Cubo Matemática Educacional
Vol. 5., N° 1, Enero 2003

Computation of Invariant Subspaces of Large and Sparse Matrices

Jan Brandts
Mathematical Institute
Universiteit Utrech
P.O. Box 80.010. 3508 TA, Utrech
The Netherlands

Abstract

In this paper we present algorithms that approximate invariant subspaces of a linear operator on a finite (but very high) dimensional space. First we will show, following Stewart and Sun (1990), that a small-enough error in an approximation for such a subspace, is the solution of a generalized Riccati equation. The solution of this Riccati equation will be approximated by Picard iterations, and we comment on convergence speed, costs and interrelations. As a by-product, we give an overview of iterative methods to solve a Sylvester equation with one large and sparse and one small and dense matrix.

Next, we will accelerate the Picard iterations in the same way as the Raleigh Quotient Iteration accelerates Shift and Invert. This results in Newton-like methods for the generalized algebraic Riccati equation. Additionally, so-called subspace acceleration will be applied, in the same way as the Arnoldi method is a subspace acceleration of the Power Method. Finally, forced by efficiency considerations, we consider the effects of inexact solution of equations at each level of the nested algorithms.

For invariant subspaces of dimension one, one of the resulting algorithms is the Jacobi-Davidson by Sleijpen and Van der Vorst (1996).

Contents

1. Motivation

The computation of invariant subspaces of a large and sparse matrix has attracted more and more attention in the recent history. The main reason for this is, that the computation of single eigenvectors can be highly numerically unstable if the corresponding eigenvalue is either not simple, or very close to another eigenvalue (Cf. Section 7.2 in [8] and Chapters IV and V in [18]). Indeed, the distance between a simple eigenvalue λ and its nearest neighboring eigenvalue appears in the denominator of upper bounds for the quality of approximations of λ.

A similar result holds for spectral invariant subspaces of hermitian matrices, i.e., the distance between the eigenvalues belonging to the one invariant subspace and those of the complementary invariant subspace, appears in the denominator of upper bounds for the approximation quality of (each one of) the invariant subspaces. In the unsymmetric case the situation is similar, though more complicated. The corresponding distance is called *separation*, a concept for which we refer to [19] and Section 3 of this paper.

To get a better understanding of the action of the linear operator and to get insight in its spectral structure, it is clarifying to try to cluster the eigenvalues in groups that are well-separated from each other and to calculate the invariant subspaces belonging to each of the clusters in a stable and efficient manner. Afterwards, computations for the restrictions of the operators to each of the invariant subspaces could be performed in a relatively stable manner.

1.1 Brief overview of current methods for subspace computations

For small and medium sized matrices, algorithms like the QR algorithm, Jacobi Rotations, Subspace Iteration, and Divide and Conquer methods are available that compute invariant subspaces often to great satisfaction. We refer to the third edition [8] of Golub and Van Loan (1996) for details on those methods and for a large collection of references to the literature. For a treatment of perturbation theory for invariant subspace approximations from the numerical point of view, Chapter V of the book [18] of Stewart and Sun (1990) is indispensable. The book [7] by Gohberg, Lancaster and Rodman (1986) is a standard reference for a more theoretical treatment of perturbation theory.

For large and very large matrices, the situation is much less satisfactory, as convergence and stability of many of the algorithms are not yet well-understood. At present, the most competitive methods to calculate invariant subspaces of large sparse matrices seem to be

- the Inexact Block Rayleigh Quotient Iteration (IBRQI), which, for the Hermitian case, was analyzed by Smit (1997) in his thesis [14] and in [15] of Smit and Paardekooper (1999); the general case was considered in [11] by Lai, Lin and Lin (1997);
- the Implicitly Restarted Arnoldi (IRA) as developed by Sorensen (1992) in [16] and well-documented by Lehoucq (1995) in his thesis [12];
- Jacobi-Davidson style QR (JDQR) algorithm by Fokkema, Sleijpen, Van der Vorst [6], based on the Jacobi-Davidson algorithm [20] by Sleijpen and Van der Vorst (1996), and well-documented by Fokkema (1996) in his thesis [5].

The recent publication date of all those references indicates that the topic is very much alive, and moreover, that the last words on this topic have not yet been spoken. Note that all three methods are based on Ritz-Galerkin projection.

1.2 Putting this exposition in context

The first of the three methods in the list above works with invariant subspaces as inseparable entities; the other two use a more flexible approach and let the subspaces expand and collapse. The methods in this paper combine different aspects of the three, in the sense that also here we work with subspaces as inseparable entities and here too, Sylvester equations are iteratively solved, as in [14]. Also, we incorporate subspace acceleration as in the IRA and JD method. To be more to the point, consider the following sketch of the IBRQI.

Given A, and X_0 with $X_0^H X_0 = I$. Set $k = 1$. $M_0 = X_0^H A X_0$.

```
While not satisfied, iterate ...
    Solve $Y_k$ from $AY_k - Y_k M_{k-1} = X_{k-1}$
    $X_k R_k = Y_k$ (QR-decomposition, orthonormalization of the columns of $Y_k$)
    $M_k = X_k^H A X_k$ (Ritz-Galerkin projection)
    $k = k + 1$
end
```

The major problem with this iteration is that the Sylvester equation in the first line in the **while** loop becomes harder and harder to solve as the eigenvalues of M_k converge to eigenvalues of A. Indeed, it is well-known that a Sylvester equation is singular if both matrices share an eigenvalue, so the conditioning of the equation becomes worse and worse as the algorithm converges, which may lead to stagnation. This problem can be approached, as in the Jacobi-Davidson method

[20], by solving for orthogonal corrections to the current approximation instead of a completely new approximation, after which only the action of A restricted to the orthogonal complement of the current invariant subspace approximation is needed. Loosely speaking, one could say that the (near) singularity of the equation is "projected away". In the single-vector case, it has been observed that this can greatly improve the performance of the method. If this single-vector case is enhanced with subspace acceleration, we arrive at the Jacobi-Davidson (JD) algorithm.

As we will show, JD neglects a mild non-linearity that may be important in case the matrix under consideration is non-normal, since exactly the non-normality is represented in the non-linear term. We will investigate the effects of extending JD by including this non-linearity, which turns the Sylvester equations to solve into generalized algebraic Riccati equations. The main drawback of considering special algorithms for highly non-normal eigenvalue problems is, of course, that standard perturbation theory tells us that the results produced by any numerical algorithm are likely to be very inaccurate. So the question of why to consider refinements of algorithms especially for those unstable cases is a legitimate one.

Since, at present, there is no satisfying block version of the JD algorithm for invariant subspaces (JDQR is a repeated single-vector algorithm), we consider our algorithms a useful contribution to the existing literature. Apart from that, stressing the link between the eigenproblem and the corresponding generalized algebraic Riccati equation, is likely to make this paper interesting for people from both research communities.

Similar to both IBRQI and JD(QR), we will also pay attention to the inexact solution of correction equations, which, in most practical situations is unavoidable. The algorithms derive much of their strength from the fact that full accuracy for the inner iteration is not often needed, or can be compensated for by an expanded search space.

1.3 Outline and discussion of this paper

We will deal with the stable and efficient computation of invariant subspaces, and the algorithms to be presented are based on the iterative solution of a generalized algebraic Riccati equation, in which the unknown P is an $(N-k)\times k$ matrix satisfying

$$BP - PM = PG^H P - C. \tag{1}$$

We assume in efficiency considerations that $k << N$, although theoretically this assumption is redundant. The solution P is strongly related to the error in an initial approximation X of an invariant subspace. The matrices C and G are of the same size as P, while B and M are given and square. All four of them depend implicitly on X. It will be reviewed in Sections (2.1) and 2.2 how (1) can be derived. Since finding the solution of (1), which uniquely exists if $||G||$ and $||C||$ are small enough with respect to the separation between B and M (Cf. Section 3), is equivalent to finding the invariant subspace, solving (1) is a non-trivial task.

There is a large and varied tradition in solving Riccati equations, especially in the fields of differential equations, differential geometry, and control theory. In those fields, however, one is mainly interested in special cases like the so-called Lyapunov equation, in which $B^H = M$. As a consequence, one typically focuses on the computation of a large part (e.g., half) of the eigendata, which drastically restricts the size of the problem that is possible to tackle in practice. Less theory seems to be available for the general case and on the case that we study in this paper, i.e., with a large matrix B for which matrix-vector multiplication is relatively inexpensive, and a small matrix M.

In the differential equations community, one typically studies solutions of the equation

$$\frac{\partial P}{\partial t} = BP - PM - PG^H P + C, \tag{2}$$

and clearly, equation (1) investigates the critical points of 2. An important observation is that traditional solution methods for algebraic Riccati equations work in the opposite direction of what we will try to pursue here: they solve the corresponding eigenvalue problem in order to get solutions of the Riccati equation, while we try to solve the eigenproblem using approximate solutions of the Riccati equation. We refer to the book [2] of Bittanti, Laub and Willems (1991) for an overview of the history of the Riccati equation, aspects from differential geometry and numerical algorithms, as well as for a large bibliography.

1.3.1 Four simple iteration schemes

As an initial attempt to solve (1), we propose to use Picard iteration to iterate to a fixed point. For this, we have to choose, basically, which appearances of P in 1 we will replace by P_{n-1} and which by P_n. Although it might be possible to consider P in the factor $G^H P$ to be an unknown of such a successive substitution process, we will not do so in this paper. This reduces the amount of possibilities to four, which we can classify in two groups of two as follows,

- Treat the quadratic form *explicitly* as $P_{n-1}G^H P_{n-1}$ or *implicitly* as $P_n G^H P_{n-1}$.
- Solve a linear system with matrix B or M, or solve a Sylvester equation with both matrices B and M.

Note that solving a system with M does not need a mathematical treatment different than the one for solving a system with B. We will stress this once more in Section 3.

The Picard iteration from those four possibilities that is numerically the most expensive per iteration step, is the successive substitution (3) below. It treats the quadratic term implicitly and solves a (different) Sylvester equation in each step,

$$BP_n - P_n\left(M + G^H P_{n-1}\right) = -C. \tag{3}$$

The approach based on iteration (3) will be pursued in Section 3.1. In Section 3.2, we study the counterpart of (3) with respect to the amount of numerical work per iteration step,

$$P_n M = \left(B - P_{n-1}G^H\right)P_{n-1} + C, \tag{4}$$

in which we only once need to compute the inverse of a small $k \times k$ matrix and in which in each iteration a multiplication with B is required. Even though we may expect that this iteration converges slower than (3), it is worthwhile considering it because each iteration step is very cheap compared to solving a Sylvester equation, as in (3). For completeness, theorems on the remaining two successive substitutions will be proved, but in less detail than the ones (3) and (4).

In Section 3.4 we will comment on the results. In Section 6, the Picard iterations are illustrated in examples in which A is a two-by-two matrix.

1.3.2 Computable error bounds for the eigenvalues

If we assume that P is a solution of the generalized Riccati equation (1), it can be shown (and it will be, in Section 2) that the eigenvalues of $M_\infty := M + G^H P$ are eigenvalues of A. Since we proposed to approximate (1) iteratively by a sequence (P_n), and since the initial approximation M is explicitly known, it is possible to compute the eigenvalues of the current approximation $M_n := M + G^H P_n$ along the way. Moreover, if we are able to find bounds for $\|P - P_n\|$, then, since

$$M_\infty = M_n + G^H(P - P_n), \tag{5}$$

we can use the Bauer-Fike Theorem or the Henrici Theorem to find upper bounds for the error in the spectrum of $M + G^H P_n$ with respect to the spectrum of M_∞. Since in many applications the dimensions of the matrix M are small, this is not merely an academic result: it is indeed possible to say something about the conditioning of an eigenvector basis of M_n and also about its deviation from normality, which are both quantities that are present in the bounds. We will discuss this in Section 3.3. See also Section 6.4 for a simple example.

1.3.3 Solving Sylvester equations

A general account on solution methods for Sylvester equations is given in Section 4. We will show in Section 4.2.1 that simple classical iteration schemes connect the implicit and explicit Picard iterations. The methods of choice for a Sylvester equation with one large and sparse matrix and one small and dense matrix, however, are (based on) Krylov subspace methods, as we will discuss in Sections 4.2.3 and 4.3. They will be used to solve an extra large system of linear equations that models the action of the Sylvester operator. In Picard iterations in which Sylvester equations with many right-hand sides have to be solved, such as in the explicit approach, information of the previous iterations might be used in the current iteration, for example by keeping (some of) the basis vectors of Krylov subspaces in memory.

1.3.4 Acceleration of the Picard iterations in a Newton-like manner

Given an initial approximation for an invariant subspace X_0, each one of the successive substitutions of Section 3 produces a sequence P_n converging to a matrix P that represents the error in X_0. Having approximations of this error available, we can, during the iteration, construct new approximations X_n that are likely to be better than the original approximation X_0. It is then possible to construct new matrices B, M, C and G^H with respect to X_n once a while or in each iteration step, and solve for the new error P. Since one of the hence accelerated Picard iterations is equivalent to the Newton method, we expect that this will lead to better convergence. Note however that the Newton method is generally not very efficient if the iteration is started to far away from the root. Some observations on this issue will be made in Section 5, and in Section 6 we present an easy example.

