UNIVERSITÀ DEGLI STUDI DI BARI

Dottorato di Ricerca in Matematica

XIX Ciclo – A.A. 2007/2008

Settore Scientifico-Disciplinare:

MAT/08 – Analisi Numerica

Tesi di Dottorato

Acceleration techniques for approximating the matrix exponential

Candidato:

Marina POPOLIZIO

Supervisore della tesi:

Prof.ssa Valeria SIMONCINI

Coordinatore del Dottorato di Ricerca:

Prof. Luciano LOPEZ

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Professor Valeria Simoncini from the University of Bologna; her guidance has been of great value in this study.

I am deeply grateful to Professor Luciano Lopez from the University of Bari for his important support throughout this work and his paternal advices.

My sincere thanks to Professor Tiziano Politi, Cinzia Elia and Alessandro Pugliese.

I owe my loving thanks to my husband Domenico, my parents and my sister.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The problem of approximating the vector

$$y = \exp(A)v \tag{1.1.1}$$

for a square matrix A and a given vector v is an important topic in numerical analysis and it has been largely investigated, starting from the 1960's.
The interest in this problem is mainly motivated by the occurrence of the vector in (1.1.1) in several applications. This aspect justifies the rich literature available for this topic and the variety of contributions given by researchers from physics or engineering communities, as well as numerical analysts.
One of the most important applications in which (1.1.1) appears is the solution of ordinary differential equations or of time dependent partial differential equations; to give an example we consider the following linear parabolic Partial Differential Equation (PDE)

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} = Lu(x,t) & x \in \Omega \\ u(x,0) = u_0, & x \in \Omega \\ u(x,t) = \sigma(x), & x \in \partial\Omega,\ t > 0 \end{cases} \tag{1.1.2}$$

with L second order partial differential operator of the elliptic type and Ω open, bounded, connected set.
By discretizing (1.1.2) with respect to space variables the problem is reduced to the following ordinary differential equation

$$\frac{dw(t)}{dt} = Aw(t), \quad w(0) = w_0$$

whose solution is readily obtained as

$$w(t) = \exp(tA)w_0$$

which has exactly the form described in (1.1.1).
There are several other cases in which the computation of (1.1.1) is required and we list some of them: the vector (1.1.1) is fundamental to describe dynamical systems, see e.g. [3], or for applications in nuclear magnetic resonance spectroscopy [50], in control theory [24], in Markov chain analysis [61], in chemical physics [51].

A first fundamental remark is that computing $\exp(A)$ and computing $\exp(A)v$ are two completely different tasks. When only the vector $\exp(A)v$ is needed it is useless and impractical computing $\exp(A)$ and consequently applying it to the vector v; several techniques are indeed available to compute the vector in (1.1.1) in a much cheaper way. As stressed in [29], this principle is the same that the one used when solving a linear system of the form $Ax = b$, for which it would be wasteful to compute A^{-1} and then to multiply it by b. In this thesis we only address the problem of computing vectors of the form (1.1.1).

A milestone in the literature on the matrix exponential operator was the paper by Moler and van Loan [45] published in 1979; it described 19 ways for computing $\exp(A)$ classified into five categories: series methods, Ordinary Differential Equations methods, polynomial methods, matrix decomposition methods and splitting methods. The authors defined them “dubious” referring to their numerical quality; however this adjective referred also to the impossibility to detect a single method which works well in all applications. Twenty five years later an updated version of the paper was published [46] with an additional twentieth one, the Krylov subspace method; in the description of this method, tailored to the computation of (1.1.1), the authors stressed the advantage of directly computing this vector without passing from the evaluation of $\exp(A)$ which in general is full, even if A is sparse.
From the publication of [45] important results have been reached in the context of the matrix exponential operator, both from a theoretical and from an algorithmic point of view. We will describe the theoretical results in the following, while we cite the Sidje's paper [60] as one of the most complete work for the implementation aspects; there the software package EXPINT was presented, with the aim of helping all researchers who need to evaluate the matrix exponential.

As mentioned before, the matrix A is often the result of a discretization or of the modelling of real problems and for this reason its dimension is very large; moreover a remarkable sparsity is often present and in several cases A is symmetric negative semidefinite, as we will assume throughout the thesis. These features of A make mandatory the use of specific numerical methods to evaluate (1.1.1) which keep computational costs and memory requirements under control. In these situations two commonly strategies are applied:

1) find a matrix H such that $\exp(H)v$ is simpler to compute than (1.1.1) and approximates it;

2) approximate the exponential with a suitable function, say g, such that $g(A)v$ is simpler to compute than (1.1.1).

A rich literature is available for both approaches; for the first class in this thesis we restrict our analysis to Krylov subspace methods, that we will describe in details in Section 2.1. Their basic idea is to project the matrix A and the vector v onto a space K_m; once the projected and restricted matrix H_m is computed, then its exponential is (easily) evaluated and, by projecting back, a suitable approximation to $\exp(A)v$ derives, as summarized in (2.1.5). The papers [29] by Gallopoulos and Saad and [58] by Saad, both published in 1992, were among the first important contributions to the theoretical analysis of Krylov subspace methods; indeed, till their publication, Krylov subspace methods were largely used by chemical physicists, starting from Nauts and Wyatt who introduced them in 1983 [51], the only motivation being their satisfactory performance. After [29] and [58] several authors devoted their attention to Krylov subspace methods and interesting results describing their behavior were presented by Druskin and Knizhnerman in [19], [20] and [21], and by Hochbruck and Lubich in [36] and [37].

