k-1} h_{i,i+1}h_{i+1,i}\right).$$

Eigenvalue	**of $\mathbf{H}_{10}$**	**of $\hat{\mathbf{H}}_{10}$**	**of $\hat{\mathbf{H}}_{10}*$**
θ_1	98.772599	300	212 + 212i
θ_2	89.609416	300	212 − 212i
θ_3	71.44979	98.772599	98.772599
θ_4	50.023387	89.609416	89.609416
θ_5	34.983112	71.44979	71.44979
θ_6	23.434997	50.023387	50.023387
θ_7	12.759248	34.983112	34.983112
θ_8	8.471585	23.434997	23.434997
θ_9	1.656424 + 0.819941i	12.759248	12.759248
θ_{10}	1.656424 − 0.819941i	8.471585	8.471585

Table 4.1: Smallest Ritz values of a non-normal test matrix before and after rank-one update with DEFSHERMOR

With these values of $y^T v_1$ and $y^T v_2$ we have two conditions for y. The others are

$$y^T \mathbf{V}_k c_j = 0, \quad 1 \le j \le k-2.$$

A concrete y is best found by first searching for a vector $z \in \mathbb{R}^k$ with

$$\begin{pmatrix} e_1^T \\ e_2^T \\ c_1^T \\ \vdots \\ c_{k-2}^T \end{pmatrix} z = \begin{pmatrix} y^T v_1 \\ y^T v_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

and then putting $y := \mathbf{V}_k z$. Thus computations are kept as much as possible in the k-dimensional subspace. We have worked out one example where the smallest complex eigenvalue pair is modified.

Example 1. Non-normal test matrix.

This matrix $\mathbf{A}$ is the same we used in Example 1 of Chapter 3. It satisfies

$$\frac{\|\mathbf{A}\mathbf{A}^T - \mathbf{A}^T\mathbf{A}\|_F}{\|\mathbf{A}\|_F} = 102080.49.$$

Our right-hand side is $b = (1, \ldots, 1)$ and $x_0 = 0$. Though there is no extremely small eigenvalue, the broad spectrum $\{1, \ldots, 100\}$ and the high non-normality may cause stagnation of GMRES(10). Deflation according to Erhel [7] needs many eigenvalue modifications to overcome stagnation: Convergence starts to be apparent when we use all 10 Ritz-vectors in the first cycle, 9 other ones in the second and 4 during the third cycle, see the curve ERHEL(10,10,9,4). Deflation with less eigenvalue modifications were not able to overcome stagnation.

On the other hand, deflation after only the initial cycle according to Algorithm 5.2.4 can give faster convergence than the repeated deflation in ERHEL(10,10,9,4).

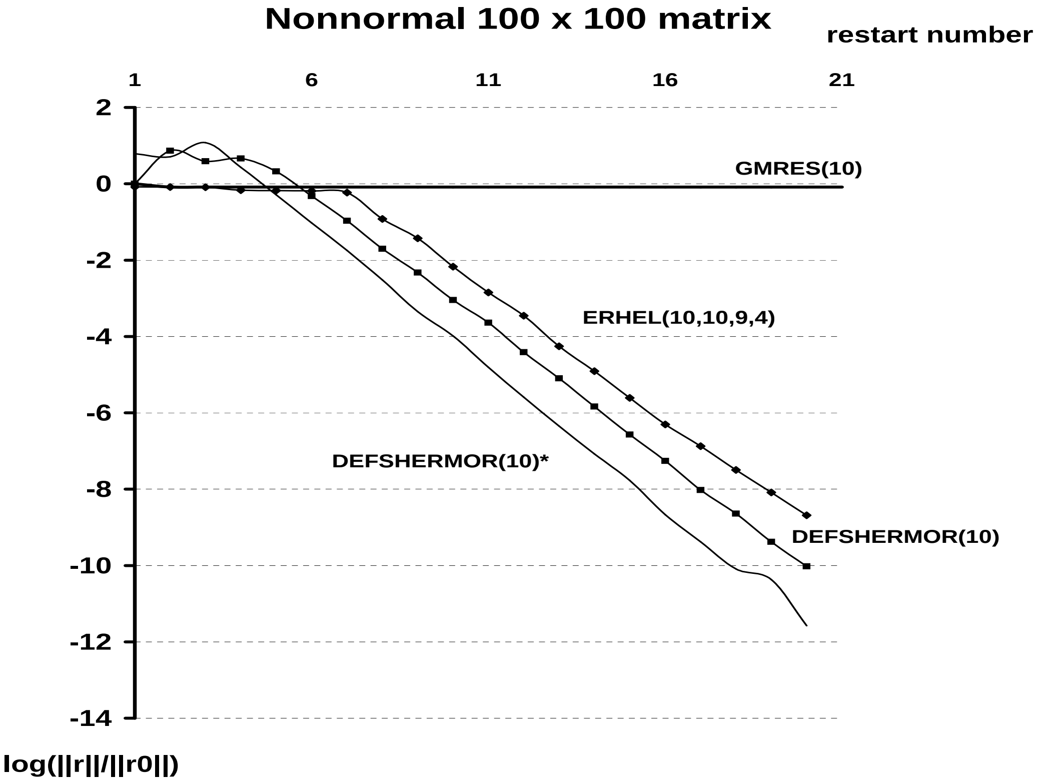

Figure 4.1: ERHEL, GMRES(10) and DEFSHERMOR

After the first cycle of 10 GMRES iterations, the Ritz values can be taken from the second column of Table 4.1. All Ritz values are real but for the complex pair $1.6564 \pm 0.8199i$. This pair badly approximates the eigenvalues 1 and 2, which causes probably the technique of Erhel not to be too effective in this case. By moving the pair to the other end of the spectrum of the Hessenberg matrix, we hope to increase the norms of the corresponding eigenvalues of $\mathbf{A} - by^T$ too. In addition, we can force the spectrum of the Hessenberg matrix to be completely real.

The best convergence is obtained when we move the small pair to be about three times as large as the Ritz value θ_1. When we choose it to equal 300, the convergence curve for the initial system after back-transformation is DEFSHERMOR(10) and can be seen below in Figure 4.1. The Ritz values for the auxiliary system after 10 iterations are displayed in the third column of Table 4.1. The 2 large Ritz va-lues approximate the largest eigenvalues of $\mathbf{A} - by^T$ very accurately, the remaining Ritz values are not such good approximations. The smallest eigenvalue of $\hat{\mathbf{A}}$ has become 1.698716. Alternatively, we moved the smallest Ritz value pair to a pair with approximately norm 300 too, but now it is the complex pair $-212 \pm 212i$. The corresponding curve is denoted by DEFSHERMOR(10)* and the first cycle Ritz values of the auxiliary system form the last column of Table 4.1. Again, the 2 large Ritz values approximate the largest eigenvalues of $\mathbf{A} - by^T$ very accurately and the remaining Ritz values are not such good approximations. But the smallest eigenvalue of $\hat{\mathbf{A}}$ is 2.833255, that is larger than in the preceding case. This probably causes this example to converge even faster.

In the sample example at the end of Chapter 5 we have applied Algorithm 5.2.4 to a Hessenberg matrix that has a completely real spectrum.

4.2 The spectrum of a given updated matrix

In this section we invert the problem of the preceding section and try to find useful relations between the spectrum of $\mathbf{A}$ and the spectrum of $\hat{\mathbf{A}} = \mathbf{A} - by^T$ for a given parameter vector $y \in \mathbb{R}^n$. Under such assumptions as diagonalizability of $\mathbf{A}$ at least theoretical information can be gained in a relatively simple manner. In Huhtanen, Nevanlinna [35] one finds spectral properties of $\hat{\mathbf{A}}$ that are related with the ones of $\mathbf{A}$ for the case where $\hat{\mathbf{A}}$ is normal and is a small rank perturbation of $\mathbf{A}$ but not necessarily a rank-one update.

For updated diagonal matrices spectral investigation is trivial and in addition useful for generalization to the case of diagonalizable matrices.

Lemma 4.2.1 *Let $\mathbf{D}$ =diag($d_1, \dots, d_n$) be a nonsingular diagonal matrix. The value d_j is an eigenvalue of $(\mathbf{D} - by^T)$ if and only if $e_j^T y = 0$ or $e_j^T b = 0$ or $d_j = d_i$ for some $i \neq j$.*

Proof: First, let $e_j^T y = 0$. Clearly, e_j is eigenvector of $\mathbf{D}$ corresponding to the eigenvalue λ_j. Thus

$$0 = (\mathbf{D} - d_j\mathbf{I})e_j = (\mathbf{D} - d_j\mathbf{I})e_j - b(y^T e_j),$$

hence

$$(\mathbf{D} - by^T)e_j = d_j e_j.$$

Next, let $e_j^T b$ equal zero. Then

$$0 = (\mathbf{D} - d_j\mathbf{I})e_j = (\mathbf{D} - d_j\mathbf{I})e_j - y(b^T e_j)$$

and therefore d_j is eigenvalue from $\mathbf{D} - yb^T$. The spectrum from $\mathbf{D} - yb^T$ equals the spectrum of its transposed matrix $\mathbf{D} - by^T$.

Finally, let $d_j = d_i$, $j \neq i$. If $e_j^T b = 0$ the result follows from the foregoing, otherwise we can execute a Givens rotation $\mathbf{G}^T$ that works on the ith and jth rows and zeroes out $e_j^T b$. Due to $d_i = d_j$ we have

$$\mathbf{G}^T\mathbf{D}\mathbf{G} = \begin{pmatrix} \mathbf{I} & & & & \\ & c & & s & \\ & & \mathbf{I} & & \\ & -s & & c & \\ & & & & \mathbf{I} \end{pmatrix} \begin{pmatrix} \ddots & & & & \\ & d_i & & & \\ & & \ddots & & \\ & & & d_j & \\ & & & & \ddots \end{pmatrix} \begin{pmatrix} \mathbf{I} & & & & \\ & c & & -s & \\ & & \mathbf{I} & & \\ & s & & c & \\ & & & & \mathbf{I} \end{pmatrix}$$

$$= \begin{pmatrix} \ddots & & & & \\ & c^2 d_i + s^2 d_j & & -scd_i + scd_j & \\ & & \ddots & & \\ & -csd_i + csd_j & & s^2 d_i + c^2 d_j & \\ & & & & \ddots \end{pmatrix} = \mathbf{D},$$

where c, s denote the Givens parameters that zero out $e_j^T b$. Hence

$$\mathbf{G}^T(\mathbf{D} - by^T)\mathbf{G} = \mathbf{D} - \mathbf{G}^T by^T \mathbf{G}$$

and $e_j^T(\mathbf{G}^T b) = 0$. By the previous case, d_j is an eigenvalue of $\mathbf{D} - \mathbf{G}^T by^T \mathbf{G}$ and therefore also of $\mathbf{D} - by^T$.
To prove the other direction, let us assume none of the previous cases occur and $d_j \in \sigma(\mathbf{D} - by^T)$. Then for some $v \in \mathbb{R}^n$

$$(\mathbf{D} - by^T)v = d_j v.$$

Thus

$$0 = e_j^T((\mathbf{D} - d_j\mathbf{I})v - b(y^Tv)) = -(y^Tv)e_j^T b.$$

Hence y^Tv must be zero and $\mathbf{D}v = d_j v$. By assumption d_j is a single eigenvalue of $\mathbf{D}$ and e_j is the corresponding unit eigenvector. Therefore $v \in \text{span}\{e_j\}$ and $0 = y^Tv = \alpha e_j^T y$ for some $\alpha \neq 0$, a contradiction. □

Theorem 4.2.2 *Let* $\mathbf{D}$ *=diag*$(d_1, \ldots, d_n)$ *be a nonsingular diagonal matrix with* $d_1 > \ldots > d_n$ *and let* b *and* y *have no zero elements. Then the roots of the rational function*

$$f(\lambda) = 1 - y^T(\mathbf{D} - \lambda\mathbf{I})^{-1}b$$

are the eigenvalues of $\mathbf{D} - by^T$.

Proof: Let λ be an eigenvalue of $\mathbf{D} - by^T$ with eigenvector $v \in \mathbb{R}^n$. Then

$$(\mathbf{D} - \lambda\mathbf{I})v - b(y^Tv) = 0.$$

By Lemma 4.2.1, $\det(\mathbf{D} - \lambda\mathbf{I}) \neq 0$. Multiplication with $y^T(\mathbf{D} - \lambda\mathbf{I})^{-1}$ yields

$$y^Tv(1 - y^T(\mathbf{D} - \lambda\mathbf{I})^{-1}b) = 0.$$

If $y^Tv = 0$ then λ is also eigenvalue of $\mathbf{D}$, which contradicts Lemma 4.2.1. Thus we obtain

$$1 - y^T(\mathbf{D} - \lambda\mathbf{I})^{-1}b = 0$$

whenever $\lambda \in \sigma(\mathbf{D} - by^T)$. On the other hand,

$$f(\lambda) := 1 - y^T(\mathbf{D} - \lambda\mathbf{I})^{-1}b =$$

$$\frac{\prod_{i=1}^n (d_i - \lambda) - y_1 b_1 \prod_{i\neq 1}^n (d_i - \lambda) - y_2 b_2 \prod_{i\neq 2}^n (d_i - \lambda) - \ldots - y_n b_n \prod_{i=1}^{n-1}(d_i - \lambda)}{\prod_{i=1}^n (d_i - \lambda)} \tag{4.8}$$

has exactly n (complex) roots. □
The result from this theorem can also be obtained by straightforward computation of the characteristic polynomial, which is relatively simple because $\mathbf{D}$ is diagonal. Clearly, the characteristic polynomial of $\mathbf{D} - by^T$ is given by

$$\prod_{i=1}^n (d_i - \lambda) - y_1 b_1 \prod_{i=2}^n (d_i - \lambda) - y_2 b_2 \prod_{i=1,\, i\neq 2}^n (d_i - \lambda) - \ldots - y_n b_n \prod_{i=1}^{n-1} (d_i - \lambda) \tag{4.9}$$

As a corollary from Theorem 4.2.2 we obtain

Corollary 4.2.3 *Let* $\mathbf{A}$ *be diagonalizable with* $\mathbf{A} = \mathbf{SDS}^{-1}$ *where* $\mathbf{D}$ *is diagonal with distinct eigenvalues. If* $\mathbf{S}^{-1}b$ *and* $\mathbf{S}^T y$ *have no zero elements, the eigenvalues of* $\mathbf{A} - by^T$ *are the roots of the function*

$$1 - y^T(\mathbf{A} - \lambda\mathbf{I})^{-1}b.$$

Proof: The spectrum of $\mathbf{A} - by^T$ equals the spectrum of

$$\mathbf{S}^{-1}(\mathbf{A} - by^T)\mathbf{S} = \mathbf{D} - \mathbf{S}^{-1}by^T\mathbf{S}.$$

By Theorem 4.2.2 the eigenvalues of $\mathbf{D} - \mathbf{S}^{-1}by^T\mathbf{S}$ are given by the roots of

$$1 - y^T\mathbf{S}(\mathbf{D} - \lambda\mathbf{I})^{-1}\mathbf{S}^{-1}b = 1 - y^T(\mathbf{S}(\mathbf{D} - \lambda\mathbf{I})\mathbf{S}^{-1})^{-1}b = 1 - y^T(\mathbf{A} - \lambda\mathbf{I})^{-1}b. \quad \square$$

This result may be useful for the theoretical investigation of $\mathbf{A} - by^T$ when spectral information of $\mathbf{A}$ is available and the parameter vector y is given.

4.2.1 Deflation with nearly normal matrices

In practice direct modification of the eigenvalues of $\mathbf{A}$ with the preceding corollary is impossible as long as we do not have the eigenvalue basis given by the columns of $\mathbf{S}$. Computation of this basis, as a well-known matter of fact, is not feasible for large n and in addition the basis can be very badly conditioned. Thus we cannot predict what elements of $\mathbf{S}^T y$ and $\mathbf{S}^{-1}b$ are susceptible to be zero.

But if $\mathbf{A}$ is normal, a *unitary* eigenvalue basis, expressed by the columns of $\mathbf{S}$, exists and the eigenvalue basis is more simple to handle than in general. For this case we can derive the following procedure.

After the k-th iteration of a GMRES process applied to a matrix $\mathbf{A}$ with $\mathbf{AS} = \mathbf{SD}$ where $\mathbf{D}$ is diagonal and $\mathbf{S}$ is unitary, let the eigenvalue eigenvector pairs of the Hessenberg matrix $\mathbf{H}_k$ from the Arnoldi process be given by (4.3). Let us assume that in equation (4.4)

$$\|\tilde{v}_{k+1}\||e_k^T c_k| \leq \|\mathbf{H}_k\|\varepsilon \tag{4.10}$$

is satisfied for some small $\varepsilon > 0$ and that the eigenvector c_k of $\mathbf{H}_k$ is real. Then we can consider the pair $(\theta_k, \mathbf{V}_k c_k)$ a good approximation to an eigenvalue eigenvector pair of $\mathbf{A}$ and this pair must be real too. We will denote it by (d_k, s_k). If we choose y to be a multiple of the normed Ritz vector corresponding to θ_k, i.e.

$$y := \rho\mathbf{V}_k c_k / \|\mathbf{V}_k c_k\| = \frac{\rho}{\|c_k\|}\mathbf{V}_k c_k$$

for some $\rho \in \mathbb{R}$, then $y \approx \rho s_k$ and $\mathbf{S}^H y \approx \rho e_k$. Thus

$$\sigma(\mathbf{A} - by^T) = \sigma(\mathbf{D} - \mathbf{S}^{-1}by^T\mathbf{S}) \approx \sigma(\mathbf{D} - \rho\mathbf{S}^H b e_k^T)$$

and the last spectrum is, with (4.9), given by the roots of

$$(d_k - \lambda - \rho(\mathbf{S}^H b)_k) \prod_{i=1,\, i\neq k}^{n} (d_i - \lambda). \tag{4.11}$$

As well as for $\mathbf{S}^H y$, we have a good approximation of $e_k^T\mathbf{S}^H b$ to our disposal, namely $e_k^T\mathbf{S}^H b = s_k^T b \approx c_k^T\mathbf{V}_k^T b/\|c_k\|$. As long as this value is not 0, we can modify the one eigenvalue d_k of $\mathbf{A}$, which is approximated by θ_k, by choice of ρ. Note that with a zero initial guess we have $c_k^T\mathbf{V}_k^T b/\|c_k\| = \beta e_1^T c_k/\|c_k\|$.