1.3.5 Relation to the Jacobi-Davidson algorithm

In Section 5, some special attention is paid to neglecting the quadratic term in (1) completely and solving (to full or less precision), the linear(ized) correction equation

$$B\hat{P} - \hat{P}M = -C, \tag{6}$$

which obviously yields an approximation $\hat{P}$ of P. Then, a new approximation of the invariant subspace can be formed and new matrices B, M, G and C computed, after which the step (6) can be repeated. In the course of this iteration, a sequence X_n of approximate invariant subspaces is formed, that will hopefully converge.

Clearly, this method can be referred to as an inexact Newton method for the Riccati equation.

Remark 1.1 Already in 1846, Jacobi [10] described a very similar method to approximate an eigenpair of a diagonally dominant matrix. After each linear correction step (6), of which he approximated the solution by two steps Jacobi iteration; however, he only updated the approximation for M. For more details, see also [20] and the references therein. ◊

It is possible to accelerate the algorithm even further by selecting from all previous spaces $X_0, \cdots, X_{n-1}$ a suitable linear combination with desirable properties in a Ritz-Galerkin like manner. For subspaces of dimension one, this gives the Jacobi-Davidson algorithm by Sleijpen and Van der Vorst [20], while for subspaces of larger dimension, it seems to be a useful generalization to invariant subspaces; one that is more natural than applying Jacobi-Davidson to a block vector.

Remark 1.2 The Davidson algorithm [4] for approximating an eigenpair μ, u with $Au = \mu u$, also incorporates subspace acceleration, but not with the correction equation (6), which takes place in the orthogonal complement of the current invariant subspace approximation, but (approximating A by its diagonal D) in the unrestricted space. Since in the latter case, the singularity of $A - \mu I$ in the direction of u is not properly taken care of, this can cause severe problems. See [20] for further details. ◊

1.3.6 Implementation matters

In order to test the resulting algorithms, it is advisable to transform the systems to solve on a more suitable basis. This is done in Section 7, in which also some observations with respect to the stability of the solution of the Sylvester equations are made. These observations originally stem from [20] and have been proved to be of great practical importance.

1.3.7 Numerical experiments

In Section 8 we will give some results of numerical experiments. We tested some of our algorithms on standard test matrices from the Matrix Market collection [13] and other notoriously difficult eigenproblems like the one for the Hilbert matrix (see Section 8.1.1). We consider algorithms with and without acceleration, and use different tolerances for the inner iterations involved. The results seem very satisfactory, but it should be noted that it is hard to make a comparison with the other algorithms because of their complicated nested structure.

2 Invariant subspaces and algebraic Riccati equations

We will now study invariant subspaces in the setting of [18]. We will use, adapt and extend their notations and results, which were used to prove perturbation theorems for invariant subspaces. In this section, we will put the emphasis on algorithmical aspects.

Remark 2.1 Throughout the paper, and if no confusion is expected, we will identify the columnspan of a matrix with the matrix itself, i.e. we talk about the matrix X as well as the subspace X.

2.1 Preliminaries

Let X be a unitary matrix approximating an invariant subspace $\hat{X}$. Consider the projection M of A on X and let R be the corresponding residual. Explicitly, this means that

$$X^H X = I, \quad M = X^H A X \quad \text{and} \quad R = AX - XM. \tag{7}$$

Let Y be a unitary matrix spanning the orthogonal complement of X. Then, $(X|Y)$ is unitary, and transformation of A from the standard basis to the basis given by the columns of $(X|Y)$ results in the definition of the blocks M, B, C and G^H in

$$A(X|Y) = (X|Y)\left[\begin{array}{c|c} M & G^H \\ \hline C & B \end{array}\right]. \tag{8}$$

Note, by comparing columns, that $AX = XM + YC$ so $C = Y^H R$. Also, $AY - YB = XG^H$, which means that XG^H is the residual corresponding to Y. In case A is hermitian, $C = G^H$, and Y is as good an approximation of an invariant subspace as X. We will now show how to find an invariant subspace assuming that the approximation X of $\hat{X}$ is "good enough". The main ideas of what follows can be found, for example, in Chapter V of [18]. We wish to repeat them here and provide some additional explanations.

2.2 Derivation of the generalized algebraic Riccati equation

Let $\hat{Y}$ be such that $(\hat{X}|\hat{Y})$ is an $N \times N$ unitary matrix. Then, because $\hat{X}$ is an invariant subspace, transformation of A to the basis $(\hat{X}|\hat{Y})$ leads to

$$A(\hat{X}|\hat{Y}) = (\hat{X}|\hat{Y})\left[\begin{array}{c|c} \hat{M} & \hat{G}^H \\ \hline 0 & \hat{B} \end{array}\right]. \tag{9}$$

for certain $\hat{M}, \hat{G}$ and $\hat{B}$. Now, $(\hat{X}|\hat{Y})$ can be constructed from X and Y as follows.

First assume that $H := X^H\hat{X}$ and $K := Y^H\hat{Y}$ are invertible (this assumption will later be translated as "$\hat{X}$ and X are close enough"). Then write

$$(\hat{X}|\hat{Y}) = (X|Y)\left[\begin{array}{c|c} X^H\hat{X} & X^H\hat{Y} \\ \hline Y^H\hat{X} & Y^H\hat{Y} \end{array}\right] =: (X|Y)\left[\begin{array}{c|c} H & QK \\ \hline PH & K \end{array}\right], \tag{10}$$

in which, clearly, $P = Y^H\hat{X}H^{-1}$ and $Q = X^H\hat{Y}K^{-1}$. As a product of two unitary matrices, the most right matrix in (10) is unitarity as well, which leads to the relations

$$\hat{X}^H\hat{X} = H^H(I + P^HP)H = I \quad \text{and} \quad \hat{Y}^H\hat{Y} = K^H(I + Q^HQ)K = I, \tag{11}$$

and

$$\hat{X}^H\hat{Y} = H^H(P^H + Q)K = 0. \tag{12}$$

Since we assumed H and K to be invertible, we can conclude from (12) that $Q = -P^H$, after which it appears that $H = (I + P^HP)^{-\frac{1}{2}}$ and $K = (I + PP^H)^{-\frac{1}{2}}$. This results in

$$\hat{X} = (X + YP)(I + P^HP)^{-\frac{1}{2}} \quad \text{and} \quad \hat{Y} = (Y - XP^H)(I + PP^H)^{-\frac{1}{2}}. \tag{13}$$

We will now determine P such that $(\hat{X}|\hat{Y})$ realizes the block Schur form (9). The only requirement is to choose P such that $\hat{Y}^HA\hat{X} = 0$, which, in terms of the blocks M, B, C and G in (9), is equivalent to the condition that P satisfies the following generalized algebraic Riccati equation,

$$BP - PM = PG^HP - C. \tag{14}$$

This equation might have several solutions. In the following section we will comment on which solution gives rise to the invariant subspace $\hat{X}$ closest to X. First we quote a result from [18].

Theorem 2.2([18]) *Suppose P satisfies (14), then,* $\sigma(M + G^HP) \subset \sigma(A)$.

Proof. By definition, $\hat{M} = \hat{X}^HA\hat{X}$. Substituting $\hat{X}$ from (13) and using the relation $BP = PM + PG^HP - C$ obtained from (14), we arrive at

$$\hat{M} = (I + P^HP)^{-\frac{1}{2}}(M + G^HP)(I + P^HP)^{-\frac{1}{2}}, \tag{15}$$

which means that, since $\sigma(\hat{M}) \subset \sigma(A)$, also the eigenvalues of $M + G^HP$ are eigenvalues of A. □

Remark 2.3 (Special case: $k = 1$) If we put $k = 1$, the theory of Section 2.2 gives us a way to transform an approximation x_1 of an eigenvector of A into a close-by exact eigenvector. First note that the non-linear equation (14) for the matrix P reduces to an equation for the vector p as follows,

$$(B - \mu I)p = p(g^Hp) - c. \tag{16}$$

Moreover, Theorem 2.2 gives,

$$\lambda = \mu + g^H p. \tag{17}$$

The one-dimensional case is special in the sense that $g^H p$ is a scalar that commutes with p. We can keep this in mind while studying the iterative methods to approximate the solution of (14) in Section 3 to come. See also Section 6 in which $k = 1$ and $n = 2$. ◊

2.3 Stability and convergence of invariant subspaces

We will now discuss the previous section in terms of stability and convergence. First, let U_1 and V_1 be $n \times k$ matrices with orthonormal columns, and let $(U_1|U_2)$ and $(V_1|V_2)$ be unitary. Then we define, as is usually done, the gap $\theta(\cdot,\cdot)$ between U_1 and V_1 by,

$$\theta(U_1, V_1) := \|U_1^H V_2\| = \|P_{U_1} - P_{V_1}\|, \tag{18}$$

where P_U and P_V are the orthogonal projections on U and V respectively. It is well-known that $\theta(U_1, V_1)$ can be interpreted as the sine of the angle between U_1 and V_1, and therefore, the condition from the previous section that $H := X^H \hat{X}$ should be invertible, is equivalent to the (not very restrictive) condition that the angle between X and $\hat{X}$ should not be $\frac{\pi}{2}$. Moreover, an easy calculation shows that K is invertible if and only if H is.

Each k dimensional invariant subspace of A that is not orthogonal to X, corresponds to (at least one) solution P of the generalized Riccati equation (14). This correspondence is expressed by the left formula in (13). From this formula, we also find an expression for the gap between X and $\hat{X}$ as follows,

$$\theta(\hat{X}, X) := \|\hat{X}^H Y\| = \|P(I + P^H P)^{-\frac{1}{2}}\| \leq \|P\|. \tag{19}$$

One could define the invariant subspace $\hat{X}$ for which $\theta(\hat{X}, X)$ is minimal to be the one "closest" to X. However, it is not clear if the $\hat{X}$ closest to X is the one that corresponds to the minimal norm solution P of (14), regardless what the inequality in (19) may suggest. Nevertheless, a useful theorem can be proved. First, we need to define the separation between two matrices.

Definition 2.4 *Define, on the space of $(N-k) \times k$ matrices, the linear Sylvester operator* $\mathbf{T}$ *associated with B and M, and consequently the separation between the matrices B and M by*

$$\mathbf{T} : Q \mapsto BQ - QM, \quad \operatorname{sep}(B, M) := \inf_{\|Q\|=1} \|\mathbf{T}(Q)\|. \tag{20}$$

Now, suppose that P satisfies (14), then clearly

$$\operatorname{sep}(B, M) = \inf_{\|Q\|=1} \|\mathbf{T}(Q)\| \leq \frac{\|\mathbf{T}(P)\|}{\|P\|} \leq \|G\|\|P\| + \frac{\|C\|}{\|P\|}. \tag{21}$$

Writing $\gamma := \|C\|, \chi := \|G\|$ and $\delta := \text{sep}(B, M)$, we conclude that the norm $p := \|P\|$ of any solution of (14) satisfies

$$\delta p \leq \chi p^2 + \gamma. \tag{22}$$

If $\delta^2 - 4\gamma\chi \leq 0$, then (22) holds for all p, and nothing can be concluded for p from this analysis. If, however, $\delta^2 - 4\chi\gamma > 0$, both roots

$$\tau_\ell := \frac{\delta}{2\chi}\left(1 - \sqrt{1 - \frac{4\chi\gamma}{\delta^2}}\right) \quad \text{and} \quad \tau_r := \frac{\delta}{2\chi}\left(1 + \sqrt{1 - \frac{4\chi\gamma}{\delta^2}}\right) \tag{23}$$

are positive and real, and (22) holds everywhere except on the open interval $\tau := (\tau_\ell, \tau_r)$. In particular this means that there exists no solution P of (14) such that $\|P\| \in \tau$. By showing that $\mathcal{N} : P \mapsto \mathbf{T}^{-1}(PG^HP - C)$ is a contraction on the ball $\mathcal{B} := \{Q \mid \|Q\| \leq \tau_\ell\}$, Stewart proved in [17] that there does indeed exists a unique solution P of (14) in $\mathcal{B}$.

