Science & Technology
Facilities Council

Technical Report

Flexible deflation in Krylov methods with Chebyshev-based polynomial filters

Mario Arioli and Daniel Ruiz

July 10th, 2009

RAL-TR-2009-014

Enquires about copyright, reproduction and requests for additional copies of this report should be addressed to:

Library and Information Services
SFTC Rutherford Appleton Laboratory
Harwell Science and Innovation Campus
Didcot
OX11 0QX
UK
Tel: +44 (0)1235 445384
Fax: +44(0)1235 446403
Email: library@rl.ac.uk

The STFC ePublication archive (epubs), recording the scientific output of the Chilbolton, Daresbury, and Rutherford Appleton Laboratories is available online at:
http://epubs.cclrc.ac.uk/

ISSN 1358-6254

Neither the Council nor the Laboratory accept any responsibility for loss or
damage arising from the use of information contained in any of their reports
or in any communication about their tests or investigation

RAL-TR-2009-014

Flexible deflation in Krylov methods with Chebyshev-based polynomial filters

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 symmetrizing preconditioning technique for solving a general (possibly unsymmetric) ill-conditioned linear system, or in domain decomposition of a numerically difficult elliptic problem.
Combining Chebyshev iterations with the Lanczos algorithm, we propose a way to identify and extract precise information related to the ill-conditioned part of the given linear system. This approach is equivalent to a flexible deflation based on Chebyshev filters. The potential of this combination, which can be related to the factorization and direct solution of linear systems, is illustrated numerically and theoretically. In particular, we also present a general theory that relates the level of filtering to the accuracy of the computed solution.

Keywords: Chebyshev polynomial, Lanczos method, Filtering.

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 20th, 2009

Contents

1 Introduction

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

$$\mathbf{Au} = \mathbf{b}\ , \qquad \mathbf{A} \in \mathbb{R}^{n \times n} \tag{1.1}$$

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, and 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, and there are case where the iterative methods are the only feasible approach. 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 symmetrizing preconditioning technique (see [14]) for solving a general (possibly unsymmetric) ill-conditioned linear system, or in domain decomposition of some numerically difficult elliptic problem. As we will see in our numerical experiments, even if the preconditioner is robust with a preconditioned system having condition number independent of n, few eigenvalues are still very small. The conjugate gradient method (and Krylov methods in general) will then have a low rate of convergence.

The technique we shall develop in the following computes the basis of an approximate invariant subspace associated with these smallest eigenvalues, and this is used to construct a projector that can extract the eigencomponents of the solution relative to this invariant subspace. This technique is based on a combination of the Lanczos method and Chebyshev filters which purpose is to *deflate* in some way the invariant subspace linked with the *non-targetted* largest eigenvalues. We want to point out immediately that our method builds some partial spectral factorization of the given iteration matrix and should be considered, as with direct methods, when more than one solution with the same matrix is required. Indeed, if the solutions corresponding to several right-hand sides appearing in sequence are required, at the cost of computing and storing few dense vectors, these subsequent solutions can be obtained in a much cheaper way.

Arioli and Ruiz [4] have already proposed a two-phase iterative approach, which first performs a partial spectral decomposition of the given matrix, and which afterwards 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 [18], [16]). 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.

This work has been furthermore developped and analysed theoretically by Balsa et.al. (see [7], [5], and [6]). From an inner-outer convergence analysis, they propose a strategy to reduce the total amount of computational work by controlling the accuracy during the solution of linear systems at each inverse iteration. They also incorporate Chebyshev polynomials as a spectral filtering tool when building the starting vectors to improve the global convergence of the algorithm. They perform an analysis of costs and benefits, in terms of floating point operations, to validate how this strategy can speed up the solution of symmetric and positive definite linear systems. For such a particular use, the computed approximate eigenvectors need not be very accurate. The proposed convergence analysis suggests a stopping criterion that enables to control explicitely the invariance degree of this information, and they illustrate this experimentally.

Golub et.al. [12] also propose a similiar technique combining the conjugate gradient method with Chebyshev filtering polynomials as preconditionners, that target some specific convergence properties of the conjugate gradient method. In their approach, the Chebyshev preconditioner is applied only to a part of the spectrum of the coefficient matrix and puts a large number of eigenvalues near one but does not degrade the distribution of the smallest ones. This procedure enables them to construct a lower dimensional Krylov basis that is very rich with respect to the smallest eigenvalues and associated eigenvectors, which can also be stored and exploited in a straightforward way to solve sequences of systems with little extra work. They illustrate experimentally that the gains can be rather effective and the cost for precomputing the Krylov basis can rapidly be compensated when solving several linear systems with the same matrix but changing right-hand sides.

As opposed to classical polynomial preconditioning techniques, where the degree of Chebyshev polynomials is fixed, we propose to monitor at each Lanczos iteration the number of Chebyshev filtering steps in order to maintain under some predetermined level the filtered eigencomponents in the computed Lanczos vectors. With respect to the Chebyshev preconditionners proposed in [12], our method is more flexible in the sense that it adapts iteratively the degree of the Chebyshev polynomials and offers the possibility to reduce substantially the computational cost in the precomputation of the targetted low dimensional Krylov basis.

Additionally, because of the uniform convergence properties of the Chebyshev polynomials, our algorithm is designed to control explicitly the relative gap between eigencomponents in the computed Krylov vectors. This eigen-componentwise relative separation of information also enables us to control a priori, with some forward error analysis, the **A**-norm of the error in subsequent solutions of linear systems with the same matrix, and is a complement to more classical backward error analysis (see e.g. [2]).

The paper is organised as follows. Section 2 motivates the proposed method, and introduces the Chebyshev filters. The algorithm is presented and discussed in Section 3. The convergence properties and the error analysis of the algorithm are given in Section 4 and Section 5 respectively. Finally, we discuss some of the numerical issues in Section 6 using selected numerical tests. Some open questions and conclusions are discussed in Section 7.

2 Chebyshev filters

As already mentioned in the introduction, we want to compute an approximation of the invariant subspace associated with the smallest eigenvalues of the given ill-conditioned SPD matrix **A**.

To do so, we start with an initial randomly generated vector **z** and we use Chebyshev polynomials in **A** to "*damp*", in this starting vector **z**, the eigencomponents associated with all the eigenvalues in some predetermined range. In the following, we shall denote by $\lambda_{\min}$ the minimum eigenvalue of **A**, and by $\lambda_{\max}$ the maximum one. We can fix, for instance, a positive number μ, with $\lambda_{\min} < \mu < \lambda_{\max}$, and decide to compute a basis the invariant subspace of **A** associated with all the eigenvalues in the range $[\lambda_{\min}, \mu]$. The computation of $\lambda_{\max}$ 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 **A**. 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 $[\lambda_{\min}, \lambda_{\max}/10]$.