When c_k is complex, however, then c_k and $\bar{c}_k$ approximate a pair of complex conjugated eigenvectors of $\mathbf{A}$. Let (4.10) hold and let the approximated eigenvalue eigenvector pairs be denoted by (d_k, s_k) and $(\bar{d}_k, \bar{s}_k)$. When we define

$$y := \rho \mathbf{V}_k c_k / \|c_k\| + \delta \mathbf{V}_k \bar{c}_k / \|c_k\|, \qquad \rho, \delta \in \mathbb{C}$$

we obtain $\mathbf{S}^H y \approx (0, \ldots, 0, \delta, \rho, 0, \ldots, 0)^T$. But if we want to keep $\mathbf{A} - by^T$ real, then so must be y, thus we have to choose $\delta = \rho \in \mathbb{R}$ and hence $\mathbf{S}^H y \approx (0, \ldots, 0, \rho, \rho, 0, \ldots, 0)^T$. The characteristic polynomial of $\mathbf{A} - by^T$ is with this choice approximated by

$$\big((d_k - \lambda)(\bar{d}_k - \lambda) - \rho e_{k-1}^T \mathbf{S}^H b(\bar{d}_k - \lambda) - \rho e_k^T \mathbf{S}^H b(d_k - \lambda)\big) \prod_{i=1,\, i \neq k-1, k}^{n} (d_i - \lambda).$$

We can estimate the influence of the choice of ρ on the eigenvalues d_k and $\bar{d}_k$ as follows: Instead of the values $e_{k-1}^T \mathbf{S}^H b$ and $e_k^T \mathbf{S}^H b$ we will use

$$e_{k-1}^T \mathbf{S}^H b = s_k^H b = \bar{s}_k^T b \approx b^T \mathbf{V}_k \bar{c}_k / \|c_k\|, \quad e_k^T \mathbf{S}^H b = \bar{s}_k^H b = s_k{}^T b \approx b^T \mathbf{V}_k c_k / \|c_k\|.$$

Let us denote real and imaginary part of θ_k by $\theta_{kR} \in \mathbb{R}$ respectively $\theta_{kI} \in \mathbb{R}$ and real and imaginary part of $b^T \mathbf{V}_k c_k / \|c_k\|$ by $b_R \in \mathbb{R}$ respectively $b_I \in \mathbb{R}$. Then the part of the characteristic polynomial we modify is close to the polynomial

$$(\theta_k - \lambda)(\bar{\theta}_k - \lambda) - \rho(b_R - b_I i)(\bar{\theta}_k - \lambda) - \rho(b_R + b_I i)(\theta_k - \lambda) =$$

$$\lambda^2 + (2b_R\rho - 2\theta_{kR})\lambda + \theta_{kR}^2 + \theta_{kI}^2 - 2\rho(b_R\theta_{kR} + b_I\theta_{kI}).$$

The roots of this polynomial stay far from zero if the determinant of this expression is far from $(2b_R\rho - 2\theta_{kR})/2$, that is if $\theta_{kR}^2 + \theta_{kI}^2 - 2\rho(b_R\theta_{kR} + b_I\theta_{kI})$ is far from $(2b_R\rho - 2\theta_{kR})^2/4$. We have

$$\theta_{kR}^2 + \theta_{kI}^2 - 2\rho(b_R\theta_{kR} + b_I\theta_{kI}) = \frac{(b_R(2\rho) - 2\theta_{kR})^2}{4} \Leftrightarrow$$

$$-4\theta_{kI}^2 + 8\rho b_I \theta_{kI} + (2\rho)^2 (b_R)^2 = 0.$$

Large enough ρ will keep the modified eigenvalues far from zero, but a too large choice is suspicious to make $\mathbf{A} - by^T$ singular. Alternatively, one can use the value between the roots for which the eigenvalues vanish (as long as it is not a double root), that is

$$\rho := \frac{-\theta_{kI} b_I}{b_R^2}.$$

With the last considerations it is of course possible to estimate all the values that we can move the complex pair to and to approximate a specific value. But the quality of these estimations depends on the quality of the Ritz values expressed by (4.10).

This deflation process can be repeated after the first cycle, yielding a matrix

$$\mathbf{A} - b(y_1 + y_2 + \ldots + y_l)^T$$

which, in the ideal case of accurate Ritz vectors, modifies the l smallest eigenvalues of $\mathbf{A}$. One complication with repeated inflation is that this auxiliary matrix does not have to be normal anymore. The further it is from normality, the less will the above described strategy work. Another complication is that with approximations at the beginning of restarts not being zero anymore, we must guarantee that

$$\|b - (\mathbf{A} - b(y_1 + y_2 + \ldots + y_l)^T)x_k\| = \|b - (\mathbf{A} - b(y_1 + y_2 + \ldots + y_{l+1})^T)x_k\|,$$

Eigenvalue	**of A**	**of $\hat{\mathbf{A}}$**, $\rho = 100$	**of $\hat{\mathbf{A}}$**, $\rho = 10$	**of $\hat{\mathbf{A}}$**, $\rho = 0.1$
θ_1	1.7428	197.2352	19.4867	1.7454
θ_2	-0.1359 + 9.416i	1.7181	1.7155	-0.1359+ 9.416i
θ_3	-0.1359 - 9.416i	-0.1359 + 9.4175i	-0.1359 + 9.417i	-0.1359 - 9.416i
θ_4	-0.1359+ 9.258i	-0.1359 - 9.4175i	-0.1359 - 9.417i	-0.1359+ 9.2589i
θ_5	-0.1359 - 9.258i	-0.1359 + 9.258i	-0.1359 + 9.2589i	-0.1359 - 9.2589i
θ_{396}	-0.0809 + 0.061i	-0.1132	-0.1132	-0.1131
θ_{397}	-0.0809 - 0.061i	0.10956	0.1095	0.1095
θ_{398}	-0.0747+ 0.0303i	-0.0809 + 0.061i	-0.0809 + 0.061i	-0.0809 + 0.061i
θ_{399}	-0.0747 - 0.0303i	-0.0809 - 0.061i	-0.0809 - 0.061i	-0.0809 - 0.061i
θ_{400}	0.0735	0.0735	0.0735	0.0735

Table 4.2: Extreme eigenvalues of PDE matrix before and after rank-one update with Algorithm 5.2.3

hence $y_{l+1}^T x_k = 0$. This can be achieved by putting for example

$$y_{l+1} := \rho \mathbf{V}_k c_k / \|\mathbf{V}_k c_k\| + \delta \mathbf{V}_k c_1 / \|\mathbf{V}_k c_1\|.$$

Then $y_{l+1}^T x_k = 0$ implies

$$\delta = -\rho \frac{x_k^T \mathbf{V}_k c_k}{x_k^T \mathbf{V}_k c_1} \frac{\|\mathbf{V}_k c_1\|}{\|\mathbf{V}_k c_k\|}, \qquad \text{if} \quad x_k^T \mathbf{V}_k c_1 \neq 0.$$

We have chosen here to modify also the largest eigenvalue because linear combination of the two might yield new eigenvectors between the extreme ones. We assumed both first and largest Ritz vector are real. Generalization for the complex case along the same lines as we have just done for zero initial guesses is straightforward.

In the algorithm we constructed from this technique (Algorithm 5.2.3) we restricted ourselves to deflation during the initial cycle. We demonstrate its effectiveness with one example for a real smallest Ritz value and one example with a smallest complex pair of Ritz values.

Example 2. Deflation of the PDE stiffness matrix of dimension 400.

This is the same matrix as the one considered in Chapter 2, it results from discretization of (2.20) on a 20 × 20 grid, has 1920 nonzero elements and $b = (0.5, \ldots, 0.5)^T$ so that with $x_0 = 0$ we have $\|r_0\| = 10$. It is close to normal with

$$\frac{\|\mathbf{A}\mathbf{A}^T - \mathbf{A}^T\mathbf{A}\|_F}{\|\mathbf{A}\|_F} \approx 0.15726.$$

and has relatively small eigenvalues, the largest one has norm 1.74288, the remaining ones lie evenly distributed in the interval $[0.1, 1]$, except for the last three ones: One complex pair $\approx -0.074724 \pm 0.0303087i$ and the real eigenvalue ≈ 0.073544, see Table 4.2 . GMRES(25) stagnates. The smallest Ritz values after 25 iterations are a complex pair: $-0.078117 \pm 0.043132i$, fairly good approximations of the small

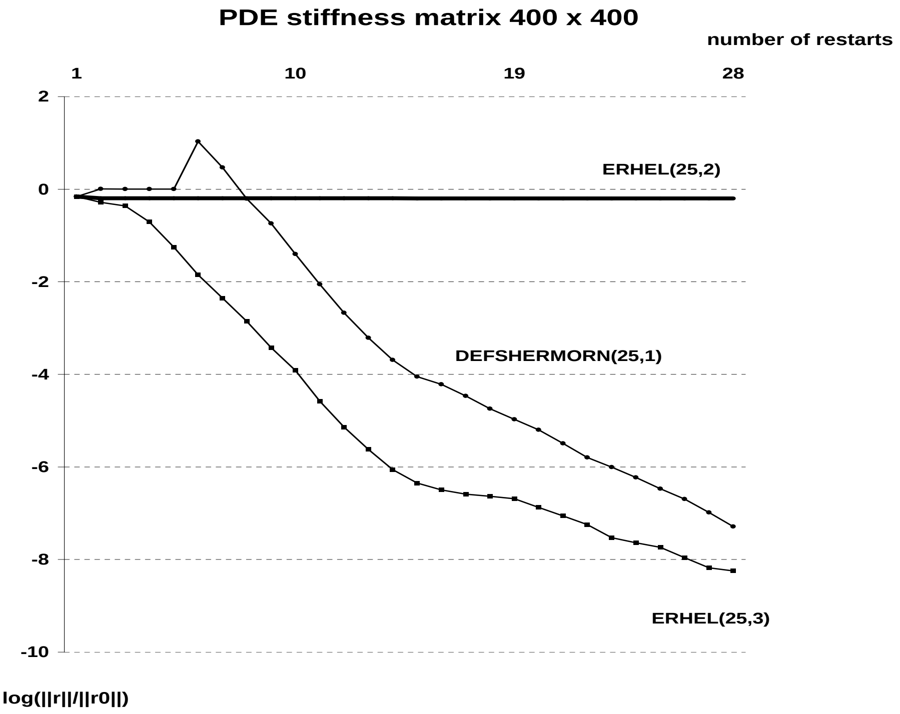

Figure 4.2: ERHEL, GMRES(25) and DEFSHERMORN

eigenvalue pair. With the DEFSHERMORN technique we can move these eigenvalues to different places in the spectrum. For the extreme large choice $\rho = 100$ (the Frobenius norm of $\mathbf{A}$ is about 9.9), we leave unchanged the single small eigenvalue ≈ 0.073544, but the second and third smallest eigenvalues are the fourth and fifth smallest of the original matrix, thus we have moved exactly the smallest complex pair. One eigenvalue of this pair has assumed the extreme large value 197.235, the second one is part of the rest of the spectrum lying in the interval $[0.1, 1]$. This is also seen in the third column of Table 4.2. For the more moderate choice $\rho = 10$, similar behaviour can be observed in the fourth column of Table 4.2. With $\rho = 0.1$ convergence is fastest, the larger values caused the denominator $1 + y^T \hat{x}_{25}$ of (2.19) at the end of restart cycles to be small and reduced the quality of the back-transformation. The curve DEFSHERMORN(25,1) in Figure 4.2 belongs to the choice $\rho = 0.1$.

Compared with another deflation technique, right preconditioning proposed in Erhel [7] (see also Chapter 1, Section 1.4 and Example 2 in Chapter 3), we notice that it behaves similarly. When executing 2 modifications, ERHEL(25,2) stagnates, whereas we needed only 2 modifications for the DEFSHERMORN curve. But ERHEL(25,3) is faster than DEFSHERMORN.

Example 3. Deflation of a bidiagonal matrix.

In this example the smallest Ritz values are not very good approximations to the smallest eigenvalues, but still deflation with Algorithm 5.2.3 works very well. It is an academic example that was constructed to test deflation techniques (i.g. Morgan uses it in [50]). The system matrix is nearly normal with

$$\frac{\|\mathbf{A}\mathbf{A}^T - \mathbf{A}^T\mathbf{A}\|_F}{\|\mathbf{A}\|_F} \approx 0.006.$$

It has diagonal entries 0.006, 0.008, 3, 4, ..., 100 and the upperdiagonal elements all equal 0.1. With $b = (1, \ldots, 1)^T$, the two very small eigenvalues 0.006 and 0.008 prevent GMRES(20) from converging. In case of applying Erhels deflative preconditioning, the first GMRES(20) cycle seems not to give satisfactory Ritz values to approximate the two smallest eigenvalues. At least ERHEL(20,2) is not able to modify them in order to overcome stagnation. Neither does successive eigenvalue modification help (ERHEL(20,1,1)). Attempts to remove three eigenvalues are necessary (see the curve ERHEL(20,2,1) in Figure 4.3), and still the order of removing is crucial (ERHEL(20,1,2) does not converge). When using Algorithm 5.2.3 with one eigenvalue modification, DEFSHERMORN(20,1), this is enough to overcome stagnation. The smallest Ritz value after 20 steps, 0.057647, is not a good approximation to the eigenvalue 0.008 at all, but still we can try to enlarge it. With $e_{99}^T\mathbf{S}^H b \approx -1.412554$ in (4.11), we have chosen $\rho = 2$ with the objective to create a second smallest eigenvalue ≈ 3, that is equally large as the third smallest eigenvalue. The result is curve DEFSHERMORN(20,1) in Figure (4.3). The one deflative step modified the 2 smallest eigenvalues to equal -0.039778 and 2.56045. The wanted eigenvalue 3 could not exactly be forced because of the poor quality of the initial Ritz values. On the other hand, the new smallest eigenvalue is quite well approximated by the new smallest Ritz value (-0.034988) and above all, the smallest eigenvalue has been considerably enlarged. The auxiliary matrix satisfies

$$\frac{\|\hat{\mathbf{A}}\hat{\mathbf{A}}^T - \hat{\mathbf{A}}^T\hat{\mathbf{A}}\|_F}{\|\hat{\mathbf{A}}\|_F} \approx 2.599856,$$

thus further deflation with the technique proposed above is susceptible to become less effective.

4.3 Open questions

Clearly, the two deflation techniques of this chapter have to be extended to application during an arbitrary number of restart cycles in order to become competitive with other techniques. We managed to prescribe initial Ritz values due to the simple form of the updated Hessenberg matrix. When the process is repeated auxiliary Hessenberg matrices are not anymore rank-one updates of the original Hessenberg matrices. But if arbitrary spectra can be achieved with rank-one update we expect it can be achieved with updates of a larger rank too. The concrete choice of the spectrum of the Hessenberg matrix of the initial cycle, which we can completely prescribe, is a question comparable with the choice of prescribed residual norms in the SHERMOR technique from Chapter 2. In fact, we have too many free parameters and do not know how to exploit them best.

As for the second technique, which is restricted to nearly normal matrices, generalization for repeated restart cycles does not seem to be complicated. But it will be a serious problem to keep the auxiliary matrix close to normal as the process proceeds.

Nevertheless, if one or both methods could be formulated for nonzero initial guesses in an attractive way, we would have to our disposal a deflation technique

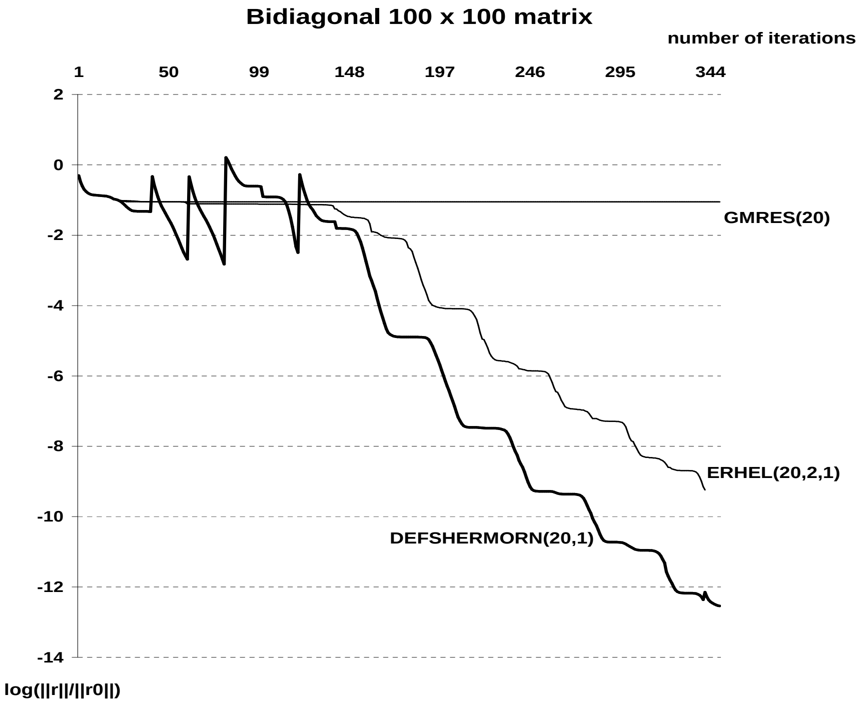

Figure 4.3: ERHEL, GMRES(20) and DEFSHERMORN

that enables inexpensive and precise eigenvalue modification. Of course, its effectiveness would depend on the quality of the Ritz values, but this is a problem all deflation methods deal with. On the other hand, our computations are kept as much as possible in the projected subspace of a small dimension say m, $m << n$, and multiplications of order n are restricted to one matrix vector product of dimension $n \times m$ per restart. This might be less expensive than the orthogonalization of n-dimensional vectors that other methods require.

