A Chebyshev-based two-stage iterative method as an alternative to the direct solution of linear systems

Mario Arioli[1] and Daniel Ruiz[2]

ABSTRACT

We consider the solution of ill-conditioned symmetric and positive definite large sparse linear systems of equations. These arise, for instance, when using some symmetrising preconditioning technique for solving a general (possibly unsymmetric) ill-conditioned linear system, or in domain decomposition of a numerically difficult elliptic problem. We are also concerned with the consecutive solution of several linear systems with the same matrix and different right-hand sides. In such cases, the consecutive runs of some iterative methods like the conjugate gradient or the block conjugate gradient algorithms might be computationally prohibitive, and it might be preferable to use direct methods which are very well suited for this type of situation.

The purpose of our study is to analyse a two-phase approach: analogously to direct methods, it includes a preliminary phase which we call a "partial spectral factorization phase", followed by a "cheap" solution phase, both only based on numerical tools that are usually exploited in iterative methods. Therefore, we need not store the given ill-conditioned matrix explicitly but we need only use it to compute matrix-vector products. This is of particular interest in the case of very large sparse systems and permits an efficient implementation on distributed memory architectures.

Combining Chebyshev iterations with the block Lanczos algorithm, we propose a way to identify and extract precise information related to the ill-conditioned part of the given linear system, and to use it for the computation of extra solutions related to the same matrix with changing right-hand sides. The potential of this combination, which can be related to the factorization and direct solution of linear systems, is illustrated experimentally in the general case of the preconditioned iterative solution of unsymmetric sparse linear systems.

Keywords: Chebyshev polynomial, Block Lanczos method, Filtering.

AMS(MOS) subject classifications: 65F05, 65F50.

Current reports available by anonymous ftp to ftp.numerical.rl.ac.uk in directory pub/reports.

[1] M.Arioli@rl.ac.uk, Rutherford Appleton Laboratory,

[2] Daniel.Ruiz@enseeiht.fr, Ecole Nationale Supérieure d'Electrotechnique, d'Electronique, d'Informatique, et d'Hydraulique de Toulouse, Institut de Recherche en Informatique de Toulouse, 2 rue Camichel, 31071 Toulouse Cedex, France.

Computational Science and Engineering Department
Atlas Centre
Rutherford Appleton Laboratory
Oxon OX11 0QX

July 1st, 2002

Contents

1 Introduction

The solution of large, sparse, and ill-conditioned linear systems of equations, such as

$$\mathbf{A}\mathbf{u} = \mathbf{b}\,, \tag{1.1}$$

arises in many scientific and engineering problems. Several of these problems come from the numerical approximation of the mathematical models of complex physical phenomena. It is important to observe that in many of these problems the matrices are very large, with the number of unknowns ranging between about 10^6 and about 10^8. Moreover, the consecutive solution of several linear systems with the same matrix and changing right-hand sides is frequently required. It is not always possible to solve the system by means of a direct solver. In many cases, the iterative methods are the only feasible approach, and, in this respect, preconditioned Krylov subspace methods are one of the most powerful techniques used.

We assume that the matrix $\mathbf{A}$ in (1.1) is symmetric and positive definite (SPD). The SPD matrix $\mathbf{A}$ may arise when using some symmetrizable preconditioning technique (see Hageman and Young, 1981) for solving a general (possibly unsymmetric) ill-conditioned linear system, or in domain decomposition of some numerically difficult elliptic problem.

In order to address the solution of (1.1) for several right-hand sides, we analyse a two-phase approach: analogously to direct methods, it includes a preliminary "factorization" phase followed by a "cheap" solution phase, both based only on numerical tools that are usually exploited in iterative methods and both only requiring matrix-vector products. This latter requirement offers the possibility of keeping the matrix in implicit form. Unfortunately, several of the most commonly used preconditioners transform the matrix $\mathbf{A}$ to a matrix which has eigenvalues in a relatively small number of clusters, but which is still ill conditioned. This situation increases the arithmetic complexity of the conjugate gradient method: the number of iterations can excessively grow. An example of the spectrum of an iteration matrix of this type can be seen in Figure 2.1.

The proposed "factorization" phase technique computes and stores a basis for the invariant subspace associated with the smallest eigenvalues of the given SPD matrix $\mathbf{A}$. In order to conserve the invariance, we use Chebyshev polynomial filters that operate on the computed basis and on the right-hand side. For the computation of any further solution, we then use the computed basis to perform a deflation on the initial residual, and either a Chebyshev method or the classical conjugate gradient algorithm to compute the remaining part of the solution. In this way, we observe that the effect of the deflation helps the classical conjugate gradient algorithm to reach super-linear convergence immediately (see van der Sluis and van der Vorst, 1986) and it is then possible to compute consecutive solutions very quickly.

Arioli and Ruiz (1995) have already proposed a similar two-phase iterative approach, which first performs a partial spectral decomposition of the given matrix, and which afterward uses the previously derived information to compute solutions for subsequent right-hand sides. They first compute and store a basis for the invariant subspace associated with the smallest eigenvalues of the given SPD matrix $\mathbf{A}$ by using inverse subspace iteration (see Wilkinson, 1965, Parlett, 1980). Moreover, to compute the sets of multiple solutions required in the inverse subspace iteration algorithm, they use the block conjugate gradient algorithm, since it is particularly well designed for simultaneously computing multiple solutions and suffers less from the particular numerical properties of the linear systems under consideration.

However, the experiments in Arioli and Ruiz (1995) give evidence that the efficiency of the algorithm depends strongly on the appropriate choice of the starting block size, which determines the number of eigenvalues from the given matrix that will be approximated. On the one hand, the amount of work in the first phase of the algorithm increases with the block size. On the other hand, bigger block sizes not only improve the rate of convergence in the second phase, but also help to reduce the number of necessary inverse iterations in Phase 1. Unfortunately, it is difficult to find the best block size without additional knowledge of the spectrum of the given matrix.

In the present paper, our first phase, based on Chebyshev polynomial filters, results in a more robust algorithm that does not depend on the choice of the size of the initial basis. We want to stress that our approach is not a polynomial preconditioning of the given linear system.

The paper is organized as follows. Section 2 motivates the proposed method, and the algorithm is presented and discussed in Section 3. The convergence properties and the error analysis of the algorithm are given in Section 4. In Section 5, we suggest the possibility of computing the smallest eigenvalues and the corresponding eigenvectors with our algorithm. General comments as well as some analysis of the properties of the method are given in Section 6. Finally, we present the numerical experiments in Section 7, and some open questions and conclusions are discussed in Section 8.

2 The proposed two-phase iterative approach

We denote by $\mathbf{M} = \mathbf{R}^T\mathbf{R}$ a given symmetric and positive definite preconditioner and its Cholesky factorization. In the following, we will use the symmetric preconditioned system

$$\mathbf{R}^{-T}\mathbf{A}\mathbf{R}^{-1}\mathbf{R}\mathbf{u} = \mathbf{R}^{-T}\mathbf{b}$$

and we will denote by $\widehat{\mathbf{A}}$ the matrix $\mathbf{R}^{-T}\mathbf{A}\mathbf{R}^{-1}$, by $\mathbf{x}$ and by $\hat{\mathbf{b}}$ respectively the vectors $\mathbf{R}\mathbf{u}$ and $\mathbf{R}^{-T}\mathbf{b}$. As already mentioned in the introduction, we want, during the first phase, to compute those eigenvectors which are associated with the smallest eigenvalues of the ill-conditioned matrix $\widehat{\mathbf{A}}$.

To do so, we start with an initial set of s randomly generated vectors (s is the block-size), and we use Chebyshev polynomials in $\widehat{\mathbf{A}}$ to "*damp*", in these starting vectors, the eigenfrequencies associated with all the eigenvalues in some predetermined range (see Figure 2.2). We can fix, for instance, a positive number $\lambda_{\min}(\widehat{\mathbf{A}}) < \mu < \lambda_{\max}(\widehat{\mathbf{A}})$, and decide to compute all the eigenvectors associated with all the eigenvalues in the range $[\lambda_{\min}(\widehat{\mathbf{A}}), \mu]$. The computation of $\lambda_{\max}(\widehat{\mathbf{A}})$ is usually not too difficult, and in some cases, a sharp upper-bound may be already available through some *a priori* knowledge of the numerical properties of $\widehat{\mathbf{A}}$. If the eigenvalues are well clustered, and this is the role of the preconditioner $\mathbf{M}$ in use, the number of remaining eigenvalues inside the predetermined range $[\lambda_{\min}(\widehat{\mathbf{A}}), \mu]$, with reasonable μ (like $\lambda_{\max}/100$, or $\lambda_{\max}/10$, for instance), should be small compared to the size of the linear system. For example, in the small case study shown in Figure 2.1, there are 19 eigenvalues inside the interval $[\lambda_{\min}, \lambda_{\max}/100]$, and 26 eigenvalues inside the interval $[\lambda_{\min}, \lambda_{\max}/10]$.

Chebyshev polynomials can be defined by the following 2-term recurrence formula (see Hageman and Young, 1981, page 46):

$$\begin{cases} T_0(\omega) = 1\,, \quad T_1(\omega) = \omega \\ T_{n+1}(\omega) = 2\omega T_n(\omega) - T_{n-1}(\omega) \qquad n \geq 1. \end{cases} . \tag{2.1}$$

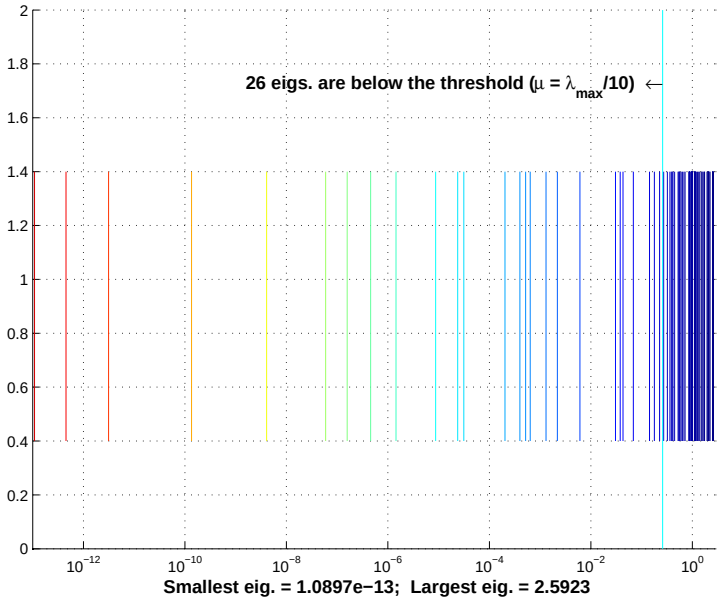


Figure 2.1: Eigenvalue distribution of a sample iteration test matrix (size of matrix is 137).

The optimal properties of Chebyshev polynomials given in Theorem 4.2.1 in Hageman and Young (1981, page 47) can be summarized as follows: if we consider $d > 1$ and

$$H_n(\omega) = \frac{T_n(\omega)}{T_n(d)},$$

then H_n has minimum l_∞ norm on the interval $[-1, 1]$ over all polynomials Q_n of degree less than of equal to n and satisfying the condition $Q_n(d) = 1$, and we have

$$\max_{\omega\in[-1,1]} H_n(\omega) = \frac{1}{T_n(d)}. \tag{2.2}$$

Now, consider the translation plus homothetic transformation:

$$\lambda \in \mathbb{R} \longmapsto \omega_\mu(\lambda) = \frac{\lambda_{\max}(\widehat{\mathbf{A}}) + \mu - 2\lambda}{\lambda_{\max}(\widehat{\mathbf{A}}) - \mu},$$

with $\lambda_{\min}(\widehat{\mathbf{A}}) < \mu < \lambda_{\max}(\widehat{\mathbf{A}})$ given above. This transformation maps $\lambda_{\max}(\widehat{\mathbf{A}})$ to -1, μ to 1, and 0 to

$$\omega_\mu(0) = d_\mu = \frac{\lambda_{\max}(\widehat{\mathbf{A}}) + \mu}{\lambda_{\max}(\widehat{\mathbf{A}}) - \mu} > 1.$$

Then, introducing

$$\mathcal{P}_n(\lambda) = \frac{T_n(\omega_\mu(\lambda))}{T_n(d_\mu)}, \tag{2.3}$$

we easily see, because of the optimal properties recalled above, that $\mathcal{P}_n(\lambda)$ has minimum l_∞ norm on the interval $[\mu, \lambda_{\max}(\widehat{\mathbf{A}})]$ over all polynomial Q_n of degree less than or equal to n

satisfying $Q_n(0) = 1$. Let us denote by

$$\widehat{\mathbf{A}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$$

the eigendecomposition of the SPD matrix $\widehat{\mathbf{A}}$, with $\mathbf{\Lambda} = \mathrm{diag}(\lambda_i)$, the matrix with the eigenvalues of $\widehat{\mathbf{A}}$ on the diagonal (in increasing order, for instance) and $\mathbf{U}$ the unitary matrix whose columns are the corresponding normalized eigenvectors of $\widehat{\mathbf{A}}$. If we multiply any vector $\mathbf{z}$, which can be decomposed in the eigenbasis of $\widehat{\mathbf{A}}$ as

$$\mathbf{z} = \sum_{i=1}^{n} \mathbf{u}_i \xi_i,$$

with $\xi_i = \mathbf{u}_i^T \mathbf{z}$, $i = 1, \ldots, n$, by the matrix $\mathcal{P}_n(\widehat{\mathbf{A}})$ we get

$$\mathbf{v} = \mathcal{P}_n(\widehat{\mathbf{A}})\mathbf{z} = \sum_{i=1}^{n} \mathbf{u}_i \left(\mathcal{P}_n(\lambda_i)\xi_i\right),$$