Theorem 2.5 *([17]) Suppose $sep(B, M)^2 - 4\|C\|\|G\| > 0$. Then there exists exactly one solution P of (14) that satisfies*

$$\|P\| \leq \tau_\ell \leq \frac{2\|C\|}{sep(B, M)}. \tag{24}$$

This solution gives rise to an invariant subspace $\hat{X}$ of A such that

$$\theta(\hat{X}, X) \leq \|P\| \leq \tau_\ell \leq \frac{2\|C\|}{\text{sep}(B, M)} \leq \frac{\text{sep}(B, M)}{2\|G\|}. \tag{25}$$

Proof. See Stewart [17] or [18] for details on the first statement. Combining (19), (24) and the condition of the theorem leads to (25). □

Summarizing, the discriminant-like condition $\text{sep}(B, M)^2 - 4\|C\|\|G\| > 0$ provides us with a ball $\mathcal{B}$ in which a unique minimal-norm solution of (14) lives. This ball is directly surrounded by a spherical layer of thickness $\tau_r - \tau_\ell$ that does not contain any other solutions. In Section 3 we will see that the relative thickness of this layer directly influences the convergence speed of Picard iterations for the Riccati equation. Note that the condition also implies that $\text{sep}(B, M)$ is strictly positive, or $\|C\|\|G\| = 0$ and an invariant subspace has been found.

3 Solving the generalized algebraic Riccati equation

Consider the non-linear equation (14), which is known as a generalized algebraic Riccati equation. Since it is equivalent to an eigenproblem, it cannot be solved directly, which necessitates the use of iterative methods. In this section we start with studying four different Picard iterations, and state conditions under which they converge. From the proofs it will, as a side product, become clear how fast they converge.

Remark 3.1 Although for diagonalizable or even Hermitian matrices, some of the results might be simplified and improved, we choose to use a general setting here. In Section 4 we will comment on these special cases. ◊

In Section 3.1 we will consider two Picard iterations that are based on solving a Sylvester equation in each step. In Section 3.2 we will consider the cheaper alternative of inverting a small matrix once (or solving linear systems with it), and applying one large matrix multiplication per iteration step. The proofs of the theorems to follow are variations of a proof in Section 2.4 of [18].

3.1 Solving a Sylvester equation per iteration step

The most sophisticated Picard iteration to approximate the solution of (14) is the following,

$$\text{given} \quad P_0 = 0, \quad \text{iterate} \quad BP_n - P_n\left(M + G^H P_{n-1}\right) = -C, \tag{26}$$

in which in each step a linear Sylvester equation needs to be solved. We present details on solution methods for linear Sylvester equations in Section 4. Note that, as P_n converges to P, the eigenvalues of $M + G^H P_n$ converge to the eigenvalues of interest (Cf. Th. 2.2). Assuming that each iteration step is performed exactly, we can state the following theorem.

Theorem 3.2 *Define the linear operator* $\mathbf{T}$ *on the space of* $(N-k) \times k$ *matrices, and the separation between the matrices* B *and* M *by,*

$$\mathbf{T}(P) = BP - PM, \quad \delta := sep(B, M) := \inf_{\|P\|=1} \|\mathbf{T}(P)\|. \tag{27}$$

Assume that $\delta > 0$. *Moreover, write* $\gamma = \|C\|$ *and* $\chi := \|G\|$. *Then, if*

$$\delta^2 - 4\gamma\chi > 0, \tag{28}$$

the implicit iteration $BP_n - P_n(M + G^H P_{n-1}) = -C$ *is convergent if* $P_0 = 0$, *and, using the notation from (23)*

$$\|P - P_n\| \leq \tau_\ell \left(\frac{\tau_\ell}{\tau_r}\right)^n. \tag{29}$$

Remark 3.3 Note that condition (28),

$$\text{sep}(B, M) - 4\|C\|\|G\| > 0, \tag{30}$$

can be interpreted as a higher dimensional equivalent of the discriminant for the quadratic equation (14) in P. ◊

Proof of Theorem 3.2. The proof consists of three steps. Firstly, we will prove that the sequence of norms $p_n := \|P_n\|$ is bounded. Secondly, we prove convergence of the sequence P_n. Thirdly, we derive our final error bound.
Step I : Since the iteration (26) reads as $P_n = \mathbf{T}^{-1}(P_nG^HP_{n-1} - C)$, for the norms p_n we find,

$$p_0 = 0 \quad \text{and} \quad p_n \leq \frac{\gamma}{\delta} + \frac{\chi}{\delta}p_np_{n-1}. \tag{31}$$

Define the sequence ξ_n by

$$\xi_0 = 0 \quad \text{and} \quad \xi_{n+1} = \frac{\gamma}{\delta - \chi\xi_n}. \tag{32}$$

This sequence is, under the condition (28), well-defined as we will show now. First note that $\xi_n = \phi(\xi_{n-1})$ with

$$\phi(\xi) := \frac{\gamma}{\delta - \chi\xi}, \quad \xi \in [0, \frac{\delta}{\chi}). \tag{33}$$

Clearly, ϕ is strictly increasing. Moreover, the quadratic equation $\phi(\xi) = \xi$ is the one in (22) which, by condition (28), has $\tau_\ell < \frac{\delta}{\chi}$ as smallest positive root. Left from this root, ϕ' has derivative smaller than one. So, the Picard iteration $\xi_n = \phi(\xi_{n-1})$ converges monotonely to τ_ℓ. From (31) and (32) we see that $p_n \leq \xi_n$ for all n, so that,

$$\forall n, 0 \leq \|P_n\| \leq \lim_{k\to\infty} \xi_k = \tau_\ell. \tag{34}$$

Step II : From (26) we find, after some rearranging of terms,

$$\mathbf{T}(P_{n+1} - P_n) = (P_{n+1} - P_n)G^HP_n + P_nG^H(P_n - P_{n-1}), \tag{35}$$

so that, after taking norms and using the bound (34) we get,

$$\|P_{n+1} - P_n\| \leq \frac{\chi}{\delta}\tau_\ell\left(\|P_{n+1} - P_n\| + \|P_n - P_{n-1}\|\right), \tag{36}$$

which, using that $\tau_\ell + \tau_r = \frac{\delta}{\chi}$, results in

$$\|P_{n+1} - P_n\| \leq \frac{\tau_\ell}{\tau_r}\|P_n - P_{n-1}\|. \tag{37}$$

So, P_n is convergent with limit P.
Step III : Using the now established existence of P, we find from (26) and (14) that

$$T(P - P_n) = (P_n - P)G^HP_{n-1} + PG^H(P_{n-1} - P),. \tag{38}$$

so that, after taking norms and using (34) again,

$$\|P - P_n\| \leq \frac{\tau_\ell\chi}{\delta}\left(\|P - P_n\| + \|P - P_{n-1}\|\right). \tag{39}$$

Clearly, this results in

$$\|P - P_n\| \leq \frac{\tau_\ell}{\frac{\delta}{\chi} - \tau_\ell}\|P - P_{n-1}\| = \frac{\tau_\ell}{\tau_r}\|P - P_{n-1}\|. \tag{40}$$

The statement follows now inductively from $\|P - P_0\| = \|P\| \le \tau_\ell$. □

3.1.1 Explicit treatment of the quadratic term

The following Picard iteration is a somewhat simpler than the previous, since the quadratic term is treated explicitly,

$$\text{given} \quad P_0 = 0, \quad \text{iterate} \quad BP_n - P_n M = P_{n-1} G^H P_{n-1} - C. \tag{41}$$

This iteration is, in a general form, considered in Section 2.4 of [18] with the purpose to prove conditions under which (14) has a unique solution. Since some steps in the proof in [18] are not optimal, we will formulate an improved version.

Theorem 3.4 *With the same notations and under the same conditions as in Theorem 3.2, we have that the explicit iteration $BP_n - P_n M = P_{n-1} G^H P_{n-1} - C$ is convergent if $P_0 = 0$, and*

$$\|P - P_n\| \le \tau_\ell \left(\frac{2\tau_\ell}{\tau_\ell + \tau_r} \right)^n. \tag{42}$$

Proof. We will only outline the proof since it contains the same elements as in the proofs of Theorem 3.2 and Theorem 3.5 to come. The successive substitution bounding the norms $p_n := \|P_n\|$ is

$$\xi_0 = 0, \quad \xi_n = \phi(\xi_{n-1}), \quad \text{with} \quad \phi(\xi) = \frac{1}{\delta} \left(\gamma + \chi \xi^2 \right). \tag{43}$$

There is again convergence of ξ_n to fixed point τ_ℓ (Cf.(23)). Since the existence of a fixed point P has already been established in Theorem 3.2, we can immediately compare P and P_n, resulting in

$$\|P - P_n\| \le \frac{2\tau_\ell}{\tau_\ell + \tau_r} \|P - P_{n-1}\|. \tag{44}$$

Since $\|P - P_0\| = \|P\| \le \tau_\ell$, the statement is now proved. □

3.2 Solving a small linear system of equations per iteration step

Another option is to go for the computationally cheapest iteration available from this setting, which means treating the quadratic term explicitly, and solving a system with the matrix M, that we assume to be much smaller than B,

$$\text{given} \quad P_0 = 0, \quad \text{iterate} \quad P_n M = \left(B - P_{n-1} G^H \right) P_{n-1} + C. \tag{45}$$

So, instead of a Sylvester equation with a large and a small matrix, we only have N linear systems with a small $k \times k$ matrix to solve. It will probably pay off to explicitly invert M.

Theorem 3.5 *Write* $\mu = \|M^{-1}\|^{-1}, \beta = \|B\|, \gamma = \|C\|$ *and* $\chi = \|G\|$. *Suppose*

$$\delta := \mu - \beta > 0, \quad \text{and} \quad \delta^2 - 4\gamma\chi > 0. \tag{46}$$

Then the explicit iteration $P_nM = (B - P_{n-1}G^H)\, P_{n-1} + C$ *is convergent if* $P_0 = 0$, *and*

$$\|P - P_n\| \le \tau_\ell \left(\frac{\beta + 2\chi\tau_\ell}{\mu}\right)^n. \tag{47}$$

Proof of Theorem 3.5 Similar as in the proof of Theorem 3.2 we can find a sequence ξ_n majorating the norms $p_n := \|P_n\|$,

$$\xi_0 = 0 \quad \text{and for all n,} \quad p_n \le \xi_n := \frac{\gamma}{\mu} + \frac{\beta}{\mu}\xi_{n-1} + \frac{\chi}{\mu}\xi_{n-1}^2. \tag{48}$$

Under the given conditions, this sequence is well-defined as we will show now. First note that $\xi_n = \phi(\xi_{n-1})$ with

$$\phi(\xi) := \frac{\gamma}{\mu} + \frac{\beta}{\mu}\xi + \frac{\chi}{\mu}\xi^2, \quad \xi \in \mathbb{R}. \tag{49}$$

The quadratic equation $\phi(\xi) = \xi$ is the one in (22). Since ϕ' is increasing for $\xi \ge 0$ and since ϕ intersects $\xi \mapsto \xi$, we have, with τ_ℓ from (23),

$$0 < \phi'(\xi) < \phi'(\tau_\ell) < 1 \quad \text{for all} \quad \xi \in [0, \tau_\ell). \tag{50}$$

So, the successive substitution $\xi_n = \phi(\xi_{n-1})$ converges monotonely to τ_ℓ (Cf.(23)). Hence,

$$\forall n, \quad 0 \le p_n \le \lim_{k\to\infty} \xi_k = \tau_\ell. \tag{51}$$

We shall use this to show convergence of P_n. Rearranging terms from (45), we get

$$(P - P_n)M = B(P - P_{n-1}) + (P_{n-1} - P)G^H P_{n-1} + PG^H(P_{n-1} - P), \tag{52}$$

so that, taking norms and using the bound (51) on the norms p_n, we arrive at

$$\begin{aligned} \|P - P_n\| &\le \frac{1}{\mu}(\beta + \chi(\|P\| + \|P_{n-1}\|))\,\|P - P_{n-1}\| \\ &\le \frac{\beta + 2\chi\tau_\ell}{\mu}\|P - P_{n-1}\|. \end{aligned} \tag{53}$$

Using that $\|P - P_0\| = \|P\| \le \tau_\ell$ we have completed the proof. □

3.2.1 Implicit treatment of the quadratic term

The last successive substitution that we will consider is one in which we solve systems with the smaller matrix in each step, but, since we treat the quadratic term implicitly, this small matrix changes in each iteration step. The iteration is

$$\text{given} \quad P_0 = 0, \quad \text{iterate} \quad P_n\left(M + G^H P_{n-1}\right) = BP_{n-1} + C. \tag{54}$$

For this iteration, we can prove the following result.