Chapter 5

Algorithms and sample experiment

It is possible to see this chapter as an appendix of the previous ones. The first part presents the algorithms we referred to in the first chapter while describing full projection methods. In principle, they concern only the basis generating algorithm for the test and projection spaces of the corresponding methods. But in some cases these basis vectors turn out to be at the same time residuals. In addition, we prove some lemma's that are immediately connected with the algorithms and needed to complete the theory of the first chapter. For all lemma's we assume the involved Krylov subspaces have full dimension. In the second part we formulate implementations of several applications of the Sherman-Morrison formula from chapters two, three and four. They all consist of modifications of the GMRES method and we discuss the computational costs they add to this method. Finally, we apply the new algorithms to a unit problem from practice.

5.1 Basis generating algorithms

Algorithm 5.1.1 Modified Gram-Schmidt Orthogonalization

Initialization: Choose a vector v_1 *with* $\|v_1\| = 1$.

Calculus:

```
    vskip1pt        do $j = 1, m$
                        $\tilde{v}_{j+1} = \mathbf{A}v_j$
                        do $l = 1, j$
                                $h_{l,j} = v_l^T \tilde{v}_{j+1}$
                                $\tilde{v}_{j+1} = \tilde{v}_{j+1} - h_{l,j} v_l$
                        enddo
                        $\gamma_{j+1} = \|\tilde{v}_{j+1}\|$
                        if $\gamma_{j+1} = 0$ stop
                        $h_{j+1,j} = \gamma_{j+1}$
                        $v_{j+1} = \tilde{v}_{j+1}/\gamma_{j+1}$
                enddo
```

Algorithm 5.1.2 Householder reflection

Initialization: Choose a vector $v; z_1 = v$.

Calculus:

```
do  $j = 1, m+1$
        Compute the Householder unit vector $w_j$ satisfying:
            1) $e_i^T w_j = 0, \quad i = 1, \ldots, j-1$ and
            2) $e_i^T(\mathbf{P}_j z_j) = 0, \quad i = j+1, \ldots, n$, where $\mathbf{P}_j = \mathbf{I}_n - 2w_j w_j^T$
        $h_{j-1} = \mathbf{P}_j z_j$
        $v_j = \mathbf{P}_1 \mathbf{P}_2 \ldots \mathbf{P}_j e_j$
        if $\|v_j\| = 0$ stop
        if $j \leq m$ then compute $z_{j+1} = \mathbf{P}_j \mathbf{P}_{j-1} \ldots \mathbf{P}_1 \mathbf{A} v_j$
enddo
```

Lemma 5.1.3 *If $v_j \neq 0$ for $j \leq m$, then Algorithm 5.1.2 generates an orthonormal sequence $\{v_1, \ldots, v_{m+1}\}$ that spans $\mathcal{K}_{m+1}(\mathbf{A}, v)$ with Arnoldi decomposition (1.12) where $\mathbf{C}_m = \mathbf{V}_m$.*

Proof: If we put

$$\Pi_j = \mathbf{P}_j \mathbf{P}_{j-1} \ldots \mathbf{P}_1,$$

the algorithm implies the relation

$$h_j = \mathbf{P}_{j+1} z_{j+1} = \mathbf{P}_{j+1} \Pi_j \mathbf{A} v_j = \Pi_{j+1} \mathbf{A} v_j.$$

Hence

$$\mathbf{A} v_j = \Pi_{j+1}^T \sum_{i=1}^{j+1} h_{ij} e_i = \sum_{i=1}^{j+1} h_{ij} \Pi_{j+1}^T e_i$$

because Π_{j+1} is orthogonal as a product of Householder reflections. Since $\mathbf{P}_k e_i = e_i$ for $i < k$, we have

$$\Pi_{j+1}^T e_i = \mathbf{P}_1 \ldots \mathbf{P}_{j+1} e_i = v_i, \quad i \leq j+1.$$

This yields $\mathbf{A} v_j = \sum_{i=1}^{j+1} h_{ij} v_i$, for $j = 1, \ldots, m$, which can be written as (1.12) when $\tilde{\mathbf{H}}_m \in \mathbb{R}^{(m+1)\times m}$ consists of the $(m+1)$ upper rows of the matrix $(h_1, \ldots, h_m)$ and with $\mathbf{V}_m := (v_1, \ldots, v_m)$. From this decomposition it can easily be seen that the columns of $\mathbf{V}_{m+1}$ span $\mathcal{K}_{m+1}(\mathbf{A}, v)$. These columns have unit norm because $\|v_j\| = \|\mathbf{P}_1 \mathbf{P}_2 \ldots \mathbf{P}_j e_j\| = \|e_j\| = 1$ and orthogonality follows from

$$v_i^T v_j = (\mathbf{P}_1 \mathbf{P}_2 \ldots \mathbf{P}_i e_i)^T \mathbf{P}_1 \mathbf{P}_2 \ldots \mathbf{P}_j e_j = e_i^T \mathbf{P}_{i+1} \ldots \mathbf{P}_j e_j = e_i^T e_j = 0, \quad i < j,$$

because $e_i^T \mathbf{P}_l = e_i^T$ for $l > i$. □

Algorithm 5.1.4 Lanczos process for symmetric matrices

```
Initialization: Choose a vector $v_1$ with $\|v_1\| = 1$; $\gamma_1 = 0$; $v_0 = 0$.
Calculus:
        do  $j = 1, m$
            $\tilde{v}_{j+1} = \mathbf{A}v_j - \gamma_j v_{j-1}$
            $\alpha_j = \tilde{v}_{j+1}^T v_j$
            $\tilde{v}_{j+1} = \tilde{v}_{j+1} - \alpha_j v_j$
            $\gamma_{j+1} = \|\tilde{v}_{j+1}\|$
            if $\gamma_{j+1} = 0$ stop
            $v_{j+1} = \tilde{v}_{j+1}/\gamma_{j+1}$
        enddo
```

Lemma 5.1.5 *If $\gamma_{j+1} \neq 0$ for all $j \leq m$ and $\mathbf{A}^T = \mathbf{A}$ then Algorithm 5.1.4 generates an orthornormal basis $\{v_1, \ldots, v_{m+1}\}$ of $\mathcal{K}_{m+1}(\mathbf{A}, v_1)$ with Arnoldi decomposition (1.12) where $\mathbf{C}_m = \mathbf{V}_m$ and where the Hessenberg matrix is tridiagonal.*

Proof: The algorithm computes unit vectors v_{j+1} with

$$\gamma_{j+1} v_{j+1} = \mathbf{A}v_j - \gamma_j v_{j-1} - v_j^T(\mathbf{A}v_j - \gamma j v_{j-1})v_j.$$

For $j = 1$ we have

$$v_1^T \gamma_2 v_2 = v_1^T(\mathbf{A}v_1 - (v_1^T \mathbf{A}v_1)v_1) = 0 = v_0^T v_2.$$

Assuming $v_i^T v_l = \delta_{il}$ for $i, l \leq j$, where j is some integer smaller than m, we obtain

$$v_j^T \gamma_{j+1} v_{j+1} = v_j^T(\mathbf{A}v_j - \gamma_j v_{j-1} - (v_j^T \mathbf{A}v_j)v_j) = 0,$$

$$\begin{aligned} v_{j-1}^T \gamma_{j+1} v_{j+1} &= v_{j-1}^T(\mathbf{A}v_j - \gamma_j v_{j-1} - (v_j^T \mathbf{A}v_j)v_j) = v_j^T \mathbf{A}v_{j-1} - \gamma_j \\ &= v_j^T(\gamma_j v_j + \gamma_{j-1} v_{j-2} + (v_{j-1}^T \mathbf{A}v_{j-1})v_{j-1}) - \gamma_j = 0, \end{aligned}$$

and

$$v_i^T \gamma_{j+1} v_{j+1} = v_i^T(\mathbf{A}v_j - \gamma_j v_{j-1} - (v_j^T \mathbf{A}v_j)v_j) = v_j^T \mathbf{A}v_i = 0, \quad i \leq j-2$$

because $\mathbf{A}v_i \in$span$\{v_0, \ldots, v_{j-1}\}$. Thus the algorithm yields a decomposition

$$\mathbf{A}(v_1, \ldots, v_m) = (v_1, \ldots, v_{m+1})\tilde{\mathbf{T}}_m,$$

where $\tilde{\mathbf{T}}_m \in \mathbb{R}^{(m+1)\times m}$ is triangular with diagonal elements $v_i^T \mathbf{A}v_i$ and two identical subdiagonals with elements $\gamma_2, \ldots, \gamma_{m+1}$. This decomposition implies span$\{v_1, \ldots, v_{m+1}\} = \mathcal{K}_{m+1}(\mathbf{A}, v_1)$. □

Algorithm 5.1.6 Bi-orthogonal Lanczos process for nonsymmetric matrices

Initialization: Choose vectors v_1 *and* w_1 *with* $v_1^T w_1 = 1$; $\beta_1 = \delta_1 = 0$; $w_0 = v_0 = 0$.

Calculus:

```
do  $j = 1, m$
        $\alpha_j = w_j^T \mathbf{A} v_j$
        $\tilde{v}_{j+1} = \mathbf{A} v_j - \alpha_j v_j - \beta_j v_{j-1}$
        $\tilde{w}_{j+1} = \mathbf{A}^T w_j - \alpha_j w_j - \delta_j w_{j-1}$
        $\delta_{j+1} = \sqrt{|\tilde{v}_{j+1}^T \tilde{w}_{j+1}|}$
        if $\delta_{j+1} = 0$ stop
        $\beta_{j+1} = \tilde{v}_{j+1}^T \tilde{w}_{j+1} / \delta_{j+1}$
        $v_{j+1} = \tilde{v}_{j+1} / \delta_{j+1}$
        $w_{j+1} = \tilde{w}_{j+1} / \beta_{j+1}$
enddo
```

Lemma 5.1.7 *If* $\tilde{v}_{j+1}^T \tilde{w}_{j+1} \neq 0$ *for* $j \leq m$ *then the columns of* $\mathbf{V}_{m+1} := (v_1, \ldots, v_{m+1})$ *span* $\mathcal{K}_{m+1}(\mathbf{A}, v_1)$*, the columns of* $\mathbf{W}_{m+1} := (w_1, \ldots, w_{m+1})$ *span* $\mathcal{K}_{m+1}(\mathbf{A}^T, w_1)$ *and*

$$\mathbf{W}_{m+1}^T \mathbf{V}_{m+1} = \mathbf{I}_{m+1}.$$

Moreover, both $\mathbf{A}\mathbf{V}_m$ *and* $\mathbf{A}^T\mathbf{W}_m$ *can be decomposed according to (1.12) where* $\mathbf{C}_m = \mathbf{V}_m$*, respectively* $\mathbf{C}_m = \mathbf{W}_m$ *and where the Hessenberg matrices are tridiagonal.*

Proof: The algorithm yields decompositions

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1} \begin{pmatrix} & \mathbf{T}_m & \\ 0 & \ldots & \delta_{m+1} \end{pmatrix}$$

and

$$\mathbf{A}^T\mathbf{W}_m = \mathbf{W}_{m+1} \begin{pmatrix} & \mathbf{T}_m^T & \\ 0 & \ldots & \beta_{m+1} \end{pmatrix},$$

where

$$\mathbf{T}_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \delta_m & \alpha_m \end{pmatrix}.$$

From them it is clear that $\operatorname{span}\{v_1, \ldots, v_{m+1}\} = \mathcal{K}_{m+1}(\mathbf{A}, v_1)$ and $\operatorname{span}\{w_1, \ldots, w_{m+1}\} = \mathcal{K}_{m+1}(\mathbf{A}^T, w_1)$. Concerning bi-orthogonality we will proceed by induction:
Assuming that $v_l^T w_i = \delta_{li}$, $1 \leq l, i \leq j$, where δ_{li} denotes the Kronecker symbol (not to be confounded with the elements δ_i from the algorithm), we have

$$v_{j+1}^T w_j = \delta_{j+1}^{-1} \left(w_j^T \mathbf{A} v_j - \alpha_j w_j^T v_j - \beta_j w_j^T v_{j-1} \right) = \delta_{j+1}^{-1} \left(w_j^T \mathbf{A} v_j - (w_j^T \mathbf{A} v_j) w_j^T v_j \right) = 0.$$

Furthermore,

$$v_{j+1}^T w_i = \delta_{j+1}^{-1}((\mathbf{A}v_j)^T w_i - \alpha_j v_j^T w_i - \beta_j v_{j-1}^T w_i) =$$
$$\delta_{j+1}^{-1}(v_j^T \mathbf{A}^T w_i - \beta_j v_{j-1}^T w_i) = \delta_{j+1}^{-1}(v_j^T(\beta_{i+1} w_{i+1} + \alpha_i w_i + \delta_i w_{i-1}) - \beta_j v_{j-1}^T w_i),$$

an expression that vanishes for $i \leq j-2$ by assumption and for $i = j-1$ holds

$$v_{j+1}^T w_{j-1} = \delta_{j+1}^{-1}(v_j^T(\beta_j w_j + \alpha_{j-1}w_{j-1} + \delta_{j-1}w_{j-2}) - \beta_j v_{j-1}^T w_{j-1})$$

$$= \delta_{j+1}^{-1}(\beta_j v_j^T w_j - \beta_j v_{j-1}^T w_{j-1}) = 0.$$

In a similar way we can prove that $w_{j+1}^T v_i = 0$ for $i \leq j$. Finally, $v_{j+1}^T w_{j+1} = 1$ by construction. □

Lemma 5.1.8 *The residuals of the BCG method applied to first and dual system (see (1.33)), can be defined with short term recurrences as follows:*

Algorithm 5.1.9 *BCG residuals with two term recurrences*

- $\xi_j = r_j^T r_j^* / p_j^T \mathbf{A}^T p_j^*$
- $r_{j+1} = r_j - \xi_j \mathbf{A} p_j, \quad r_{j+1}^* = r_j^* - \xi_j \mathbf{A}^T p_j^*$
- $\phi_j = r_{j+1}^T r_{j+1}^* / r_j^T r_j^*$
- $p_{j+1} = r_{j+1} + \phi_j p_j, \quad p_{j+1}^* = r_{j+1}^* + \phi_j p_j^*$

Proof: Let us consider the BCG projection method described in chapter 1. If we generate bases with Algorithm 5.1.6 and apply Proposition 1.2.5, the procedure would consist of the following steps:

- Execute the jth step of the Bi-orthogonal Lanczos process (Algorithm 5.1.6), yielding δ_{j+1}, β_{j+1} and the basis vectors v_{j+1}, w_{j+1}. Compute also $\alpha_{j+1} = v_{j+1}^T \mathbf{A} w_{j+1}$.
- Define recursively the parameters connected with the LDU decomposition of $\mathbf{T}_{j+1}$ from Lemma 5.1.7 proposed in Proposition 1.2.5:
  $l_{j+1} = \delta_{j+1}/d_j$, $u_{j,j+1} = \beta_{j+1}/d_j$, $d_{j+1} = \alpha_{j+1} - \delta_{j+1}\beta_{j+1}/d_j$ and $\zeta_j = -\delta_{j+1}\zeta_{j-1}/d_{j+1}$.
- With the help of the auxiliary sequence $\{s_0, \ldots, s_j\}$ define the residual vector: $s_j = v_{j+1} - u_{j,j+1}s_{j-1}$,
  $r_{j+1} = r_j - \zeta_j \mathbf{A} s_j$.

In this algorithm the starting vector for the bi-orthogonalization process must be $v_1 := r_0$. Then the sequence of residual vectors is closely connected with the sequence $\{v_1, \ldots, v_{j+1}\}$. In fact, $r_j \in \mathcal{K}_{j+1}(\mathbf{A}, r_0)$ can be written as $r_j = \sum_{i=1}^{j+1} \alpha_i v_i$ for some $\alpha_i \in \mathbb{R}$, and the Galerkin condition of the BCG projector yields $\alpha_i = 0$ for all i but for $i = j+1$. Thus $r_j = \|r_j\| v_{j+1}$ and with $p_j := \|r_j\| s_j$, $p_0 := r_0$ the last point of the above recurrence becomes:

- With the help of the auxiliary sequence $\{p_0, \ldots, p_j\}$ define the residual vector: $p_j = r_j - u_{j,j+1}\frac{\|r_j\|}{\|r_{j-1}\|}p_{j-1}$,
  $r_{j+1} = r_j - \frac{\zeta_j}{\|r_j\|}\mathbf{A} p_j$.