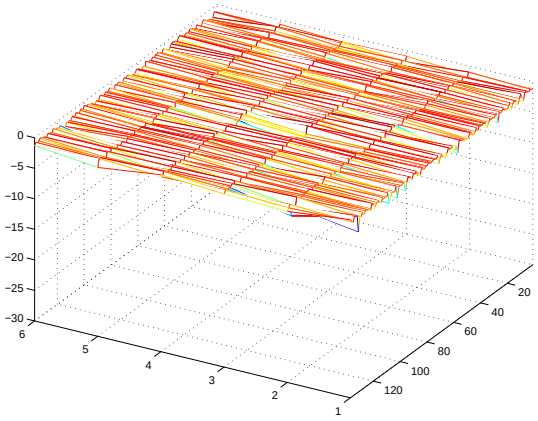
which shows that the eigencomponents of the resulting vector $\mathbf{v}$ are close to that of the initial vector $\mathbf{z}$ for all i such that λ_i is close to 0 and relatively much smaller for large enough degree n and all i such that $\lambda_i \in [\mu, \lambda_{\max}(\widehat{\mathbf{A}})]$. This is how we can easily damp the eigenfrequencies of any vector in the range $[\mu, \lambda_{\max}(\widehat{\mathbf{A}})]$ using classical Chebyshev iteration. This is illustrated in Figure 2.2 on a set of 6 randomly generated vectors whose eigencomponents corresponding to eigenvalues in the range $[\lambda_{\max}/10, \lambda_{\max}]$ have been damped close to machine precision. The number of Chebyshev iterations needed to reach some level ε for eigencomponents associated with eigenvalues in the range $[\mu, \lambda_{\max}]$, starting with normalized random vectors, is directly linked with the ratio $\lambda_{\max}/\mu$ (see Hageman and Young, 1981 and Golub and Kent, 1989), and can be very easily monitored using (2.2).

We can interpret the use of Chebyshev polynomials as a *filtering* tool that increases the degree of collinearity with some selected eigenvectors. Then, after "*filtering*" the initial starting vectors, we obtain a set of s vectors with eigencomponents below a certain level ε for those eigenvalues in the range $[\mu, \lambda_{\max}(\widehat{\mathbf{A}})]$, and relatively much bigger eigencomponents linked with the smallest eigenvalues in $\widehat{\mathbf{A}}$. Of course, the *a priori* choice of the block-size s may not correspond to or at least be greater than, the number k of remaining eigenvalues outside the interval $[\mu, \lambda_{\max}(\widehat{\mathbf{A}})]$. In the case $k \leq s$, a simple Ritz decomposition of the resulting s-dimensional filtered basis (see Parlett, 1980) would enable us to get easily the k wanted eigenvectors with a residual directly linked to the filtering level ε. To overcome the difficulties related to the case $k > s$, we propose to use a Block-Lanczos type of approach (see O'Leary, 1980, Underwood, 1975, Golub and Underwood, 1977 for details on such techniques) to build a Krylov basis starting with these filtered vectors and to stop when appropriate.

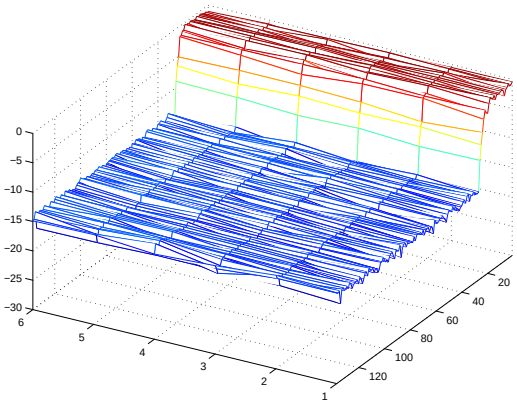
One of the main drawbacks with a Lanczos/Block-Lanczos algorithm is that it does not maintain the nice property of the filtered vectors, and gradually (and rather quickly, indeed) the Lanczos vectors may again have eigencomponents all about the same level. In Figure 2.3, we show (in a logarithmic scale) the eigendecomposition of the Krylov basis of size 30 obtained after 5 steps of block Lanczos algorithm with block-size 6, starting with the initial "*filtered*" set of 6 vectors whose eigencomponents are shown in Figure 2.2. From this experiment, it can be seen that "*near*" invariance with respect to the eigensubspace linked to the k smallest eigenvalues can be completely lost, and the termination of the Block Lanczos algorithm may not occur

after building a basis of dimension close to k (k being equal to 26 in the present experiment). It must be mentioned beforehand that this problem is inherent to the Lanczos/Block Lanczos iteration, and is not at all due to any kind of loss of orthogonality in the Lanczos basis. To produce the results indicated in Figure 2.3, we have indeed performed a full re-orthogonalization of the Lanczos vectors at each iteration, and this is explicitly illustrated by the second graph in Figure 2.3 which shows the scalar products (in a logarithmic scale) between each pair of computed Lanczos vectors.

To maintain the level of the unwanted eigenfrequencies in the orthonormal block Krylov basis under ε, we propose to perform, at each Block-Lanczos iteration, a few extra Chebyshev iterations on the newly generated Block Lanczos vectors $\mathbf{V}^{(k+1)}$. In this way, "*near*" invariance can be maintained. Figure 2.4 illustrates the benefit of this additional filtering step performed at each Lanczos iteration on the same test case as the one of Figure 2.3. It must be mentioned, beforehand, that the nice property of the block-Krylov spaces, which makes the projected matrix $\mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V}$ block-tridiagonal, is lost after re-filtering the current block $\mathbf{V}^{(k+1)}$. Therefore, it is necessary to orthogonalize at each iteration the filtered vectors against all the previously constructed ones, as in an Orthodir process for instance. This is the reason why we denote this process as "*Lanczos/Orthodir*" in subsequent figures, meaning only that we cannot simplify the orthogonalization step when computing the next block of Krylov vectors and iterate with



Eigencomponents of the starting set of orthonormal vectors (6 vectors, randomly generated, and orthonormalized).



Eigencomponents of this set of vectors after 51 Chebyshev iterations (the resulting 6 vectors have also been re-orthonormalized).

- The eigenvectors of $\mathbf{A}$ are indexed on the X-axis (from 1 to 136) in increasing order of their corresponding eigenvalue.
- The indexes (from 1 to 6) of the vectors in the current set are indicated on the Y-axis.
- The Z-axis indicates the logarithm of the absolute values of the eigencomponents in each of the 6 vectors.

Figure 2.2: Use of Chebyshev polynomials in $\widehat{\mathbf{A}}$ to damp eigencomponents associated with the eigenvalues of $\widehat{\mathbf{A}}$ in the range $[\mu, \lambda_{\max}(\widehat{\mathbf{A}})]$ $(\mu = \lambda_{\max}(\widehat{\mathbf{A}})/10)$.

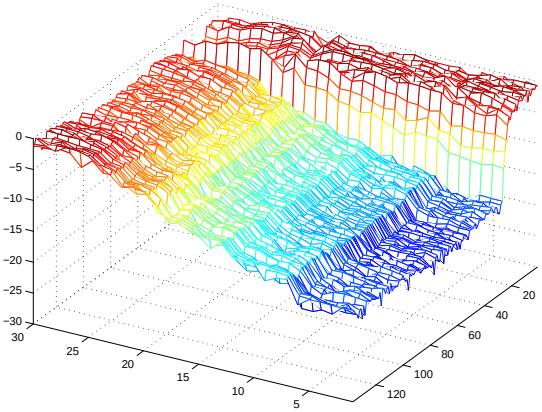
a simple 3 term recurrence formula in the usual way, as in the classical Lanczos algorithm. Finally, it is important to notice that the number of extra filtering Chebyshev iterations at each Lanczos step is less than the number of Chebyshev iterations performed at the starting point, since the degradation of the information is very gradual until the very last steps when *near* invariance with respect to $\mathbf{U}_1$ must be lost.

Once the *near*-invariant subspace linked to the smallest eigenvalues is obtained, we can use it for the computation of further solutions. The idea is to perform an oblique projection of the initial residual ($\hat{\mathbf{r}}_0 = \hat{\mathbf{b}} - \widehat{\mathbf{A}}\mathbf{x}_0$) onto this *near* invariant subspace in order to get the eigencomponents in the solution corresponding to the smallest eigenvalues, viz.

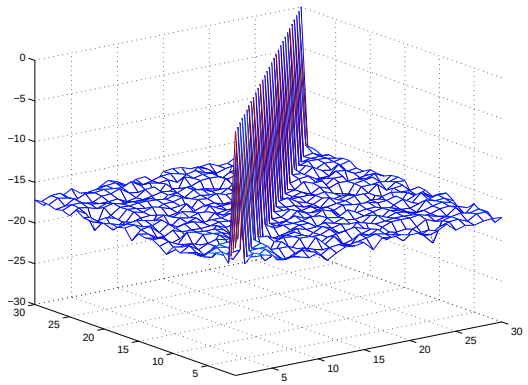
$$\begin{aligned} \hat{\mathbf{r}}_1 &= \hat{\mathbf{r}}_0 - \widehat{\mathbf{A}}\mathbf{V}\left(\mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V}\right)^{-1}\mathbf{V}^T\hat{\mathbf{r}}_0, \\ \text{with} \quad \mathbf{x}_1 &= \mathbf{x}_0 + \mathbf{V}\left(\mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V}\right)^{-1}\mathbf{V}^T\hat{\mathbf{r}}_0. \end{aligned} \tag{2.4}$$

To compute the remaining part of the solution vector $\widehat{\mathbf{A}}\mathbf{x}_2 = \hat{\mathbf{r}}_1$, one can then use the classical Chebyshev algorithm with eigenvalue bounds given by μ and $\lambda_{\max}(\widehat{\mathbf{A}})$, as explained by Hageman and Young (1981, Chapter 4).

Note that, if $\mathbf{V}$ exactly spans the invariant subspace generated by $\mathbf{U}_1$, the Chebyshev iteration and the oblique projection steps in the solution phase can be performed, though sequentially, in any order *a priori*. However, since $\mathit{Span}(\mathbf{V})$ is only an approximation of this invariant subspace, we prefer to perform the Chebyshev step first, followed then by the oblique projection, because this enables us to increase the accuracy of the oblique projection by "*minimizing*" the influence of the eigencomponents in $\mathbf{U}_2$ relative to those in $\mathbf{U}_1$ in the scalar products $\mathbf{V}^T\hat{\mathbf{r}}$



The Z-axis indicates the logarithm of the absolute values of the eigencomponents in each Lanczos vector.



The Z-axis indicates the logarithm of the absolute values of the scalar products between each pair of Lanczos vectors.

Figure 2.3: Eigendecomposition of the Krylov basis obtained after 5 block-Lanczos steps (Block Lanczos with block size 6 and full classical Gram-Schmidt re-orthogonalization). The iteration matrix in use is the one with spectrum given in Figure 2.1.

in (2.4), in particular when the filtering level ε is not close to machine precision. This is illustrated by the results in Figure 2.5, where the solution for a given right-hand side vector $\widehat{\mathbf{b}}$ (corresponding to an exact solution vector $\mathbf{x}^*$ with random entries) is computed in two steps, with the use of Chebyshev iterations first and the oblique projection applied to the resulting intermediate residual.

3 The algorithm

The algorithm we introduced in the previous section is constructed in the same spirit as direct methods, with a factorization phase, which we call a "*partial spectral factorization phase*", followed by a cheap "*solution phase*" in two steps, but it offers the possibility of not building the matrix $\widehat{\mathbf{A}}$ explicitly, since only matrix-vector multiplications are needed at every stage of the algorithm.