Theorem 3.6 *With the same notations and under the same conditions as in Theorems 3.5, we have that the implicit iteration* $P_n\left(M + G^H P_{n-1}\right) = BP_{n-1} + C$ *is convergent if* $P_0 = 0$, *and*

$$\|P - P_n\| \le \frac{\gamma}{\mu}\left(1 - \frac{1 - \frac{\mu-\beta}{\mu+\beta}\sqrt{1-\rho}}{1 + \frac{\mu-\beta}{\mu-\beta}\sqrt{1-\rho}}\right)^{-1}\left(\frac{1 - \frac{\mu-\beta}{\mu+\beta}\sqrt{1-\rho}}{1 + \frac{\mu-\beta}{\mu+\beta}\sqrt{1-\rho}}\right)^{n}. \tag{55}$$

Proof. Again, we will only outline the proof since it contains the same elements as in the proofs of Theorem 3.2 and Theorem 3.5. The successive substitution bounding the norms $p_n := \|P_n\|$ is here

$$\xi_0 = 0, \quad \xi_n = \phi(\xi_{n-1}), \quad \text{with} \quad \phi(\xi) = \frac{\gamma + \beta\xi}{\mu - \chi\xi}. \tag{56}$$

There is convergence of ξ_n to the smallest fixed point τ from (23), giving

$$\begin{aligned} &\|P_{n+1} - P_n\| \\ &\le \frac{1}{\mu}\left(\left[\beta + \frac{\delta}{2}\left(1 - \sqrt{1-\rho}\right)\right]\|P_n - P_{n-1}\| + \frac{\delta}{2}\left(1 - \sqrt{1-\rho}\right)\|P_{n+1} - P_n\|\right). \end{aligned} \tag{57}$$

Rearranging the terms we can prove that we have a Cauchy sequence, after which the statement follows from $\|P_1 - P_0\| = \|CM^{-1}\| \le \gamma/\mu$. □

Remark 3.7 As in the case of Theorem 3.2, we could work with $M_n := M + G^H P_n$ and get a result in terms of the norms $\mu_n := \|M_n^{-1}\|^{-1}$. The following iteration

$$P_0 = 0, \quad \text{iterate} \quad P_n M_{n-1} = BP_{n-1} + C, \tag{58}$$

is equivalent to (54). We will, however, not pursue this possibility. ◊

3.3 Computable error bounds for the eigenvalues

As we already mentioned in Section 1.3.2, it is possible to monitor the progress of the successive substitution methods by computing the eigenvalues of the matrix $M_n := M + G^H P_n$. The sequence (M_n) converges to $M_\infty := M + G^H P$, whose eigenvalues are a subset of the eigenvalues of A (see Theorem 2.2). The difference between the two can be easily written down,

$$M_\infty - M_n = G^H(P - P_n). \tag{59}$$

Any theorem on the perturbation of eigenvalues of a matrix can now be applied to this situation, since we have developed bounds on the norms $\|P - P_n\|$ for each of the four successive substitution methods in Theorems 3.2, 3.4, 3.5 and 3.6. We will highlight two of them, of which the first one is a general result. It includes the concept *deviation from normality* $\nu(A)$ of a (general) matrix A, which is defined as follows.

Definition 3.8 (Departure from normality) Suppose that $AQ = QT$ is the Schur decomposition of A for which the norm of the upper triangular part N of T is minimal. Then $\nu(A) := \|N\|$ is called the $\|\cdot\|$*-departure from normality* of A.

Theorem 3.9 (A Henrici corollary) *Let $\chi := \|G\|$, and let $\nu(M_n)$ be the L_2-departure from normality of the matrix M_n. Then for each eigenvalue λ of M_∞ there exist an eigenvalue μ of M_n such that*

$$|\lambda - \mu| \leq \max\left(\theta, \theta^{\frac{1}{k}}\right), \quad \text{where} \quad \theta = \|P - P_n\|_2 \left(\chi \sum_{j=0}^{k-1} \nu(M_n)^j\right). \tag{60}$$

Proof. The theorem is a trivial corollary of Theorem 7.2.3 in [8]. □

For diagonalizable M_n, we have the well-known Bauer-Fike theorem. Note that for normal M_n (this is, unitarily diagonalizable) the results from Theorem 3.9 and Theorem 3.10 can be seen to overlap, by putting $\nu(M_n) = 0$ and $\kappa_p(Q) = 1$.

Theorem 3.10 (A Bauer-Fike corollary) *Let $\chi := \|G\|$ and $p \in [1, \infty)$. Suppose M_n is diagonalizable, and that Q_n is such that $Q_n^{-1} M_n Q_n$ is diagonal. Denote the p-norm condition number of Q_n by $\kappa_p(Q_n)$. Then for each eigenvalue λ of M_∞ there exist an eigenvalue μ of M_n such that*

$$|\lambda - \mu| \leq \kappa_p(Q_n)\chi\|P - P_n\|_p. \tag{61}$$

Proof. We refer to [8] and the references therein for the Bauer-Fike Theorem. The statement of the theorem here is just a trivial corollary. □

In Section 6.4 we will illustrate, using a simple example, that the bounds from the two theorems above can, in some circumstances, indeed be estimated in a relatively inexpensive way. This will result from back-transformation to the original basis as will be shown in Section 7, and the fact that during the iterations (either successive substitutions or Krylov subspace iterations), much information about spectral properties of the operators involved, becomes available. Apart from that, *extrapolation* of the sequence P_n is also a topic of interest. Indeed, asymptotically, the difference $P_{n+1} - P_n$ behaves like a geometric sequence, especially after taking norms. More research in this direction is needed.

3.4 Discussion of the results

In Sections 3.1 and 3.2 we have proved error bounds (implying convergence) for four different successive substitution methods for approximating the solution P of the non-linear Sylvester equation (14). The methods have been classified according to how the quadratic term was treated (implicitly or explicitly) and according to the type of equation to be solved (standard linear system or Sylvester equation). We will discuss the results.

3.4.1 The different conditions of the theorems in different norms

First note, that although the conditions $\delta > 0$ and $\rho < 1$ seem to be the same in all four theorems, there is a difference in the definition of the number δ. For the Sylvester equations in Section 3.1, δ was the separation between B and M, while for the linear systems considered in Section 3.2, it was a difference of norms. In the case of a Hermitian matrix A, which leads to Hermitian B and M, the two quantities can be easily written down if specific norms are used.

Proposition 3.11 **([18], Th. 2.3 and Th. 3.1)** *Let B and M be square matrices, then the separation between B and M in the Frobenius norm satisfies*

$$sep(B, M) \leq \min\{|\lambda_B - \lambda_M| \mid \lambda_B \in \sigma(B) \quad \text{and} \quad \lambda_M \in \sigma(M)\}, \qquad (62)$$

while equality holds if M and B are Hermitian.

This result implies that, in the Hermitian case, the iterations (26) and (41) can be applied as long as the spectra of M and B are disjoint, so, also if they interlace. In that case one might expect that convergence of a block-iteration could be as slow as the slowest single vector iteration. In Section 4 we will show that, fortunately, this does not always need to be the case.

Consider on the other hand a situation in which M contains clusters of eigenvalues, and that the mutual distances within such a cluster are smaller than the distance between the spectra of B and M. Then the block-algorithm is a clear improvement over the multiple application of the single-vector variant.

We will now turn to the successive substitutions (45) and (54), in which systems are solved. The easiest norm to use seems the L_2 norm, since the condition $\|M^{-1}\|^{-1} - \|B\| > 0$ reduces in the Hermitian case to

$$|\max \sigma(B)| < |\min \sigma(M)|. \qquad (63)$$

This, however, does not admit interlacing spectra (although in different norms, different results might be obtained). Note that (63) can also be realized by a simple translation if the convex hulls of the spectra are disjoint sets. Alternatively, one could choose, by symmetry of formulation (not of the matrices) to solve systems with B instead of with M. We note that an approach based on *harmonic Ritz values* might be worthwhile considering in case one is interested in approximating (clusters of) interior eigenvalues. We refer to [20] for details on harmonic Ritz values in this context.

3.4.2 The convergence speed and the amount of numerical work

Comparing the theorems 3.2, 3.4, 3.5 and 3.6, we note that the upper bounds for the Sylvester equation approach of Section 3.1 are better than those for the linear system approach of Section 3.2, and also that the bounds for an implicitly treated quadratic term are better than those for an explicitly quadratic term. This could have been predicted on beforehand. It is also clear that the linear system approach can be very much cheaper per iteration step than the Sylvester equation approach (although one should realize that the most expensive part in each step is probably the multiplication with the larger matrix). Also, explicit methods are cheaper than implicit methods, in which the small matrix changes in every iteration step.

Remark 3.12 Similarity transformations can be used to improve the convergence estimates, although this hardly has any practical value. Defining $U := VBV^{-1}$ and $T := W^{-1}TW$ with non-singular V and W, we find, with $Z := VPW$, that (14) transforms into

$$UZ - ZT = Z\left(W^{-1}G^{H}V^{-1}\right) - VCW. \tag{64}$$

Applying the convergence theorems to a transformed system can lead to better values for the parameters that determine the convergence, although only for unitary transformations V and W we can transform the resulting estimates for $\|Z - Z_n\|$ back to estimates for $\|P - P_n\|$. ◊

Before we can make a fair comparison of the costs of the four methods, we will need to concentrate on methods for solving Sylvester equations. We will see that solving them to full accuracy can be very expensive. However, approximating their solution by means of one or more steps of an iterative method can be feasible as well (see also the IBRQI in [14]). As a matter of fact, we will see that some of those inexact solution methods will reduce the Sylvester equation approach of Section 3.1 to a linear system approach.

4 Iterative methods for the Sylvester equation

In this section we will concentrate on solving the general Sylvester equation $FZ - ZT = E$ by means of iterative methods. We will follow the lines that develop iterative methods for linear systems $Ax = b$ and adapt them to the Sylvester equation setting. Note that we may assume that T is upper triangular, since, using a Schur decomposition $TQ = Q\hat{T}$ of T, the equation $FZ - ZT = E$ transforms to

$$F(ZQ) - (ZQ)\hat{T} = EQ. \tag{65}$$

Therefore, in the following we will try to solve $FZ - ZT = E$ assuming that T is upper triangular (note that we did not want to introduce new notations and will continue to work with F, T and E). Before that, though, we will consider the special case that T is even on diagonal form (either as a result of a unitary or a similarity transformation).

Remark 4.1 Throughout Section 4 we assume the matrix T or M to be much smaller than B or F, so that the costs for computing a Schur form for T or M is negligible. ◊

4.1 The diagonalizable and Hermitian case

Suppose that the matrix T in $FZ - ZT = E$ is diagonal, then solving this Sylvester equation is (mathematically) equivalent to solving k independent linear systems. In spite of that, neither one of the iterations from Section 3.1 reduces to a set of k independent vector iterations because of interactions within the non-linear term.

First reconsider iteration (41). Assuming that $M = WDW^{-1}$ diagonalizes M, we can rewrite this iteration as

$$BZ_n - Z_nD = -CW + Z_{n-1}\left(W^{-1}G^H\right)Z_{n-1}, \quad \text{where} \quad Z_n := P_nW. \tag{66}$$

Since the quadratic term is treated explicitly, we can use the same diagonalization of M throughout the whole iteration. Note however, that due to the presence of the quadratic term in the right hand side, the block-iteration (66) is, in general *not* equivalent to k single vector iterations.

Neither iteration (26) cannot be interpreted as k single vector iterations by the diagonalization of the smaller matrix. Indeed, per iteration step, k independent linear systems can be solved, avoiding the difficulties of solving Sylvester equations in which the Schur factor has a non-trivial upper triangular part. But since the matrix $M_n := M + G^HP_n$ changes in each iteration step and depends on the complete iterate P_n, a total decoupling into k independent vector iterations is not possible.

Remark 4.2 Note that diagonalization of the matrices M and $M_n := M + G^H P_n$ in the successive substitutions of Section 3.2 leads to a similar result. There is a step-by-step decoupling of the equations, which can be practically very useful in the explicit iteration (54), but there is not a full decoupling into k independent vector iterations. This is probably exactly what makes that the block algorithms perform better than when k single vector iterations are applied. ◊

4.2 Basic iterative method for the Sylvester equation

Any iterative algorithm for solving the Sylvester equation will, essentially, have the following structure. Given an initial guess Z_0 for the solution Z, we calculate the residual $R_0 = E + Z_0T - FZ_0$, put $k = 0$ and iterate

$$\text{solve } U_k \text{ approximately and cheaply from } \quad FU_k - U_kT = R_k, \qquad (67)$$

$$C_k = FU_k - U_kT, \quad R_{k+1} = R_k - C_k, \quad Z_{k+1} = Z_k + U_k, \quad k = k+1. \qquad (68)$$

If U_k is solved exactly from the *residual correction equation* (67), then $Z_{k+1} = Z$. Otherwise, the hope is that the algorithm will produce a sequence Z_k that eventually converges to Z. Of course, there are a multitude of methods to solve the residual correction equation only approximately. We start by showing a simple one in Section 4.2.1 because it establish a connection between the Sylvester equation approach of Section 3.1 and the linear system approach of Section 3.2. Then we move on to variations based on the Bartels-Stewart algorithm [1] in Section 4.2.2 and comment on the use of Krylov Subspace methods in Section 4.2.3.

4.2.1 Linear system approximation

The classical idea in linear system theory for the iterative solution of $Ax = b$, is to split the linear operator $A = M - N$ such that solving systems with M is easy, and then to iterate $Mx_{k+1} = Nx_k + b$ to the fixed point x. The Richardson, Jacobi and Gauss-Seidel algorithms are instances of this method, and we refer to [9] for details. We can apply a similar approach in the Sylvester setting, resulting in Algorithm 4.1 .