Similarly, two-term recurrences for the projector $\wp_j^*$ of the dual system can be defined with the help of the Bi-orthogonal Lanczos process when we initialize with $w_1 := r_0^*$, the initial residual of the dual system. This projector projects onto $\mathbf{A}^T \mathcal{K}_j(\mathbf{A}^T, r_0^*)$, orthogonal to $\mathcal{K}_j(\mathbf{A}, r_0)$. When we extract from the LDU composition of $\mathbf{T}_{j+1}$ from Lemma 5.1.7 an auxiliary sequence $\{s_0^*, \ldots, s_j^*\}$ by putting $\mathbf{S}_j^* := \mathbf{W}_{j+1}\mathbf{L}_j^{-T}$, exploit $r_j^* = \|r_j^*\| w_{j+1}$, and define $p_j^* := \|r_j^*\| s_j^*$, $p_0^* := r_0^*$, the algorithm would be

- Execute the jth step of the Bi-orthogonal Lanczos process (Algorithm 5.1.6), yielding δ_{j+1}, β_{j+1} and the basis vectors v_{j+1}, w_{j+1}. Compute also $\alpha_{j+1} = v_{j+1}^T \mathbf{A} w_{j+1}$.
- Define recursively the parameters connected with the LDU decomposition of $\mathbf{T}_{j+1}$ from Lemma 5.1.7:
  $l_{j+1} = \delta_{j+1}/d_j$, $u_{j,j+1} = \beta_{j+1}/d_j$, $d_{j+1} = \alpha_{j+1} - \delta_{j+1}\beta_{j+1}/d_j$ and $\zeta_j^* = -\beta_{j+1}\zeta_{j-1}^*/d_{j+1}$.

- With the help of the auxiliary sequence $\{p_0^*,\dots,p_j^*\}$ define the residual vector: $p_j^* = r_j^* - l_{j+1}\frac{\|r_j^*\|}{\|r_{j-1}^*\|}p_{j-1}^*$,
  $r_{j+1}^* = r_j^* - \frac{\zeta_j^*}{\|r_j^*\|}\mathbf{A}^T p_j^*$.

The auxiliary sequences of first and dual system are related by the underlying LDU decomposition. With the abbreviations $\mathbf{P}_j = (p_0,\dots,p_j)$ and $\mathbf{P}_j^* = (p_0^*,\dots,p_j^*)$, $\mathbf{S}_j = (s_0,\dots,s_j)$ and $\mathbf{S}_j^* = (s_0^*,\dots,s_j^*)$, we obtain

$$\begin{aligned}(\mathbf{P}_j^*)^T\mathbf{A}\mathbf{P}_j = \operatorname{diag}(\|r_0^*\|,\dots,\|r_j^*\|)(\mathbf{S}_j^*)^T\mathbf{A}\mathbf{S}_j\operatorname{diag}(\|r_0\|,\dots,\|r_j\|) =\\ \operatorname{diag}(\|r_0^*\|,\dots,\|r_j^*\|)\mathbf{L}_{j+1}^{-1}\mathbf{W}_{j+1}^T\mathbf{A}\mathbf{V}_{j+1}\mathbf{U}_{j+1}^{-1}\operatorname{diag}(\|r_0\|,\dots,\|r_j\|) =\\ \operatorname{diag}(\|r_0^*\|,\dots,\|r_j^*\|)\mathbf{D}_{j+1}\operatorname{diag}(\|r_0\|,\dots,\|r_j\|),\end{aligned}$$

a diagonal matrix. Due to this „**A**-bi-orthogonality" property and the bi-orthogonality of the two residual vector sequences, we have

$$-\zeta_{j-1}l_{j+1}\|r_j^*\|d_j = -\frac{\zeta_{j-1}}{\|r_{j-1}\|}l_{j+1}\frac{\|r_j^*\|}{\|r_{j-1}^*\|}\|r_{j-1}^*\|d_j\|r_{j-1}\| =$$

$$-\frac{\zeta_{j-1}}{\|r_{j-1}\|}(p_j^* + l_{j+1}\frac{\|r_j^*\|}{\|r_{j-1}^*\|}p_{j-1}^*)^T\mathbf{A}p_{j-1} = (r_{j-1} - \frac{\zeta_{j-1}}{\|r_{j-1}\|}\mathbf{A}p_{j-1})^T r_j^* = r_j^T r_j^*$$

Hence,

$$-\frac{\|r_j^*\|}{\|r_{j-1}^*\|}l_{j+1} = -\frac{\|r_j^*\|l_{j+1}\delta_j d_j}{l_j d_j\|r_{j-1}^*\|d_{j-1}} = \frac{\zeta_{j-1}l_{j+1}\|r_j^*\|d_j}{\zeta_{j-2}l_j\|r_{j-1}^*\|d_{j-1}} = \frac{r_j^T r_j^*}{r_{j-1}^T r_{j-1}^*} \tag{5.1}$$

because of the recurrences for l_j and ζ_{j-2}. Analogue considerations yield

$$-\zeta_{j-1}^* u_{j,j+1}\|r_j\|d_j = r_j^T r_j^*,$$

and

$$-\frac{\|r_j\|}{\|r_{j-1}\|}u_{j,j+1} = \frac{r_j^T r_j^*}{r_{j-1}^T r_{j-1}^*}. \tag{5.2}$$

Finally,

$$\frac{\zeta_j^*}{\|r_j^*\|} = \frac{-d_{j+1}\zeta_j^*}{-\beta_{j+1}}\frac{u_{j,j+1}d_j}{d_{j+1}\|r_j^*\|} = \frac{-\zeta_{j-1}^* u_{j,j+1}\|r_j\|d_j}{\|r_j^*\|d_{j+1}\|r_j\|} = \frac{r_j^T r_j^*}{p_j^T\mathbf{A}^T p_j^*}, \tag{5.3}$$

and in a similar way

$$\frac{\zeta_j}{\|r_j\|} = \frac{r_j^T r_j^*}{p_j^T\mathbf{A}^T p_j^*}. \tag{5.4}$$

With (5.1), (5.2), (5.3) and (5.4) we can now reformulate the third point of the above algorithms and combine both algorithms to obtain the desired one. □

Algorithm 5.1.10 Look-ahead Lanczos process for nonsymmetric matrices

Initialization: Choose vectors v_1 *and* w_1 *with* $\|v_1\| = \|w_1\| = 1$; $\mathbf{V}_0 = \mathbf{D}_0 = \mathbf{W}_0 = \emptyset$; $v_0 = w_0 = 0$, $\mathbf{V}_1 = (v_1)$, $\mathbf{W}_1 = (w_1)$, $\mathbf{D}_1 = \mathbf{W}_1^T\mathbf{V}_1$; $k_1 = 1$, $i = 1$, $\rho_1 = \xi_1 = 1$.

Calculus:

```
do  $j = 1, m$
        if $\mathbf{D}_i$ is nonsingular then
                $\tilde{v}_{j+1} = \mathbf{A}v_j - \mathbf{V}_i\mathbf{D}_i^{-1}\mathbf{W}_i^T\mathbf{A}v_j - \mathbf{V}_{i-1}\mathbf{D}_{i-1}^{-1}\mathbf{W}_{i-1}^T\mathbf{A}v_j$
                $\tilde{w}_{j+1} = \mathbf{A}^Tw_j - \mathbf{W}_i\mathbf{D}_i^{-T}\mathbf{V}_i^T\mathbf{A}^Tw_j - \mathbf{W}_{i-1}\mathbf{D}_{i-1}^{-T}\mathbf{V}_{i-1}^T\mathbf{A}^Tw_j$
                $k_{i+1} = j + 1; \quad i = i + 1; \quad \mathbf{V}_i = \mathbf{W}_i = \emptyset$
        else
                $\tilde{v}_{j+1} = \mathbf{A}v_j - v_j - \mathbf{V}_{i-1}\mathbf{D}_{i-1}^{-1}\mathbf{W}_{i-1}^T\mathbf{A}v_j$
                $\tilde{w}_{j+1} = \mathbf{A}^Tw_j - w_j - \mathbf{W}_{i-1}\mathbf{D}_{i-1}^{-T}\mathbf{V}_{i-1}^T\mathbf{A}^Tw_j$
                if $j \neq k_i + 1$ then
                        $\tilde{v}_{j+1} = \tilde{v}_{j+1} - v_{j-1}/\rho_j$
                        $\tilde{w}_{j+1} = \tilde{w}_{j+1} - w_{j-1}/\xi_j$
                endif
        endif
        $\rho_{j+1} = ||\tilde{v}_{j+1}||$
        $\xi_{j+1} = ||\tilde{w}_{j+1}||$
        if $\rho_{j+1} = 0$ or $\xi_{j+1} = 0$ stop
        $v_{j+1} = \tilde{v}_{j+1}/\rho_{j+1}$
        $w_{j+1} = \tilde{w}_{j+1}/\xi_{j+1}$
        $\mathbf{V}_i = (\mathbf{V}_i, v_{j+1}); \quad \mathbf{W}_i = (\mathbf{W}_i, w_{j+1}); \quad \mathbf{D}_i = \mathbf{W}_i^T\mathbf{V}_i$
enddo
```

Lemma 5.1.11 *If* ρ_j *and* ξ_j *do not vanish for any* $j \leq m+1$, *then* $\operatorname{span}\{v_1, \ldots, v_{m+1}\} = \mathcal{K}_{m+1}(\mathbf{A}, v_1)$ *and* $\operatorname{span}\{w_1, \ldots, w_{m+1}\} = \mathcal{K}_{m+1}(\mathbf{A}^T, w_1)$ *and* $\mathbf{A}(v_1, \ldots, v_m)$ *can be decomposed in the form (1.12) with a block-tridiagonal Hessenberg matrix and with* $\mathbf{C}_m = (v_1, \ldots, v_m)$. *Moreover, the algorithm divides the bases* $\{v_1, \ldots, v_m\}$ *and* $\{w_1 \ldots, w_m\}$ *of* $\mathcal{K}_m(\mathbf{A}^T, \tilde{v_1})$ *into* i *blocks*

$$\mathbf{V}_l = (v_{k_l}, v_{k_l+1}, \ldots, v_{k_{l+1}-1}), \quad \mathbf{W}_l = (w_{k_l}, w_{k_l+1}, \ldots, w_{k_{l+1}-1}), \quad l = 1, 2, \ldots, i-1, \tag{5.5}$$

$$\mathbf{V}_i = (v_{k_i}, v_{k_i+1}, \ldots, v_k), \qquad \mathbf{W}_i = (w_{k_l}, w_{k_i+1}, \ldots, w_k),$$

where

$$1 = k_1 < k_2 < \ldots < k_i \leq m < k_{i+1},$$

with the properties

$$\mathbf{W}_j^T\mathbf{V}_l = 0, \quad \text{for } j \neq l, \qquad \mathbf{W}_j^T\mathbf{V}_l = \mathbf{D}_l, \quad \text{for } j = l, \qquad j, l = 1, 2, \ldots, i,$$

where

$$\mathbf{D}_l \quad \text{is nonsingular for} \quad l = 1, 2, \ldots, i-1, \quad \mathbf{D}_i \quad \text{is nonsingular for} \quad k = k_{i+1} - 1.$$

With $\mathbf{D}_m$ being the block-diagonal matrix with the blocks $(\mathbf{D}_1, \dots, \mathbf{D}_i)$ on its diagonal, we thus have the block-bi-orthogonality property

$$(w_1, \dots, w_m)^T (v_1, \dots, v_m) = \mathbf{D}_m. \tag{5.6}$$

Proof: It is not difficult to see the algorithm decomposes $\mathbf{A}(v_1, \dots, v_m)$ and $\mathbf{A}^T(w_1, \dots, w_m)$ in

$$\mathbf{A}(v_1, \dots, v_m) = (v_1, \dots, v_{m+1})\tilde{\mathbf{H}}_m \qquad \text{and} \quad \mathbf{A}^T(w_1, \dots, w_m) = (w_1, \dots, w_{m+1})\tilde{\mathbf{H}}_m^T,$$

where $\tilde{\mathbf{H}}_m$ is block tridiagonal and has the form

$$\tilde{\mathbf{H}}_m = \begin{pmatrix} \alpha_1 & \beta_2 & 0 & \dots & 0 \\ \gamma_2 & \alpha_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_i \\ 0 & \dots & 0 & \gamma_i & \alpha_i \\ 0 & \dots & & \dots & \rho_{m+1} \end{pmatrix},$$

for some block matrices α_l, β_l and γ_l. If we put $g_l = k_{l+1} - k_l$ for $l = 1, \dots, i-1$, and $\tilde{g}_i = m - k_i$, then the blocks α_l, β_l a γ_l of $\tilde{\mathbf{H}}_m$ have respective dimensions $g_l \times g_l, g_{l-1} \times g_l$, and $g_l \times g_{l-1}$ for $l \le i-1$. The matrices α_i, β_i and γ_i corresponding to the current blocks have dimensions respectively $\tilde{g}_i \times \tilde{g}_i, g_{i-1} \times \tilde{g}_i$, and $\tilde{g}_i \times g_{i-1}$. The blocks have the following form:

$$\alpha_l = \begin{pmatrix} * & * & 0 & \dots & 0 & * \\ \rho_{k_l+1} & * & \ddots & \ddots & \vdots & \vdots \\ 0 & \rho_{k_l+2} & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & * & * \\ \vdots & & \ddots & \ddots & * & * \\ 0 & \dots & \dots & 0 & \rho_{k_{l+1}-1} & * \end{pmatrix},$$

where $*$ denotes an entry that is not necessarily 0, and

$$\gamma_l = \begin{pmatrix} 0 & \dots & 0 & \rho_{k_l} \\ \vdots & \ddots & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & 0 \end{pmatrix}.$$

The blocks β_l are in general full matrices. Thus $\tilde{\mathbf{H}}_m$ is upper Hessenberg and therefore $\operatorname{span}\{v_1, \dots, v_{m+1}\} = \mathcal{K}_{m+1}(\mathbf{A}, v_1)$ and $\operatorname{span}\{w_1, \dots, w_{m+1}\} = \mathcal{K}_{m+1}(\mathbf{A}^T, w_1)$.

We will show the block-orthogonality by induction. Trivially, $\mathbf{W}_1^T\mathbf{V}_0 = 0 = \mathbf{W}_2^T\mathbf{V}_0$. We get $\mathbf{W}_2^T\mathbf{V}_1 = 0$ when $w_I^T\mathbf{V}_1 = 0$ for $I = k_2, \dots, k_3 - 1$. This indeed holds due to

$$w_{k_2}^T\mathbf{V}_1 = \frac{1}{\xi_{k_2}}(w_{k_2-1}^T\mathbf{A} - w_{k_2-1}^T\mathbf{A}\mathbf{V}_1\mathbf{D}_1^{-1}\mathbf{W}_1^T)\mathbf{V}_1 = \frac{1}{\xi_{k_2}}(w_{k_2-1}^T\mathbf{A}\mathbf{V}_1 - w_{k_2-1}^T\mathbf{A}\mathbf{V}_1) = 0,$$

because $\mathbf{W}_1^T\mathbf{V}_1 = \mathbf{D}_1$ and

$$w_{k_2+1}^T\mathbf{V}_1 = \frac{1}{\xi_{k_2+1}}(w_{k_2}^T\mathbf{A} - w_{k_2}^T - w_{k_2}^T\mathbf{A}\mathbf{V}_1\mathbf{D}_1^{-1}\mathbf{W}_1^T)\mathbf{V}_1 = 0.$$

In addition,

$$w_{k_2+I}^T\mathbf{V}_1=\frac{1}{\xi_{k_2+I}}(w_{k_2+I-1}^T\mathbf{A}-w_{k_2+I-1}^T-\frac{1}{\xi_{k_2+I-1}}w_{k_2+I-2}^T-w_{k_2+I-1}^T\mathbf{A}\mathbf{V}_1\mathbf{D}_1^{-1}\mathbf{W}_1^T)\mathbf{V}_1=0$$

for $1<I<k_3-k_2$.
Similarly, $\mathbf{W}_0^T\mathbf{V}_1=0=\mathbf{W}_0^T\mathbf{V}_2$ and we have $\mathbf{W}_1^T\mathbf{V}_2=0$ if and only if $\mathbf{W}_1^T(\tilde{v}_{k_2},\dots,\tilde{v}_{k_3-1})=0$. Indeed,

$$\mathbf{W}_1^T\tilde{v}_{k_2}=\mathbf{W}_1^T(\mathbf{A}v_{k_2-1}-\mathbf{V}_1\mathbf{D}_1^{-1}\mathbf{W}_1^T\mathbf{A}v_{k_2-1})=0\,;$$

$$\mathbf{W}_1^T\tilde{v}_{k_2+1}=\mathbf{W}_1^T(\mathbf{A}v_{k_2}-v_{k_2}-\mathbf{V}_1\mathbf{D}_1^{-1}\mathbf{W}_1^T\mathbf{A}v_{k_2})=0\,;$$

$$\mathbf{W}_1^T\tilde{v}_{k_2+I}=\mathbf{W}_1^T(\mathbf{A}v_{k_2+I-1}-v_{k_2+I-1}-v_{k_2+I-2}/\rho_{k_2+I-1}-\mathbf{V}_1\mathbf{D}_1^{-1}\mathbf{W}_1^T\mathbf{A}v_{k_2+I-1})=0,$$

for $1<I<k_3-k_2$.
Now let us assume $\mathbf{W}_j^T\mathbf{V}_l=0$ for $j\neq l$, where $j,l\leq I-1$ and $I\geq 3$.
We first show that $\mathbf{W}_I^T\mathbf{V}_j=0$ pro $j<I$. This is true as far as $w_l^T\mathbf{V}_j=0$ for $l=k_I,\dots,k_{I+1}-1,\quad j<I$. Thus we have the following cases:

- $l=k_I, j=I-1$:

$$w_{k_I}^T\mathbf{V}_{I-1}=\frac{1}{\xi_{k_I}}(w_{k_I-1}^T\mathbf{A}-w_{k_I-1}^T\mathbf{A}\mathbf{V}_{I-1}\mathbf{D}_{I-1}^{-1}\mathbf{W}_{I-1}^T-w_{k_I-1}^T\mathbf{A}\mathbf{V}_{I-2}\mathbf{D}_{I-2}^{-1}\mathbf{W}_{I-2}^T)\mathbf{V}_{I-1}=$$

$$\frac{1}{\xi_{k_I}}(w_{k_I-1}^T\mathbf{A}\mathbf{V}_{I-1}-w_{k_I-1}^T\mathbf{A}\mathbf{V}_{I-1})=0.$$