To "*filter out*" the unwanted eigenvalues in the range $[\mu, \lambda_{\max}]$, and focus only on the few remaining ones in $[\lambda_{\min}, \mu]$, we consider Chebyshev polynomials, which can be defined by the following 2-term recurrence formula (see [14, page 46]):

$$\left\{ \begin{array}{ll} T_0(\omega) = 1\,, \quad T_1(\omega) = \omega & \\ T_{k+1}(\omega) = 2\omega T_k(\omega) - T_{k-1}(\omega) & \quad k \geq 1. \end{array} \right. . \tag{2.2}$$

The optimal properties of Chebyshev polynomials given in Theorem 4.2.1 in [14, page 47] can be summarised as follows: if we consider $d > 1$ and

$$\mathcal{F}_k(\omega) = \frac{T_k(\omega)}{T_k(d)},$$

then $\mathcal{F}_k$ has minimum l_∞ norm on the interval $[-1, 1]$ over all polynomials Q_k of degree less than of equal to n and satisfying the condition $Q_k(d) = 1$, and we have

$$\max_{\omega \in [-1,1]} |\mathcal{F}_k(\omega)| = \frac{1}{T_k(d)}. \tag{2.3}$$

We will denote in the following the values $T_k(d)$ by σ_k. These values can easily computed from the recurrence (2.2). Now, consider the translation plus homothetic transformation:

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

with

$$d_\mu = \frac{\lambda_{\max} + \mu}{\lambda_{\max} - \mu} > 1 \quad \text{and} \quad \alpha = \frac{2}{\lambda_{\max} - \mu},$$

and $\lambda_{\min} < \mu < \lambda_{\max}$ as given above. This transformation maps $\lambda_{\max}$ to -1, μ to 1, and 0 to $\omega_\mu(0) = d_\mu > 1$. Then, introducing

$$\mathcal{F}_k(\lambda) = \frac{T_k(\omega_\mu(\lambda))}{T_k(d_\mu)}, \tag{2.4}$$

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

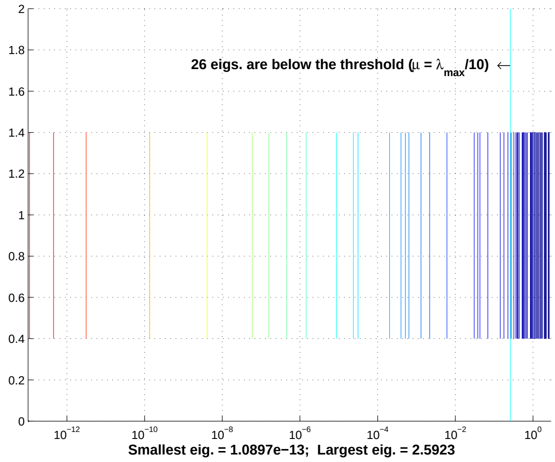


Figure 2.1: Eigenvalue distribution of a sample iteration test matrix (size of matrix is 137).

$Q_k(0) = 1$. Finally, from (2.2), we have

$$\begin{cases} \mathcal{F}_0(\lambda) = 1, \qquad \mathcal{F}_1(\lambda) = \dfrac{d_\mu - \alpha\lambda}{\sigma_1} = 1 - \dfrac{2\lambda}{\lambda_{\max} + \mu} \\ \sigma_{k+1}\mathcal{F}_{k+1}(\lambda) = 2\sigma_k\Big(d_\mu - \alpha\lambda\Big)\mathcal{F}_k(\lambda) - \sigma_{k-1}\mathcal{F}_{k-1}(\lambda), \end{cases} \tag{2.5}$$

which gives the recurrence formula to compute $\mathcal{F}_k(\lambda)$.

Let us denote by

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

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

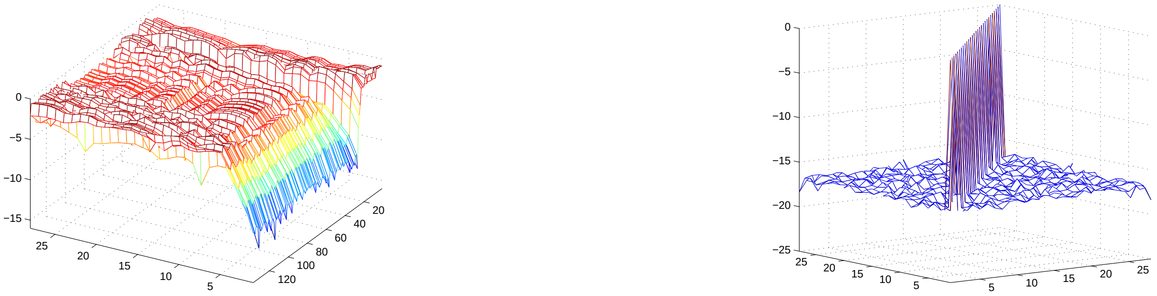
$$\mathbf{v} = \mathcal{F}_k(\mathbf{A})\mathbf{z} = \sum_{i=1}^{n} \mathbf{u}_i \left(\mathcal{F}_k(\lambda_i)\xi_i\right),$$

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 (since $\mathcal{F}_k(0) = 1$), and relatively much smaller for large enough degree n and all i such that $\lambda_i \in [\mu, \lambda_{\max}]$. This is how we can easily damp the eigencomponents of any vector in the range $[\mu, \lambda_{\max}]$ using classical Chebyshev iteration. 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 [14] and [11]), and can be very easily monitored using (2.3).

We can interpret the use of Chebyshev polynomials as a *filtering* tool that increases the degree of colinearity with some selected eigenvectors. Indeed, after "*filtering*" the initial starting vector, we obtain a vector with eigencomponents below a certain level ε for those eigenvalues in the range $[\mu, \lambda_{\max}]$, and relatively much bigger eigencomponents linked with the smallest eigenvalues in $\mathbf{A}$.

Of course, to be efficient, this approach relies on the fact that the spectrum of $\mathbf{A}$ is mostly concentrated in the range $[\mu, \lambda_{\max}]$, with reasonable value of μ (like $\lambda_{\max}/100$, or $\lambda_{\max}/10$, for instance) so that the number of Chebyshev iterations required to reach a given filtering level ε is not too large. In that respect, a preliminary preconditioner applied to $\mathbf{A}$, with classical preconditioning techniques, may be used so that the number of remaining eigenvalues inside the predetermined range $[\lambda_{\min}, \mu]$ (with some reasonable μ) is small compared to the size of the linear system.

One of the main drawbacks with the Lanczos algorithm is that it does not maintain this nice property of the filtered vectors, and gradually (and rather quickly, indeed) the Lanczos vectors may again have eigencomponents all about the same level. It must be mentioned beforehand that this problem is inherent to the Lanczos iteration, and is not at all due to any kind of loss of orthogonality in the Lanczos basis. In Figure 2.2, we show (in a logarithmic scale) the eigendecomposition of the Krylov basis obtained after 30 steps of the Lanczos algorithm starting with a vector initially "*filtered*" to a level close to machine precision with Chebyshev polynomial in the range $[\lambda_{\max}(\mathbf{A})/10, \lambda_{\max}]$.