Proposition 4.3 *Algorithm 4.1 converges to the solution Z of $FZ - ZT = E$ if the product of spectral radii $\rho(F)\rho(T^{-1})$ of F and T^{-1} is smaller than one* □.

Now, apply one step of Algorithm 4.1 with start value $Z_0 = 0$ to approximate the solution P_n of each iteration step of (26). Then the approximating sequence $\hat{P}_n$ thus obtained is exactly the sequence P_n from (54).

```
ALGORITHM 4.1: Classical Method for the Sylvester equation.
    input: $F, T, E, Z_0$, tolerance
    $R_0 = E - (FZ_0 - Z_0T)$
    $k = 0$
    while $\|R_k\| >$ tolerance
    $U_kT = -R_k$
    $R_{k+1} = -FU_k$
    $Z_{k+1} = Z_k + U_k$
    $k = k + 1$
end (while)
```

Instead of having to combine the convergence theorem (Theorem 3.2) for the sequence (26) with the convergence of Algorithm 4.1 given by Proposition 4.3, we already proved convergence of this combined method directly in Theorem 3.5. Note that the condition for convergence in Proposition 4.3 is in fact the same as in Theorem 3.6.

Remark 4.4 It is probably better not to start Algorithm 4.1 with $Z_0 = 0$, but with the previously found value of P_{n-1}. ◊

Remark 4.5 Note that if we do not update the non-linear term $G^H P_{n-1}$ in (54), the sequence P_n still converges, but then to the solution of one iteration step of iteration (26). ◊

Remark 4.6 Similarly, iteration (41) can be seen as arising from (45) in which in each step, one step of Algorithm 4.1 is used to approximate the solution of the Sylvester equation. ◊

4.2.2 The Bartels-Stewart algorithm

A second idea is to sequentially solve the columns u_j of U_k as follows. Using that $T = (t_{ij})$ is upper triangular and assuming that $u_1, \cdots, u_{j-1}$ have already been calculated, we find,

$$(F - t_{jj}I)u_j = R_k e_j + \sum_{i=1}^{j-1} t_{ij}u_i. \qquad (69)$$

One can choose to solve (69) approximately by replacing F by a diagonal or triangular matrix, or a product of triangular factors. The idea of solving a Sylvester equation with F and T *both* upper triangular using the recurrence (69) is due to

Bartels and Stewart [1], so, what we have just suggested is to use the Bartels-Stewart algorithm to approximate the solution of the residual correction equation (67). The resulting algorithm is Algorithm 4.2.

Clearly, a sequential algorithm solving the columns of U_k will suffer from error propagation in the sense that if u_i is only solved approximately, one cannot expect any of the columns $u_j, j > i$, to be more accurate that u_i. Therefore it is important to use such a solution algorithm for example, as it is done here, in an inner loop, such that the outer iteration might correct the effects of errors made in the inner loop.

```
ALGORITHM 4.2: Classical Method with Bartels-Stewart residual correction.
    input: $F$, upper triangular part $K$ of $F, T, E, Z_0$, tolerance
    $R_0 = E - (FZ_0 - Z_0T)$
    $k = 0$
    while $\|R_k\| >$ tolerance
        $KU_k - U_kT = R_k$
        $C_k = FU_k - U_kT$
        $R_{k+1} = R_k - C_k$
        $Z_{k+1} = Z_k + U_k$
        $k = k + 1$
    end (while)
```

4.2.3 Krylov subspace methods

Of course, one can also employ Krylov subspace methods for linear systems of equations to approximate each equation in (69). It is worth mentioning that, denoting by $K^p(A, v)$ the p-dimensional Krylov subspace of the matrix A and startvector v,

$$\forall p \in \mathbb{N}, \forall t \in \mathbb{R}, \quad K^p(A, v) = K^p(A - tI, v), \tag{70}$$

so that a Krylov subspace built to approximate the first of the equations in (69), can be used in the consecutive equations in (69) as well. As a matter of fact, also in future residual correction equations, corresponding to further iterations of the successive substitution methods (26) and (41), the same Krylov subspaces could be successfully employed. This is because the matrix B does not change during the entire successive substitution.

4.3 Krylov subspace methods for the Sylvester equation

In Section 4.2 we have seen a nested iteration scheme with a very simple outer iteration to approximate the solution of the Sylvester equation. The inner iterations suggested in Sections 4.2.2 and 4.2.3 may seem (and may be) to sophisticated in comparison to this outer iteration. Of course one could just skip the whole outer iteration and apply, for example, the Krylov subspace approach to sequentially solve the columns of the unknown matrix Z as in (69), but, as we already noted, this has the problem that errors made in the first columns propagate to the other columns. This effect might become highly undesirable if T has a large departure from normality, i.e., if the strict upper triangular part of T is very heavy compared to the diagonal (See Definition 3.8). The following class of algorithms suffers (in general) much less from a non-proportional distribution of errors over the columns of Z, but the price to pay is clear too; they are computationally more expensive.

4.3.1 Geometrical interpretation

It is not hard to adapt Krylov subspace methods for the solution of linear systems directly to the Sylvester equation itself. It does imply, however, that we will have to work with systems of the size $(n-k)k \times (n-k)k$, since we need to identify the matrices Z_k (and others) of size $(n-k) \times k$ with vectors of length $(n-k)k$. The linear Sylvester operator $\mathbf{T}$ (Cf.Th.3.2) operates on such vectors and can be expressed as the $(n-k)k \times (n-k)k$ matrix

$$I_k \otimes F - T \otimes I_{n-k}, \tag{71}$$

where I_q is the $q \times q$ identity matrix and $\otimes$ the Kronecker product, which is defined as follows,

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & & \vdots \\ a_{n1}B & \cdots & a_{nn}B \end{bmatrix}, \tag{72}$$

where $A = (a_{ij})$ is an $(n-k) \times (n-k)$ matrix and B a $k \times k$ matrix. In our applications, the *extra large* matrix $A \otimes B$ does not need to be formed explicitly since in Krylov subspace methods it suffices to have its action available. And we do have this action available, because if we define a function **vec** from the space of $(n-k) \times k$ matrices to the space of $(n-k)k$ vectors by

$$\mathbf{vec}(Z) = \mathbf{vec}\left(\left[\ z_1 \mid \cdots \mid z_k\ \right]\right) = \left(z_1^H, \cdots, z_k^H\right)^H, \tag{73}$$

it holds for the Sylvester operator that

$$\mathit{vec}(\mathbf{T}(Z)) = \mathbf{vec}(FZ - ZT) = (I_k \otimes F - T \otimes I_{n-k})\,\mathbf{vec}(Z). \tag{74}$$

4.3.2 Building a Krylov subspace of long vectors

The heart of Krylov subspace methods is formed by residual correction in an expanding Krylov subspace of which an orthogonal basis is maintained during the iteration. Consider for example the residual correction $R_{k+1} = R_k - C_k$ in (67). The correction C_k will in general not be the optimal correction of R_k in the span αC_k, and since we know that $\mathbf{T}(\alpha C_k) = \alpha C_k$ we could have corrected with any multiple of αC_k instead, and update Z_k in (68) with αU_k accordingly. Identifying the matrices involved with vectors using the function **vec**, the optimal correction is

$$R_{k+1} = R_k - \alpha C_k \quad \text{with} \quad \alpha = \frac{\mathbf{vec}(C_k)^H \mathbf{vec}(R_k)}{\mathbf{vec}(C_k)^H \mathbf{vec}(C_k)}. \tag{75}$$

Note that

$$\mathbf{vec}(A)^H \mathbf{vec}(B) = \quad \text{trace}(A^H B), \quad \text{so,} \quad \mathbf{vec}(A)^H \mathbf{vec}(A) = \|A\|_F^2, \tag{76}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Continuing to correct in subspaces on which the action of $\mathbf{T}^{-1}$ is known, and which grow in each step, leads to algorithms like GMRES and GCR.

Remark 4.7 As a matter of fact, we could have introduced the inner product

$$(A, B) := \text{trace}\,(A^H B) \tag{77}$$

on the space of $(n-k) \times k$ matrices as to derive the Krylov subspace methods without any reference to Kronecker products and the function **vec**. We chose to present the geometrical interpretation of Section 4.3.1 as well. ◊

For completeness, we will give the GCR algorithm for the Sylvester equation below, using the notation $\mathbf{T}(Z)$ for the Sylvester action on Z and the inner product $(\cdot,\cdot)$ from Remark 4.7. The operator $\mathbf{S}$ is a preconditioner for $\mathbf{T}$ and can be chosen as any of the previous approximation methods.

In particular we stress that when in each successive substitution step the same Sylvester operator $\mathbf{T}$ is used, it will be worthwhile to re-use the Krylov subspace built in the previous step. It can be used to correct the initial residual in this 'old' space, but even more interesting seems its application as preconditioner. This might reduce the number of iterations of the Krylov subspace needed as the successive substitution progresses.

```
ALGORITHM 4.3: Preconditioned Generalized Conjugate Residuals
    input: $\mathbf{T}$, $\mathbf{S}$, $E$, $Z_0$, Tolerance
    $R_k = E - \mathbf{T}(Z_0)$
    $k = 0$
    while $||R_k||_F >$ tolerance
      $U_k = \mathbf{S}^{-1}(R_k)$
      $C_k = \mathbf{T}(U_k)$
      for $i = 0, \cdots, d-1$ do
        $\beta_{i+1} = (C_i, C_k)/\sigma_i$
        $C_k = C_k - \beta_{i+1}C_i$
        $U_k = U_k - \beta_{i+1}U_i$
      end for
      $\sigma_k = ||C_k||_F^2$
      $\alpha_k = (C_k, R_k)/\sigma_k$
      $Z_{k+1} = Z_k + \alpha_k U_k$
      $R_{k+1} = R_k - \alpha_k C_k$
      $k = k + 1$
    end (while)
```

Also if the Sylvester operator changes in each step, the information from the previous iteration step can be used as preconditioning for the new equation, i.e., the operator $\mathbf{S}$ could be the Krylov subspace approximation from the previous step.

Having now available, at this point, four successive substitution methods for solving (14), and moreover, various ways to tackle the linear Sylvester equations that arise in two of those iterations, we have the basic ingredients ready for a class of workable algorithms. Before we will test them, we will consider how to accelerate these algorithms.

5 Acceleration of the algorithms

In this section we will consider a logical extension of the successive substitution methods introduced in Section 3. In these methods, given an approximation X for an invariant subspace $\hat{X}$ and an orthogonal matrix Y (see Section 2), we produced a sequence P_n that converged to a matrix P using the non-linear correction equation (14). Then, P was used to correct X to $\hat{X} = X + YP$. So far, we did not comment on the fact that during the successive substitutions, intermediate approximations $X_n := X + YP_n$ can be produced and that those can be used as an initial approximation for essentially the same iteration, but now with different matrices B, M, G and C in (14). We will study this attempt to accelerate the algorithms in Section 5.1.

In Section 5.2 we will comment on how to incorporate subspace acceleration, which means that we will not only use P_n to find a new approximate subspace, but all the P_j that were produced in the previous iterations as well. Here too, the hope is to speed up the algorithms, and in particular to improve the convergence in the initial steps.

Finally, in Section 5.3 we will comment on the close relation of the resulting algorithms to the Jacobi-Davidson method of Sleijpen and Van der Vorst [20] in the case of invariant subspaces of dimension one.

5.1 Acceleration by basis transformation

Given a sequence P_n, defined by a successive substitution from Section 3 and converging to P, we will, in view of (13) define

$$X_n := X + YP_n \quad \text{and} \quad Y_n := Y - XP_n^H, \tag{78}$$

so that with $P_0 = 0$ we have $X_0 := X$ and $Y_0 := Y$. Moreover, let

$$B_n := Y_n^H AY_n, \quad M_n := X_n^H AX_n, \quad C_n := Y_n^H AX_n \quad \text{and} \quad G_n := X_n^H AY_n. \tag{79}$$

The norms of the matrices involved, and also the norm of the Sylvester operator $\mathbf{T}_n : Z \mapsto B_n Z - ZM_n$, are parameters that determine the upper bounds for the successive substitutions, as stated in the corresponding theorems in Section 3. Therefore, it might be an improvement to compute, at a certain point, the matrices M_n, B_n, C_n and G_n and continue to iterate on the new nonlinear Sylvester equation obtained this way,

$$B_n P - PM_n = PG_n^H P - C_n. \tag{80}$$

Unfortunately, it is not clear whether this new equation (80) really has better convergence properties for the corresponding successive substitutions than (14), and the contrary may very well be the case. In particular, this may happen when successive substitution steps are not computed in full precision, as will most often be the case in practical situations.