- $l=k_I, j=I-2$:

$$w_{k_I}^T\mathbf{V}_{I-2}=\frac{1}{\xi_{k_I}}(w_{k_I-1}^T\mathbf{A}-w_{k_I-1}^T\mathbf{A}\mathbf{V}_{I-1}\mathbf{D}_{I-1}^{-1}\mathbf{W}_{I-1}^T-w_{k_I-1}^T\mathbf{A}\mathbf{V}_{I-2}\mathbf{D}_{I-2}^{-1}\mathbf{W}_{I-2}^T)\mathbf{V}_{I-2}=$$

$$\frac{1}{\xi_{k_I}}(w_{k_I-1}^T\mathbf{A}\mathbf{V}_{I-2}-w_{k_I-1}^T\mathbf{A}\mathbf{V}_{I-2})=0$$

- $l=k_I, j<I-2$:

$$w_{k_I}^T\mathbf{V}_j=\frac{1}{\xi_{k_I}}(w_{k_I-1}^T\mathbf{A}-w_{k_I-1}^T\mathbf{A}\mathbf{V}_{I-1}\mathbf{D}_{I-1}^{-1}\mathbf{W}_{I-1}^T-w_{k_I-1}^T\mathbf{A}\mathbf{V}_{I-2}\mathbf{D}_{I-2}^{-1}\mathbf{W}_{I-2}^T)\mathbf{V}_j=$$

$$w_{k_I-1}^T\mathbf{A}\mathbf{V}_j=w_{k_I-1}^T(\mathbf{A}v_{k_j},\dots,\mathbf{A}v_{k_{j+1}-1}),$$

  where $\mathbf{A}v_{k_j+J}=\tilde{v}_{k_j+J+1}+*v_{k_j+J}+*v_{k_j+J-1}+\mathbf{V}_{j-1}(\mathbf{D}_{j-1}^{-1}\mathbf{W}_{j-1}^T\mathbf{A}v_{k_j+J})\in\operatorname{span}\{\mathbf{V}_{j-1},\mathbf{V}_j\}$ for some not necessarily 0 values $*$ when $J<k_{j+1}-k_j-1$, and for $J=k_{j+1}-k_j-1$ we obtain

$$\mathbf{A}v_{k_{j+1}-1}=\tilde{v}_{k_{j+1}}+\mathbf{V}_j(\mathbf{D}_j^{-1}\mathbf{W}_j^T\mathbf{A}v_{k_{j+1}-1})+\mathbf{V}_{j-1}(\mathbf{D}_{j-1}^{-1}\mathbf{W}_{j-1}^T\mathbf{A}v_{k_{j+1}-1})\in\operatorname{span}\{\mathbf{V}_{j-1},\mathbf{V}_j,\mathbf{V}_{j+1}\}.$$

  The expression $j+1$ equals at most $I-2$ and $w_{k_I-1}\in\mathbf{W}_{I-1}$. Hence with the assumption of the induction we always have $w_{k_I-1}^T\mathbf{A}\mathbf{V}_j=0$.

- $l=k_I+1, j=I-1$:

$$w_{k_I+1}^T\mathbf{V}_{I-1}=\frac{1}{\xi_{k_I+1}}(w_{k_I}^T\mathbf{A}-w_{k_I}^T-w_{k_I}^T\mathbf{A}\mathbf{V}_{I-1}\mathbf{D}_{I-1}^{-1}\mathbf{W}_{I-1}^T)\mathbf{V}_{I-1}=$$

$$\frac{1}{\xi_{k_I+1}}(w_{k_I}^T\mathbf{A}\mathbf{V}_{I-1}-w_{k_I}^T\mathbf{A}\mathbf{V}_{I-1})=0.$$

- $l = k_I + 1, j \leq I - 2$:

  $$w^T_{k_I+1}\mathbf{V}_j = \frac{1}{\xi_{k_I+1}}(w^T_{k_I}\mathbf{A} - w^T_{k_I} - w^T_{k_I}\mathbf{A}\mathbf{V}_{I-1}\mathbf{D}^{-1}_{I-1}\mathbf{W}^T_{I-1})\mathbf{V}_j$$

  $$= \frac{1}{\xi_{k_I+1}}(w^T_{k_I}\mathbf{A}\mathbf{V}_j),$$

  where, as above, $\mathbf{A}\mathbf{V}_j \in \text{span}\{\mathbf{V}_{j-1}, \mathbf{V}_j, \mathbf{V}_{j+1}\}$. This means $w^T_{k_I+1}\mathbf{V}_j = 0$ due to the foregoing cases.

- $l = k_I + J, j = I - 1$ with $1 < J < k_{I+1} - k_I$:

  $$w^T_{k_I+J}\mathbf{V}_{I-1} =$$

  $$\frac{1}{\xi_{k_I+J}}(w^T_{k_I+J-1}\mathbf{A} - w^T_{k_I+J-1} - \frac{1}{\xi_{k_I+J-1}}w^T_{k_I+J-2} - w^T_{k_I+J-1}\mathbf{A}\mathbf{V}_{I-1}\mathbf{D}^{-1}_{I-1}\mathbf{W}^T_{I-1})\mathbf{V}_{I-1} =$$

  $$\frac{1}{\xi_{k_I+J}}(w^T_{k_I+J-1}\mathbf{A}\mathbf{V}_{I-1} - w^T_{k_I+J-1}\mathbf{A}\mathbf{V}_{I-1}) = 0.$$

- $l = k_I + J, j \leq I - 2$ with $1 < J < k_{I+1} - k_I$:

  $$w^T_{k_I+J}\mathbf{V}_j =$$

  $$\frac{1}{\xi_{k_I+J}}(w^T_{k_I+J-1}\mathbf{A} - w^T_{k_I+J-1} - \frac{1}{\xi_{k_I+J-1}}w^T_{k_I+J-2} - w^T_{k_I+J-1}\mathbf{A}\mathbf{V}_{I-1}\mathbf{D}^{-1}_{I-1}\mathbf{W}^T_{I-1})\mathbf{V}_j =$$

  $$\frac{1}{\xi_{k_I+J}}(w^T_{k_I+J-1}\mathbf{A}\mathbf{V}_j),$$

  where, as above, $\mathbf{A}\mathbf{V}_j \in \text{span}\{\mathbf{V}_{j-1}, \mathbf{V}_j, \mathbf{V}_{j+1}\}$, implying $w^T_{k_I+J}\mathbf{V}_J = 0$ because of the preceding cases. The second half of the induction, i.e. $\mathbf{W}^T_j\mathbf{V}_I = 0$ for $j < I$, can be proven similarly.

Algorithm 5.1.12 TRANSPOSE FREE LANCZOS PROCESS FOR NONSYMMETRIC MATRICES

Initialization: Choose vectors q_0 *and* q^*_0 ; $p_0 = c_0 = q_0$.

Calculus:

```
do  $j = 0, m$
        $\alpha_{2j} = q^T_{2j}q^*_0/(\mathbf{A}p_{2j})^T q^*_0$
        $\alpha_{2j+1} = \alpha_{2j}$
        $c_{2j+1} = c_{2j} - \alpha_{2j}\mathbf{A}p_{2j}$
        $q_{2j+1} = q_{2j} - \alpha_{2j}\mathbf{A}c_{2j}$
        $q_{2j+2} = q_{2j+1} - \alpha_{2j+1}\mathbf{A}c_{2j+1}$
        if $\alpha_{2j} = 0$ stop
        $\beta_{2j} = q^T_{2j+2}q^*_0/q^T_{2j}q^*_0$
        $c_{2j+2} = q_{2j+2} + \beta_{2j}c_{2j+1}$
        $p_{2j+2} = c_{2j+2} + \beta_{2j}(c_{2j+1} + \beta_{2j}p_{2j})$
enddo
```

Lemma 5.1.13 *If $\alpha_i \neq 0$ for $i \leq 2m$ then Algorithm 5.1.12 generates sequences $\{c_0, \dots, c_{2m+2}\}$ and $\{q_0, \dots, q_{2m+2}\}$ that both span $\mathcal{K}_{2m+3}(\mathbf{A}, q_0)$. Moreover, the sequences $\{c_0, \dots, c_{2m+1}\}$ and $\{q_0, \dots, q_{2m+1}\}$ both span $\mathcal{K}_{2m+2}(\mathbf{A}, q_0)$.*

Proof: By induction.
Trivially, span$\{c_0\}$ =span$\{q_0\} = \mathcal{K}_1(\mathbf{A}, q_0)$.
Let us assume span$\{c_0, \dots, c_{2i}\}$ =span$\{q_0, \dots, q_{2i}\} = \mathcal{K}_{2i+1}(\mathbf{A}, q_0)$ for all $i \leq j$ and for some $j \leq m$, and

$$p_{2j} = \sum_{i=0}^{2j} \delta_i \mathbf{A}^i q_0, \quad \delta_{2j} \neq 0,$$

for some $\delta_i \in \mathbb{R}$, which holds for $j = 0$.
Then

$$c_{2j+1} = c_{2j} - \alpha_{2j}\mathbf{A}p_{2j} = \sum_{i=0}^{2j} \gamma_i \mathbf{A}^i q_0 - \alpha_{2j}\mathbf{A}(\sum_{i=0}^{2j} \delta_i \mathbf{A}^i q_0), \quad \delta_{2j} \neq 0,$$

for some $\gamma_i \in \mathbb{R}$. Hence span$\{c_0, \dots, c_{2j+1}\} = \mathcal{K}_{2j+2}(\mathbf{A}, q_0)$. Moreover,

$$q_{2j+1} = q_{2j} - \alpha_{2j}\mathbf{A}c_{2j} \in \mathcal{K}_{2j+2}(\mathbf{A}, q_0) \setminus \mathcal{K}_{2j+1}(\mathbf{A}, q_0),$$

because $\alpha_{2j} \neq 0$ and $\{c_0, \dots, c_{2j}\}$ is an ascending basis of $\mathcal{K}_{2j+1}(\mathbf{A}, q_0)$. Hence span$\{q_0, \dots, q_{2j+1}\} = \mathcal{K}_{2j+2}(\mathbf{A}, q_0)$. Similarly, we obtain from

$$q_{2j+2} = q_{2j+1} - \alpha_{2j+1}\mathbf{A}c_{2j+1} \in \mathcal{K}_{2j+3}(\mathbf{A}, q_0) \setminus \mathcal{K}_{2j+2}(\mathbf{A}, q_0)$$

that span$\{q_0, \dots, q_{2j+2}\} = \mathcal{K}_{2j+3}(\mathbf{A}, q_0)$. Furthermore,

$$c_{2j+2} = q_{2j+2} + \beta_{2j}c_{2j+1} \in \mathcal{K}_{2j+3}(\mathbf{A}, q_0) \setminus \mathcal{K}_{2j+2}(\mathbf{A}, q_0),$$

because $\{q_0, \dots, q_{2j+2}\}$ is an ascending basis of $\mathcal{K}_{2j+3}(\mathbf{A}, q_0)$. This shows that span$\{c_0, \dots, c_{2j+2}\} = \mathcal{K}_{2j+3}(\mathbf{A}, q_0)$. Finally,

$$p_{2j+2} = c_{2j+2} + \beta_{2j}(c_{2j+1} + \beta_{2j}p_{2j}),$$

proving that

$$p_{2j+2} = \sum_{i=0}^{2j+2} \delta_i \mathbf{A}^i q_0, \quad \delta_{2j+2} \neq 0,$$

for some $\delta_i \in \mathbb{R}$. □

Lemma 5.1.14 *The sequence $\{q_0, \dots, q_{2j}\}$ generated by Algorithm 5.1.12 with $q_0 := r_0$ has the property that when the jth BCG residual has the form*

$$\overset{\text{BCG}}{r_j} = \rho_j(\mathbf{A})r_0, \tag{5.7}$$

for some polynomial ρ_j of degree j with $\rho_j(0) = 1$, then

$$q_{2j} = (\rho_j(\mathbf{A}))^2 r_0.$$

Proof: Let us square, in Algorithm 5.1.9, the polynomials for both residuals and auxiliary sequences, assuming

$$p_j = \pi_j(\mathbf{A})r_0 \tag{5.8}$$

for some polynomial π_j of degree j with $\pi_j(0) = 1$. If we exploit the recurrences given in Algorithm 5.1.9, then

$$\rho_{j+1}^2(t) = \rho_j(t)^2 - 2\xi_j t\pi_j(t)\rho_j(t) + \xi_j^2 t^2 \pi_j^2(t)$$

and

$$\pi_{j+1}^2(t) = \rho_{j+1}(t)^2 + 2\phi_j\rho_{j+1}(t)\pi_j(t) + \phi_j^2\pi_j^2(t).$$

Short recurrences for the squared polynomials can be obtained when we introduce a second auxiliary vector sequence, namely

$$s_j := \rho_{j+1}(\mathbf{A})\pi_j(\mathbf{A})r_0.$$

Together with

$$q_j := \rho_j^2(\mathbf{A})q_0, \qquad p'_j := \pi_j^2(\mathbf{A})q_0$$

and with $q_0 = r_0$ we obtain the recurrences

$$q_{j+1} = q_j - \xi_j\mathbf{A}(2q_j + 2\phi_{j-1}s_{j-1} - \xi_j\mathbf{A}p'_j),$$

$$s_j = q_j + \phi_{j-1}s_{j-1} - \xi_j\mathbf{A}p'_j,$$

$$p'_{j+1} = q_{j+1} + 2\phi_j s_j + \phi_j^2 p'_j.$$

In Algorithm 5.1.9, it is obvious that the polynomial for r_j from (5.7) also defines the dual residual r_j^*, namely through $r_j^* = \rho_j(\mathbf{A}^T)r_0^*$ and the same holds for the polynomials π_j with $\pi_j(0) = 1$ that express the auxiliary sequences in terms of their initial vector:

$$\text{if} \quad p_j = \pi_j(\mathbf{A})r_0, \qquad \text{then} \quad p_j^* = \pi_j(\mathbf{A}^T)r_0^* \tag{5.9}$$

and vice versa. With these BCG polynomials, we can define both ξ_j and ϕ_j from Algorithm 5.1.9 in terms of the squared polynomials q_j and p'_j:

$$\phi_j = \frac{(r_{j+1}^*)^T r_{j+1}}{(r_j^*)^T r_j} = \frac{(r_0^*)^T(\rho_{j+1}^2(\mathbf{A})r_0)}{(r_0^*)^T(\rho_j^2(\mathbf{A})r_0)} = \frac{(r_0^*)^T q_{j+1}}{(r_0^*)^T q_j},$$

$$\xi_j = \frac{(r_j^*)^T r_j}{(p_j^*)^T \mathbf{A} p_j} = \frac{(r_0^*)^T(\rho_j^2(\mathbf{A})r_0)}{(r_0^*)^T(\mathbf{A}\pi_j^2(\mathbf{A})r_0)} = \frac{(r_0^*)^T q_j}{(r_0^*)^T \mathbf{A} p'_j}.$$

Note that both coefficients can be defined without transposing the matrix $\mathbf{A}$. This is the original motivation of squaring BCG polynomials. A last simplification to the recurrence can be made by using the vectors $c_j := q_j + \phi_{j-1}s_{j-1}$. The resulting algorithm, initialized with $p'_0 = c_0 = q_0$, $s_0 = 0$ and $\phi_0 = 0$, would have the form

- $\xi_j = (q_0^*)^T q_j / (q_0^*)^T \mathbf{A} p'_j$
- $s_j = c_j - \xi_j \mathbf{A} p'_j$
- $q_{j+1} = q_j - \xi_j \mathbf{A}(c_j + s_j)$
- $\phi_j = (q_0^*)^T q_{j+1} / (q_0^*)^T q_j$
- $c_{j+1} = q_{j+1} + \phi_j s_j$
- $p'_{j+1} = q_{j+1} + \phi_j(2s_j + \phi_j p'_j)$

Algorithm 5.1.12 now follows by multiplying all indexes by two, by putting $c_{2j+1} := s_{2j}$ and dividing the computation of the vector q_{2j+2} into two steps, by replacing p'_{2j} by p_{2j} and finally with $\alpha_{2j} := \xi_{2j}$, $\beta_{2j} := \phi_{2j}$ for all j. □

Algorithm 5.1.15 Transpose free Lanczos process for nonsymmetric matrices with stabilization parameters ω_{2j}

Initialization: Choose vectors q_0 *and* q_0^* *;* $p_{-2} = c_{-2} = 0\,;\alpha_{-2} = 0\,;\omega_{-2} = \rho_{-2} = 1$.

Calculus:

```
do  $j = 0, m$
        $\rho_{2j} = q_{2j}^T q_0^*$
        if $\rho_{2j-2}\omega_{2j-2} = 0$ stop
        $\beta_{2j-2} = (\alpha_{2j-2}\rho_{2j})/(\rho_{2j-2}\omega_{2j-2})$
        $p_{2j} = q_{2j} + \beta_{2j-2}p_{2j-2} - \beta_{2j-2}\omega_{2j-2}c_{2j-2}$
        $c_{2j} = \mathbf{A}p_{2j}$
        $\gamma_{2j} = c_{2j}^T q_0^*$
        if $\gamma_{2j} = 0$ stop
        $\alpha_{2j} = \rho_{2j}/\gamma_{2j}$
        $q_{2j+1} = q_{2j} - \alpha_{2j}c_{2j}$
        $c_{2j+1} = \mathbf{A}q_{2j+1}$
        $\omega_{2j} = c_{2j+1}^T q_{2j+1}/c_{2j+1}^T c_{2j+1}$
        $q_{2j+2} = q_{2j+1} - \omega_{2j}c_{2j+1}$
enddo
```

Lemma 5.1.16 *If* $\gamma_{2i} \neq 0$*,* $\rho_{2i} \neq 0$ *and* $\omega_{2i} \neq 0$ *for* $0 \leq i \leq m$ *then Algorithm 5.1.15 generates a sequence* $\{q_0, \ldots, q_{2m+2}\}$ *that spans* $\mathcal{K}_{2m+3}(\mathbf{A}, q_0)$ *and a sequence* $\{c_0, \ldots, c_{2m+1}\}$ *spanning* $\mathbf{A}\mathcal{K}_{2m+2}(\mathbf{A}, q_0)$.