Eigencomponents of the computed Lanczos basis, starting with a filtered initial vector (Chebyshev filter with $\mu = \lambda_{\max}/10$).

Logarithm of the absolute values of the scalar products between each pair of Lanczos vectors (e.g. no loss of orthogonality).

- *on the left subplot, the eigencomponents are indexed on the X-axis (from 1 to 136) in increasing order of their corresponding eigenvalue,*
- *the indexes (from 1 to 28) of the vectors in the current Lanczos basis are indicated on the Y-axis,*
- *the Z-axis indicates the logarithm of the absolute values of the eigencomponents in each of the 30 computed Lanczos vectors.*

Figure 2.2: Eigendecomposition of the Krylov basis obtained after 28 Lanczos steps (with full reorthogonalization). The iteration matrix in use is the one with spectrum given in Figure 2.1.

From the small test experiment of Figure 2.1, it can be seen that "*near*" invariance with respect to those eigenvectors linked to the 26 eigenvalues in the range $[\lambda_{\min}, \lambda_{\max}/10]$ can be completely lost, and the termination of the Lanczos algorithm may not occur after building a basis of dimension close to 26. 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 these results, 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 2.2 which shows the scalar products (in a logarithmic scale) between each pair of computed Lanczos vectors. The reasons for this will be analyzed and identified more theoretically in Section 4, and we first introduce in the next section the modifications we incorporate to the Lanczos algorithm to compensate this inherent behavior and maintain the filtered eigencomponents under some fixed level.

3 The algorithm

The algorithm we shall detail in this section is decomposed 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 $\mathbf{A}$ explicitly, since only matrix-vector multiplications are needed at every stage of the algorithm.

To maintain, during the spectral factorization, the value of the filtered eigencomponents in the computed Krylov basis under some level ε, we propose to perform, at each Lanczos iteration, a few extra Chebyshev iterations on the newly generated Lanczos vector $\mathbf{v}^{(k+1)}$. In this way, "*near*" invariance can be maintained. To describe the algorithm, let us first denote by

$$\mathbf{z} = \text{Chebyshev\_Filter}(\mathbf{y}, \varepsilon, [\mu, \lambda_{\max}], \mathbf{A})$$

the application of a Chebyshev polynomial in $\mathbf{A}$ to the vector $\mathbf{y}$, viz. $\mathbf{z} = \mathcal{F}_k(\mathbf{A})\mathbf{y}$, where $\mathcal{F}_k$ is defined as in (2.4) 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 cut-off eigenvalue μ, with $\lambda_{\min} < \mu < \lambda_{\max}$, and the filtering level $\varepsilon \ll 1$, we can then detail the various algorithmic steps of this partial spectral factorization phase in Algorithm 1.

It must be mentioned, beforehand, that the nice property of the Krylov spaces, which makes tridiagonal the projected matrix $\mathbf{V}^T\mathbf{A}\mathbf{V}$ in the Lanczos method, is lost after re-filtering the current $\mathbf{v}^{(k+1)}$. We cannot simplify the orthogonalization step when computing the next Krylov vectors and iterate with a simple 3 term recurrence formula in the usual way, as in the classical Lanczos algorithm, and it is therefore necessary to orthogonalize at each iteration the filtered vectors against all the previously constructed ones. However, if the number k of eigenvalues in the range $[\lambda_{\min}, \mu]$ is small, the algorithm is an inverse-free technique that will build directly a basis of dimension k, and with a precise control on the eigencomponents relative to eigenvalues outside that range. Figure 3.3 illustrates the benefit of these additional filtering steps performed at each Lanczos iteration on the same test case as the one of Figure 2.2.

Note also that the starting point in Algorithm 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 vector may have a much smaller norm depending on how many eigenvalues in $\mathbf{A}$ remain outside the damping interval $[\mu, \lambda_{\max}]$. Then, the orthonormalization step that occurs after filtering this vector may increase the actual level ε of the filtered eigencomponents. The purpose of the second filtering step is to ensure that the filtering level of these eigencomponents in the starting vector is actually very close to ε. In Section 4, we will discuss why this can be necessary and, in Section 5, we will justify the chosen stopping criterion threshold. We have also forced a double process of filtering in algorithm 3, where we repeated the steps (3.6) twice. Indeed, the application of Chebyshev polynomial filtering to $\mathbf{v}^{(k+1)}$ in (3.6) 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 μ. 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 (3.6) helps to guaranty that $\|\mathbf{\Psi}\| \leq \varepsilon\sqrt{m(n-m)}$ (where m is the rank of $\mathbf{\Psi}$), at least until convergence is not about to be reached. Moreover, we will show in Section 6 how this heuristic has a favourable cost/benefit.

We want to stress that our approach is not a polynomial preconditioning of the given linear system. We exploit Chebyshev polynomials as a spectral filtering tool to perform implicitly some sort of deflation with respect to the invariant subspace linked with the largest eigenvalues in the given coefficient matrix $\mathbf{A}$. Chebyshev filtering polynomials, in that respect, do not present the same nice properties as projectors used commonly in such deflation, which have explicit eigenvalues equal to 0 and 1. Still, they can partly mimic these properties, and offer an alternative to achieve the same behavior in the Lanczos iteration as with deflation techniques, but without having to effectively compute the basis vectors associated with this deflation. In Section 4, we analyse in more details the theoretical aspects in this algorithm and discuss how to monitor appropriately its convergence.

Once the *near*-invariant subspace linked to the smallest eigenvalues is obtained, we can use it for the computation of further solutions. Several techniques have been proposed in the literature