Remark 5.1 The convergence of $sep(B_n, M_n)$ to $sep(B, M)$ does not need to be monotone. Therefore, it might be that $sep(B_{n+1}, M_{n+1}) < sep(B_n, M_n)$, and also that this negative effect is not compensated by small enough residuals C_{n+1} and G_{n+1}^H. So, also the upper bounds for the convergence might become worse.◊

5.1.1 The accelerated algorithm

Taking the previous in account, we propose Algorithm 5.1 to accelerate the successive substitutions. It is based on tracing the only quantity that supplies us with immediate, though not always reliable, information about the convergence to P of a successive substitution $P_{k+1} := \phi(P_k)$ for (80), i.e., the residual

$$S_k := B_n P_k - P_k M_n + C_n - P_k G_n^H P_k. \tag{81}$$

The magnitude relative to $\|S_0\|$ of the residual $\|S_k\|$, can be an indication of to which extent P_k has become a better approximation to P than P_0.

Note that there are at least three levels of iteration in Algorithm 5.1. Apart from the outer and inner iterations, which are clearly distinguishable in the form of two **while** loops, there is in general also an iteration present in the step $P_{k+1} = \phi(P_k)$. This iteration can be the preconditioned Generalized Conjugate Residual method (or, mathematically equivalent: preconditioned GMRES) and henceforth there might even be a fourth level of iteration if the preconditioner is an iterative method, in some sense also when **S** in Algorithm 4.3 is the approximate Sylvester operator obtained by the GCR algorithm in the previous step.

```
ALGORITHM 5.1: Accelerated Succesive Substitution
    input: $A, X_0.\varepsilon_1, \varepsilon_2$
    $n = 0,\ k = 0,\ P_0 = 0$
    choose $Y_0$ and compute $B_0, M_0, C_0, G_0$
    while $\|C_n\| > \varepsilon_1 \|C_0\|$
        $S_k \ = B_n P_k - P_k M_n + C_n - P_k G_n^H P_k$
        while $\|S_k\| > \varepsilon_2 \|S_0\|$
            $P_{k+1} = \phi(P_k)$
            $k \ = k + 1$
            $S_k \ = B_n P_k - P_k M_n + C_n - P_k G_n^H P_k$
        end (while)
        $X_{n+1} = X_n + Y_n P_k$
        choose $Y_{n+1}$ and compute $B_{n+1}, M_{n+1}, C_{n+1}, G_{n+1}$
        $P_0 \ = P_k$
        $k \ = 0$
        $n \ = n + 1$
    end (while)
```

Note that only in the successive substitution (80) with $n = 0$, the initial approximation $P_0 = 0$ is used. In the successive substitution with index $n \geq 1$, the (generally better) final approximation of the substitution with index $n - 1$ is used.

5.2 Subspace acceleration

We are now ready to discuss an additional way to accelerate the algorithm, which is *subspace acceleration.* Subspace acceleration is that what turns the power method into the Arnoldi method (see for example [8] for both methods). The main idea is to use all the information obtained in previous iteration steps to optimize the next, just as in the Arnoldi method, a Ritz-Galerkin procedure is applied to the complete span of all vectors that are the result from the application of A, while in the power method this is only done with the span of the last vector thus obtained. Here we will discuss two ways to incorporate subspace acceleration in Algorithm 5.1.

- Each approximate subspace X_n is seen as an inseparable entity, and a Ritz Galerkin procedure for extra large vectors is applied to select from $X_0, \cdots X_n$ the linear combination with residual (with respect to some extra large matrix) orthogonal to their span (see Section 4.3 for the concept of Krylov Subspace methods for extra long vectors).
- Each approximate subspace X_n is seen as a set of k vectors of length N, and a Ritz-Galerkin procedure is applied to select from the total of kN of those vectors from $X_1, \cdots, X_n$ the best k that approximate the invariant subspace under consideration.

In the coming two sections we will highlight both these methods.

5.2.1 A Ritz-Galerkin procedure with extra large vectors

As we have seen in Section 4.3, we can identify $(n-k) \times k$ matrices with $(n-k)k$ vectors by means of the mapping **vec**. This proved to be useful in the development of Krylov Subspace methods for the Sylvester equation, using also the notion of Kronecker products of matrices. The same ingredients can lead to a way to accelerate Algorithm 4.3 as follows.

Let Q be an orthogonal matrix spanning an invariant subspace for the matrix A, and write $AQ = QS$, hence defining S. Let D be the diagonal matrix with the eigenvalues of S on its diagonal. Then, in the notations of Section 4.3 it holds that

$$(I_k \otimes A - D \otimes I_n)\,\mathbf{vec}(Q) = 0, \tag{82}$$

so the **vec** of the eigenspace that we wish to approximate, is an eigenvector of an extra large matrix, belonging to its eigenvalue zero. It is possible to use in Algorithm 5.1 the current approximations of the eigenvalues of S and to set up a Ritz-Galerkin method, by projecting the extra large matrix on the span of all previously obtained extra long vectors $\mathbf{vec}(X_j)$.

Remark 5.2 Since convergence of the eigenvector(s) belonging to eigenvalue zero is now preferred (see 82), one could work with *harmonic Ritz values* to obtain a more regular convergence pattern than one would have when applying one of our algorithms to an interior eigenvalue. Also in [20], the use of harmonic Ritz values is encouraged (as well as explained). Because of the complicated and specialized nature of this topic, we will not consider it in this paper. $\Diamond$

5.2.2 A Ritz-Galerkin procedure based on Schur vectors

Instead of the approach in Section 5.2.1 one could choose an easier alternative. Assuming that approximations $X_0, \cdots, X_{n-1}$ for the invariant subspace have been obtained, one could, instead of applying a Ritz-Galerkin procedure to X_{n-1} only (as in Algorithm 5.1), apply it to (a part of) the complete set $X_0, \cdots, X_{n-1}$. For this purpose, it will be convenient to orthogonalize X_n column by column to all previous columns of all previous X_j, such that $(X_0| \cdots |X_n)$ is an orthogonal matrix. Then the Ritz-Galerkin projection can take place, and from the resulting set of Ritz data, a suitable selection of data to represent the new approximation X_n of the invariant subspace can be made.

Remark 5.3 We suggest here to construct a Schur decomposition of the projected matrix and select the k Schur vectors corresponding to the Ritz values closest to some pre-defined *target values*, similar to what is done in Chapter 6 of [5]. ◊

5.3 Jacobi-Davidson as one step of a successive substitution

As mentioned in Section 1.3.5, the Jacobi-Davidson algorithm of Sleijpen and Van der Vorst [20] can be embedded in our class of algorithms. In their approach, not the non-linear correction equation (14) is iteratively solved, but the *linear* correction equation

$$B\hat{P} - \hat{P}M = -C. \tag{83}$$

It is not hard to see that doing one step of the successive substitution (26) is equivalent to solving (83). This is, in particular, caused by the starting value $P_0 = 0$ in (26). Also, since we have already seen (in Section 4.2.1) that the application of one step of Algorithm 4.1 to approximate a step of iteration (26) is equivalent to iteration (54), the Jacobi-Davidson algorithm in which in each step the solution of (83) is approximated by one step of Algorithm 4.1, is also equivalent to iteration (54).

One of the essential differences between Jacobi-Davidson, and a subspace accelerated and basis-transformation accelerated successive substitution for (14) is, that no matter how accurately (83) is solved, there is a limit to the accuracy of the next invariant subspace obtained. Conversely, at least in theory, if (14) is solved exactly, we immediately have the exact invariant subspace.

This difference can be important if we have to decide *a priori* how much effort we want to invest into solving the correction equation. Given a certain amount of 'computational effort', it might be that (83) is *over-solved*, while (14) can, in principle, never be solved too accurately. Of course, in the latter case one has to decide how much effort to put in each of the successive substitutions, whereas with Jacobi-Davidson, this is not an issue.

5.3.1 Another interpretation for Jacobi-Davidson

The difference between the correction equation for Jacobi-Davidson (83) and the non-linear correction equation (14) is, that the term PG^HP is neglected. This can also be interpreted as *assuming that* $G^H = 0$ and therefore, that Y is an invariant subspace for A. In the Hermitian case, this would mean that the residual C for X is assumed to be zero as well, which is not too bad since we assumed, in fact, that X is a good initial approximation for $\hat{X}$. So, as convergence to the invariant subspace progresses, the neglected term converges to zero. In the non-symmetric (non-normal) case, however, the term G^H can have any size, and is strongly related to the departure from normality (See Def.3.8) of the matrix A. Even though Jacobi-Davidson may well converge, it might be an improvement (and even a big improvement in case A is highly non-normal) to include the term PG^HP and apply a number of successive substitution steps on (14).

Alternatively, in the non-symmetric case, it can be that G^H indeed vanishes without C having to vanish, in case Y is an invariant subspace and X is not. In that (unrealistic, though instructive) situation, solving the linear correction equation exactly leads to the invariant subspace $\hat{X}$ since the linear and non-linear correction equations coincide.

Remark 5.4 The conclusion is that this all asks for an approach in which not the *orthogonal complement* of the invariant subspace plays a central role, but its *spectral complement*. Assuming that $\hat{X}$ is a simple invariant subspace, A can be *block diagonalized* by a similarity transformation that is in general not unitary, which results in a *spectral resolution* of A (see [18]). It will be topic of further research to find out to which extent it is possible to work with the spectral resolution, since two obvious problems immediately arise:

- The spectral complement is unknown and often of very high dimension,
- The favorable stability properties of Jacobi-Davidson might be lost.

In such an approach, if successful, both C and G^H would converge to zero, and the cubic convergence of the Jacobi-Davidson algorithm for Hermitian matrices, might be restored for non-normal matrices as well.

6 Illustration of some of the methods

We will now present some very simple examples to illustrate the mechanisms behind some of the methods and algorithms so far. They concern the computation of an eigenpair of a two by two symmetric matrix, which, for even more simplicity, is assumed to be diagonal (although this is no further restriction for the method). Some comments on other types of matrices will be made along the way.

6.1 Initial approximation and transformation

We will start with the following situation,

$$A := \begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix} \quad \text{and} \quad (x_0|y_0) := \frac{1}{\sqrt{1+\varepsilon^2}} \begin{pmatrix} 1 & \varepsilon \\ \varepsilon & -1 \end{pmatrix}. \tag{84}$$

So, we have an initial approximation x_0 of the eigenvector $(1,0)^T$ and a vector y_0 orthogonal to x_0, and both x_0 and y_0 are of unit length. Transforming the matrix A on the basis x_0, y_0, we get

$$\hat{A} : \begin{pmatrix} m & c \\ c & b \end{pmatrix} \quad \text{where} \quad m = \frac{a_1 + a_2\varepsilon^2}{1+\varepsilon^2}, b = \frac{a_2 + a_1\varepsilon^2}{1+\varepsilon^2} \quad textand \quad c = \frac{(a_1 - a_2)\varepsilon}{1+\varepsilon^2}.$$

In case $a_1 \neq a_2$, the nonlinear equation (14) reduces to finding the scalar p for which

$$cp^2 + (m - b)p - c = 0, \quad \text{or,} \quad \varepsilon p^2 + (1 - \varepsilon^2)p - \varepsilon = 0. \tag{85}$$

Clearly, the solution that we are interested in is $p = \varepsilon$. This is the only solution that is an attractor in the successive substitution. Starting with $p_0 = 0$, the sequence p_n converges to $p = \varepsilon$ for all ε with $|\varepsilon| < 1$ for all ε with $|\varepsilon| < 1$.

6.2 Analysis of iteration (26) for the quadratic equation in p

For our simple model problem, we will look more closely at the implicit successive substitution method (26) as developed in Section 3.1. This gives the iteration

$$p_{n+1} = \phi(p_n) \quad \text{where} \quad \phi(\xi) = \frac{c}{m - b + c\xi} = \frac{\varepsilon}{1 - \varepsilon^2 + \varepsilon\xi}. \tag{86}$$

The convergence properties of this successive substitution clearly do not depend on $a_1 - a_2$, but this is due to the simplicity of the problem; in fact, $bp - pm$ is a scalar multiple of cp, which normally is not the case. In Figure 1, the convergence is displayed in the usual way. On the left, $\varepsilon = 0.8$ and on the right, $\varepsilon = 0.6$. Note the difference in convergence speed for the two values, as indicated by the following proposition.

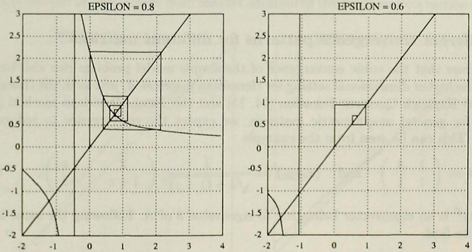

Figure 1. Convergence of the successive substitutions for $\varepsilon = 0.8$ (left) and $\varepsilon = 0.6$ (right). The scale in both pictures is the same.

Proposition 6.1 *The successive substitution converges linearly with asymptotic convergence factor*

$$\lim_{n\to\infty} \frac{p_{n+1} - p_n}{p_n - p_{n-1}} = \phi'(\varepsilon) = -\varepsilon^2. \tag{87}$$

Proof. Standard, using $p_{n+1} = \phi(p_n)$ and the mean value theorem. □.