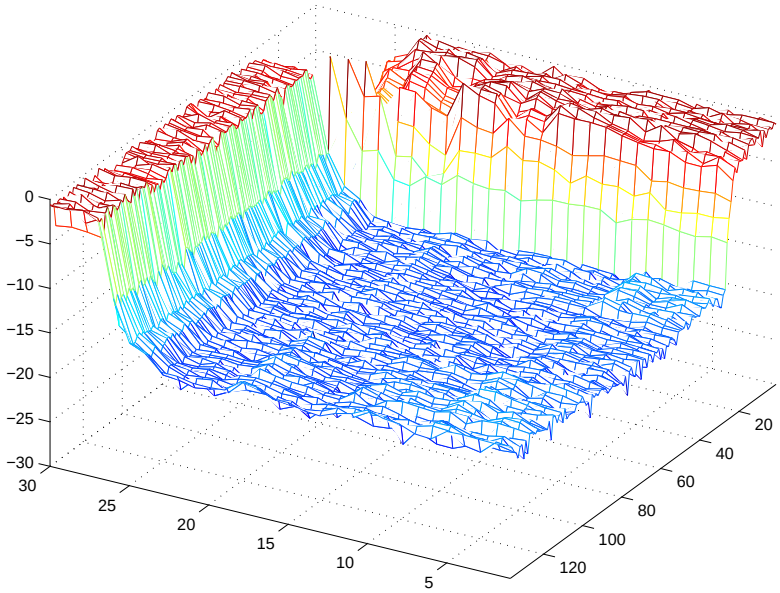
Let us first denote by

$$\mathbf{Z} = \text{Chebyshev\_Filter}(\mathbf{Y}, \varepsilon, [\mu, \lambda_{\max}], \widehat{\mathbf{A}})$$

the application of a Chebyshev polynomial in $\widehat{\mathbf{A}}$ to $\mathbf{Y}$, viz.

$$\mathbf{Z} = \mathcal{P}_n(\widehat{\mathbf{A}})\mathbf{Y},$$

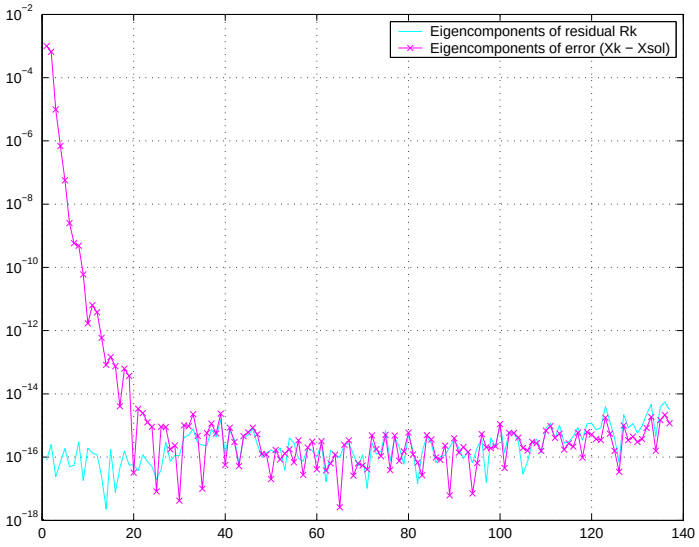
where $\mathcal{P}_n$ is defined as in (2.3) with a degree such that it is L_∞ norm over the interval $[\mu, \lambda_{\max}]$ is less than ε. Using this notation, and fixing the block size to s, the cut-off eigenvalue μ,



The Z-axis indicates the logarithm of the absolute values of the eigencomponents in each Lanczos/Orthodir vector.

Figure 2.4: Eigendecomposisiton of the Krylov basis obtained after 5 block-Lanczos/Orthodir steps with additional Chebyshev filtering at each Lanczos iteration, starting with the same initial filtered set of vectors as that corresponding to the experiments in Figure 2.3.

The first part of the solution is obtained with Chebyshev iterations (58 in this case), whose purpose are to damp the eigencomponents of vector $\hat{\mathbf{b}}$ associated with the eigenvalues in the predetermined range $[\mu, \lambda_{\max}(\widehat{\mathbf{A}})]$.



The second part of the solution is obtained with an oblique projection of the current residual onto the subspace linked to the computed Lanczos basis.

Figure 2.5: Computation of the solution of a linear system with a right-hand side vector $\hat{\mathbf{b}}$ corresponding to a given random exact solution vector $\mathbf{x}^*$.

$\lambda_{\min}(\widehat{\mathbf{A}}) < \mu < \lambda_{\max}(\widehat{\mathbf{A}})$, and the filtering level $\varepsilon \ll 1$, we can then describe the partial spectral factorization phase as

Algorithm 3.1 (Chebyshev-based Partial Spectral Factorization phase)

$$
\begin{aligned}
&\mathbf{P}^{(0)} = \text{random}(n, s); \quad \text{and } \mathbf{Q}^{(0)} = \text{orthonormalize}(\mathbf{P}^{(0)}) \\
&\mathbf{Z}^{(0)} = \text{Chebyshev\_Filter}(\mathbf{Q}^{(0)}, \varepsilon, [\mu, \lambda_{\max}], \widehat{\mathbf{A}}) \\
&[\mathbf{Y}^{(0)}, \mathbf{\Sigma}_1^{(0)}, \mathbf{W}] = \text{SVD}(\mathbf{Z}^{(0)}, 0); \quad \text{and } \delta_0 = \sigma_{\min}(\mathbf{\Sigma}_1^{(0)}) \\
&\widehat{\mathbf{Z}}^{(0)} = \text{Chebyshev\_Filter}(\mathbf{Y}^{(0)}, \delta_0, [\mu, \lambda_{\max}], \widehat{\mathbf{A}}); \quad \text{and } \mathbf{V}^{(0)} = \text{orthonormalize}(\widehat{\mathbf{Z}}^{(0)}) \\
&\mathbf{V} = \mathbf{V}^{(0)}; \quad \text{and set } \delta_2 = 1 \\
&\text{for } k = 0, 1, 2, \ldots, \text{ until convergence do :} \\
&\quad \mathbf{P}^{(k+1)} = \widehat{\mathbf{A}}\mathbf{V}^{(k)} - \mathbf{V}\mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V}^{(k)} \\
&\quad [\mathbf{V}^{(k+1)}, \mathbf{\Sigma}_1^{(k+1)}, \mathbf{W}] = \text{SVD}(\mathbf{P}^{(k+1)}, 0); \\
&\quad \text{set } \delta_1 = \sigma_{\min}(\mathbf{\Sigma}_1^{(k+1)})/\lambda_{\max} \text{ and } \delta = \max(\varepsilon, \delta_1 \times \delta_2) \\
&\quad \text{for } i = 1, 2 \\
&\qquad \left.\begin{aligned}
&\mathbf{Z}^{(k+1)} = \text{Chebyshev\_Filter}(\mathbf{V}^{(k+1)}, \delta, [\mu, \lambda_{\max}], \widehat{\mathbf{A}}) \\
&\mathbf{Y}^{(k+1)} = \mathbf{Z}^{(k+1)} - \mathbf{V}\mathbf{V}^T\mathbf{Z}^{(k+1)} \\
&[\mathbf{V}^{(k+1)}, \mathbf{\Sigma}_2^{(k+1)}, \mathbf{W}] = \text{SVD}(\mathbf{Y}^{(k+1)}, 0) \\
&\text{set } \ \delta_2 = \sigma_{\min}(\mathbf{\Sigma}_2^{(k+1)}) \ \text{ and } \ \delta = \delta_2
\end{aligned}\right\} \qquad (3.1) \\
&\quad \text{end} \\
&\quad \mathbf{V} = [\mathbf{V}; \mathbf{V}^{(k+1)}]
\end{aligned}
$$

The notation SVD in the algorithm above describes the Singular Value Decomposition, as for instance in $[\mathbf{Y}^{(0)}, \mathbf{\Sigma}_1^{(0)}, \mathbf{W}] = \text{SVD}(\mathbf{Z}^{(0)}, 0)$, and must be understood as

$$\mathbf{Z}^{(0)} = \mathbf{Y}^{(0)}\mathbf{\Sigma}_1^{(0)}\mathbf{W}^T$$

where $\mathbf{Y}^{(0)}$ s an $n \times s$ matrix, $\mathbf{\Sigma}_1^{(0)}$ is an $s \times s$ diagonal matrix with the singular values of $\mathbf{Z}^{(0)}$ on the diagonal, and $\mathbf{W}$ is an $s \times s$ orthonormal matrix which we do not consider afterwards.

Note also that the starting point in Algorithm 3.1 implies two steps of filtering of the initial randomly generated basis. The issue there is that randomly generated vectors have eigencomponents all about the same size (at least on average) and, thus, after the first filtering step, the resulting vectors may have a much smaller norm depending on how many eigenvalues in $\widehat{\mathbf{A}}$ remain outside the damping interval $[\mu, \lambda_{\max}]$. Then, the orthonormalization step that occurs after filtering these vectors may increase the actual level ε of the damped eigencomponents. The purpose of the second filtering step is to ensure that the filtering level of these eigencomponents in the starting basis is actually very close to ε.

To describe the solution phase, we also denote by

$$[\hat{\mathbf{r}}_1, \mathbf{x}_1] = \text{Chebyshev\_Solve}(\hat{\mathbf{r}}_0, \mathbf{x}_0, \delta, [\mu, \lambda_{\max}], \widehat{\mathbf{A}})$$

the application of Chebyshev polynomial in $\widehat{\mathbf{A}}$ the purpose of which is to reduce by a factor of δ the eigencomponents in $\hat{\mathbf{r}}_0$ associated with the eigenvalues in the range $[\mu, \lambda_{\max}]$, providing thus the resulting residual $\hat{\mathbf{r}}_1 = \mathcal{P}_n(\widehat{\mathbf{A}})\hat{\mathbf{r}}_0$ and the corresponding update $\mathbf{x}_1$ such that $\hat{\mathbf{b}} - \widehat{\mathbf{A}}\mathbf{x}_1 = \hat{\mathbf{r}}_1$.

Algorithm 3.2 (Solution phase)

For any right-hand side vector $\hat{\mathbf{b}}$, set $\mathbf{x}_0$ and $\hat{\mathbf{r}}_0 = \hat{\mathbf{b}} - \widehat{\mathbf{A}}\mathbf{x}_0$,

and perform the two following consecutive steps:

$$[\hat{\mathbf{r}}_1, \mathbf{x}_1] = \text{Chebyshev\_Solve}(\hat{\mathbf{r}}_0, \mathbf{x}_0, \varepsilon, [\mu, \lambda_{\max}], \widehat{\mathbf{A}})$$
$$\hat{\mathbf{r}} = \hat{\mathbf{r}}_1 - \widehat{\mathbf{A}}\mathbf{V}\left(\mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V}\right)^{-1}\mathbf{V}^T\hat{\mathbf{r}}_1, \quad \text{and } \mathbf{x} = \mathbf{x}_1 + \mathbf{V}\left(\mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V}\right)^{-1}\mathbf{V}^T\hat{\mathbf{r}}_1$$

It is also possible to iterate on that solution phase, and improve the solution with iterative refinement in the usual way.

4 Monitoring and convergence analysis

Let us first analyse in more detail why the eigencomponents in the Krylov vectors that were initially damped under some level ε must increase so strongly at each block Lanczos step, and how we may maintain the desired level of "*filtering*" in these eigencomponents.

Consider, for instance, that we have built a current block-Krylov orthonormal basis $\mathbf{V} = [\mathbf{V}^{(0)}, \ldots, \mathbf{V}^{(k)}]$ with the property that

$$\begin{aligned} \mathbf{V} &= \mathbf{U}_1\mathbf{\Gamma} + \mathbf{U}_2\mathbf{\Psi}, \text{ and } \|\mathbf{\Psi}\| \leq \varepsilon, \\ \text{with } \widehat{\mathbf{A}} &= \mathbf{U}_1\mathbf{\Lambda}_1\mathbf{U}_1^T + \mathbf{U}_2\mathbf{\Lambda}_2\mathbf{U}_2^T, \end{aligned} \tag{4.1}$$

$\mathbf{\Lambda}_1$ being the diagonal matrix made with all eigenvalues of $\widehat{\mathbf{A}}$ less than μ and $\mathbf{U}_1$ the set of corresponding eigenvectors in matrix form, and $\mathbf{\Lambda}_2$ and $\mathbf{U}_2$ the complementary corresponding matrices.

Now, the next step in the block Lanczos process is to build

$$\mathbf{P}^{(k+1)} = \widehat{\mathbf{A}}\mathbf{V}^{(k)} - \mathbf{V}\mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V}^{(k)}$$

and to orthonormalize the set of s vectors $\mathbf{P}^{(k+1)}$ to get the next entry $\mathbf{V}^{(k+1)}$ in the block Krylov orthonormal basis $\mathbf{V}$. Using the decomposition (4.1), we can then write

$$\begin{aligned} \mathbf{P}^{(k+1)} &= \mathbf{U}_1\left(\mathbf{\Lambda}_1\mathbf{\Gamma}^{(k)} - \mathbf{\Gamma}\mathbf{\Gamma}^T\mathbf{\Lambda}_1\mathbf{\Gamma}^{(k)} - \mathbf{\Gamma}\mathbf{\Psi}^T\mathbf{\Lambda}_2\mathbf{\Psi}^{(k)}\right) \\ &+ \mathbf{U}_2\left(\mathbf{\Lambda}_2\mathbf{\Psi}^{(k)} - \mathbf{\Psi}\mathbf{\Psi}^T\mathbf{\Lambda}_2\mathbf{\Psi}^{(k)} - \mathbf{\Psi}\mathbf{\Gamma}^T\mathbf{\Lambda}_1\mathbf{\Gamma}^{(k)}\right), \end{aligned} \tag{4.2}$$

where $\mathbf{\Gamma}^{(k)}$ and $\mathbf{\Psi}^{(k)}$ are the sub-blocks in $\mathbf{\Gamma}$ and $\mathbf{\Psi}$ corresponding to the subset of vectors $\mathbf{V}^{(k)}$ in $\mathbf{V}$.