Proof: By induction.
Note that by assumption we have $\alpha_{2i} \neq 0$ and $\beta_{2i} \neq 0$ for $0 \leq i \leq m$. For $j = 0$ we have $p_0 = q_0$, $c_0 = \mathbf{A}q_0$, $q_1 = q_0 - \alpha_0\mathbf{A}q_0$, $c_1 = \mathbf{A}(q_0 - \alpha_0\mathbf{A}q_0)$ and $q_2 = q_1 - \omega_0 c_1$. Hence $\{q_0, q_1, q_2\}$ is an ascending basis of $\mathcal{K}_3(\mathbf{A}, q_0)$ and $\{c_0, c_1\}$ an ascending basis of $\mathbf{A}\mathcal{K}_2(\mathbf{A}, q_0)$.

Now for some $j \leq m$ we assume $\text{span}\{q_0, \ldots, q_{2i}\} = \mathcal{K}_{2i+1}(\mathbf{A}, q_0)$, $\text{span}\{c_0, \ldots, c_{2i-1}\} = \mathbf{A}\mathcal{K}_{2i}(\mathbf{A}, q_0)$ for all $i \leq j$ and

$$p_{2j-2} \in \mathcal{K}_{2j-1}(\mathbf{A}, q_0),$$

which holds for $j = 1$. We have

$$c_{2j} = \mathbf{A}\left(q_{2j} + \beta_{2j-2}p_{2j-2} - \beta_{2j-2}\omega_{2j-2}c_{2j-2}\right) \in \mathbf{A}\mathcal{K}_{2j+1}(\mathbf{A}, q_0) \setminus \mathbf{A}\mathcal{K}_{2j}(\mathbf{A}, q_0).$$

Therefore $\text{span}\{c_0, \ldots, c_{2j}\} = \mathbf{A}\mathcal{K}_{2j+1}(\mathbf{A}, q_0)$. Even so,

$$q_{2j+1} = q_{2j} - \alpha_{2j}c_{2j} \in \mathcal{K}_{2j+2}(\mathbf{A}, q_0) \setminus \mathcal{K}_{2j+1}(\mathbf{A}, q_0),$$

hence $\text{span}\{q_0, \ldots, q_{2j+1}\} = \mathcal{K}_{2j+2}(\mathbf{A}, q_0)$. Furthermore,

$$c_{2j+1} = \mathbf{A}q_{2j+1} \in \mathbf{A}\mathcal{K}_{2j+2}(\mathbf{A}, q_0) \setminus \mathbf{A}\mathcal{K}_{2j+1}(\mathbf{A}, q_0),$$

thus $\text{span}\{c_0, \ldots, c_{2j+1}\} = \mathbf{A}\mathcal{K}_{2j+2}(\mathbf{A}, q_0)$. Finally,

$$q_{2j+2} = q_{2j+1} - \omega_{2j}c_{2j+1} \in \mathcal{K}_{2j+3}(\mathbf{A}, q_0) \setminus \mathcal{K}_{2j+2}(\mathbf{A}, q_0),$$

implying $\text{span}\{q_0, \ldots, q_{2j+2}\} = \mathcal{K}_{2j+3}(\mathbf{A}, q_0)$ and, to complete the induction,

$$p_{2j} = q_{2j} + \beta_{2j-2}p_{2j-2} - \beta_{2j-2}\omega_{2j-2}c_{2j-2} \in \mathcal{K}_{2j+1}(\mathbf{A}, q_0).$$

□

Lemma 5.1.17 *Algorithm 5.1.15 with $q_0 := r_0$ generates a sequence $\{q_0, \ldots, q_{2m+2}\}$ with the property that if BCG residuals are given by (5.7), then*

$$q_{2j} = \left(\prod_{i=0}^{j-1} (\mathbf{I}_n - \omega_{2i}\mathbf{A}) \right) \rho_j(\mathbf{A}) r_0, \qquad j \leq m+1.$$

Proof: Let us define the stabilizing polynomial

$$\kappa_j(t) := \left(\prod_{i=0}^{j-1} (1 - \omega_{2i} t) \right)$$

and use the polynomial π_j from (5.8) to express the auxiliary vectors p_j of Algorithm 5.1.9. For polynomials κ_j we have the relation

$$\kappa_{j+1}(\mathbf{A}) = (\mathbf{I}_n - \omega_{2j}\mathbf{A})\kappa_j(\mathbf{A}).$$

If we define

$$q_{2j} := \kappa_j(\mathbf{A})\rho_j(\mathbf{A})r_0, \qquad p'_{2j} := \kappa_j(\mathbf{A})\pi_j(\mathbf{A})r_0,$$

and use Algorithm 5.1.9 where we replace ξ_j by α_j and ϕ_j by β_j, then we obtain the two-term recurrences

$$p'_{2j+2} = \kappa_{j+1}(\mathbf{A})\pi_{j+1}(\mathbf{A})r_0 =$$
$$\kappa_{j+1}(\mathbf{A})\rho_{j+1}(\mathbf{A})r_0 + \beta_j(\mathbf{I}_n - \omega_{2j}\mathbf{A})\kappa_j(\mathbf{A})\pi_j(\mathbf{A})r_0 = q_{2j+2} + \beta_j(p'_{2j} - \omega_{2j}\mathbf{A}p'_{2j})$$

and similarly

$$q_{2j+2} = (q_{2j} - \alpha_j\mathbf{A}p'_{2j}) - \omega_{2j}\mathbf{A}(q_{2j} - \alpha_j\mathbf{A}p'_{2j}).$$

In contrast with the CGS polynomials, our BCG coefficients α_j and β_j cannot immediately be calculated from q_{2j} and p'_{2j}. The coefficient β_j is given by $\frac{r_{j+1}^T r_{j+1}^*}{r_j^T r_j^*}$ and the value $r_j^T r_j^*$ can be written as follows. The coefficient for the highest order term of the BCG polynomial ρ_j is equal to $(-1)^j \alpha_0\alpha_1 \ldots \alpha_{j-1}$ and we have

$$r_j^* = (-1)^j \alpha_0\alpha_1 \ldots \alpha_{j-1} (\mathbf{A}^T)^j r_0^* + w_j^*,$$

for some $w_j^* \in \mathcal{K}_j(\mathbf{A}^T, r_0^*)$. With the orthogonality condition of the BCG projection we obtain

$$r_j^T r_j^* = (-1)^j \alpha_0\alpha_1 \ldots \alpha_{j-1} r_j^T (\mathbf{A}^T)^j r_0^*.$$

Similarly, the coefficient for the highest order term of the polynomial κ_j is equal to $(-1)^j \omega_0\omega_2 \ldots \omega_{2j-2}$ and we have

$$\kappa_j(\mathbf{A}^T)r_0^* = (-1)^j \omega_0\omega_2 \ldots \omega_{2j-2} (\mathbf{A}^T)^j r_0^* + w_j^{**},$$

for some $w_j^{**} \in \mathcal{K}_j(\mathbf{A}^T, r_0^*)$. Thus

$$\delta_j := q_{2j}^T r_0^* = r_j^T \kappa_j(\mathbf{A}^T) r_0^* = (-1)^j \omega_0\omega_2 \ldots \omega_{2j-2} r_j^T (\mathbf{A}^T)^j r_0^*.$$

We can now express coefficients β_j in terms of coefficients δ_j:

$$\beta_j = \frac{r_{j+1}^T r_{j+1}^*}{r_j^T r_j^*} = -\alpha_j \frac{r_{j+1}^T (\mathbf{A}^T)^{j+1} r_0^*}{r_j^T (\mathbf{A}^T)^j r_0^*} = (\frac{\delta_{j+1}}{\delta_j})(\frac{\alpha_j}{\omega_{2j}}).$$

In a similar way it is possible to write the coefficient α_j with the help of δ_j. Per definition,

$$\alpha_j = \frac{(\rho_j(\mathbf{A})r_0)^T \rho_j(\mathbf{A}^T)r_0^*}{(\mathbf{A}\pi_j(\mathbf{A})r_0)^T \pi_j(\mathbf{A}^T)r_0^*}.$$

The leading coefficients for $\rho_j(\mathbf{A}^T)r_0^*$ and $\pi_j(\mathbf{A}^T)r_0^*$ are identical, as can be seen from Algorithm 5.1.9, and therefore, with the orthogonality conditions of the BCG residuals and auxiliary sequences,

$$\alpha_j = \frac{(\rho_j(\mathbf{A})r_0)^T \rho_j(\mathbf{A}^T)r_0^*}{(\mathbf{A}\pi_j(\mathbf{A})r_0)^T \rho_j(\mathbf{A}^T)r_0^*} = \frac{(\rho_j(\mathbf{A})r_0)^T \kappa_j(\mathbf{A}^T)r_0^*}{(\mathbf{A}\pi_j(\mathbf{A})r_0)^T \kappa_j(\mathbf{A}^T)r_0^*} = \frac{(r_0^*)^T \kappa_j(\mathbf{A})\rho_j(\mathbf{A})r_0}{(r_0^*)^T \mathbf{A}\kappa_j(\mathbf{A})\pi_j(\mathbf{A})r_0}.$$

Since $p'_{2j} = \kappa_j(\mathbf{A})\pi_j(\mathbf{A})r_0$, this yields

$$\alpha_j = \frac{\delta_j}{(r_0^*)^T \mathbf{A} p'_{2j}}.$$

Algorithm 5.1.15 now results from the above recurrences if we replace δ_j by ρ_{2j}, r_0^* by q_0^* and r_{2j} by q_{2j}, multiply the indexes of α and β by two, replace p'_{2j} by p_{2j}, divide the computation of the residual into two steps and define a second auxiliary vector sequence through $c_{2j} := \mathbf{A}p_{2j}$ and $c_{2j+1} := \mathbf{A}q_{2j+1}$. □

5.2 Implementations applying rank-one update

We display here several implementations that try to improve the restarted GMRES method with the help of the Sherman-Morrison formula according to the theory of chapters two, three and four. In most cases we have formulated an interactive algorithm, that is an algorithm where the user has the possibility to influence the calculus during the execution of the programm. For example, the user can prescribe residual norms of the auxiliary matrix or prescribe eigenvalues. Of course, modifications for fixed prescribed values are straightforward. All algorithms assume full dimensional Krylov subspaces, i.e. the involved Modified Gram Schmidt orthogonalization process does not break down. In addition, we assume back-transformation with (2.19) does not lead to overflow due to too small denominators.

The algorithms are followed by a brief computational and cost storage comparison with restarted GMRES. In this context, one should note that it is never needed to explicitly compute $\hat{\mathbf{A}}$, only matrix-vector products with $\hat{\mathbf{A}}$ are required. While multiplying with $\hat{\mathbf{A}} = \mathbf{A} - by^T$, respectively $\hat{\mathbf{A}} = \mathbf{A} - \mathbf{A}dy^T$, its special form can be exploited and the advantages of operations with structured matrices are preserved.

Algorithm 5.2.1 SHERMOR(m,k) - restarted GMRES with a predefined auxiliary system

Input: $\mathbf{A}\ldots$ *matrix;* $b\ldots$ *right-hand side;* $\varepsilon\ldots$ *tolerance for residual norm;*
$m\ldots$ *number of steps after which GMRES is restarted;*
$k\ldots$ *number of prescribed residual norms at the first cycle.*

Initialization: $y = \mathbf{0}$; $x_0 = \mathbf{0}$; $r_0 = b$; $\beta = \|r_0\|$; $r = r_0$; $g_1 = \beta e_1$; $init = 0$.

```
while $\|r\|/\|r_0\| > \varepsilon$ do
    do $i = 1, m$
        $\tilde{v} = (\mathbf{A} - by^T)v_i$
        do $j = 1, i$
            $h_{j,i} = v_j^T \tilde{v}$
            $\tilde{v} = \tilde{v} - h_{j,i} v_j$
        enddo
        $h_{i+1,i} = \|\tilde{v}\|, \quad v_{i+1} = \tilde{v}/h_{i+1,i}$
        if $i \le k$ and $init = 0$ then
            read $\|r_i\|$
            $\alpha_i = \frac{\sqrt{\frac{1-(\|r_i\|/\|r_{i-1}\|)^2}{(\|r_i\|/\|r_{i-1}\|)^2}} h_{i+1,i} - \sum_{j=1}^{i} c_{j-1} h_{j,i} \prod_{l=j}^{i-1}(-s_l)}{-\beta \prod_{l=1}^{i-1}(-s_l)}$
            $h_{1,i} = h_{1,i} - \beta\alpha_i$
        endif
        compute the Givens parameters $c_i$ and $s_i$ that zero out $h_{i+1,i}$
        $\tilde{\mathbf{H}}_i = \begin{pmatrix} \mathbf{I}_{i-1} & 0 & \ldots \\ 0 & c_i & s_i \\ \vdots & -s_i & c_i \end{pmatrix} \cdot \tilde{\mathbf{H}}_i \quad g_i = \begin{pmatrix} \mathbf{I}_{i-1} & 0 & \ldots \\ 0 & c_i & s_i \\ \vdots & -s_i & c_i \end{pmatrix} \cdot g_i$
        if $i = k$ and $init = 0$ then
            $y = \mathbf{V}_k \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix}$
            $init = 1$
        endif
    enddo
    $w_m = (\mathbf{H}_m)^{-1} g_m$
    $x_0 = x_0 + \mathbf{V}_m w_m$
    $r = (b - \mathbf{A}x_0)/(1 + y^T x_0)$
    $v_1 = r/\|r\|$
endwhile
$x_0 = x_0/(1 + y^T x_0)$
```

In comparison with the classical restarted GMRES method, additional storage costs of this algorithm consist of only one more n-dimensional vector ($y \in \mathbb{R}^n$) and during the initial cycle the necessity to store k Givens sines and k Givens cosines in order to compute the conditions α_i, which also need to be stored until the kth iteration. These conditions are obtained without computations of order n, but the vector y results from a matrix-vector product of dimension $n \times k$, thus asking for nk multiplications. At the end of the first cycle we need the number $1+y^T x_0$ to multiply it with $b-\mathbf{A}x_0$, which costs $2n$ more multiplications and these extra costs return at the end of every restart. In addition, the matrix-vector product with $\mathbf{A} - by^T$ is $2n$ multiplications more expensive than the product with $\mathbf{A}$. At the very end of the process, one has to update x_0 which costs also $2n$ multiplications. Thus if we need C restarts until convergence, the total number of extra multiplications, except for negligible ones whose number is independent from n, equals

$$nk + 2n + 4Cn + 2n = (k + 4 + 4C)n.$$

Algorithm 5.2.2 ALGORITHM PSHERMOR(M,K) - RESTARTED GMRES WITH MINIMIZATION OF GIVENS SINES

Input: $\mathbf{A}\ldots$ *matrix;* $b\ldots$ *right-hand side;* $\varepsilon\ldots$ *tolerance for residual norm;*
$m\ldots$ *number of steps in every restart;* $x_0\ldots$ *nonzero initial guess;*
$k\ldots$ *number of Givens sine minimizations at the beginning of every cycle.*

Initialization: $y = \mathbf{0}$; $r_0 = b - \mathbf{A}x_0$; $\beta = \|r_0\|$; $r = r_0$.

```
while $\|r\|/\|r_0\| > \varepsilon$ do
        do $i = 1, m$
            $\mathbf{A} = \mathbf{A} - \mathbf{A}x_0 y^T$
            $\tilde{v} = \mathbf{A}v_i$
            do  $j = 1, i$
                $h_{j,i} = v_j^T \tilde{v}$
                $\tilde{v} = \tilde{v} - h_{j,i}v_j$
            enddo
            if  $i \le k$ then
                $h_i^* = v_i^T \mathbf{A}x_0$
                compute $\alpha_i$, minimizing root of $s_i^2(\alpha_i)$
                $h_{j,i} = h_{j,i} - h_j^*\alpha_i,\ j \le i$
                $\tilde{v} = \tilde{v} - \alpha_i(\mathbf{A}x_0) - \sum_{j=1}^{i} h_{j,i}v_j$
            endif
            $h_{i+1,i} = \|\tilde{v}\|, \quad v_{i+1} = \tilde{v}/h_{i+1,i}$
            if  $i = k$ then
                compute $y \in \mathbb{R}^n$ satisfying $\begin{pmatrix} v_1^T \\ \vdots \\ v_k^T \\ x_0^T \end{pmatrix} y = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \\ 0 \end{pmatrix}$
            endif
        enddo
        compute $w_m \in \mathbb{R}^m$ minimizing $\|\beta e_1 - \tilde{\mathbf{H}}_m w\|$
        compute $x_m := x_0 + \mathbf{V}_m w_m$
        $x_0 = x_m - (y^T x_m)x_0$
        $y = \mathbf{0}$
        $r = b - \mathbf{A}x_0$
        $v_1 = r/\|r\|$
        endwhile
```

Computational costs of this algorithm are slightly higher than for GMRES(m): One extra matrix-vector product is needed at every restart when the vector d is not chosen to be equal to the actual approximation. During every restart we need to compute k extra values $v_i^T \mathbf{A} d$ and to update the values $h_{j,i}$ and $\tilde{v}$ in dependency of the choice of α_i, costing $n(k^2 + 5k + 2)/2$ multiplications. To compute y we need $2kn + n$ operations to orthogonalize d against $v_1, \ldots, v_k$ and one matrix vector-product of dimension $n \times k$. The minimizer α_i can be computed with additional operations that are dependent from n of order $2kn + n(k+1)k/2$. Finally, iteration numbers larger than k require matrix vector-products with the auxiliary matrix, which costs $2n$ more multiplications than with the original matrix. As for the storage costs, they will be the same as for GMRES(m) at every restart, except for the storage of three more n-dimensional vectors, y, d and $\mathbf{A}d$, and k values $v_i^T \mathbf{A} d$.