6.2.1 Acceleration of the successive substitution

The convergence in both cases is not monotone although the norms of the error $\|p - p_n\|$ do decrease monotonely. This, and the large difference in convergence rate for the two values of ε suggest to accelerate the method according to Section 5.1. To analyze the effects of this acceleration, we compute explicitly what happens if we update b, m and c from the newly obtained approximation of the eigenvector.

Suppose we start iteration (86) with $p_0 = 0$. Then, updating the subspaces according to (78) leads to

$$x_1 = \frac{x_0 + p_1 y_0}{\sqrt{1+p_1^2}} \quad \text{and} \quad y_1 = \frac{y_0 - p_1 x_0}{\sqrt{1+p_1^2}}, \quad \text{where} \quad p_1 = \frac{\varepsilon}{1-\varepsilon^2}. \tag{88}$$

Substituting the value for p_1 in the expressions for x_1 and y_1 gives

$$(x_1|y_1) := \frac{1}{\sqrt{1+\varepsilon^6}} \begin{pmatrix} 1 & -\varepsilon^3 \\ -\varepsilon^3 & -1 \end{pmatrix}. \tag{89}$$

This expression can in turn be seen as initial approximation for the same eigenproblem, and the result is that the accelerated successive substitution is cubically convergent in this particular case,

$$sin\angle(\hat{x}, x_n) = \mathcal{O}(\varepsilon^{3^n}), \quad (n \to \infty). \tag{90}$$

6.2.2 Different convergence patterns for different matrices

We stress that the cubic convergence of the simple model problem can also be expected in the higher dimensional setting for Hermitian matrices A, just as in the (Inexact) Accelerated Rayleigh Quotient Iteration [14, 15] and the Jacobi-Davidson method [20]. Similarly, for general unsymmetric matrices, we expect the convergence to reduce to quadratic. This can be seen from the example

$$A := \begin{pmatrix} 1 & 4 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad (x_0|y_0) := \frac{1}{\sqrt{4+(1+\varepsilon)^2}} \begin{pmatrix} 2 & 1+\varepsilon \\ 1+\varepsilon & -2 \end{pmatrix}, \tag{91}$$

where $(2,1)^T$ is an eigenvector belonging to eigenvalue 3 of A. Following the same lines as before, we find

$$p_1 = \frac{\varepsilon(\varepsilon+2)}{5\varepsilon+4} \quad \text{and} \quad x_1 = \frac{x_0 + p_1 y_0}{\sqrt{1+p_1^2}} = \alpha \begin{pmatrix} 2 \\ 1+\mathcal{O}(\varepsilon^2) \end{pmatrix}, \tag{92}$$

where α is a scalar such that the resulting vector has unit length. The term $\mathcal{O}(\varepsilon^2)$ is sharp, it cannot be improved.

In case of a double eigenvalue (note that this is not included in our theory of Section 3) the situation can be even worse. Consider the example

$$A := \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad (x_0|y_0) := \frac{1}{\sqrt{1+\varepsilon^2}} \begin{pmatrix} 1 & \varepsilon \\ \varepsilon & -1 \end{pmatrix}. \tag{93}$$

where $(1,0)^T$ is an eigenvector belonging to the double eigenvalue 1 of A. Following the same lines again, we find the optimal result,

$$p_1 = \frac{1}{2}\varepsilon \quad \text{and} \quad x_1 = \frac{x_0 + p_1 y_0}{\sqrt{1+p_1^2}} = \alpha \left(\begin{array}{c} 1 \\ \mathcal{O}(\varepsilon) \end{array} \right), \tag{94}$$

which means that the accelerated successive substitution only converges linearly, but with reduction factor smaller than one half. Note that the unaccelerated algorithm converges extremely slowly because $\phi'(\varepsilon) = 1$. Finally, in case of a double eigenvalue and an eigenspace of dimension two, the algorithm converges in one step, since all p satisfy the correction equation $0p = 0$.

Remark 6.2 In all cases, the convergence can suffer if the inexactness of the solution methods is too large, although often, linear convergence with high speed remains, as we will show in Section 6.3. $\diamond$

6.3 Analysis of the iteration (41) for the quadratic equation in p

We will now repeat the analysis of the previous example for the iteration (41), of which we noted that it can often be (in the high dimensional setting) much cheaper to perform. Also, since we have seen in Section 4.2.1 that (41) can be interpreted as resulting from the inexact solution of iteration (26), it is interesting to see how much the cubic convergence proved in (89) suffers from inexact solution of the nonlinear correction equation.

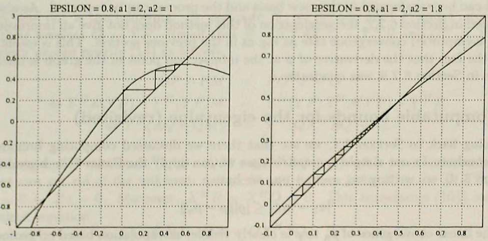

Figure 2. Convergence of the successive substitutions for $\varepsilon = 0.8$ and $a_1 = 2, a_2 = 1$ (left) and $a_1 = 2, a_1 = 1.8$ (right).

The successive substitution becomes in our simple example the following,

$$p_{n+1} = \phi(p_n), \quad \text{where} \quad \phi(\xi) = \frac{1}{m}\left(b\xi - bc\xi^2 + c\right). \tag{95}$$

It should be noted that there is no cancellation like in the previous section due to the specific form of the term $b - m$. So here, the iteration really depends on the entries a_1 and a_2. In Figure 2 below, two pictures display the convergence for $\varepsilon = 0.8$ in both pictures, while $a_1 = 2, a_2 = 1$ in the left picture, and $a_1 = 2, a_1 = 1.8$ in the right. The parabola becomes more flat if the ratio a_1/a_2 tends to one (from below), and the convergence becomes slower. This can be compensated by a smaller ε.

Proposition 6.3 *The successive substitution (95) converges linearly with asymptotic convergence factor*

$$\lim_{n\to\infty} \frac{p_{n+1}-p_n}{p_n-p_{n-1}} = \phi'(\varepsilon) = \frac{b(1-2c\varepsilon)}{m} = \mathcal{O}(\frac{a_2}{a_1}\varepsilon), \quad (\varepsilon \to 0). \tag{96}$$

Proof. Standard, using $p_{n+1} = \phi(p_n)$ and the mean value theorem. □.

In spite of the fact that both iterations (86) and (95) are linearly convergent, this does not automatically imply that the accelerated versions of both iterations should behave similarly. This can be shown by taking again $p_0 = 0$, such that

$$x_1 = \frac{x_0 + p_1 y_0}{\sqrt{1+p_1^2}} \quad \text{and} \quad y_1 = \frac{y_0 - p_1 x_0}{\sqrt{1+p_1^2}}, \quad \text{where} \quad p_1 = \frac{(a_1-a_2)\varepsilon}{a_1 + a_2^2\varepsilon^2}. \tag{97}$$

Substituting the value for p_1 in the expressions for x_1 and y_1 only gives, as opposed to (89),

$$(x_1|y_1) := \frac{1}{\sqrt{1+(\frac{a_2}{a_1}\varepsilon)^2}} \begin{pmatrix} 1 & \frac{a_2}{a_1}\varepsilon \\ \frac{a_2}{a_1}\varepsilon & -1 \end{pmatrix}. \tag{98}$$

Now, A can be transformed to this new basis and the process can be repeated. As already mentioned in Section 6.2.2, the acceleration of this method does not give an improvement of the (asymptotic) convergence rate as big as in the previous section. This is essentially due to the fact that the derivative of ϕ at the intersection point in the graph is linear in ε, while in Section 6.2 it was quadratic.

6.4 Computable bounds for the eigenvalue (revisited)

Going back to Section 3.4, we see that there we discussed computing bounds for the eigenvalues. In our simple symmetric case we can apply the Bauer-Fike bound from Theorem 3.10, so, writing $m_n := m + gp_n$ we have,

$$|a_1 - m_n| \leq |g|\|p - p_n\|. \tag{99}$$

Once the linear convergence of p_n to p is clearly visible, extrapolation based on geometric series can be applied to find an estimate for p and hence for $\|p - p_n\|$. Indeed, we could try to use the following error estimation,

$$p - p_n = \sum_{k=n}^{\infty} p_{k+1} - p_k \approx (p_{n+1} - p_n)\sum_{k=0}^{\infty} \phi'(\varepsilon)^k = \frac{p_{n+1}-p_n}{1-\phi'(\varepsilon)}. \tag{100}$$

In the right-hand side picture of Figure 2, even though convergence is slow, it is already very much linear from the beginning.

Remark 6.4 Note that, apart from error estimates for the approximations m_n of the eigenvalue based on (99) above, one can also accept the extrapolation as approximation of the eigenvalue. Of course, in general there is no error estimate available for this extrapolate. ◊

We conclude by noting that although cubic and quadratic convergence are interesting and desirable, it might not be so bad to have very regular linear convergence. Apart from the fact that methods with linear convergence are as a rule numerically much less expensive, the possibility to extrapolation might turn it into an option worthwhile considering. For details on all kinds of extrapolation methods, we refer to the book by Brezinski and Zaglia [3].

7 Practical considerations

In the end, it is theoretically clear that all we need to do is to solve P from the non-linear equation (14) and to form the matrix $M + C^H P$ of which the eigenvalues are the eigenvalues belonging to the invariant subspace spanned by the columns of $X + YP$. This, however, assumes that we have the matrices M, B and C readily available, and also the matrix Y. Indeed, it is possible to form a matrix Y with the desired properties, and to project A on the column span of Y to obtain B and so on, but in particular when k is small and n large, this procedure is unacceptably expensive.

7.1 Back-transformation to the original basis

A way out is the following. To compute $\hat{X}$, we do not need P and Y explicitly, only their product $Q := YP$. Recall that $B = Y^H AY, C = Y^H R$ and $G^H = X^H AY$. Therefore we can rewrite (14) as follows,

$$\begin{aligned} BP - PM &= PG^H P - C \\ \Leftrightarrow Y^H A(YP) - Y^H(YP)M &= Y^H(YP)X^H A(YP) - Y^H R \\ \Leftrightarrow Y^H(AQ - QM) &= Y^H(QX^H AQ - R). \end{aligned} \tag{101}$$

The orthogonality relation $Y^H(AQ - QM - QX^H AQ + R) = 0$ is basis-independent, so now we can get rid of the unknown matrix Y and replace it by $Z := I - XX^H$, since $Y^H z = 0 \leftrightarrow Zz = 0$. Moreover, $X^H R = X^H Q = 0$. This transforms (101) into the equivalent equation

$$ZAQ - QM = QX^H AQ - R. \tag{102}$$

This equation only involves the given matrices A and X, and M and R, which are relatively cheap to compute. After solving Q from (102), the sum $X + Q$ can be formed which has the same column span as $\hat{X}$.

7.2 Stability issues

The orthogonality $X^H Q = 0$ is a property of the solution Q, and if we try to solve (102) iteratively, it is not guaranteed that the iterates share that property. For stability reasons, it is best to work in the orthogonal complement of the column span of X during the whole iteration process, and therefore we will work with the practical correction equation

$$(ZAZ)(ZQ) - (ZQ)(M + X^H A(ZQ)) = -R. \tag{103}$$

The extra matrices Z have been put there to indicate that during an iterative procedure, all iterates will be projected on Y. Note that ZAZ is singular, so only its action should be used, and one should be careful with preconditioning.

Remark 7.1 The values that determine the convergence speed of the upper bounds of the successive substitutions, as given in Section 3, do not change under the transformation performed.

8 Numerical experiments

We will now perform some numerical experiments to illustrate the algorithms. In Section 8.1 we will consider the combination of the successive substitution (26) with the Krylov subspace solver for Sylvester equations from Section 4.3.2. In Section 8.2 we will accelerate this algorithm.

8.1 No acceleration.

We applied the successive substitution (26) and solved the linear Sylvester equation in each step by GCR according to Section 4.3.2, to relative accuracy α_1. So, given an initial residual S_0, the outer iteration continues until for the residual S_k it holds that $\|S_k\| \leq \alpha_1 \|S_0\|$. Similarly, the relative accuracy for each GCR solve we denote by α_2.

We started with an approximation of the invariant subspace that was a random perturbation of the exact invariant subspace with maximum relative size t per matrix entry. So, each entry x_{ij} of the matrix representing the invariant subspace was randomly perturbed within the range $[(1-t)x_{ij}, (1+t)x_{ij}]$.

8.1.1 The Hilbert matrix

Let $A = (a_{ij})$, where $a_{ij} := 1/(i+j-1)$ be the Hilbert matrix of size 100 × 100. This is a notorious example of an extremely bad conditioned matrix. We will first approximate the largest five eigenvalues $\mu_{96}, \cdots, \mu_{100}$, then the ones $\mu_{86}, \cdots, \mu_{90}$. The convergence is plotted in Figure 3 below. We started $t = 0.1$ away from the exact solution, and $\alpha_1 = \alpha_2 = 10^{-10}$.