If we assume also, for simplicity, that $\varepsilon \leq 10^{-8}$ so that we can neglect to a first approximation the terms in $\mathcal{O}(\|\mathbf{\Psi}\|^2)$, we easily see that the updating process in (4.2) reduces to a first part in $\mathbf{U}_1$, given by

$$\mathbf{U}_1 \left(\mathbf{\Lambda}_1 \mathbf{\Gamma}^{(k)} - \mathbf{\Gamma}\mathbf{\Gamma}^T \mathbf{\Lambda}_1 \mathbf{\Gamma}^{(k)} \right),$$

and corresponds simply to the block Krylov update one would obtain when working directly with a *projected* matrix whose spectrum corresponds to $\mathbf{\Lambda}_1$, and a part in $\mathbf{U}_2$,

$$\mathbf{U}_2 \left(\mathbf{\Lambda}_2 \mathbf{\Psi}^{(k)} - \mathbf{\Psi}\mathbf{\Gamma}^T \mathbf{\Lambda}_1 \mathbf{\Gamma}^{(k)} \right),$$

whose norm must stay very close to $\|\mathbf{\Psi}\| \leq \varepsilon$ since the maximum eigenvalue in $\mathbf{\Lambda}_1$ is less than the minimum one in $\mathbf{\Lambda}_2$. Therefore, the cancellation due to the orthogonalization process (4.2) occurs mostly within the part in $\mathbf{U}_1$ and, additionally, since all eigenvalues in $\mathbf{\Lambda}_1$ are less than μ, the norm of the resulting part in $\mathbf{U}_1$ must also decrease by a factor of $(\mu/\lambda_{\max}(\widehat{\mathbf{A}}))$ relatively to the part in $\mathbf{U}_2$. These two combined effects are the responsible for the "*staircase*" increase that one can observe in Figure 2.3 in the eigencomponents relative to $\mathbf{U}_2$ of the Krylov iterates.

For these reasons, we have introduced the extra Chebyshev filtering steps (3.1) after the actual block Lanczos step in algorithm 3.1, in order to recover the above described loss of information and maintain the norm of $\mathbf{\Psi}^{(k+1)}$ $(= \mathbf{U}_2^T \mathbf{V}^{(k+1)})$ in the next set of Lanczos vectors close to that of $\mathbf{\Psi}^{(k)}$, and recursively close to the actual value of $\|\mathbf{\Psi}^{(0)}\|$ in the initial set of filtered vectors $\mathbf{V}^{(0)}$.

Finally, we have forced a double process of filtering in algorithm 3, where we repeated the steps (3.1) twice. Indeed, the application of Chebyshev polynomial filtering to $\mathbf{V}^{(k+1)}$ in (3.1) may also induce a loss of information, in particular when the vectors $\mathbf{V}^{(k+1)}$ have mostly large eigencomponents associated with eigenvalues close to the cut-off value mu. In this case, and because of the continuity of the Chebyshev polynomials over the whole interval $[\lambda_{\min}, \lambda_{\max}]$, the resulting filtered vectors $\mathbf{Z}^{(k+1)}$ may have a norm that gets close to δ (e.g. the current level of re-filtering on the interval $[\mu, \lambda_{\max}]$). This can be detected in the actual value of δ_2 after normalization of the resulting vectors, and applying a second time the filtering process (3.1) helps to guaranty that $\|\mathbf{\Psi}\| \leq \varepsilon\sqrt{m}$ (where m is the rank of $\mathbf{\Psi}$), at least until convergence is not about to be reached. Moreover, we will show in Section 7 how this heuristic has a favourable cost/benefit.

Convergence or *near* invariance with respect to $\mathbf{U}_1$ can be detected when the new vectors that are built become mostly collinear to the unwanted eigenvectors (in $\mathbf{U}_2$), meaning that the eigenbasis we look for is contained in the current block Lanczos basis and resulting in an update $\mathbf{Y}^{(k+1)}$ in (3.1) with a norm close to ε. This means that some of the filtered vectors, after being orthogonalized against the previously computed basis $\mathbf{V}$, still get a norm close to ε and thus must become collinear to the subspace generated by $\mathbf{U}_2$ (as described in the previous section). It must be mentioned that similar testing on δ_1 may not indicate convergence. Indeed, there is some inherent ill-conditioning that may occur within the aforementioned block Krylov vectors $\mathbf{P}^{(k+1)}$ when convergence is about to be reached (see, for instance, Arioli, Duff, Ruiz and Sadkane, 1995), but still, the computed orthonormal basis $\mathbf{V}$ does not have a dimension greater than or equal to that of the invariant subspace $\mathbf{U}_1$ and iterations must be continued.

Another way of testing convergence or *near* invariance with respect to $\mathbf{U}_1$ could also be to generate an extra random vector at the beginning, to filter it under the level ε and normalize it, and to test when appropriate (e.g. when δ_2 becomes small) if the norm of the orthogonal

projection of this vector onto the orthogonal complement of the computed basis $\mathbf{V}$ is close to ε or not. This way of performing the convergence tests is directly connected to the results obtained in the error analysis detailed in the next section.

4.1 Error Analysis

In the Solution phase we perform an oblique projection of the filtered residual. This implies that we operate within $\mathbb{R}^n$ with scalar product $\mathbf{x}^T\widehat{\mathbf{A}}\mathbf{x}$. Therefore, the residual is a linear form belonging to the dual space, and the natural norm of the dual space is $(\hat{\mathbf{r}}^T\widehat{\mathbf{A}}^{-1}\hat{\mathbf{r}})^{1/2}$. We observe that:

$$||\hat{\mathbf{r}}||_{\widehat{\mathbf{A}}^{-1}} = ||\mathbf{x}^* - \mathbf{x}||_{\widehat{\mathbf{A}}}.$$

The value $||\hat{\mathbf{r}}||_{\widehat{\mathbf{A}}^{-1}}$ can be evaluated by using the following expression:

$$\begin{aligned}
\widehat{\mathbf{A}}^{-\frac{1}{2}}\hat{\mathbf{r}} &= (\mathbf{I} - \widehat{\mathbf{A}}^{\frac{1}{2}}\mathbf{V}(\mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V})^{-1}\mathbf{V}^T\widehat{\mathbf{A}}^{\frac{1}{2}})\widehat{\mathbf{A}}^{-\frac{1}{2}}\mathcal{P}_n(\widehat{\mathbf{A}})\hat{\mathbf{r}}_0 \\
&= (\mathbf{I} - \widehat{\mathbf{A}}^{\frac{1}{2}}\mathbf{V}(\mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V})^{-1}\mathbf{V}^T\widehat{\mathbf{A}}^{\frac{1}{2}})\widehat{\mathbf{A}}^{\frac{1}{2}}\mathcal{P}_n(\widehat{\mathbf{A}})\mathbf{e}_0 \\
&= \wp\widehat{\mathbf{A}}^{\frac{1}{2}}\mathbf{v}||\mathcal{P}_n(\widehat{\mathbf{A}})\mathbf{e}_0||_2,
\end{aligned}$$

where $\wp = (\mathbf{I} - \widehat{\mathbf{A}}^{\frac{1}{2}}\mathbf{V}(\mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V})^{-1}\mathbf{V}^T\widehat{\mathbf{A}}^{\frac{1}{2}})$, $\mathbf{e}_0 = \widehat{\mathbf{A}}^{-1}\hat{\mathbf{r}}_0 = \mathbf{x}^* - \mathbf{x}_0$, and

$$\mathbf{v} = \mathcal{P}_n(\widehat{\mathbf{A}})\hat{\mathbf{r}}_0/||\mathcal{P}_n(\widehat{\mathbf{A}})\mathbf{e}_0||_2. \tag{4.1.1}$$

Thus, because $||\mathcal{P}_n(\widehat{\mathbf{A}})||_2 \leq 1$ and $||\mathbf{e}_0||_2 \leq ||\widehat{\mathbf{A}}^{-\frac{1}{2}}||_2||\hat{\mathbf{r}}_0||_{\widehat{\mathbf{A}}^{-1}}$ we have

$$||\hat{\mathbf{r}}||_{\widehat{\mathbf{A}}^{-1}} \leq ||\wp\widehat{\mathbf{A}}^{\frac{1}{2}}\mathbf{v}||_2||\hat{\mathbf{r}}_0||_{\widehat{\mathbf{A}}^{-1}}||\widehat{\mathbf{A}}^{-\frac{1}{2}}||_2. \tag{4.1.2}$$

Moreover, we can introduce the following representation for $\mathbf{v}$:

$$\begin{aligned}
\mathbf{v} &= \mathbf{V}\zeta + (\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v} \\
\zeta &= \arg\min ||\mathbf{V}\mathbf{y} - \mathbf{v}||_2.
\end{aligned}$$

Then the following relations hold:

$$\begin{aligned}
||\wp\widehat{\mathbf{A}}^{\frac{1}{2}}\mathbf{v}||_2 &= ||\wp\widehat{\mathbf{A}}^{\frac{1}{2}}(\mathbf{V}\zeta + (\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v})||_2 \\
&= ||\wp\widehat{\mathbf{A}}^{\frac{1}{2}}(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v}||_2 \\
&\leq ||\wp||_2||\widehat{\mathbf{A}}^{\frac{1}{2}}||_2||(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v}||_2 \\
&\leq ||\widehat{\mathbf{A}}^{\frac{1}{2}}||_2||(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v}||_2.
\end{aligned}$$

Finally, from (4.1.2) we have:

$$\frac{||\hat{\mathbf{r}}||_{\widehat{\mathbf{A}}^{-1}}}{||\hat{\mathbf{r}}_0||_{\widehat{\mathbf{A}}^{-1}}} \leq ||\widehat{\mathbf{A}}^{\frac{1}{2}}||_2||\widehat{\mathbf{A}}^{-\frac{1}{2}}||_2||(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v}||_2. \tag{4.1.3}$$

The following theorem gives an upper bound for $||(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v}||_2$ in terms of the filtering level ε.

THEOREM 4.1 *Let $\mathbf{U}_1 \in \mathbb{R}^{n\times m}$ be the matrix of the eigenvectors of $\widehat{\mathbf{A}}$ corresponding to $\mathbf{\Lambda}_1$ and $\mathbf{U}_2 \in \mathbb{R}^{n\times(n-m)}$ be the matrix of the remaining eigenvectors of $\widehat{\mathbf{A}}$. Let $\mathbf{V} \in \mathbb{R}^{n\times\ell}$ be the full basis generated by Algorithm 3.1 using a filtering level ε and $\mathbf{v}$ be the vector defined by (4.1.1). If $m\varepsilon \ll 1$ and $\ell \geq m$ then*

$$||(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v}||_2 \leq 2\varepsilon\sqrt{m} + 2\varepsilon^2 m.$$

Proof. We give the proof in the case $n - \ell \geq \ell \geq m$: the case $n - \ell < \ell$ can be proved making few and evident adjustments on the matrix $\mathbf{\Sigma}$ in the CS-decomposition that appears in the following. The filtering process at the start and during the algorithm computes a matrix $\mathbf{V}$ and vector $\mathbf{v}$ such that their representations in the eigenvector basis $\mathbf{U} = [\mathbf{U}_1; \mathbf{U}_2]$ of $\widehat{\mathbf{A}}$ have the form: $\mathbf{V} = \mathbf{U}\mathbf{H}$ and $\mathbf{v} = \mathbf{U}\hat{\mathbf{v}}$, with $\mathbf{H}^T\mathbf{H} = \mathbf{I}$, and

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{\bullet 1} & \mathbf{H}_{\bullet 2} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{matrix} \}m \\ \}n-m \end{matrix}, \quad \overbrace{}^{m} \overbrace{}^{\ell-m}$$

$$\hat{\mathbf{v}} = \begin{bmatrix} \hat{\mathbf{v}}_1 \\ \hat{\mathbf{v}}_2 \end{bmatrix} \begin{matrix} \}m \\ \}n-m \end{matrix}.$$

Moreover, each entry in $\mathbf{H}_{21}$ and $\hat{\mathbf{v}}_2$ is bounded by ε and we have

$$\begin{aligned} ||\mathbf{H}_{21}||_2 &\leq ||\mathbf{H}_{21}||_F \leq \sqrt{m}\varepsilon \\ ||\hat{\mathbf{v}}_2||_2 &\leq \varepsilon. \end{aligned}$$