Algorithm 5.2.3 DEFSHERMORN(M,1) - RESTARTED GMRES WITH AUXILIARY SYSTEM DEFLATION AT THE INITIAL CYCLE

```
Input: $\mathbf{A}$ ... matrix;  $b$ ... right-hand side;  $\varepsilon$ ... tolerance for residual norm;
    $m$ ... number of steps after which GMRES is restarted;

Initialization: $y = \mathbf{0}$; $x_0 = \mathbf{0}$; $r_0 = b$; $\beta = \|r_0\|$; $r = r_0$; $init = 0$.
    while $\|r\|/\|r_0\| > \varepsilon$ do
        do $i = 1, m$
            $\tilde{v} = (\mathbf{A} - by^T)v_i$
            do $j = 1, i$
                $h_{j,i} = v_j^T \tilde{v}$
                $\tilde{v} = \tilde{v} - h_{j,i} v_j$
            enddo
            $h_{i+1,i} = \|\tilde{v}\|, \quad v_{i+1} = \tilde{v}/h_{i+1,i}$
        enddo
        compute $w_m \in \mathbb{R}^m$ minimizing $\|\beta e_1 - \tilde{\mathbf{H}}_m w\|$
        $x_0 = x_0 + \mathbf{V}_m w_m$
        $r = (b - \mathbf{A}x_0)/(1 + y^T x_0)$
        $v_1 = r/\|r\|$
        if $init = 0$ then
            $init = 1$
            compute $(d_m, c_m)$, the smallest eigenvalue-vector pair of $\mathbf{H}_m$
            if $d_m$ is real then
                read $\hat{d}_m$
                if $e_1^T c_m \neq 0$ then
                    $\rho = \frac{d_m - \hat{d}_m}{\beta e_1^T c_m}$
                    $y = \rho \mathbf{V}_m c_m$
                else
                    $y = 0$
                endif
            else
                read $\rho$
                $y = 2\rho/\|c_m\| \mathbf{V}_m \mathrm{Re}(c_m)$
            endif
        endif
    endwhile
    $x_0 = x_0/(1 + y^T x_0)$
```

Additional storage costs of this algorithm compared with the classical restarted GMRES method consist of only one more n-dimensional vector ($y \in \mathbb{R}^n$) and during the initial cycle it is necessary to store the entire Hessenberg matrix $\mathbf{H}_m$ in order to compute its smallest eigenvalue-eigenvector pair. This pair can for example be obtained by application of the inverse power method and does not ask for computations of order n. The vector y results from a matrix-vector product of dimension $n \times m$, thus asking for nm multiplications. At the end of every restart cycle we need the number $1 + y^T x_0$ to multiply it with $b - \mathbf{A}x_0$, which costs $2n$ more multiplications. In addition, the matrix-vector product with $\mathbf{A} - by^T$ is $2n$ multiplications more expensive than the product with $\mathbf{A}$. At the very end of the process, one has to update x_0 which costs also $2n$ multiplications. Thus if we need C restarts until convergence, the total number of extra multiplications, except for negligible ones whose number is independent from n, equals

$$nm + 4Cn + 2n = (m + 2 + 4C)n.$$

Algorithm 5.2.4 DEFSHERMOR(M) - RESTARTED GMRES WITH PRESCRIBED FIRST CYCLE RITZ VALUES

Input: $\mathbf{A}\ldots$ *matrix;* $b\ldots$ *right-hand side;* $\varepsilon\ldots$ *tolerance for residual norm;* $m\ldots$ *number of steps after which GMRES is restarted;*

Initialization: $y = \mathbf{0}$; $x_0 = \mathbf{0}$; $r_0 = b$; $\beta = \|r_0\|$; $r = r_0$; $g_1 = \beta e_1$; $init = 0$.

```
10 while $\|r\|/\|r_0\| > \varepsilon$ do
    do $i = 1, m$
        $\tilde{v} = (\mathbf{A} - by^T)v_i$
        do $j = 1, i$
            $h_{j,i} = v_j^T \tilde{v}$
            $\tilde{v} = \tilde{v} - h_{j,i} v_j$
        enddo
        $h_{i+1,i} = \|\tilde{v}\|, \quad v_{i+1} = \tilde{v}/h_{i+1,i}$
        if $init = 0$ then
            compute $(\theta_j, c_j)$, $1 \le j \le m$, the eigenvector-eigenvalue pairs
            of $\mathbf{H}_m$ ordered from largest to smallest norm
            if $Im(\theta_m) = 0$ then
                read $\hat{\theta}_m$
                $\alpha_1 = (\sum_{j=1}^m h_{j,j} - \hat{\theta}_m - \sum_{j=1}^{m-1} \theta_j)/\beta$
                compute $z \in \mathbb{R}^m$ satisfying $\begin{pmatrix} e_1^T \\ c_1^T \\ \vdots \\ c_{m-1}^T \end{pmatrix} z = \begin{pmatrix} \alpha_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$
            else
                read $\mathrm{Re}(\hat{\theta}_m)$, $\mathrm{Im}(\hat{\theta}_m)$
                $\gamma_1 = \frac{2(\mathrm{Re}(\theta_m) - \mathrm{Re}(\hat{\theta}_m))}{\beta}$
                $\gamma_2 = \frac{\alpha_{m-4} - (\hat{h}_{1,1}+1)\sum_{j=2}^m h_{j,j} - \sum_{i,l=2,\, i<l}^{m-1} h_{i,i}h_{l,l} + \sum_{i=1}^{m-1} h_{i,i+1}h_{i+1,i}}{h_{2,1}\beta}$
                compute $z \in \mathbb{R}^m$ satisfying $\begin{pmatrix} e_1^T \\ e_2^T \\ c_1^T \\ \vdots \\ c_{m-2}^T \end{pmatrix} z = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$
            endif
            $y = \mathbf{V}_m z$
            $init = 1$
            goto 10
        endif
    enddo
```

```
        compute $w_m \in \mathbb{R}^m$ minimizing $\|\beta e_1 - \tilde{\mathbf{H}}_m w\|$
        $x_0 = x_0 + \mathbf{V}_m w_m$
        $r = (b - \mathbf{A}x_0)/(1 + y^T x_0)$
        $v_1 = r/\|r\|$
    endwhile
    $x_0 = x_0/(1 + y^T x_0)$
```

Computational costs depending from n of DEFSHERMOR are exactly the same as for DEFSHERMORN. Costs that depend only from m are higher, because all eigenvalues and eigenvectors of the Hessenberg matrix $\mathbf{H}_m$ are needed. The same can be said about storage costs: During the initial cycle one needs to save *all* Ritz values and the eigenvectors of $\mathbf{H}_m$.

Algorithm 5.2.5 LOCAL(M,K) - RESTARTED GMRES WITH AUXILIARY SYSTEM LOCALLY MINIMIZING k INITIAL STEPS

```
Input: $\mathbf{A}\ldots$ matrix;  $b\ldots$ right-hand side;  $\varepsilon\ldots$ tolerance for residual norm;
    $m\ldots$ number of steps in every restart;  $x_0\ldots$ nonzero first guess;
    $k\ldots$ number of minimized residual norms at the first cycle.

Initialization: $y = \mathbf{0}$; $r_0 = b - \mathbf{A}x_0$; $\alpha_0 = -\frac{r_0^T b}{\|b\|^2}$; $r_0 = r_0 + \alpha_0 b$; $\beta = \|r_0\|$; $r = r_0$;
    $g_1 = \beta e_1$; $init = 0$.
    while $\|r\|/\|r_0\| > \varepsilon$ do
        do $i = 1, m$
            $\tilde{v} = (\mathbf{A} - by^T)v_i$
            do  $j = 1, i$
                $h_{j,i} = v_j^T \tilde{v}$
                $\tilde{v} = \tilde{v} - h_{j,i}v_j$
            enddo
            if  $i \leq k$ and $init = 0$ then
                $\alpha_i = \frac{b^T \mathbf{A} v_i}{\|b\|^2}$
                $\tilde{v} = \tilde{v} - \alpha_i b$
            endif
            $h_{i+1,i} = \|\tilde{v}\|$,    $v_{i+1} = \tilde{v}/h_{i+1,i}$
            if  $i = k$ and $init = 0$ then
                compute $y \in \mathbb{R}^n$ satisfying $\begin{pmatrix} x_0^T \\ v_1^T \\ \vdots \\ v_k^T \end{pmatrix} y = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix}$
                $init = 1$
            endif
        enddo
        compute $w_m \in \mathbb{R}^m$ minimizing $\|\beta e_1 - \tilde{\mathbf{H}}_m w\|$
        $x_0 = x_0 + \mathbf{V}_m w_m$
        $r = (b - \mathbf{A}x_0)/(1 + y^T x_0)$
        $v_1 = r/\|r\|$
    endwhile
    $x_0 = x_0/(1 + y^T x_0)$
```

In this algorithm we have assumed $x_0 \notin \mathcal{K}_k(\hat{\mathbf{A}}, \hat{r}_0)$.

In comparison with the classical restarted GMRES method, additional storage costs of this algorithm consist of only one more n-dimensional vector ($y \in \mathbb{R}^n$) and during the initial k iterations the conditions α_i. These conditions are obtained with $2n$ multiplications and the updated vector $\tilde{v}$ asks for n more products. The vector y is best obtained by orthogonalizing x_0 against $v_1, \ldots, v_k$. We achieve this by putting

$$\tilde{v}_{k+1} = x_0 - \sum_{j=1}^{k} (v_j^T x_0) v_j$$

and we denote the wanted normalized vector $\tilde{v}_{k+1}$ by v_{k+1}. Then y must satisfy

$$\begin{pmatrix} & \mathbf{I}_k & & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ v_1^T x_0 & \ldots & v_k^T x_0 & \|\tilde{v}_{k+1}\| \end{pmatrix} \begin{pmatrix} v_1^T \\ \vdots \\ v_{k+1}^T \end{pmatrix} y = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \\ \alpha_0 \end{pmatrix}.$$

The simplest choice is

$$y := (v_1, \ldots, v_{k+1}) \begin{pmatrix} & \mathbf{I}_k & & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ -v_1^T x_0 / \|\tilde{v}_{k+1}\| & \ldots & -v_k^T x_0 / \|\tilde{v}_{k+1}\| & 1/\|\tilde{v}_{k+1}\| \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \\ \alpha_0 \end{pmatrix}.$$

The orthogonalization of x_0 costs $2n(k+1)$ products and the computation of y costs as much. At the end of the first cycle we need the number $1 + y^T x_0$ to multiply it with $b - \mathbf{A}x_0$, which costs $2n$ more multiplications and these extra costs return at the end of every restart. In addition, the matrix-vector product with $\mathbf{A} - by^T$ is $2n$ multiplications more expensive than the product with $\mathbf{A}$. At the very end of the process, one has to update x_0 which costs also $2n$ multiplications. Thus if we need C restarts until convergence, the total number of extra multiplications equals

$$3nk + 4n(k+1) + 2n + 4Cn + 2n = (7k + 8 + 4C)n.$$

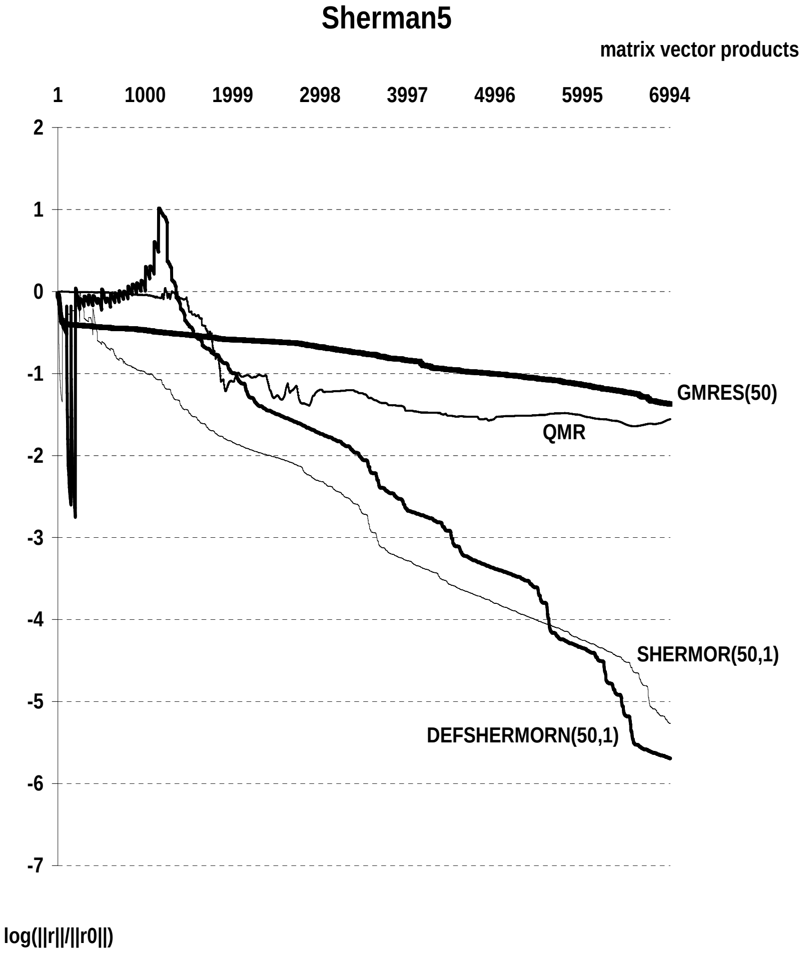

Figure 5.1: QMR, GMRES(50), SHERMOR and DEFSHERMORN applied to sample experiment

5.3 Sample numerical experiment

We conclude this chapter by comparing the effectiveness of all new algorithms from the previous section to a numerical experiment from practice. The matrix is taken from the Matrix Market collection. It has dimension 3312 and 20793 nonzero elements. It is non-normal and satisfies

$$\frac{\|\mathbf{A}\mathbf{A}^T - \mathbf{A}^T\mathbf{A}\|_F}{\|\mathbf{A}\|_F} = 2252.9.$$

The initial guess is zero and as right-hand side we used $b = (0.01, \ldots, 0.01)$ and thus $\|r_0\| \approx 0.5755$. This system is hard to solve for restarted GMRES. Only restart parameters larger than 50 begin to yield convergence. Neither does the QMR method converge satisfactory, although the first 1000 iterations (that is 2000 matrix vector products) seem promising. When we define an auxiliary system with 4 prescribed residual norms, $\|r_1\| := 0.005$, $\|r_2\| := 0.004$, $\|r_3\| := 0.003$ and $\|r_4\| := 0.002$, apply GMRES(50) to this second system and back-transform with the Sherman-Morrison formula, the process converges quickly. The curve for the original system after back-transformation is denoted by SHERMOR(50,4).

Eigenvalue	**of A**	**of $\mathbf{A} - by^T$**
λ_{3306}	0.908005	0.908007
λ_{3307}	0.847002	0.846968
λ_{3308}	0.618836	0.618634
λ_{3309}	0.579574	0.579266
λ_{3310}	0.402658	0.399875
λ_{3311}	0.125445	0.371408
λ_{3312}	0.046925	0.118764

Table 5.1: Smallest eigenvalues of Sherman5 before and after rank-one update with DEFSHERMOR

The spectrum of this matrix (computed with the QZ method) consists of 1631 evenly distributed real eigenvalues in the interval $[597.528315, 1.280845]$ and the remaining eigenvalues are all equal to 1, except for the smallest 7 eigenvalues. Their values can be taken from the second column of the table below. As the last eigenvalue is especially small compared to the rest of the spectrum, one expects this value to be the main factor that hampers convergence, as we have seen for GMRES(50). After 50 iterations the smallest eigenvalue of the generated Hessenberg matrix $\mathbf{H}_{50}$, the smallest Ritz value, equals about 0.67 and is not an approximation of λ_{3312} but rather of λ_{3308}. Still we can, with Algorithm 5.2.4, DEFSHERMOR, modify this smallest Ritz value and hope that the resulting, larger Ritz value also forces the corresponding eigenvalue of the $n \times n$ matrix to be larger. For example, we can compute a vector y such that the smallest eigenvalue 0.67 of the Hessenberg matrix for $\mathbf{A}$ moves to the other end of the spectrum of $\mathbf{A}$ to become the eigenvalue 600 of the Hessenberg matrix for $\mathbf{A} - by^T$ and all other Ritz values are left unchanged. The influence that such Ritz values have on the spectrum of $\mathbf{A} - by^T$ can be described as follows: Large eigenvalues of $\mathbf{A}$ are very much the same for $\mathbf{A} - by^T$ except for the new largest eigenvalue 600, all eigenvalues 1 remain, but the seven small eigenvalues $\mathbf{A}$ are modified. This is shown in the last column of Table (5.1). The smaller the eigenvalue, the more it has been modified and above all, the very last one has been considerably enlarged, it is about 10 times larger. Moreover, the resulting process does not stagnate anymore as can be seen in Figure 5.2.