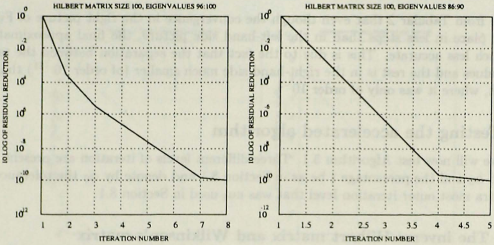

Figure 3. Converge of the successive substitution (26) with GCR for the Hilbert matrix of dimension 100.

The performance is excellent. Only eight (left picture) and five (right picture) outer iterations are sufficient to reach a relative residual reduction of α_1. In Tabular 1 we present the exact eigenvalues, the approximations, and the absolute errors in those approximations for the largest five eigenvalues of A (corresponding to the left graph in Figure 3).

approximate values	exact values	absolute errors
$2.182696097757422e+00$	$2.182696097757424e+00$	$1.776356839400250e-15$
$8.214455605561981e-01$	$8.214455605561967e-01$	$1.443289932012704e-15$
$2.185958823706972e-01$	$2.185958823706963e-01$	$8.881784197001252e-16$
$4.929225104310325e-02$	$4.929225104310336e-02$	$1.110223024625157e-16$
$1.003181218354683e-02$	$1.003181218355605e-02$	$9.220055274816730e-15$

Tabular 1. Accuracy of the approximations of $\mu_{96}, \cdots, \mu_{100}$.

In Tabular 2 we present the exact eigenvalues, the approximations, and the absolute errors in those approximations for the eigenvalues $\mu_{86}, \cdots, \mu_{90}$ of A (corresponding to the right graph in Figure 3).

approximate values	exact values	absolute errors
$1.788722429011753e-07$	$1.788722433072537e-07$	$4.060783945837891e-16$
$2.412650483324968e-08$	$2.412649126353820e-08$	$1.356971147926129e-14$
$4.472343364861278e-11$	$4.569865037083792e-11$	$9.752167222251419e-13$
$3.113442812051378e-09$	$3.113349338012415e-09$	$9.347403896297142e-14$
$3.845863150109427e-10$	$3.850229418596463e-10$	$4.366268487035981e-13$

Tabular 2. Accuracy of the approximations of $\mu_{86}, \cdots, \mu_{90}$.

We see from Tabular 2 that even though the convergence in the right picture of Figure 3 took place in less steps than in the left-hand side picture, the final approximations are much less accurate. This is due to the fact that the separation between the target eigenvalues and the rest is in the right-hand side much smaller (of order 10^{-11}) than on the left, where it was only of order 10^{-2}.

8.2 Testing the accelerated algorithm

We will now test Algorithm 5.1. Three different levels of iteration are present. Let α_1 and α_2 and the percentage t be as in Section 8.1 and denote by α_0 the tolerance for the extra most outer iteration level that was not used in Section 8.1.

8.2.1 The inverse Hilbert matrix and Wilkinson's matrix

First we took for A the inverse of the Hilbert matrix of size 100×100, and approximated the largest five eigenvalues. Then we took the famous Wilkinson's matrix of size 200×200, and also here approximated the largest five eigenvalues.

The convergence plots are in Figure 4 below. We started $t = 0.1$ away from the exact solution, and $\alpha_1 = \alpha_2 = 10^{-3}$.

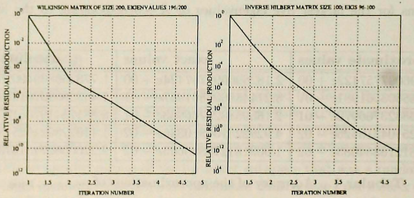

Figure 4. Convergence of the successive substitution (26) with GCR for a Wilkinson (left) and an inverse Hilbert matrix (right).

The algorithm performs very well and already after a few iteration steps, the relative residual reduction of $\alpha_0 = 10^{-10}$ is realized.

8.2.2 The SHERMAN4 matrix

In the following experiment is A the SHERMAN4 matrix from the Harwell-Boeing collection, that can be found in [13]. This matrix has size 1104×1104 and is unsymmetric with real eigenvalues. The parameters for the three iteration levels were set on $\alpha_0 = 10^{-10}, \alpha_1 = 0.5$ (with a maximum of 5 successive substitutions) and $\alpha_2 = 0.1$ (with a maximum of 30 GCR iterations).

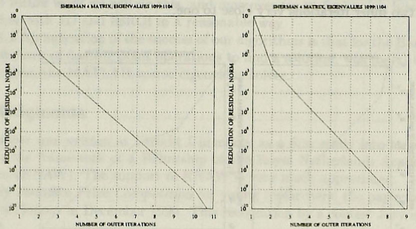

Figure 5. Convergence of the basis transformation accelerated successive substitution (26) with GCR for the SHERMAN 4 matrix of dimension 1104.

The results for the approximation of the six largest eigenvalues are depicted in Figure 5. On the left we took $t = 0.1$ and on the right $t = 1$. In both cases, the convergence was again excellent. The reason to take large tolerances for α_1 and α_2 is obvious; we did not apply any kind of preconditioning within GCR which means that solving systems will become problematic. On the other hand, if we would take α_1 too small, we would not be illustrating the acceleration, but purely the successive substitution again as in Section 8.1. Also, as an effect of the very inexact GCR solve, it might be that convergence of the successive substitution stagnates.

In Tabular 3 below, we see again the same data as in the previous tabulars, for the case in which $t = 1$. And again it can be seen that the approximations of the eigenvalues are very good, in spite of the fact that the initial approximation was relatively far away from the exact invariant subspace.

approximate values	exact values	absolute errors
$6.649656408021296e + 01$	$6.649656408021302e + 01$	$1.808331262509455e - 10$
$6.427368830619677e + 01$	$6.427368830619666e + 01$	$1.153637185780099e - 10$
$6.234470612362478e + 01$	$6.234470612362486e + 01$	$1.421085471520200e - 14$
$6.129003463774111e + 01$	$6.129003463774112e + 01$	$7.815970093361102e - 14$
$5.959818367933947e + 01$	$5.959818367922411e + 01$	$1.136868377216160e - 13$
$5.821585364878595e + 01$	$5.821585364896678e + 01$	$5.684341886080801e - 14$

Tabular 3. Accuracy of the approximations of the largest six eigenvalues of SHERMAN4.

8.2.3 The PORES 2 matrix

In our final experiment we tried to obtain worse convergence by putting the tolerances of the two inner iterations very close to one.

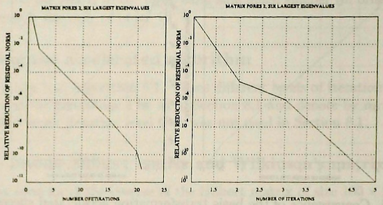

Figure 6. Convergence of the basis transformation accelerated successive substitution (26) with GCR for the PORES matrix of dimension 1104.

This had indeed the desired effect, as is clearly visible in the left picture in Figure 6. The matrix is the PORES 2 matrix from the Harwell-Boeing collection, which is real unsymmetric and of size 1228 × 1228. We approximated the largest six eigenvalues.

On the left, we took $\alpha_1 = 0.9$ and $\alpha_2 = 0.5$. As before, $\alpha_0 = 10^{-10}$ and $t = 0.1$, and it took twenty iteration steps to solve the problem to the desired accuracy. On the right we took $\alpha_1 = 10^{-2}$ and $\alpha_2 = 10^{-3}$, which appeared already to be small enough for very fast convergence in a few iteration steps. The exact and approximate eigenvalues are tabulated in Tabular 4.

-approximate values	-exact values	absolute errors
$1.682505953488249e+07$	$1.682505953488348e+07$	$9.946525096893311e-07$
$9.744661832185134e+06$	$9.744661832185108e+06$	$2.607703208923340e-08$
$9.742904198321559e+06$	$9.742904198322691e+06$	$1.132488250732422e-06$
$5.279641583582438e+06$	$5.279641583565828e+06$	$1.660920679569244e-05$
$4.596517485010833e+06$	$4.596517458224008e+06$	$2.678682561963797e-02$
$4.595293666371528e+06$	$4.595293704177797e+06$	$3.780626878142357e-02$

Tabular 4. Accuracy of the approximations of the largest six eigenvalues of PORES 2.

8.3 Conclusions and remarks

Even though we did not present many experiments, it is clear that we have developed a flexible method to approximate invariant subspaces and the corresponding eigenvalues. More experimenting with larger matrices is needed to prove the real value of this approach. Also, we did not yet experiment with subspace acceleration (which was not really needed in the examples showed), nor did we use any form of preconditioning in GCR. In future work we will do this, and moreover try to find a way how to compare the algorithm with JD and IBRQI in a reasonable way.

Remark 8.1 An indication of the success of our algorithms is that for the Harwell-Boeing cases, the amount of floating point operations to find a solution, was (only) three to four times more than the MATLAB sparse eigenvalue solver. ◊

Acknowledgments

This paper was written at Utrecht University as a first study into large eigenvalue problems in the framework of a Fellowship of the Royal Netherlands Academy of Arts and Sciences (KNAW). The support of KNAW is very much appreciated.

In the yearly *Babuška Prize Competition* for young researchers, the paper was awarded a Diploma of Honour in December 1999. This recognition by a contest Jury is very much appreciated.

Finally, the author wishes to thank Gerard Sleijpen [20] and Paul Smit [14, 15] for interesting suggestions and discussions on the topic.

References

[1] **Bartels, R.H. and Stewart, G.W** , *Solution of the equation* $AX + XB = C$, Comm. ACM 15, pp. 820–826, (1972).

[2] **Bittanti, S.; Laub, A.J. and Willems, J.C. (Eds.)**, *The Riccati Equation, Communications and Control Engineering Series*, Springer-Verlag, Berlin, (1991).

[3] **Brezinski, C. and Zaglia, M.R.**, *Extrapolation methods. Theory and practice*, North Holland, Amsterdam, (1991).

[4] **Davidson, E.R.**, *The iterative computation of a few of the lowest eigenvalues and corresponding eigenvectors of large real symmetric matrices*, J. Comput. Phys., 17, pp. 87–94, (1975).

[5] **Fokkema, D.R.**, *Subspace methods for linear, nonlinear, and eigen problems*, PhD-thesis Utrecht University, (1996).

[6] **Fokkema, D.R.; Sleijpen, G.L.G. and van der Vorst, H.A.**, *Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils*, SIAM J. Sc. Comput., 20(1), pp 94–125, (1998).

[7] **Gohberg, I.; Lancaster, P. and Rodman, L.**, *Invariant subspaces of matrices with applications*, Wiley, New York, (1986).

[8] **Golub, G.H. and van Loan, C.F.**, *Matrix computations, third edition*, The John Hopkins University Press, (1996).

[9] **Hackbusch, W.**, *Iterative solution of large sparse systems of equations* Applied Mathematical Sciences, Vol. 95, Springer- Verlag, New York, U.S.A., (1993).

[10] **Jacobi, C.G.J.**, *Ueber ein leichtes Verfahren, die in der Theorie der Saecularstoerungen vorkommenden Gleichungen numerisch aufzuloesen*, J. Reine und Angew. Math., pp. 51–94, (1846).

[11] **Lai,Lai, Y.L.; Lin, K.Y. and Lin, W.W.**, *An inexact inverse iteration for large sparse eigenvalue problems*, Num. Lin. Alg. Appl., 4(5), pp. 425–437, (1997).

[12] **Lehouqc, R.B.**, *Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration*, Ph.D. thesis, Rice University, Houston, Texas, (1995).

[13] **Matrix, Market**, *http://math.nist.gov/MatrixMarket/*

[14] **Smit, P.** *Numerical analysis of eigenvalue algorithms based on subspace iterations.* PhD thesis, Catholic University Brabant, Netherlands, (1997).

[15] **Smit, P. and Paardekooper, M.H.C.** *The effects of inexact solvers in algorithms for symmetric eigenvalue problems*, Lin. Alg. Appl., 287, pp. 337–357, (1999).

[16] **Sorensen, D.C.**, *Implicit Application of Polynomial Filters in a k-step Arnoldi Method*, SIAM J. Matrix Anal. Appl. Vol 13, pp. 357–385, (1992).

[17] **Stewart, G.W.**, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Review, Vol. 15(4), (1973).

[18] **Stewart, G.W and Sun, J.G.**, *Matrix Perturbation Theory*, Academic Press, London, (1990).

[19] **Varah, J.M.**, *On the separation of two matrices*, SIAM J. Num. Anal. 16, pp. 212–222, (1979).

[20] **Sleijpen, G.L.G. and van der Vorst, H.A.**, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Applic. 17, pp. 401–425, (1996).