Owing to the orthogonality of $\mathbf{H}$, we have that

$$\begin{aligned} \mathbf{I} - \mathbf{V}\mathbf{V}^T &= \mathbf{U}(\mathbf{I} - \mathbf{H}\mathbf{H}^T)\mathbf{U}^T = \mathbf{U}(\mathbf{I} - [\mathbf{H}_{\bullet 1}\mathbf{H}_{\bullet 2}]\,[\mathbf{H}_{\bullet 1}\mathbf{H}_{\bullet 2}]^T)\mathbf{U}^T \\ &= \mathbf{U}(\mathbf{I} - \mathbf{H}_{\bullet 1}\mathbf{H}_{\bullet 1}^T - \mathbf{H}_{\bullet 2}\mathbf{H}_{\bullet 2}^T)\mathbf{U}^T \\ &= \mathbf{U}(\mathbf{I} - \mathbf{H}_{\bullet 1}\mathbf{H}_{\bullet 1}^T)(\mathbf{I} - \mathbf{H}_{\bullet 2}\mathbf{H}_{\bullet 2}^T)\mathbf{U}^T \\ &= \mathbf{U}(\mathbf{I} - \mathbf{H}_{\bullet 2}\mathbf{H}_{\bullet 2}^T)(\mathbf{I} - \mathbf{H}_{\bullet 1}\mathbf{H}_{\bullet 1}^T)\mathbf{U}^T \end{aligned}$$

Under the hypothesis $n - \ell \geq \ell \geq m$, the CS-decomposition of $\mathbf{H}_{\bullet 2}$ takes the form (Stewart 1982):

$$\mathbf{H}_{\bullet 1} = \mathbf{W}\mathbf{\Sigma}\mathbf{Q}^T$$

where $\mathbf{W}^T\mathbf{W} = \mathbf{W}\mathbf{W}^T = \mathbf{I}$, $\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$, and

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & 0_{m\times(n-m)} \\ 0^T_{m\times(n-m)} & \mathbf{W}_2 \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} \mathbf{C} \\ 0_{n-2m,m} \\ \mathbf{S} \end{bmatrix},$$

with $\mathbf{C} = diag(c_1, \ldots, c_m)$, $\mathbf{S} = diag(s_1, \ldots, s_m)$, and $\mathbf{C}^2 + \mathbf{S}^2 = \mathbf{I}$. Moreover, because $||\mathbf{H}_{21}||_2 \leq \sqrt{m}\varepsilon$ then $||\mathbf{S}||_2 \leq \sqrt{m}\varepsilon$ and $\sqrt{1 - m\varepsilon^2} \leq ||\mathbf{C}||_2 = ||\mathbf{H}_{11}||_2 \leq 1$ owing to the hypothesis $m\varepsilon \ll 1$.

Therefore, we have

$$\begin{aligned}(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{v} &= \mathbf{U}(\mathbf{I}-\mathbf{H}_{\bullet 2}\mathbf{H}_{\bullet 2}^T 1)(\mathbf{I}-\mathbf{H}_{\bullet 1}\mathbf{H}_{\bullet 1}^T)\hat{\mathbf{v}} \\ &= \mathbf{U}(\mathbf{I}-\mathbf{H}_{\bullet 2}\mathbf{H}_{\bullet 2}^T)\mathbf{W}(\mathbf{I}-\mathbf{\Sigma}\mathbf{\Sigma}^T)\begin{bmatrix}\mathbf{W}_1^T\hat{\mathbf{v}}_1\\ \mathbf{W}_2^T\hat{\mathbf{v}}_2\end{bmatrix}.\end{aligned}$$

From the CS-decomposition, it follows that

$$(\mathbf{I}-\mathbf{\Sigma}\mathbf{\Sigma}^T) = \begin{bmatrix}\mathbf{I}-\mathbf{C}^2 & 0 & -\mathbf{CS}\\ 0 & \mathbf{I}_{n-2m} & 0\\ -\mathbf{CS} & 0 & \mathbf{I}-\mathbf{S}^2\end{bmatrix} = \begin{bmatrix}\mathbf{S}^2 & 0 & -\mathbf{CS}\\ 0 & \mathbf{I}_{n-2m} & 0\\ -\mathbf{CS} & 0 & \mathbf{C}^2\end{bmatrix}.$$

Then, we have

$$(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{v} = \mathbf{U}(\mathbf{I}-\mathbf{H}_{\bullet 2}\mathbf{H}_{\bullet 2}^T)\mathbf{W}\begin{bmatrix}\mathbf{S}^2\mathbf{W}_1^T\hat{\mathbf{v}}_1 - \mathbf{CS}\mathbf{W}_2^T\hat{\mathbf{v}}_2 \\ \begin{bmatrix}0 & 0\\ -\mathbf{CS} & 0\end{bmatrix}\mathbf{W}_1^T\hat{\mathbf{v}}_1 + \begin{bmatrix}\mathbf{I}_{n-2m} & 0\\ 0 & \mathbf{C}^2\end{bmatrix}\mathbf{W}_2^T\hat{\mathbf{v}}_2\end{bmatrix}. \quad (4.1.4)$$

Finally, from (4.1.4) we obtain

$$\begin{aligned}||(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{v}||_2 &\leq \left|\left|\begin{array}{c} m\varepsilon^2\|\hat{\mathbf{v}}_1\|_2 + \sqrt{m}\varepsilon||\hat{\mathbf{v}}_2||_2 \\ \sqrt{m}\varepsilon||\hat{\mathbf{v}}_1||_2 + ||\hat{\mathbf{v}}_2||_2\end{array}\right|\right|_2 \quad (4.1.5)\\ &\leq \left|\left|\begin{array}{c} m\varepsilon^2||\hat{\mathbf{v}}_1||_2 + \sqrt{m}\varepsilon||\hat{\mathbf{v}}_2||_2 \\ \sqrt{m}\varepsilon||\hat{\mathbf{v}}_1||_2 + ||\hat{\mathbf{v}}_2||_2\end{array}\right|\right|_1. \quad (4.1.6)\end{aligned}$$

The thesis follows from the bounds $||\hat{\mathbf{v}}_2||_2 \leq \varepsilon$ and $\sqrt{1-\varepsilon^2} \leq ||\hat{\mathbf{v}}_1||_2 \leq 1$. □

Finally, from the previous theorem and (4.1.3), we have

$$\frac{||\hat{\mathbf{r}}||_{\widehat{\mathbf{A}}^{-1}}}{||\hat{\mathbf{r}}_0||_{\widehat{\mathbf{A}}^{-1}}} \leq 4\sqrt{m}\varepsilon||\widehat{\mathbf{A}}^{\frac{1}{2}}||_2||\widehat{\mathbf{A}}^{-\frac{1}{2}}||_2. \quad (4.1.7)$$

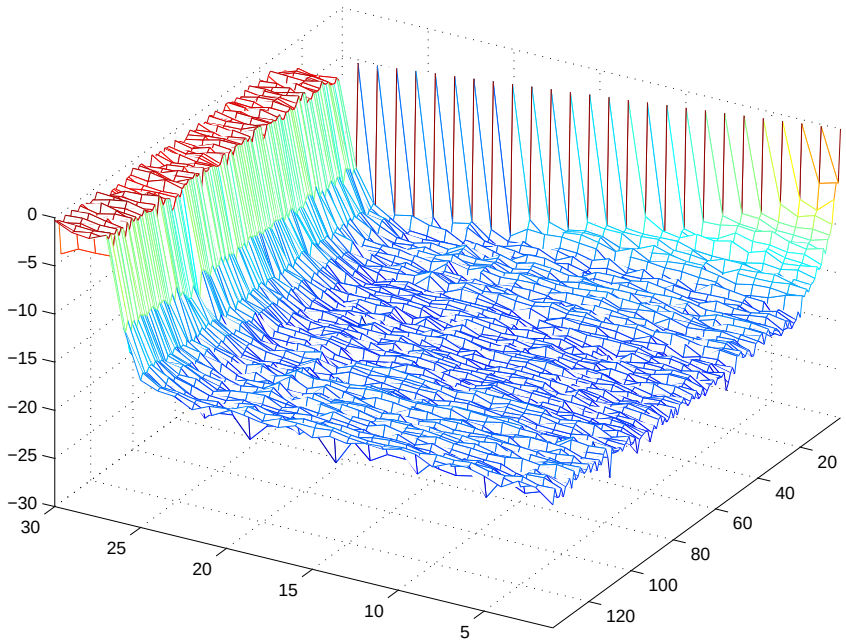
5 Computing eigenvalues and eigenvectors

Since we have computed, in the first phase of the algorithm, an orthonormal basis for a subspace very collinear to the invariant subspace associated with the smallest eigenvalues in $\widehat{\mathbf{A}}$, we may also use it to approximate these smallest eigenvalues and associated eigenvectors.

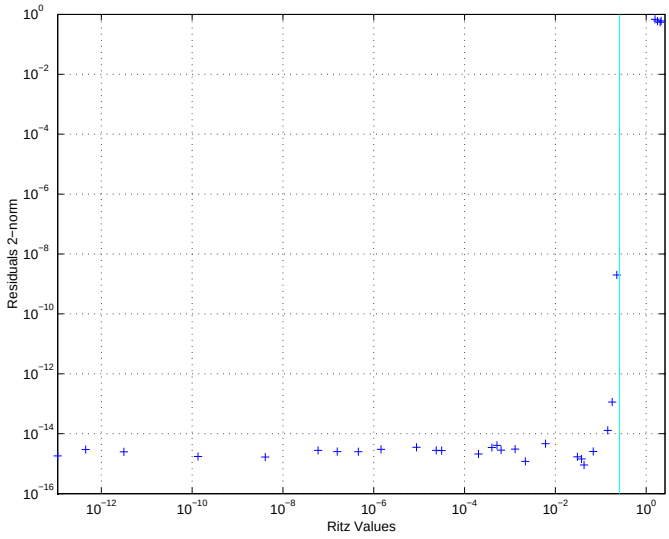
To this end, we simply construct the projected matrix

$$\widehat{\mathbf{A}}_1 = \mathbf{V}^T\widehat{\mathbf{A}}\mathbf{V}, \quad (5.1)$$

and compute it's eigenvalue decomposition to obtain finally (combining with $\mathbf{V}$) the Schur values and associated Schur vectors. As can be seen in Figure 5.1, the resulting approximation can be of relatively good quality, in particular if the filtering level ε has been set close to machine precision as in our test example.



The Z-axis indicates the logarithm of the absolute values of the eigencomponents in each Schur vector.



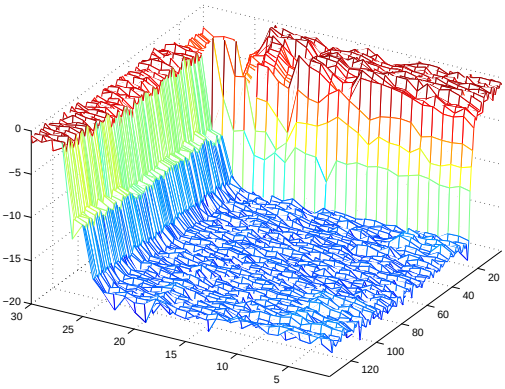
The residuals $\mathbf{r}_j$, $j = 1, \ldots, 30$, for each Schur vector $\mathbf{v}_j$ and associated Ritz value λ_j, correspond to $\mathbf{r}_j = \mathbf{A}\mathbf{v}_j - \lambda_j \mathbf{v}_j$.

Figure 5.1: Eigencomponents and convergence of the Schur vectors obtained out of the computed "filtered" Lanczos/Orthodir basis.

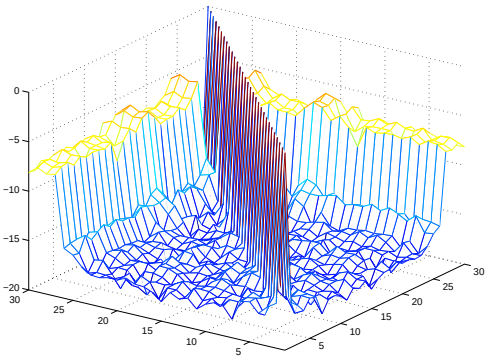
We also mention that, to obtain good residuals in the approximation of these eigenvalues and eigenvectors, it is necessary to re-orthogonalize the final block of computed vectors $\mathbf{V}^{(k+1)}$ against all the previous ones since the very last step in Algorithm 3.1, when convergence is reached, implies very unstable computations. At any rate, if one is only interested in the solution of linear systems, this re-orthogonalization is not at all necessary. The reasons for that is simply that the eigenvalue computation above relies on the similarity transformation (5.1), which in turn requires that the basis $\mathbf{V}$ be orthonormal. With respect to that, the solution of linear systems implies the projection step detailed in Algorithm 3.2, and exploits only the fact that the range of the vectors $\mathbf{V}$ includes a "*good*" approximation of the invariant subspace we actually try to factorize. In Figure 5.2, one can indeed see that the "*quality*" of the computed subspace with respect to filtering of the unwanted eigencomponents is as good as in Figure 2.4, but that some loss of orthogonality occurs for the last computed vectors in that basis. The above mentioned "*quality*" of the computed subspace will be discussed in more details in Section 6.