If we apply the other deflation technique (DEFSHERMORN) to this problem, we must realize this technique was meant for nearly normal matrices whereas Sherman5 is far from normal. Nevertheless we can try to do something: With the smallest Ritz value 0.67 we have a (poor) approximation of λ_{3308}. Because $e_{3308}^T \mathbf{S}^H b$ yields in this case a value of about 0.547, one expects to be able to shift λ_{3308} to somewhere in the middle of the spectrum (about 50) when $\rho := -100$ in (4.11). The smallest eigenvalues of $\mathbf{A} - by^T$ with this choice are seen Table (5.2). They are less changed than with DEFSHERMOR, which corresponds to the theory. But the non-normality of the matrix prevented the theory from working concerning modification of the eigenvalue λ_{3308}, it is still present. The smallest eigenvalue has become essentially larger, which accelerates the system. The smallest Ritz value 0.4 has become the value 2.934199.

This is an example where local minimization spoils the slow convergence of GMRES(50). Even minimization of all 50 Givens sines during the first cycle does not

Eigenvalue	**of A**	**of A** $- by^T$
λ_{3306}	0.908005	0.908005
λ_{3307}	0.847002	0.847033
λ_{3308}	0.618836	0.658457
λ_{3309}	0.579574	0.615082
λ_{3310}	0.402658	0.571775
λ_{3311}	0.125445	0.401679
λ_{3312}	0.046925	0.120158

Table 5.2: Smallest eigenvalues of Sherman5 before and after rank-one update with DEFSHERMORN

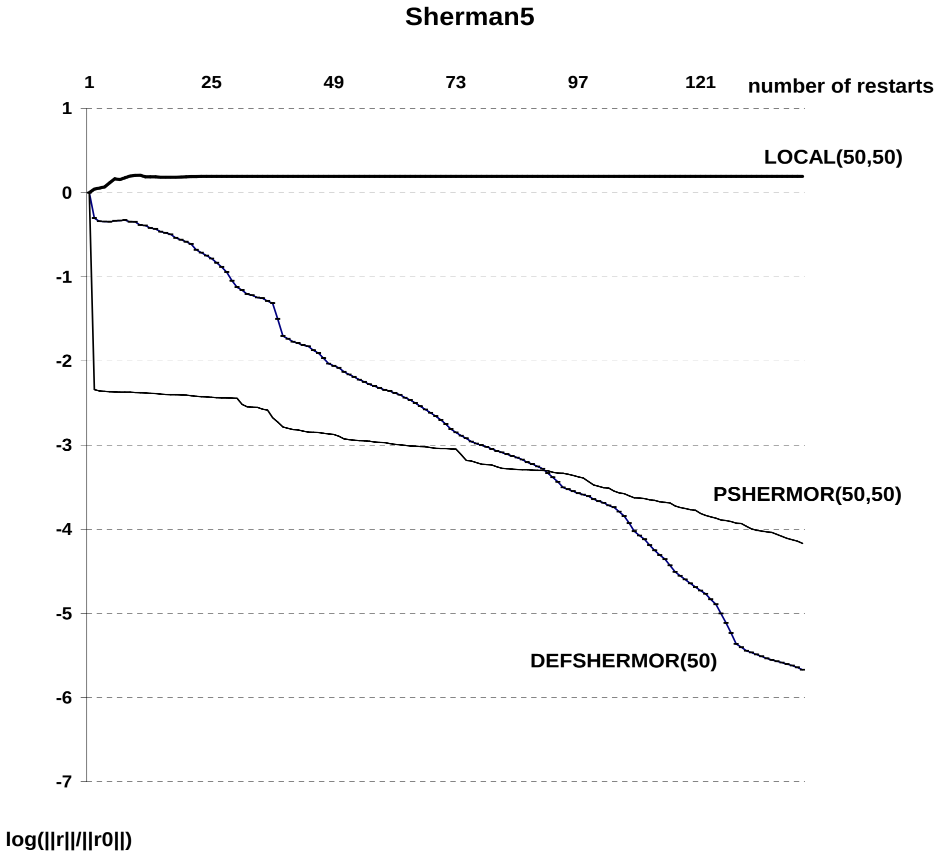

Figure 5.2: DEFSHERMOR, PSHERMOR and local minimization applied to sample experiment

yield a system that converges faster than the original system. In contrary, iterates after back-transformation converge even slower than the iterates of GMRES(50) (compare with Figure 5.1).

On the other hand, Givens sine minimization by means of preconditioning, as proposed in PSHERMOR, is able to accelerate convergence. As seen in previous examples with this algorithm, the exact number of minimizations per cycle has an important influence on the process. The results we obtained by minimizing the first 35 sines at every cycle, are seen in Figure 5.2. Typically, the minimization works best at the beginning of the process, later on convergence is moderate.

Summarizing, we observe that for this random problem from practice, with a sparse, non-symmetric and non-normal matrix of dimension 3312, all the techniques proposed in this thesis accelerate the convergence of restarted GMRES with the exception of local minimization. Our experience has shown that the SHERMOR procedure is the most powerful tool to overcome stagnation in general. On the other hand, the preconditioning technique PSHERMOR gives smoother convergence curves. When confronted with problems that are known to be hampered by unfavorable spectral properties, the DEFSHERMOR algorithm is able to improve these properties and the same can be achieved with DEFSHERMORN when the system matrix is close to normal.

From these observations it seems worth to undertake further investigation of our acceleration techniques. Apart from open questions pointed out in the preceding chapters, we want to emphasize that we focussed in this thesis on a very specific exploitation of the Sherman-Morrison formula, namely rank-one update with one of the rectangular matrices being equal or close to the right-hand side. As already this choice appeared to be able to overcome stagnation in several ways, we expect many other applications of small rank update with the Sherman-Morrison formula to be effective in the context of accelerating restarted projection methods. It would be very interesting to work this out.

Bibliography

[1] Arnoldi WE. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quart. Appl. Math.* 1951; 9: 17–29.

[2] Arioli M, Pták V, Strakoš Z. Krylov Sequences of maximal length and convergence of GMRES. *B.I.T.* 1998; 38(4): 636–643.

[3] Baglama J, Calvetti D, Golub GH, Reichel L. Adaptively Preconditioned GMRES Algorithms. *SIAM J. Sci. Comput.* 1998; 20(1): 243–269.

[4] Benzi M, Tůma M. A sparse approximate inverse preconditioner for nonsymmetric linear systems. *SIAM J. Sci. Comput.* 1998; 19(3): 968–994.

[5] Benzi M, Gander MJ, Golub GH. Optimization of the Hermitian and Skew-Hermitian splitting iteration for saddle point problems. *B.I.T.* 2002; 43(1): 001–013.

[6] Bunch JR, Nielsen CP, Sorensen DC. Rank-One Modification of the Symmetric Eigenvalueproblem. *Numer. Math* 1978; 31: 31-48.

[7] Burrage K, Erhel J, Pohl B. Restarted GMRES Preconditioned by Deflation. *Journal of Computational and Applied Mathematics* 1996; 69: 303–318.

[8] Chan TF, Wan WL. Analysis of Projection Methods for Solving Linear Systems with Multiple Right-hand sides. *Technical report CAM-94-26* 1994; University of California at Los Angeles, Departement of Mathematics, Los Angeles, CA.

[9] Chapman A, Saad Y. Deflated and augmented Krylov subspace techniques. *Numerical Linear Algebra with Applications* 1997; 4(1): 043–066.

[10] Crivelli L, Farhat C, Roux FX. Extending substructure Based Iterative Solvers to Multiple Load and Repeated Analyses. *Technical report* 1993, Center for Space Structures and Controls, CO.

[11] De Sturler E. Truncation strategies for optimal Krylov subspace methods. *SIAM J. Numer. Anal.* 1999; 36(3): 864–889.

[12] Dongarra JJ, Sorensen DJ. A Fully Parallel Algorithm for the Symmetric Eigenvalue Problem. *SIAM J. Sci. and Stat. Comp.* 1987; 8: 139–154.

[13] Duintjer Tebbens EJ, Zítko J. Comparing the QMR and the augmented GMRES algorithm. *Proceedings of contributed papers WDS ´01* , J. Šafránková, ed. matfyzpress, Prague 2001; 156–161.

[14] Duintjer Tebbens EJ. An Application of the Sherman-Morrison formula to the GMRES method. *Proceedings of the Conference ,,Conjugate Gradient Algorithms and Finite Element Methods"* Springer-Verlag, Berlin 2004; 69–92.

[15] Eiermann M, Ernst OG. Geometric Aspects in the Theory of Krylov Subspace Methods. *Acta Numerica* 2000; 001–061.

[16] Eiermann M, Ernst OG, Schneider O. Analysis of acceleration strategies for restarted minimal residual methods. *Journal of Computational and Applied Mathematics* 2000; 123: 261–292.

[17] Eirola T, Nevanlinna O. Accelarating with rank-one updates. *Lin. Alg. Appl.* 1989; 121: 511–520.

[18] Eisenstat SC, Elman HC, Schultz MH. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM Journal on Numerical Analysis.* 1983; 2: 345–357.

[19] Elman HC. *Iterative methods for large sparse nonsymmetric systems of linear equations.* Ph.D. Thesis, Computer Science Department, Yale University, New Haven, CT, 1982.

[20] Fiedler M. *Special matrices and their Applications in Numerical Mathematics.* Nijhoff, Dordrecht, 1986.

[21] Fischer PF. Projection Techniques for Iterative Solution of Ax=b with Succesive Right-hand Sides. *Technical report TR-93-90* 1993; ICASE, Hampton, VA.

[22] Fischer B, Reichel L. A stable Richardson iteration method for complex linear systems. *Numer. Math.* 1988; 54: 225–242.

[23] Fokkema DR, Sleijpen GL, Van der Vorst HA. BiCGstab(l) and other hybrid BiCG methods. *SIAM J. Sci. Statist. Comput.* 1992; 13: 631–644.

[24] Freund WF, Nachtigal NM. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numer. Math.* 1991; 60: 315–339.

[25] Freund WF. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems. *SIAM J. Sci. Comput.* 1993; 14(2): 470–482.

[26] Freund WF, Golub GH, Nachtigal NM. Iterative solution of linear systems. *Acta Numerica* 1991; 57–100.

[27] Freund WF, Gutknecht MH, Nachtigal NM. An implementation of the look-ahead Lanczos process for non-Hermitian matrices. *SIAM J. Sci. Comput.* 1993; 14: 137–158.

[28] Golub GH, Van Loan CHF. *Matrix Computations.* The John Hopkins University Press, Baltimore, MD, 1984.

[29] Goossens S, Roos D. Ritz and Harmonic Ritz Values and the convergence of FOM and GMRES *Numerical Linear Algebra with Applications* 1999; 6: 281–293.

[30] Greenbaum A, Strakoš Z. Matrices that generate the same Krylov Residual Spaces. In *Recent Advances in Iterative Methods, G.H. Golub, A. Greenbaum and M. Luskin, eds.* 1994; 50: 95–118.

[31] Greenbaum A, Pták V, Strakoš Z. Any nonincreasing convergence curve is possible for GMRES. *SIAM J. Matrix Anal. Appl.* 1996; 17(3): 465–469.

[32] Gutknecht HM, Strakoš Z. Accuracy of two three-term and three two-term recurrences for Krylov subspace solvers. *SIAM J. Matrix Anal. Appl.* 201; 22: 213–229.

[33] Householder AS. *The Theory of Matrices in Numerical Analysis.* Blaisdell, New York, 1964.

[34] Hestenes MR, Stiefel E. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.* 1952; 49: 409–436.

[35] Huhtanen M, Nevanlinna O. Minimal decompositions and iterative methods. *Numer. Math.* 2000; 86: 257–281.

[36] Ipsen ICF. A Different Approach to Bounding the Minimal Residual Norm in Krylov Methods. *Technical report, CRSC, NC* 1998 ; Department of Mathematics, North Carolina State University, Raleigh.

[37] Ipsen ICF. Expressions and bounds for the GMRES residual. *B.I.T.* 2000; 40: 524–535.

[38] Jea KC, Young DM. Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods. *Lin. Alg. Appl.* 1980; 34: 159–194.

[39] Joubert W. A robust GMRES-based adaptive polynomial preconditioning algorithm for nonsymmetric linear systems. *SIAM J. Sci. Comput.* 1994; 15: 427–439.

[40] Kane JH, Keyes DE, Prasad KG. GMRES for sequentially Multiple Nearby Systems. *Technical report* 1995; Old Dominion University.

[41] Kharchenko SA, Yeremin AY. Eigenvalue translation based preconditioners for the GMRES(k) method. *Numer. Linear Algebra Appl.* 1995; 2: 51–77.

[42] Krylov AN. On the numerical solution of the equation by which the frequency of small oscillations is determined in technical problems. *Isz. Akad. Nauk SSSR Ser. Fiz.-Math.* 1931; 4: 491–539.

[43] Knobloch P, Tobiska L. The P_1^{mod} element: A new nonconforming finite element for convection-diffusion problems. *SIAM J. Numer. Anal.* 2003; 41(2): 436–456.

[44] Lanczos C. Solution of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards* 1952; 49: 33–53.

[45] Lehoucq R. *Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration.* Ph.D. Thesis, Rice University, Houston, 1995.

[46] Liesen J, Rozložník M, Strakoš Z. Least squares residuals and minimal residual methods. *SIAM J. Sci. Comput.* 2002; 23(5): 1503–1525.

[47] Liesen J, Tichý P. The worst case GMRES for normal matrices. Accepted for publication in *B.I.T.*, 2003.

[48] Liu ZA, Parlett BN, Taylor DR. A Look-Ahead Lanczos Algorithm for Unsymmetric Matrices. *Mathematics of computation* 1985; 44(169): 105–124.

[49] Matrix Market, http://math.nist.gov/MatrixMarket/. The matrix market is a service of the Mathematical and Computational Sciences Division of the Information Technology Laboratory of the National Institute of Standards and Technology.

[50] Morgan RB. A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Anal. Appl.* 1995; 16(4): 1154–1171.

[51] Morgan RB. Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations. *SIAM J. Matrix Anal. Appl.* 2000; 21(4): 1112–1135.

[52] Nicolaides RA. Deflation of conjugate gradients with applications to boundary value poblems. *SIAM J. Numer. Anal. Appl.* 1987; 24: 355–365.

[53] Paige CC, Saunders M. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.* 1975; 12: 617–629.

[54] Paige CC, Strakoš Z. Residual and backward error bounds in minimum residual Krylov subspace methods. *SIAM J. Sci. Comput.* 2002; 23(6): 1899–1924.

[55] Rozložník M. *Numerical stability of the GMRES method.* Ph.D. Thesis, Academy of Sciences of the Czech Republic, 1997.

[56] Saad Y, Schultz MH. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* 1986; 7: 856–869.

[57] Saad Y. *Numerical Methods for Large Eigenvalue Problems.* Halstead Press, New York, 1992.

[58] Saad Y. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Stat. Comput.* 1993; 14: 461–469.

[59] Saad Y. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, 1996.

[60] Saad Y. Analysis of augmented Krylov subspace methods. *SIAM J. Matrix Anal. Appl.* 1997; 18(7): 435-449.

[61] Saad Y. Further Analysis of Minimum Residual Iterations. *Numer. Linear Algebra Appl.* 2000; 7: 67–93.

[62] Sonneveld P. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* 1989; 10(1): 36–52.

[63] Sorensen D.C. Implicit application of polynomial filters in k-step Arnoldi method. *SIAM J. Matrix Anal. Appl.* 1992; 13(1): 357–385.

[64] Strakoš Z, Tichý P. On error estimation in the conjugate gradient method and why it works in finite precision. *Electronic Transactions on Numerical Analysis (ETNA)* 2002; 13: 56–80.

[65] Tebbens JD, Zítko J. Adaptivní předpodmínění metody GMRES. *Proceedings of the conference ,,Programy a algoritmy numerické matematiky 11"* Matematický Ústav AV ČR, 2002; 293-311.

[66] Tichý P. *O některých otevřených problémech v krylovovských metodách.* Ph.D. Thesis, Charles University Prague, 2002.

[67] Van der Vorst HA. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.* 1992; 13: 631–644.

[68] Van der Vorst HA, Vuik C. The superlinear convergence behavior of GMRES. *Journal of Computational and Applied Mathematics* 1993; 48: 327–341.

[69] Van der Vorst HA, Vuik C. GMRESR: A family of nested GMRES methods. *Numer. Linear Algebra Applications* 1994; 1: 369–386.

[70] Vinsome PKW. ORTHOMIN, an iterative method for solving sparse sets of simultaneous linear equations. *Proc. Fourth Symposium on Reservoir Simualtion, Society of Petroleum Engineers of AIME* 1976; 149–159.

[71] Wagner B. *Kurze Rekursionen bei präkonditionierten CKS-Verfahren.* Internal report 54/95, University of Karlsruhe, Computer center, Postfach 6980, D-76128 Karlsruhe, Germany 1995. Diploma thesis.

[72] Walker HF. Implementation of the GMRES method using Householder transformations. *SIAM J. Sci. Stat. Comput.* 1988; 9: 52–163.

[73] Walker HF. Implementations of the GMRES method. *Computer Physics Communication* 1989; 53: 311–320.

[74] Walker HF, Zhou L. A simpler GMRES. *Numer. Linear Algebra Appl.* 1994; 1: 571–581.

[75] Weiss R. *Convergence behavior of generalized conjugate gradient methods.* Ph.D. Thesis, University of Karlsruhe, 1990.

[76] Weiss R. Minimization properties and short recurrences for Krylov subspace methods. *Electronic Transactions on Numerical Analysis (ETNA)* 1994; 2: 57–75.

[77] Young DM. *Iterative Solution of Large linear Systems.* Academic Press, New York, San Francisco, London, 1971.

[78] Zítko J. Generalization of convergence conditions for a restarted GMRES. *Numer. Linear Algebra Appl.* 2000; 7: 117–131.

[79] Zítko J. Using successive approximations for improving the convergence of GMRES method. *Applications of mathematics* 1998; 43(5): 321–350.