Algorithm 1 (Chebyshev-based Partial Spectral Factorization (Chebyshev-PSF))

```
$\mathbf{p}^{(0)} = \text{random}(n,1);$   and $\mathbf{q}^{(0)} = \dfrac{\mathbf{p}^{(0)}}{||\mathbf{p}^{(0)}||};$
$\mathbf{z}^{(0)} = \text{Chebyshev\_Filter}(\mathbf{q}^{(0)}, \varepsilon, [\mu, \lambda_{\max}], \mathbf{A});$
$\hat{\mathbf{q}}^{(0)} = \dfrac{\mathbf{z}^{(0)}}{||\mathbf{z}^{(0)}||};$   and set $\delta_0 = ||\mathbf{z}^{(0)}||$ and $k = 0;$
$\hat{\mathbf{z}}^{(0)} = \text{Chebyshev\_Filter}(\hat{\mathbf{q}}^{(0)}, \delta_0, [\mu, \lambda_{\max}], \mathbf{A});$
$\mathbf{V} = \mathbf{v}^{(0)} = \dfrac{\hat{\mathbf{z}}^{(0)}}{||\hat{\mathbf{z}}^{(0)}||};$   and set $\delta_2 = 1;$
$\mathbf{w} = \mathbf{A}\mathbf{v}^{(0)};$   and set $\mathbf{G} = \mathbf{g}^{(1)} = \mathbf{V}^T\mathbf{w};$
while $\delta_2 > \varepsilon\sqrt{k(n-k)}$ do :
    $\mathbf{p}^{(k+1)} = \mathbf{w} - \mathbf{V}\mathbf{g}^{(k+1)};$ $\mathbf{v}^{(k+1)} = \dfrac{\mathbf{p}^{(k+1)}}{||\mathbf{p}^{(k+1)}||};$
    set $\delta_1 = ||\mathbf{p}^{(k+1)}||/\lambda_{\max}$ and $\delta = \max(\varepsilon, \delta_1 \times \delta_2);$
    for $i = 1, 2$     % the second round is optional %
        $\mathbf{z}^{(k+1)} = \text{Chebyshev\_Filter}(\mathbf{v}^{(k+1)}, \delta, [\mu, \lambda_{\max}], \mathbf{A});$
        $\mathbf{y}^{(k+1)} = \mathbf{z}^{(k+1)} - \mathbf{V}\mathbf{V}^T\mathbf{z}^{(k+1)};$
        $\mathbf{v}^{(k+1)} = \dfrac{\mathbf{y}^{(k+1)}}{||\mathbf{y}^{(k+1)}||};$                                   (3.6)
        set  $\delta_2 = ||\mathbf{y}^{(k+1)}||$  and  $\delta = \delta_2;$
        if  $(\delta_2 \geq 0.1)$,  break;   % re-filtering is not useful %    (3.7)
    end
    $\mathbf{V} = [\mathbf{V}; \mathbf{v}^{(k+1)}];$     and $\mathbf{w} = \mathbf{A}\mathbf{v}^{(k+1)};$
    $\mathbf{g}^{(k+1)} = \mathbf{V}^T\mathbf{w};$     and $\mathbf{G} = \begin{bmatrix} & & & g_1^{(k+1)} \\ & \mathbf{G} & & \vdots \\ & & & g_k^{(k+1)} \\ g_1^{(k+1)} & \cdots & g_k^{(k+1)} & g_{k+1}^{(k+1)} \end{bmatrix}.$
set $k = k + 1;$
end.
```

that consist of either updating the preconditioner or enforcing conjugate gradient to work in the orthogonal complement of an invariant subspace associated with the smallest eigenvalues. In [10] and [9], Giraud et.al. exploit the Chebyshev-based Partial Spectral Factorization algorithm

(Algorithm 1 above) to generate an orthonormal basis of a near-invariant subspace of $\mathbf{A}$ associated with the smallest eigenvalues. They vary, in particular, the level of filtering from 10^{-16} to 10^{-2} and use the resulting information in combination with different solution techniques to compare their behaviour and numerical efficiency with respect to the quality of the near-invariant basis.

Amongst these various solution techniques, one of our favorite remains what they called INIT-CHEBYSHEV and which resumes in using the classical Chebyshev algorithm with eigenvalue bounds given by μ and $\lambda_{\max}$, as explained in [14, Chapter 4], to compute a first part of the solution, and in performing an oblique projection of the residual onto the pre-computed *near* invariant subspace in order to get the eigencomponents in the solution corresponding to the smallest eigenvalues. The nice feature of this solution technique is that it exploits classical Chebyshev polynomials which do not require scalar products, as with conventional Krylov solvers, and presents therefore a good potential for parallel computation in distributed memory environments. Additionally, the uniform convergence properties of Chebyshev polynomials also enables a forward error analysis in this solution technique, and this will be the topic of Section 5.

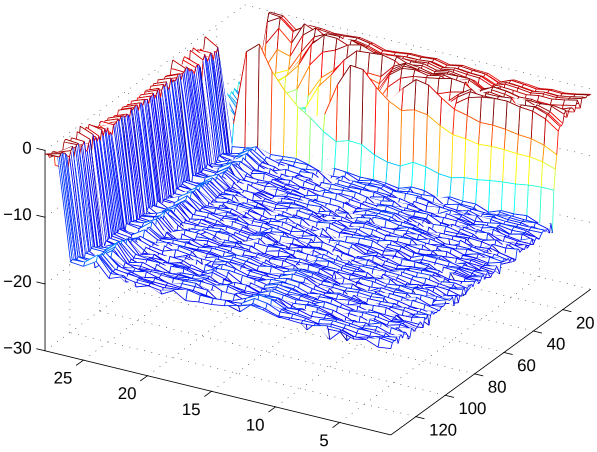
To describe this solution phase, we also denote by

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

the application of Chebyshev polynomial in $\mathbf{A}$ the purpose of which is to reduce by a factor of δ the eigencomponents in $\mathbf{r}_0$ associated with the eigenvalues in the range $[\mu, \lambda_{\max}]$, providing thus the resulting residual $\mathbf{r}_1 = \mathcal{F}_k(\mathbf{A})\mathbf{r}_0$ and the corresponding update $\mathbf{x}_1$ such that $\mathbf{b} - \mathbf{A}\mathbf{x}_1 = \mathbf{r}_1$. The polynomials $1 - \mathcal{F}_k$ are homogeneous and

$$1 - \mathcal{F}_k(\lambda) = \lambda \mathcal{G}_{k-1}(\lambda).$$

From this, and with the recurrence formula (2.5), it is easy to derive an equivalent 3-term recurrence formula that can be used to construct $\mathcal{G}_{k-1}(\mathbf{A})\mathbf{r}_0$ and which corresponds to $\mathbf{x}_1$ above. For technical details, we refer to [9, § 3.1] where this recurrence formula is given explicitely. Using this shortcut, the solution phase can be summarized as



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

Figure 3.3: Eigendecomposisiton of the Krylov basis obtained after 28 iterations of the Chebyshev-PSF algorithm starting with the same initial filtered set of vectors as that corresponding to the experiments in Figure 2.2.