This technique may also be used as an alternative to Lanczos with shift and invert type of techniques for computing the eigenvalues (and corresponding eigenvectors) of a symmetric matrix that are close to a given value. Indeed, Algorithm 3.1 enables the computation of the smallest eigenvalues and associated eigenvectors of a symmetric and positive definite matrix. Therefore, combining it with some appropriate polynomial transformation of a given symmetric matrix (see Fischer, 1996) may enable us to compute those eigenvalues (and corresponding eigenvectors) close to some predetermined value, and this without the need for inversion.

In a more general way, we can extend the "*filtering*" ideas developed in the previous sections to unsymmetric matrices. In Arnoldi-Chebyshev type of algorithms, see Saad, 1992, Chapter VII for instance, the main idea is to determine some Chebyshev polynomials $\mathcal{P}_k$ of degree k that act on a certain part of the spectrum of the given matrix $\mathbf{A}$, usually an ellipse in the complex plane containing all unwanted eigenvalues, and that basically damp the modulus of eigenvalues inside



The Z-axis indicates the logarithm of the absolute values of the eigencomponents in each Lanczos/Orthodir vector.



The Z-axis indicates the logarithm of the absolute values of the scalar products between each pair of Lanczos/Orthodir vectors.

Figure 5.2: Eigendecomposition of the Krylov basis obtained after 5 block Lanczos/Orthodir steps without any re-orthogonalization. (5 steps of Block Lanczos/Orthodir with block size 6 and Chebyshev filtering).

the predetermined ellipse with respect to those outside. This can be related to the "*filtering*" of eigencomponents, when applying the Chebyshev polynomials $\mathcal{P}_k(\mathbf{A})$ in $\mathbf{A}$ to some set of vectors. The idea we suggest is then to develop some type of Arnoldi algorithm with Chebyshev filtering after each orthonormalization step, as in Algorithm 3.1. The major change between the two approaches is to exploit polynomials in $\mathbf{A}$ not to transform the eigenvalues of the iteration matrix, but as an intermediate filtering step within the process of building the Krylov basis that helps to enlarge the cosines of the principal angles between the computed basis and some specific invariant subspace of $\mathbf{A}$, making the two more collinear.

6 Influence of the different parameters

The algorithm we have developed in Section 3, including both the partial spectral factorization and solution phases, depends on three different parameters which correspond to the choice of the starting block size s, of the cut-off eigenvalue μ, and of the filtering level ε under which we try to maintain those eigencomponents relative to all eigenvalues bigger than μ in the computed Krylov vectors. In the rest of this section, we will try to analyse the influence of these three parameters on the behaviour of the complete algorithm and discuss in more detail the different issues that we have illustrated in the previous sections.

6.1 The choice of the starting block size s

The value of s only concerns the first phase in the algorithm (Algorithm 3.1), and not the second one (Algorithm 3.2). It influences mostly the number of Block-Lanczos steps, which will be roughly about k/s, k being equal to the number of eigenvalues outside the damping interval $[\mu, \lambda_{\max}(\widehat{\mathbf{A}})]$ and corresponding to the dimension k of the Krylov basis generated at the end of the partial spectral factorization phase.

We did not observe a big impact when varying s on the actual total number of matrix-vector products induced by the re-filtering steps in the Block-Lanczos/Orthodir iterations. Basically, what we observed is that, before reaching convergence, the sum over the iterations of the number of re-filtering steps times the block size s does not vary a lot with different values of s. It is only at the last step, when convergence is reached, that the block size s may have some impact, simply because we must filter down to the level ε again all the s vectors in the last block, and the larger is s, the larger is the amount of work in this last re-filtering step.

Increasing the value of s may also have an impact on the MFlops rate, because of the possibility of using Level 3 BLAS kernels (if $s \neq 1$) within the first phase. This can help to reduce by quite a reasonable factor the total execution time in the Partial Spectral Factorization phase (see Dongarra, Du Croz, Duff and Hammarling, 1990, Anderson, Bai, Bischof, Demmel, Dongarra, Du Croz, Greenbaum, Hammarling, McKenney, Ostrouchov and Sorensen, 1992).

6.2 The choice of the cut-off eigenvalue μ

The second parameter μ splits the spectrum of the matrix $\mathbf{QA}$ in two subsets and fixes the dimension k of the invariant subspace that will be computed in the partial spectral factorization phase. It also determines the convergence rate of the classical Chebyshev iterations that will be performed, both when filtering the Lanczos vectors and when computing the solution in

the second phase of the algorithm. The determination of a good value for μ needs a good approximation of $\lambda_{\max}$. The latter can be easily obtained by few steps of the power method.

In Table 6.1, we can observe these combined effects on our small test example, with three different values for the parameter μ. The rapid change in the Chebyshev rate of convergence with smaller values of μ induces much more Chebyshev filtering steps at each iteration, and this could only be counterbalanced by a very strong reduction in the total number of iterations in the first phase of the algorithm. In other words, it is worth reducing the value of μ only if there is a very strong clustering of eigenvalues in the spectrum of the iteration matrix $\widehat{\mathbf{A}}$ and if the change in μ helps to reduce by a large amount the dimension of the invariant subspace that will be approximated. This is of course not the case in our small case-study example.

In that respect, preconditioning is a key issue that may help to enforce a strong clustering in the spectrum of the iteration matrix. The spectrum of our test example is extracted from that of an iteration matrix obtained with the use of the Block-Cimmino method applied to an unsymmetric matrix from the *Harwell-Boeing* collection of sparse matrices (see Duff, Grimes and Lewis, 1989). This type of preconditioning technique (see Arioli, Duff, Noailles and Ruiz, 1992, Arioli et al., 1995, for instance) transforms in general the original matrix $\mathbf{A}$ into a symmetric and positive definite matrix whose eigenvalues are in a relatively small number of clusters but still ill-conditioned.

However, the method we propose in this article can address very badly conditioned problems and offers the possibility of using a wider range of preconditioning techniques, still with good clustering properties but without the constraint of reducing the condition number at the same time.

	Number of Chebyshev Iterations					
	$\mu = \lambda_{\max}/5$		$\mu = \lambda_{\max}/10$		$\mu = \lambda_{\max}/100$	
Block Krylov	Value of ε		Value of ε		Value of ε	
Iteration	10^{-14}	10^{-8}	10^{-14}	10^{-8}	10^{-14}	10^{-8}
Start	35 + 2	20 + 3	51 + 4	30 + 3	165 + 14	96 + 14
1	10	9	15	15	59	56
2	11	11	20	18	88	90
3	16	15	32	30	165	96
4	25	20	43	30	-	-
5	35	20	-	-	-	-

In the case $\mu = \lambda_{\max}/5$, there are 33 eigenvectors to capture.
In the case $\mu = \lambda_{\max}/10$, there are 26 eigenvectors to capture.
In the case $\mu = \lambda_{\max}/100$, there are 19 eigenvectors to capture.

Table 6.1: Comparison of the number of Chebyshev filtering steps for different values of the filtering level and different bounds for the damping interval. (Block Lanczos/Orthodir with block size 6).

6.3 The choice of the filtering level ε

The results shown in the previous sections have been obtained with a value of ε close to machine precision, just to illustrate that this is actually feasible. In Table 6.1, we compare the number of Chebyshev iterations at each filtering step with two values of ε. We observe that these mostly differ in the first and last filtering steps, and that during the intermediate stages when building the near-invariant subspace, the number of Chebyshev iterations do not vary much with the choice of ε. Indeed, the intermediate Chebyshev iterations simply aim to recover some potential increase in the level of filtering when we orthogonalize the current Lanczos/Orthodir vectors with respect to the previous ones and, as described in Section 4, this may not be influenced by the choice of the filtering level at least for values of ε less than the square root of machine precision.

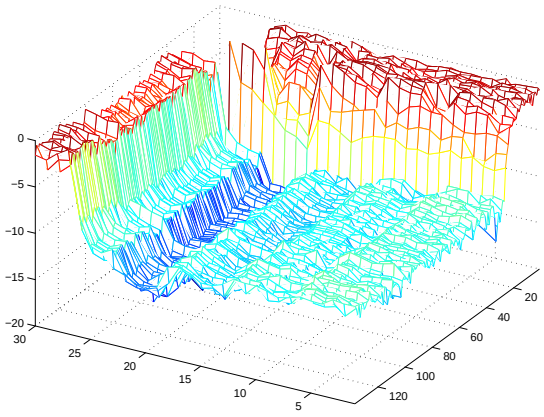
The first filtering step is obviously directly linked to the choice of the filtering level ε since its purpose is to filter some random starting set of vectors under that level. At the final stage also, when convergence or near invariance with respect to $\mathbf{U}_1$ is reached in the partial spectral factorization phase, some vectors in the last computed set have become strongly collinear to the unwanted invariant subspace generated by $\mathbf{U}_2$, and a full re-filtering similar to the starting one is required again. It is then easy to understand why smaller values for ε imply more Chebyshev iterations in these first and last filtering stages.

Additionally, the choice of the filtering level ε has a direct impact on the "*quality*" of the computed near-invariant subspace after completion of the first phase. This is illustrated by the eigencomponents with respect to $\mathbf{U}_2$ of the corresponding Schur vectors (see Figures 5.1 and 6.1), which are all close to the value of ε. This means, in particular, that for eigenvalue and/or eigenvector computation, small values of ε might be required since the perturbation in the approximate corresponding eigenvectors will not be smaller than the value of ε. The experiments tend to indicate that the residuals of these Schur vectors all have indeed a norm close to ε.

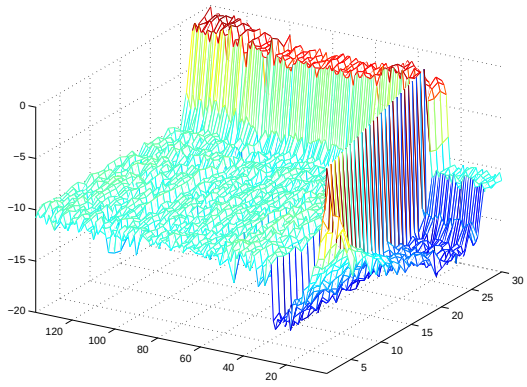
As discussed in Section 4.1, the filtering level with respect to $\mathbf{U}_2$ in the computed near-invariant subspace also influences the "*numerical quality*" of the computed solution in the second phase of the algorithm. Figure 6.2 shows the eigencomponents of the error and residual obtained after completion of the solution phase, with a filtering level with respect to $\mathbf{U}_2$ equal to 10^{-8} in both phases of the algorithm. The right-hand side vector $\hat{\mathbf{b}}$ corresponds to an exact solution with random entries. We can observe that the eigencomponents relative to $\mathbf{U}_2$ in the residual are all close to ε, and that those relative to $\mathbf{U}_1$ are even smaller and close to ε^2, with the eigencomponents in the error vector being simply divided by their corresponding eigenvalue. This observation is reflected by the error analysis in Section 4.1 and the results in equation (4.1.5) where we can see block-wise that the projected filtered vector $(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v}$ has eigencomponents of order ε^2 with respect to $\mathbf{U}_1$ and of order ε with respect to $\mathbf{U}_2$. Of course, this result holds for an orthogonal projection and not an oblique projection such as that actually performed in the solution phase.

The choice ε equal to square root of machine precision is adequate when an "*a priori*" information on the condition number $\kappa(\widehat{\mathbf{A}}) = \|\widehat{\mathbf{A}}\|\|\widehat{\mathbf{A}}^{-1}\|$ is not available. However, this choice can be quite conservative. Inequality (4.1.7) gives the possibility to choose the value of ε as a function of a threshold τ we want to impose on the scaled dual norm of the residual

$$\frac{||\hat{\mathbf{r}}||_{\widehat{\mathbf{A}}^{-1}}}{||\hat{\mathbf{r}}_0||_{\widehat{\mathbf{A}}^{-1}}} \leq \tau.$$



The Z-axis indicates the logarithm of the absolute values of the eigencomponents in each Lanczos/Orthodir vector.



The Z-axis indicates the logarithm of the absolute values of the eigencomponents in each Schur vector.

Figure 6.1: Eigendecomposition of the Krylov basis and corresponding Schur vectors obtained with a value of the filtering level ε of 10^{-8}. (5 steps of Block Lanczos/Orthodir with block size 6 and Chebyshev filtering).

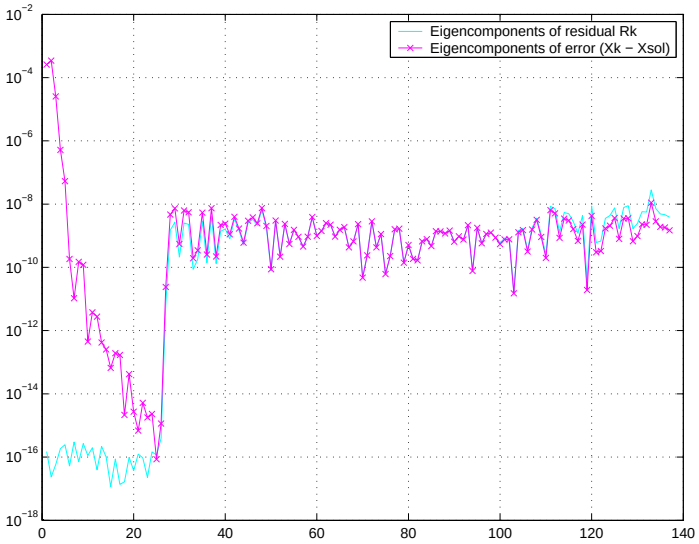


Figure 6.2: Computation of the solution of a linear system with a right-hand side vector $\hat{\mathbf{b}}$ corresponding to a given random exact solution vector $\mathbf{x}^*$. The filtering level ε has been fixed at 10^{-8} in both phases of the algorithm.