Algorithm 2 (Solution phase (Init-Chebyshev))

```
For any right-hand side vector $\mathbf{b}$, set $\mathbf{x}_0$ and $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$,
and perform the two following consecutive steps:
    $[\mathbf{r}_1, \mathbf{x}_1] = \text{Chebyshev\_Solve}(\mathbf{r}_0, \mathbf{x}_0, \varepsilon, [\mu, \lambda_{\max}], \mathbf{A})$
    $\mathbf{r} = \mathbf{r}_1 - \mathbf{A}\mathbf{V}\mathbf{G}^{-1}\mathbf{V}^T\mathbf{r}_1$,   and  $\mathbf{x} = \mathbf{x}_1 + \mathbf{V}\mathbf{G}^{-1}\mathbf{V}^T\mathbf{r}_1$
```

In Figure 3.4, we illustrate the behaviour of the solution phase error, on our small example, for the values $\varepsilon = 10^{-8}$ and $\varepsilon = 10^{-16}$. In Section 5, we will prove a general result that establishes an upper bound of the error in the energy norm (i.e. the norm of the non preconditioned $\mathbf{A}$). The numerical results of Figure 3.4, are consistent with the theoretical upper bound and show its tightness. It is also possible to iterate on that solution phase, and improve the solution with iterative refinement in the usual way. Finally, we can observe that, in Algorithm 1, we may store only the upper triangular part of matrix $\mathbf{G}$ and this will be enough for performing the Cholesky decomposition of $\mathbf{G}$ in Algorithm 2.

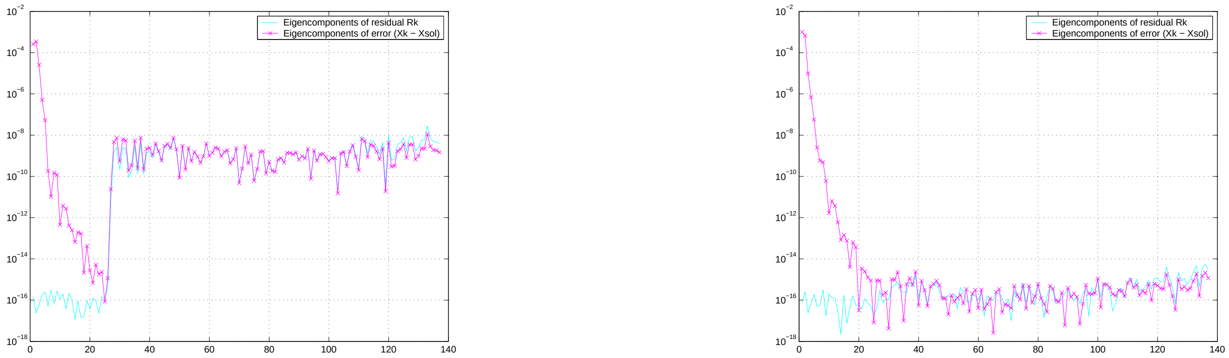


Figure 3.4: Computation of the solution of a linear system with a right-hand side vector $\mathbf{b}$ corresponding to a given random exact solution vector $\mathbf{x}^*$. The filtering level ε has been fixed at 10^{-8} for the left plot and 10^{-16} for the right plot in both phases of the algorithm.

4 An analysis of Lanczos and filtering interaction

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

If $\mathbf{V}$ exactly spans the invariant subspace associated with all eigenvalues in the range $[\lambda_{\min}, \mu]$, the Chebyshev iteration and the oblique projection steps in this solution phase can be performed, though sequentially, in any order *a priori*. However, since *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 corresponding to the eigenvalues greater or equal to μ in the inner product $\mathbf{V}^T\mathbf{r}_1$ above. This can be of particular interest when the filtering level ε is not close to machine precision.

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

$$\begin{array}{rl} & \mathbf{V} = \mathbf{U}_1\mathbf{\Gamma} + \mathbf{U}_2\mathbf{\Psi}, \text{ and } \|\mathbf{\Psi}\| \leq \varepsilon, \\ \text{with} & \mathbf{A} = \mathbf{U}_1\mathbf{\Lambda}_1\mathbf{U}_1^T + \mathbf{U}_2\mathbf{\Lambda}_2\mathbf{U}_2^T, \end{array} \tag{4.8}$$

$\mathbf{\Lambda}_1$ being the diagonal matrix made with all eigenvalues of $\mathbf{A}$ 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 Lanczos process is to build

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

and to orthonormalize the set of s vectors $\mathbf{p}^{(k+1)}$ to get the next entry $\mathbf{v}^{(k+1)}$ in the Krylov orthonormal basis $\mathbf{V}$. Using the decomposition (4.8), we can then write

$$\begin{array}{rcl} \mathbf{p}^{(k+1)} & = & \mathbf{U}_1\left(\mathbf{\Lambda}_1\boldsymbol{\gamma}^{(k)} - \mathbf{\Gamma}\mathbf{\Gamma}^T\mathbf{\Lambda}_1\boldsymbol{\gamma}^{(k)} - \mathbf{\Gamma}\mathbf{\Psi}^T\mathbf{\Lambda}_2\boldsymbol{\psi}^{(k)}\right) \\ & + & \mathbf{U}_2\left(\mathbf{\Lambda}_2\boldsymbol{\psi}^{(k)} - \mathbf{\Psi}\mathbf{\Psi}^T\mathbf{\Lambda}_2\boldsymbol{\psi}^{(k)} - \mathbf{\Psi}\mathbf{\Gamma}^T\mathbf{\Lambda}_1\boldsymbol{\gamma}^{(k)}\right), \end{array} \tag{4.9}$$

where $\boldsymbol{\gamma}^{(k)}$ and $\boldsymbol{\psi}^{(k)}$ are the columns in $\mathbf{\Gamma}$ and $\mathbf{\Psi}$ corresponding to the vectors $\mathbf{v}^{(k)}$ in $\mathbf{V}$.

Let us assume also, for simplicity, that $\varepsilon \leq 10^{-8}$ so that we can neglect to a first approximation the terms in $\mathcal{O}(\|\mathbf{\Psi}\|^2)$ The factor of $\mathbf{U}_1$ in (4.9) reduces to

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

and corresponds simply to the Krylov update one would obtain when working directly with a *projected* matrix whose spectrum corresponds to $\mathbf{\Lambda}_1$. The factor of $\mathbf{U}_2$ in (4.9) reduces to

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

where the term $\mathbf{\Lambda}_2\boldsymbol{\psi}^{(k)}$ is dominant 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 (4.9) 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 $(\mu/\lambda_{\max})$ relatively to the part in $\mathbf{U}_2$. These two combined causes are responsible for the "*staircase*" effect that one can observe in Figure 2.2 in the eigencomponents relative to $\mathbf{U}_2$ of the Krylov iterates.

For these reasons, we have introduced the extra Chebyshev filtering steps (3.6) after the actual Lanczos step in algorithm 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 $\|\boldsymbol{\psi}^{(0)}\|$ in the initial set of filtered vectors $\mathbf{v}^{(0)}$.

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 Lanczos basis and resulting in an update $\mathbf{y}^{(k+1)}$ in (3.6) 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).