In this case, given an approximation of the square root of the condition number of the matrix $\widehat{\mathbf{A}}$, we can choose ε as

$$\varepsilon = \frac{\tau}{\sqrt{\kappa(\widehat{\mathbf{A}})}}.$$

In this way, one may even expect to obtain a solution as good as with a filtering level close to machine precision but without the expense of computing a basis with such a high numerical quality.

Furthermore, Algorithm 3.1 gives the possibility to compute "*a posteriori*" a good approximation of $\lambda_{\min}$ that, combined with the $\lambda_{\max}$ approximation computed for the determination of μ, can be used to improve the $\kappa(\widehat{\mathbf{A}})$ approximation. Therefore, we can validate our choice of the value of ε *a posteriori*. If the value is inadequate to guaranty an error less than τ in (4.1.7) ($\varepsilon > \tau/(\kappa(\widehat{\mathbf{A}})^{1/2})$, we can recover the wanted results by applying a final Chebyshev filtering process to $\mathbf{V}$ followed by a re-orthogonalization:

$$\begin{aligned}
&\mathbf{Z}^{(k+1)} = \text{Chebyshev_Filter}(\mathbf{V}, \tau\kappa(\widehat{\mathbf{A}})^{-1/2}/\varepsilon, [\mu, \lambda_{\max}], \widehat{\mathbf{A}}) \\
&[\mathbf{V}, \mathbf{\Sigma}, \mathbf{W}] = \text{SVD}(\mathbf{Z}, 0)
\end{aligned}$$

Then, we can set ε to the new value and apply Algorithm 3.2.

Following this discussion and that of the previous sections, we may also expect to be capable of improving the solution using iterative refinement in the usual way, provided that the filtering level ε is less than the square root of the condition number of the iteration matrix $\widehat{\mathbf{A}}$.

7 Numerical experiments

We generated our test problem using **pdetool**© under **Matlab**©. Let Ω be a simply connected bounded polygonal domain in $\mathbb{R}^2$, defined by a closed curve Γ. In the following, we will denote by $H^1(\Omega)$ the space of all distributions $u(x)$ defined in Ω that satisfy

$$||u||_{1,\Omega} = \left(\int_\Omega |\nabla u(x)|^2 dx + \int_\Omega |u(x)|^2 dx \right)^{1/2} < +\infty \,.$$

Finally, we will denote by $H_0^1(\Omega)$ the closure of the space of all infinitely differentiable functions with compact support in Ω in $H^1(\Omega)$, and by $H^{-1}(\Omega)$ the dual space of $H_0^1(\Omega)$. Let

$$a(u,v) = \int_\Omega \mathfrak{K}(x) \nabla u \cdot \nabla v dx, \quad \forall u, v \in H_0^1(\Omega) \tag{7.1}$$

be a continuous and coercive bilinear form: $\forall u, v \in H_0^1(\Omega)$, $\exists \gamma \in \mathbb{R}_+$ and $\exists \mathcal{M} \in \mathbb{R}_+$ such that

$$\gamma ||u||_{1,\Omega}^2 \leq \quad a(u,u) \tag{7.2}$$

$$a(u,v) \quad \leq \mathcal{M} ||||, ||||, \,, \tag{7.3}$$

and $L(v) = \int_\Omega f(x) v(x) dx$ be a continuous linear functional, $L(v) \in H^{-1}(\Omega)$. Using the hypotheses stated above the problem

$$\begin{cases} \text{Find } u \in H_0^1(\Omega) \text{ such that} \\ a(u,v) = L(v), \ \ \forall v \in H_0^1(\Omega), \end{cases} \tag{7.4}$$

has a unique solution. Our test problem is define on a L-shape domain Ω of $\mathbb{R}^2$ and we chose boundary condition zero.

In Fig. 7.1, we plot the geometry of the domain Ω. In problem (7.4), we choose the functional $L(v) = \int_\Omega 10v(x)dv,\ \forall v \in H_0^1(\Omega)$, and in the bilinear form (7.1), the function $\mathfrak{K}(x) \in L^\infty(\Omega)$ takes different values in each subdomain:

$$\mathfrak{K}(x) = \begin{cases} 1 & x \in \Omega \setminus \{\Omega_1 \cup \Omega_2 \cup \Omega_3\}, \\ 10^6 & x \in \Omega_1, \\ 10^4 & x \in \Omega_2. \end{cases}$$

Using **pdetool**©, we generated a mesh satisfying the usual regularity conditions of Ciarlet

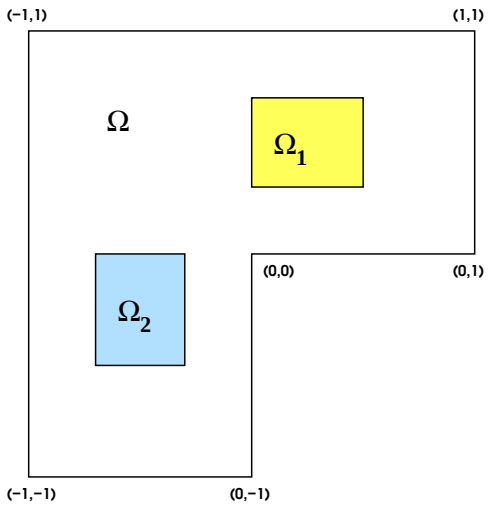


Figure 7.1: Geometry of the domain Ω.

(Ciarlet 1978, page 132) and we computed a finite-element approximation of problem (7.4) with the use of continuous piece-wise linear elements. The approximated problem is equivalent to the following system of linear equations:

$$\mathbf{A}\mathbf{u} = \mathbf{b}. \tag{7.5}$$

In our mesh, the largest triangle has an area of 3.123×10^{-4}, therefore, the resulting linear system (7.5) has 16256 triangles, 8289 nodes, and 7969 degrees of freedom.

Moreover, we used three kinds of preconditioners: the classical Jacobi diagonal matrix, $\mathbf{M} = diag(\mathbf{A})$, the incomplete Cholesky decomposition of $\mathbf{A}$ with zero fill-in, and the incomplete Cholesky decomposition of $\mathbf{A}$ with drop tolerance 10^{-2} (Greenbaum 1997, Meurant 1999). Using the incomplete Cholesky decompositions, we computed the upper triangular matrix $\mathbf{R}$ such that $\mathbf{M} = \mathbf{R}^T\mathbf{R}$. In Table 7.1, we report on the values of the condition numbers $\kappa(\mathbf{A})$ and $\kappa(\mathbf{M}^{-1}\mathbf{A})$ for both problems, and for the Jacobi and incomplete Cholesky with zero fill-in. The condition numbers of the preconditioned matrices $\mathbf{M}^{-1}\mathbf{A}$ for the second problem are are still very high, and only the incomplete Cholesky preconditioner with drop tolerance 10^{-2} is an effective choice. For these class of elliptic problems when finite-element method is used, Arioli (2003) showed that the threshold τ in (4.1.7) can be set to $\mathcal{O}(h) = \mathcal{O}(\sqrt{6.246 \times 10^{-4}})$ the square root of twice

$\mathbf{M}$	$\kappa(\mathbf{M}^{-1}\mathbf{A})$	ε	$\lambda_{\min}$	$\lambda_{\max}$
$\mathbf{I}$	$2.6\ 10^{9}$	$4\ 10^{-7}$	$3.7\ 10^{-3}$	$9.6\ 10^{6}$
Jacobi	$6.8\ 10^{8}$	$1\ 10^{-6}$	$3.1\ 10^{-9}$	2.08
Inc. Cholesky(0)	$9.4\ 10^{7}$	$3\ 10^{-6}$	$1.7\ 10^{-8}$	1.6
Inc. Cholesky(10^{-2})	$6.2\ 10^{6}$	$1\ 10^{-5}$	$1.8\ 10^{-7}$	1.1

Table 7.1: Estimates for $\kappa(\mathbf{M}^{-1}\mathbf{A})$, $\lambda_{\min}$, $\lambda_{\max}$.

$\mu = \lambda_{\max}/\gamma$	Preconditioner $\mathbf{M}$			
γ	Identity	Jacobi scaling	Inc. Cholesky(0)	Inc. Cholesky(10^{-2})
10^{9}	3			
10^{8}	41			
10^{7}	>200			
10^{3}		3		
500		5		
200		18		
100		43	3	
50		89	11	
20		>200	32	
10			68	3
5			157	9
2			>200	40

Table 7.2: Number of eigenvalues in $[\lambda_{\min}, \mu]$.

the area of the largest triangle in the mesh. This choice will allow to achieve a final error in energy norm of the same order as the final finite-element approximation error. Own to (4.1.7), we fixed the value of ε as

$$\varepsilon = \frac{\tau}{\sqrt{\kappa(\mathbf{M}^{-1}\mathbf{A})}}.$$

In Table 7.1, we exhibit the chosen values of ε computed by the previous expression, and in Table 7.2 we exhibit the number of eigenvalues in $[\lambda_{\min}, \mu]$ for selected values of the parameter μ and for each preconditioner. From the data, we can see that the original matrix is very badly scaled, and the Jacobi preconditioner is able to cluster the eigenvalues even if the matrix has still few small eigenvalues.

In Table 7.3, we summarize the behaviour of Algorithm 3.1 plus Algorithm 3.2 in computing the solution. We can observe, for different values of the parameter μ, the total number of Chebyshev filtering iterations performed in the spectral factorization phase, as well as the actual size of the computed Lanczos basis. We also indicate the number of Chebyshev iterations that have been performed in the solution phase, as well as the final energy norm of the error corresponding to the computed solution. The number between parenthesis are those obtained for a value of the filtering level ε fixed to 10^{-8}, the other values being obtained with the level ε indicated in Table 7.1.

In the experiments, $\mathbf{u}^{(k)}$ is the computed value at iteration k of our algorithm and we use the energy norm for the vectors:

$$\|\mathbf{y}\|_{\mathbf{A}} = \sqrt{\mathbf{y}^T\mathbf{A}\mathbf{y}}.$$

Finally, we assume that the solution $\mathbf{u}$ computed by a direct solver applied to (7.5) is exact, and we assume that $\mathcal{E}$, the energy norm of the solution $\tilde{\mathbf{u}}$ on the finer mesh with ≈ 129409 degrees of freedom, is a good approximation of $\sqrt{a(u,u)}$, the energy norm of the solution $u(x)$

$\mu = \lambda_{\max}/\gamma$	Spectral Factorization		Solution phase	
γ	Tot. Chebyshev Iterations	Size $\mathbf{V}$	Chebyshev Iterations	Error Energy Norm
Jacobi				
1000	1030 (1004)	3	231	$7\ 10^{-3}$ ($2.6\ 10^{-5}$)
500	1101 (1114)	5	163	$6\ 10^{-4}$ ($7.0\ 10^{-6}$)
200	2234 (2615)	19	103	$2.5\ 10^{-4}$ ($4\ 10^{-6}$)
Inc. Cholesky(0)				
100	433 (248)	5 (3)	68	$7.4\ 10^{-3}$ ($1.8\ 10^{-5}$)
50	462 (503)	9	48	$3\ 10^{-3}$ ($1.3\ 10^{-5}$)
Inc. Cholesky(10^{-2})				
10	55 (70)	3	19	$8.2\ 10^{-3}$ ($3.3\ 10^{-6}$)