5 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 $\mathbf{x}^T\mathbf{A}\mathbf{x}$. Therefore, the residual is a linear form belonging to the dual space, and the natural norm of the dual space is $(\mathbf{r}^T\mathbf{A}^{-1}\mathbf{r})^{1/2}$. We observe that:

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

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

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

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

$$\mathbf{v} = \mathcal{P}_k(\mathbf{A})\mathbf{e}_0/||\mathcal{P}_k(\mathbf{A})\mathbf{e}_0||_2. \tag{5.10}$$

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

$$||\mathbf{r}||_{\mathbf{A}^{-1}} \leq ||\wp\mathbf{A}^{\frac{1}{2}}\mathbf{v}||_2||\mathbf{r}_0||_{\mathbf{A}^{-1}}||\mathbf{A}^{-\frac{1}{2}}||_2. \tag{5.11}$$

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:

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

Finally, from (5.11) we have:

$$\frac{||\mathbf{r}||_{\mathbf{A}^{-1}}}{||\mathbf{r}_0||_{\mathbf{A}^{-1}}} \leq ||\mathbf{A}^{\frac{1}{2}}||_2||\mathbf{A}^{-\frac{1}{2}}||_2||(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v}||_2. \tag{5.12}$$

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 1 *Let $\mathbf{U}_1 \in \mathbb{R}^{n\times m}$ be the matrix of the eigenvectors of $\mathbf{A}$ corresponding to $\mathbf{\Lambda}_1$ and $\mathbf{U}_2 \in \mathbb{R}^{n\times(n-m)}$ be the matrix of the remaining eigenvectors of $\mathbf{A}$. Let $\mathbf{V} \in \mathbb{R}^{n\times\ell}$ be the full basis generated by Algorithm 1 using a filtering level ε and $\mathbf{v}$ be the vector defined by (5.10). If $c(n,m)\varepsilon \ll 1$ ($c(n,m) = \sqrt{(n-m)m}$) and $\ell \geq m$ then*

$$||(\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v}||_2 \leq 2\varepsilon c(n,m)(1 + \varepsilon c(n,m)).$$

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 $\mathbf{A}$ 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

$$\mathbf{H} \; = \left[\begin{array}{cc} \mathbf{H}_{\bullet 1} & \mathbf{H}_{\bullet 2} \end{array} \right] = \begin{array}{c} \overbrace{\phantom{\mathbf{H}_{11}}}^{m} \; \overbrace{\phantom{\mathbf{H}_{12}}}^{\ell - m} \\ \left[\begin{array}{cc} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{array} \right] \end{array} \quad \begin{array}{l} \}m \\ \}n-m \end{array},$$

$$\hat{\mathbf{v}} = \left[\begin{array}{c} \hat{\mathbf{v}}_1 \\ \hat{\mathbf{v}}_2 \end{array} \right] \begin{array}{l} \}m \\ \}n-m \end{array}.$$

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

$$\begin{array}{rcl} ||\mathbf{H}_{21}||_2 & \leq & ||\mathbf{H}_{21}||_F \leq c(n,m)\varepsilon \\ ||\hat{\mathbf{v}}_2||_2 & \leq & \sqrt{n-m}\varepsilon. \end{array}$$

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

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

Under the hypothesis $n-\ell \geq \ell \geq m$, the CS-decomposition of $\mathbf{H}_{\bullet 1}$ takes the form [17]:

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

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

$$\mathbf{W} = \left[\begin{array}{cc} \mathbf{W}_1 & 0_{m\times(n-m)} \\ 0^T_{m\times(n-m)} & \mathbf{W}_2 \end{array} \right], \quad \mathbf{\Sigma} = \left[\begin{array}{c} \mathbf{C} \\ 0_{n-2m,m} \\ \mathbf{S} \end{array} \right],$$

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 $||\mathbf{H}_{21}||_2 \leq c(n,m)\varepsilon$ then $||\mathbf{S}||_2 \leq c(n,m)\varepsilon$ and $\sqrt{1 - c^2(n,m)\varepsilon^2} \leq ||\mathbf{C}||_2 = ||\mathbf{H}_{11}||_2 \leq 1$ owing to the hypothesis $c(n,m)\varepsilon \ll 1$.

Therefore, we have

$$\begin{array}{rcl} (\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{v} & = & \mathbf{U}(\mathbf{I} - \mathbf{H}_{\bullet 2}\mathbf{H}_{\bullet 2}^T 1)(\mathbf{I} - \mathbf{H}_{\bullet 1}\mathbf{H}_{\bullet 1}^T)\hat{\mathbf{v}} \\ & = & \mathbf{U}(\mathbf{I} - \mathbf{H}_{\bullet 2}\mathbf{H}_{\bullet 2}^T)\mathbf{W}(\mathbf{I} - \mathbf{\Sigma}\mathbf{\Sigma}^T) \left[\begin{array}{c} \mathbf{W}_1^T\hat{\mathbf{v}}_1 \\ \mathbf{W}_2^T\hat{\mathbf{v}}_2 \end{array} \right]. \end{array}$$

From the CS-decomposition, it follows that

$$(\mathbf{I} - \mathbf{\Sigma}\mathbf{\Sigma}^T) \; = \left[\begin{array}{ccc} \mathbf{I} - \mathbf{C}^2 & 0 & -\mathbf{C}\mathbf{S} \\ 0 & \mathbf{I}_{n-2m} & 0 \\ -\mathbf{C}\mathbf{S} & 0 & \mathbf{I} - \mathbf{S}^2 \end{array} \right] = \left[\begin{array}{ccc} \mathbf{S}^2 & 0 & -\mathbf{C}\mathbf{S} \\ 0 & \mathbf{I}_{n-2m} & 0 \\ -\mathbf{C}\mathbf{S} & 0 & \mathbf{C}^2 \end{array} \right].$$

Then, we have

$$(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{v} = \\ \mathbf{U}(\mathbf{I}-\mathbf{H}_{\bullet 2}\mathbf{H}_{\bullet 2}^T)\mathbf{W}\left[\begin{array}{c} \mathbf{S}^2\mathbf{W}_1^T\hat{\mathbf{v}}_1 - \mathbf{C}\mathbf{S}\mathbf{W}_2^T\hat{\mathbf{v}}_2 \\ \left[\begin{array}{cc} 0 & 0 \\ -\mathbf{C}\mathbf{S} & 0 \end{array}\right]\mathbf{W}_1^T\hat{\mathbf{v}}_1 + \left[\begin{array}{cc} \mathbf{I}_{n-2m} & 0 \\ 0 & \mathbf{C}^2 \end{array}\right]\mathbf{W}_2^T\hat{\mathbf{v}}_2 \end{array}\right]. \tag{5.13}$$

Finally, from (5.13) we obtain

$$\begin{aligned} ||(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{v}||_2 &\leq \left|\left|\begin{array}{c} c^2(n,m)\varepsilon^2\|\hat{\mathbf{v}}_1||_2 + c(n,m)\varepsilon||\hat{\mathbf{v}}_2||_2 \\ c(n,m)\varepsilon||\hat{\mathbf{v}}_1||_2 + ||\hat{\mathbf{v}}_2||_2 \end{array}\right|\right|_2 \\ &\leq \left|\left|\begin{array}{c} c^2(n,m)\varepsilon^2||\hat{\mathbf{v}}_1||_2 + c(n,m)\varepsilon||\hat{\mathbf{v}}_2||_2 \\ c(n,m)\varepsilon||\hat{\mathbf{v}}_1||_2 + ||\hat{\mathbf{v}}_2||_2 \end{array}\right|\right|_1. \end{aligned} \tag{5.14}$$

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 (5.12), we have

$$\frac{||\mathbf{r}||_{\mathbf{A}^{-1}}}{||\mathbf{r}_0||_{\mathbf{A}^{-1}}} \leq 4c(n,m)\varepsilon||\mathbf{A}^{\frac{1}{2}}||_2||\mathbf{A}^{-\frac{1}{2}}||_2. \tag{5.15}$$

The choice ε equal to machine precision is adequate when an "*a priori*" information on the condition number $\kappa(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$ is not available. However, this choice can be quite conservative. Inequality (5.15) 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

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

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

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

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.