Table 7.3: Summary of the results of Algorithm 3.1 plus Algorithm 3.2

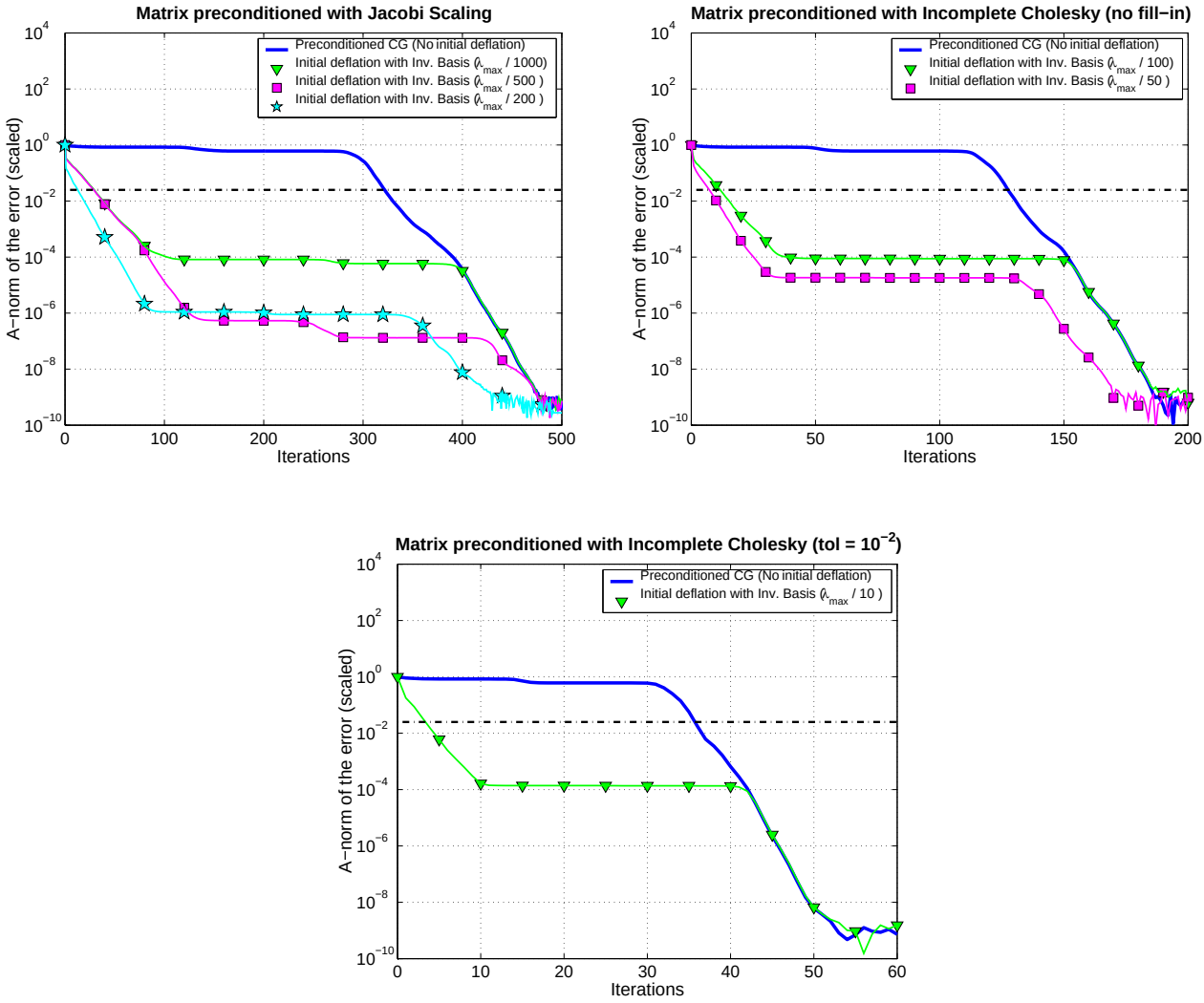


Figure 7.2: Convergence history of the preconditioned conjugate gradient method with a starting guess obtained with an oblique projection of the right-hand side vector onto the orthogonal complement of the computed near-invariant basis $\mathbf{V}$ (function of μ).

of the continuous problem (7.4). We then consider

$$\mathfrak{E}(\mathbf{u}) = \sqrt{1 - \frac{\mathbf{u}^T\mathbf{A}\mathbf{u}}{\mathcal{E}^2}},$$

which is equal in this case to $3.6\,10^{-2}$, as a good estimate of the finite-element error (Arioli 2003). We can observe in Table 7.3, that the level of the energy norm of the error for our computed solution is always below $\mathfrak{E}(\mathbf{u})$. Finally, we point out that $\mathfrak{E}(\mathbf{u}) = \mathcal{O}(h)$ which justifies our choice for the value of τ above.

It is also possible to make a different choice for the final phase. We have used, for instance, the conjugate gradient method in combination with the Chebyshev-filtering technique to solve the problem. To take advantage of the information computed in the first phase within the conjugate gradient method, it is sufficient to project the initial residual by the oblique projection (2.4) onto the invariant subspace represented by $\mathbf{V}$, viz.

$$\mathbf{u}^{(0)} = \mathbf{R}^{-1}\mathbf{V}\left(\mathbf{V}^T\mathbf{R}^{-T}\mathbf{A}\mathbf{R}^{-1}\mathbf{V}\right)^{-1}\mathbf{V}^T\mathbf{R}^{-T}\mathbf{b},$$

in order to get the eigencomponents in the solution corresponding to the smallest eigenvalues as described in Section 2. Then, we can apply straightforward the conjugate gradient method on the preconditioned system with the above starting guess $\mathbf{u}^{(0)}$. We point out that the matrix $\mathbf{V}^T\mathbf{R}^{-T}\mathbf{A}\mathbf{R}^{-1}\mathbf{V}$ can be computed during the spectral factorization in Algorithm 3.1.

In Figure 7.2, we exhibit the conjugate gradient method curves of the error measured in energy norm respectively for the Jacobi, incomplete Cholesky with no fill-in, and incomplete Cholesky with drop tolerance 10^{-2} preconditioners. We also plot the curve of the errors measured in the energy norm when the conjugate gradient method is used without the deflation of the initial residual. We can observe that the CG method benefits directly from this initial deflation since the number of iterations to reach the actual value of τ (indicated by the horizontal dotted line on the graphs) is much less. We can also see that the linear rates of convergence in the CG method vary with the value of μ, and this is simply due to the fact that the Conjugate Gradient method with initial deflation behaves as if the "*active*" set of eigenvalues in the right-hand side was reduced to only those belonging to the interval $[\mu, \lambda_{\max}]$.

The gains obtained for one solution are of about 300 iterations in the case of the Jacobi scaling, 110 iterations with incomplete Cholesky with no fill-in, and about 30 iterations in the case of the incomplete Cholesky preconditioner with drop tolerance 10^{-2}. We can then see that within 7 successive solutions in the worst case, we can counterbalance the extra cost in terms of matrix-vector operations paid when building the near-invariant basis in the partial spectral factorization phase. This amortization is even quicker when the initial preconditioner manages to cluster well the eigenvalues in $\widehat{\mathbf{A}}$, as for instance with incomplete Cholesky with drop tolerance $10^{[}-2]$ in the case of this PDE test problem. It is also clear that the conjugate gradient method achieves in much less iterations than the Chebyshev based solution phase the same level in the energy norm of the error. Obviously, the conjugate gradient method minimizes explicitly this energy norm in the course of the iterations, whereas the Chebyshev semi-iterative method minimizes the $\mathcal{L}_\infty$ norm of the residuals over the interval $[\mu, \lambda_{\max}]$. Nevertheless, the Chebyshev semi-iterative method involves only matrix-vector products but no dot products, and this can be of some advantage in particular in parallel distributed memory environments.

8 Conclusion

We have introduced a two-phase iterative-based approach for the consecutive solution of several ill-conditioned linear systems involving the same matrix and changing right-hand sides. The method is based on partial spectral factorization of the given matrix, and involves the use of Chebyshev polynomials as a filtering tool. The preliminary experiments indicate that the proposed technique has a good numerical potential.

Moreover, this algorithm is based only on kernels commonly used in iterative methods that enable us to keep the given ill-conditioned matrix in implicit form. It requires only matrix-vector products plus some vector updates, but no dot-products. This remark is also of some importance in the context of parallel computing, and in particular in distributed memory environments.

In the previous section, we partially addressed the influence of the different parameters, the block-size s, the cut-off value μ, and the filtering level ε. As already mentioned, we still need to investigate in deeper detail the influence of the filtering level ε on the numerical properties of the algorithm. Concerning the choice of the cut-off eigenvalue μ, which splits the spectrum of the matrix in two, the experiments tend to indicate that it is better not to choose μ too small because it will result in a strong increase in the overall number of Chebyshev iterations. However, it is of some importance that μ falls in between two clusters of eigenvalues and is not in the middle of a cluster. Indeed, in the latter case, it could be interesting to move the value of μ slightly to incorporate more eigenvectors into the unwanted invariant subspace $\mathbf{U}_2$ and to decrease substantially the dimension of the subspace that will be approximated. On the one hand, moving μ from the middle of a cluster to its border may not induce a big change in the Chebyshev convergence rate and, on the other hand, if the cluster contains a lot of eigenvalues, then moving μ may help to reduce the dimension of the invariant subspace $\mathbf{U}_1$ significantly and, consequently, the total number of operations to perform. At present, the cut-off eigenvalue μ is fixed *a priori* and does not change afterwards. We plan to investigate if some of the ideas from the adaptive Chebyshev iterative method (see Hageman and Young, 1981 and Golub and Kent, 1989) can be incorporated to adapt the value of μ slightly during the process.

We did not present many experiments with different test examples because our main target was to introduce the use of Chebyshev polynomials as a powerful filtering tool in the context of Krylov subspace techniques, and to illustrate and discuss the potential and limitations of this approach by means of some representative examples. Our plan is now to test this algorithm on more realistic examples, to study its behaviour, its cost and/or efficiency in the context of parallel computing, and to compare it with other well known techniques.

As mentioned in Section 5, we can extend the idea of additional Chebyshev filtering in the context of eigenvalue computation, and in particular for the design of an inverse-free Krylov technique for the computation of eigenvalues with smallest modulus. Similar ideas can also be contemplated for extracting a subset of the smallest singular values of a matrix, or for computing its null-space. Finally, following the discussion of Fischer (1996, Section 3.3), we can build Chebyshev polynomials that act on two disjoint intervals and focus in this way on a precise sub-part of the spectrum of the given matrix. A natural application of such extensions can be the case of symmetric indefinite matrices.

References

Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S. and Sorensen, D. (1992), *LAPACK : A portable linear algebra library for high-performance computers*, SIAM, Philadelphia.

Arioli, M. (2003), 'A stopping criterion for the conjugate gradient algorithm in a finite element method framework', *Numer. Math.* **DOI: 10.1007/s00211-003-0500-y**, Electronic Edition.

Arioli, M. and Ruiz, D. (1995), Block conjugate gradient with subspace iteration for solving linear systems, *in* S. D. Margenov and P. S. Vassilevski, eds, 'Iterative Methods in Linear Algebra, II. Volume 3 in the IMACS Series in Computational and Applied Mathematics. Proceedings of The Second IMACS International Symposium on Iterative Methods in Linear Algebra.'.

Arioli, M., Duff, I. S., Noailles, J. and Ruiz, D. (1992), 'A block projection method for sparse matrices', *SIAM J. Scient. Statist. Comput.* **13**, 47–70.

Arioli, M., Duff, I. S., Ruiz, D. and Sadkane, M. (1995), 'Block Lanczos techniques for accelerating the Block Cimmino method', *SIAM J. Scient. Statist. Comput.* **16**, 1478–1511.

Ciarlet, P. (1978), *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, The Netherlands.

Dongarra, J. J., Du Croz, J., Duff, I. S. and Hammarling, S. (1990), 'Algorithm 679. A set of Level 3 Basic Linear Algebra Subprograms.', *ACM Transactions on Mathematical Software* **16**, 1–17.

Duff, I. S., Grimes, R. G. and Lewis, J. G. (1989), 'Sparse matrix test problems', *ACM Transactions on Mathematical Software* **15**, 1–14.

Fischer, B. (1996), *Polynomial Based Iteration Methods for Symmetric Linear Systems*, Wiley-Teubner.

Golub, G. H. and Kent, M. D. (1989), 'Estimates of eigenvalues for iterative methods', *Math. Comp.* **53**(188), 619–626.

Golub, G. H. and Underwood, R. (1977), The block Lánczos method for computing eigenvalues, *in* J. R. Rice, ed., 'Mathematical Software Symposium, University of Wisconsin, Madison, 1977. Mathematical Software III: Proceedings of a Symposium Conducted by the Mathematics Research Center, the University of Wisconsin, Madison, March 28–30, 1977', number 39 *in* 'Publication of the Mathematics Research Center, the University of Wisconsin, Madison', Academic Press, New York, pp. 361–377.

Greenbaum, A. (1997), *Iterative Methods for Solving Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Hageman, L. A. and Young, D. M. (1981), *Applied Iterative Methods*, Academic Press, New York and London.

Meurant, G. (1999), *Computer Solution of Large Linear Systems*, Vol. 28 of *Studies in Mathematics and its Application*, North-Holland, Amsterdam, The Netherlands.

O'Leary, D. P. (1980), 'A generalized conjugate gradient algorithm for solving class of quadratic programming problems', *Linear Algebra and its Applications* **34**, 371–399.

Parlett, B. N. (1980), *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ.

Saad, Y. (1992), *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK.

Stewart, G. W. (1982), 'An algorithm for computing the CS decomposition of a partitioned orthonormal matrix', *Numer. Math.* **40**, 297–306.

Underwood, R. (1975), An iterative block lanczos method for the solution of large sparse symmetric eigenproblems, PhD Thesis STAN-CS-75-496, Computer Science Department, Stanford, California.

van der Sluis, A. and van der Vorst, H. A. (1986), 'The rate of convergence of conjugate gradients', *Numer. Math.* **48**, 543–560.

Wilkinson, J. H. (1965), *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, England.