Remark 1 *Another way, directly connected to Theorem 1, of testing convergence or* near *invariance with respect to $\mathbf{U}_1$ in Algorithm 1 is:*

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.

6 Practical and numerical issues

6.1 A small example

The small example used in order to illustrate some numerical aspects of the algorithm has been generated by extracting a 137 × 137 submatrix from a more complex problem. The example is not completely artificial and it can be downloaded at the home page of the authors[1]. In Figure 2.1, we present the spectrum of this matrix and we poit out that only 26 eigenvalues are within the interval $[\lambda_{min}, \lambda_{max}/10] = [1.0897\ 10^{-13}, 2.5923/10]$. In Table 6.1, we illustrate the behaviour of Algorithm 1 on our small test example for both $\varepsilon = 1.0\ 10^{-8}$ and $\varepsilon = 2.2\ 10^{-16}$ and $\mu = \lambda_{max}/10$.

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 (see Table 6.1) until the very last steps when the dimension of the invariant subspace associated with all eigenvalues in the range $[\lambda_{\min}, \mu]$ is about to be reached.

	Filtering level : ε = 2.2e-16				Filtering level : ε = 1.0e-8			
	1^{st} filter step		2^{nd} filter step		1^{st} filter step		2^{nd} filter step	
Basis Vector	Cheb. iter.	Value of δ_2	Cheb. iter.	Value of δ_2	Cheb. iter.	Value of δ_2	Cheb. iter.	Value of δ_2
$\mathbf{v}^{(0)}$	60	–	2	–	31	–	3	–
$\mathbf{v}^{(1)}$	14	0.89			12	0.64		
$\mathbf{v}^{(2)}$	11	0.64			10	0.67		
$\mathbf{v}^{(3)}$	10	0.80			10	0.58		
$\mathbf{v}^{(4)}$	10	0.56			10	0.79		
$\mathbf{v}^{(5)}$	10	0.61			10	0.67		
$\mathbf{v}^{(6)}$	10	0.85			12	0.88		
$\mathbf{v}^{(7)}$	11	0.60			10	0.66		
$\mathbf{v}^{(8)}$	14	0.99			13	0.19		
$\mathbf{v}^{(9)}$	15	0.95			10	0.25		
$\mathbf{v}^{(10)}$	13	0.19			14	0.98		
$\mathbf{v}^{(11)}$	10	0.26			15	0.98		
$\mathbf{v}^{(12)}$	15	0.99			16	0.45		
$\mathbf{v}^{(13)}$	17	0.99			12	5.6e-2	6	0.72
$\mathbf{v}^{(14)}$	20	0.76			9	0.12		
$\mathbf{v}^{(15)}$	15	1.8e-2	7	0.26	17	1.00		
$\mathbf{v}^{(16)}$	9	0.37			24	1.00		
$\mathbf{v}^{(17)}$	20	1.00			21	1.00		
$\mathbf{v}^{(18)}$	27	1.00			27	1.00		
$\mathbf{v}^{(19)}$	29	1.00			27	0.29		
$\mathbf{v}^{(20)}$	28	1.00			22	1.2e-2	9	1.00
$\mathbf{v}^{(21)}$	33	0.92			17	1.2e-3	12	5.7e-2
$\mathbf{v}^{(22)}$	27	6.0e-4	13	1.00	13	2.8e-2	7	8.6e-2
$\mathbf{v}^{(23)}$	19	5.3e-4	13	0.58	31	1.00		
$\mathbf{v}^{(24)}$	20	1.00			31	1.00		
$\mathbf{v}^{(25)}$ $(*)$	49	1.00			31	7.4e-4	13	1.00
$\mathbf{v}^{(26)}$ $(\dagger)$	58	3.2e-16	58	3.7e-16	31	8.2e-9	31	8.2e-9
$\mathbf{v}^{(27)}$ $(\dagger)$	60	4.9e-17	60	9.7e-17	31	8.0e-9	31	8.1e-9

Table 6.1: History of convergence for the small sample of size 137, $\mu = \lambda_{max}/10$

[1] http://www.numerical.rl.ac.uk/people/marioli/ArioliRuiz/small137.mat

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 Krylov 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.

Remark 2 *If we sum the filtering steps for $\hat{\mathbf{v}}_1$ to $\hat{\mathbf{v}}_2$ during the anaysis phase for both $\varepsilon = 10^{-16}$ and $\varepsilon = 10^{-8}$ in Table 6.1, we have respectively $489 = 456 + 33$ and $471 = 424 + 47$ before convergence is achieved. Therefore, the global cost of matrix by vector product increases as function of ε only during the initialization phase and the solution phase.*

As discussed in Section 5, 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 3.4 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 $\mathbf{b}$ 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 5 and the results in equation (5.14) 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.

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 $\mathbf{A}$.

6.2 A PDE example

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

$$a(u,v) = \int_\Omega \mathfrak{K}(x) \nabla u \cdot \nabla v dx, \quad \forall u, v \in H_0^1(\Omega) \tag{6.16}$$

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

$$\gamma ||u||^2_{1,\Omega} \leq \; a(u,u) \tag{6.17}$$

$$a(u,v) \; \leq \mathcal{M}||u||_{1,\Omega}||v||_{1,\Omega} \; , \tag{6.18}$$

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

$$\begin{cases} \text{Find } u \in H_0^1(\Omega) \text{ such that} \\ a(u,v) = L(v), \;\; \forall v \in H_0^1(\Omega), \end{cases} \tag{6.19}$$

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. 6.5, we plot the geometry of the domain Ω. In problem (6.19), we choose the functional $L(v) = \int_\Omega 10v(x)dv$, $\forall v \in H_0^1(\Omega)$, and in the bilinear form (6.16), 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 [8, page 132] and we computed a finite-element approximation of problem (6.19) with the use of continuous piece-wise linear elements. The approximated problem is equivalent to the following system of linear equations:

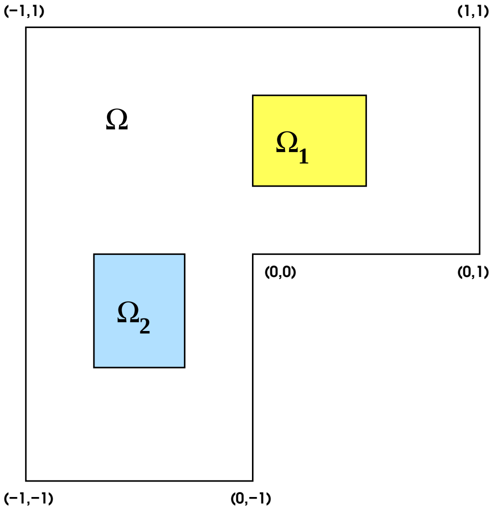


Figure 6.5: Geometry of the domain Ω.

$$\mathbf{Au} = \mathbf{b}. \tag{6.20}$$

In our mesh, the largest triangle has an area of 3.123×10^{-4}, therefore, the resulting linear system (6.20) 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} [13, 15]. Using the incomplete

$\mathbf{M}$	$\kappa(\mathbf{M}^{-1}\mathbf{A})$	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 6.2: 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 6.3: Number of eigenvalues in $[\lambda_{\min},\ \mu]$.

Cholesky decompositions, we computed the lower triangular matrix $\mathbf{L}$ such that $\mathbf{M} = \mathbf{L}\mathbf{L}^T$. In Table 6.2, we report on the values of the condition numbers $\kappa(\mathbf{A})$ and $\kappa(\mathbf{M}^{-1}\mathbf{A})$. The condition numbers of the preconditioned matrices $\mathbf{M}^{-1}\mathbf{A}$ 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 [1] showed that the threshold τ in (5.15) can be set to $\mathcal{O}(h) = \mathcal{O}(\sqrt{6.246 \times 10^{-4}})$ the square root of twice 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 (5.15), we fixed the value of ε as

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

In Table 6.2, we exhibit the chosen values of ε computed by the previous expression, and in Table 6.3 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 6.4, we summarize the behaviour of Algorithm 1 plus Algorithm 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

$\mu = \lambda_{\max}/\gamma$	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 6.4: Summary of the results of Algorithm 1 plus Algorithm 2

the filtering level ε fixed to 10^{-8}, the other values being obtained with the level ε indicated in Table 6.2.

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 (6.20) 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 $u(x)$ of the continuous problem (6.19). 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 [1]. We can observe in Table 6.4, 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.

6.3 Solution phase and multiple right-hand sides

It is also possible to make a different choice for the final solution phase. We can use, 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 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 $\mathbf{u}^{(0)}$. In [10], an estensive experimentation of this approach is presented. 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 1.

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 algorithm without any deflation can be computationally prohibitive.

From the costs of the spectral factorization illustrated in Table 6.4, 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 $\mathbf{A}$, as for instance with incomplete Cholesky with drop tolerance 10^{-2} in the case of this PDE test problem. However, it is also clear from the results presented in [10] 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 L_∞ 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.

7 Conclusion

We have introduced a two-phase iterative-based approach for the solution of ill-conditioned linear systems. The method is based on a deflation process that identifies the ill-conditioned part of the 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.

We plan to investigate if some of the ideas from the adaptive Chebyshev iterative method (see [11, 14]) can be incorporated to adapt the value of μ slightly during the process. At present, the cut-off eigenvalue μ is fixed *a priori* and does not change afterwards. However, the experiments 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. Moreover, it is important 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 and, consequently, the total number of operations to perform.

References

[1] M. Arioli, *A stopping criterion for the conjugate gradient algorithm in a finite element method framework*, Numer. Math., DOI: 10.1007/s00211-003-0500-y (2003), pp. 1–24.

[2] M. Arioli, I. S. Duff, and D. Ruiz, *Stopping criteria for iterative solvers*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 138–144.

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

[4] M. Arioli and D. Ruiz, *Block conjugate gradient with subspace iteration for solving linear systems*, in 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., S. D. Margenov and P. S. Vassilevski, eds., 1995.

[5] C. Balsa, M. Daydé, R. Guivarch, J. Palma, and D. Ruiz, *Monitoring the Block Conjugate Gradient Convergence within the Inexact Inverse Subspace Iteration*, in 6th International Conference on Parallel Processing and Applied Mathematics, Poznan, Poland, 11/09/05-14/09/05, Lecture Notes in Computer Sciences, septembre 2005.

[6] C. Balsa, M. Daydé, J. Palma, and D. Ruiz, *Improving the Numerical Simulation of an Airflow Problem with the BlockCGSI Algorithm*, in International Conference on Vector and Parallel Processing (VECPAR), Rio de Janeiro, Brésil, 10/07/2006-13/07/2006, no. 4395 in LNCS, http://www.springerlink.com/, 2006, Springer-Verlag, pp. 281–291.

[7] C. Balsa, J. Palma, and D. Ruiz, *Partial Spectral Information from Linear Systems to Speed-up Numerical Simulations in Computational Fluid Dynamics* , in High Performance Computing for Computational Science – VECPAR'2004. Sixth International Conference , Valencia, Espanha, 28/06/04-30/06/04, Springer-Verlag., J. Dongarra, V. Hernandez, and J. Palma, eds., Berlin, juin 2004, Lecture Notes in Computer Science. Selected Papers and Invited Talks from VECPAR'2004. Springer-Verlag, pp. 703–719.

[8] P. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, The Netherlands, 1978.

[9] L. Giraud, D. Ruiz, and A. Touhami, *Krylov and polynomial iterative solvers combined with partial spectral factorization for spd linear systems*, in VECPAR, M. J. Daydé, J. Dongarra, V. Hernández, and J. M. L. M. Palma, eds., vol. 3402 of Lecture Notes in Computer Science, Springer, 2004, pp. 637–656.

[10] L. Giraud, D. Ruiz, and A. Touhami, *A comparative study of iterative solvers exploiting spectral information for SPD systems*, SIAM Journal on Scientific Computing, 27 (2006), pp. 1760–1786.

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

[12] G. H. Golub, D. Ruiz, and A. Touhami, *A Hybrid Approach Combining Chebyshev Filter and Conjugate Gradient for Solving Linear Systems with Multiple Right-Hand Sides*, SIAM Journal on Matrix Analysis and Applications, 29 (2007), pp. 774–795.

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

[14] L. A. Hageman and D. M. Young, *Applied Iterative Methods*, Academic Press, New York and London, 1981.

[15] G. Meurant, *Computer Solution of Large Linear Systems*, vol. 28 of Studies in Mathematics and its Application, Elsevier/North-Holland, Amsterdam, The Netherlands, 1999.

[16] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ., 1980.

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

[18] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, England, 1965.