Variants to these methods have been proposed in the recent years, due to the increased size of the matrices A stemming from the applications; Moret and Novati [49], for example, combined the use of the *Restricted Denominator* rational forms proposed by Nørsett [52] with Krylov subspace methods; interestingly van den Eshof and Hochbruck [68], independently of [49], resorted exactly to the same technique by trying to compute $\exp(A)v$ by means of an auxiliary matrix, as described in (2.4.3). This approach, known as *shift and invert*, will be a key element of our analysis and we will describe it in details in Section 2.4.

Another strategy to make Krylov subspace methods effective for large matrices was presented by Eiermann and Ernst in [22], with the name of *Restarted*

Krylov and we will describe it in Section 2.3; also Hochbruck and Hochstenbach proposed in [35] a new approach whose key ingredient are the Krylov subspace methods.
Among the other techniques belonging to the first class mentioned above, we cite the approach proposed by Castillo and Saad in [15], dating back to 1998. The strategy consists in approximating A by means of a *preconditioner* M such that $\exp(M)v$ is simpler to compute and two strategies were proposed to recover $\exp(A)v$ from the computed vector $\exp(M)v$, one involving Krylov subspace approximations to integrals, the other based on a Generalized Runge Kutta method.
Among the methods of the first class not involving Krylov subspace approximations we cite the approach proposed by Lu in [43]. The idea is to reduce A to a tridiagonal matrix T by an orthogonal similarity transformation Q, to evaluate $\exp(T)$ by means of Chebyshev approximation and then to recover $\exp(A)$ by suitable multiplications with the matrix Q; however, due to computational costs which are $O(10/3n^3)$, where n is the dimension of A, this approach may be applied only when n is modest.

For the methods in 2) we will describe in details the *rational* approximations to exp, specifically Padé and Chebyshev, for which the books by Baker and Graves-Morris [6] and by Petrushev and Popov [54] gave a complete description; these approximations were also cited in the survey by Moler and van Loan [46] and Higham addressed in [34] the problem of efficiently implementing the Padé approximation. In the context of Chebyshev approximation the results by Carpenter, Ruttan and Varga in [14] and by Cody, Meinardus and Varga [16] gave important contributions to the detection of the best rational approximation to exp. Recently Trefethen and coauthors [67] studied connections among quadrature fomulas and rational approximations to exp, while Lopez and Simoncini [42] analyzed in depth the connections among Krylov subspace methods and rational approximations to the exponential. The combined use of rational approximations and Krylov subspace methods allowed Gallopoulos and coauthors [7] to evaluate an approximation to $\exp(A)v$ in parallel architecture. Recently Frommer and Simoncini addressed in [27] implementation problems related to the use of rational approximations for approximating matrix functions.
Druskin and Knizhnerman [19] and Tal-Ezer [66] proposed a polynomial approximation to the exponential based on its Chebyshev expansion with turned out to be useful to derive error estimates for the Krylov subspace approximation to $\exp(A)v$; years later Bergamaschi and Vianello in [11] offered nu-

merical experiments for this approach, with comparisons with other available algorithms.
Several interpolation procedures have been proposed to approximate the exponential and the choice of the nodes played an important role in this context: in [47], for example, Moret and Novati considered a polynomial approximation based on the zeros of Faber polynomials.

The main result of this thesis is that when g is a rational function and H is the projection of A onto a Krylov space, then the two classes of methods described in $1), 2)$ before are strictly related and behave similarly. This unifying framework allowed us to better analyze the methods of the two classes, especially the recently acceleration technique *shift and invert* proposed in [49] and [68]. We found that this technique may be viewed as a special form of preconditioning, thus allowing the use of theoretical results on this issue.
On the other hand, in the context of rational approximation to the exponential, we derived an acceleration technique based on a *real valued* method proposed in [5] for solving complex systems with only real arithmetic.

The rigorous definition of $\exp(A)$ is necessary before addressing all the practical aspects: several choices are possible, since different definitions of a generic matrix function have been presented during the years, the first work being probably the paper by Rinehart [56].
One of the simplest representation for $\exp(A)v$ is based on the symmetric Schur decomposition of A, which results in the expression

$$\exp(A)v = Q\exp(\Lambda)Q^T v$$

where Q is a real orthogonal matrix and Λ is the diagonal matrix with the eigenvalues $\lambda_1, \dots, \lambda_n$ of A as diagonal entries; the matrix $\exp(\Lambda)$ is readily computed since it only requires the evaluation of the diagonal entries $\exp(\lambda_j)$, $j = 1, \dots, n$.
Another straightforward formulation for $\exp(A)v$ is based on the Taylor series expansion of $\exp(z)$, for a generic scalar z, which in matrix formulation reads

$$\exp(A)v = v + Av + \frac{1}{2}A^2v + \frac{1}{3!}A^3v + \dots;$$

the convergence of this series is guaranteed for any square matrix. Interestingly, this power expansion justifies, in some sense, the use of the Krylov space $K_m(A, v)$, defined in (2.1.1), which is spanned exactly by $v, Av, A^2v, \dots, A^{m-1}v$.

When dealing with Krylov subspace methods we will use another definition for $\exp(A)$, which we recall here:

Definition 1.1.1. Assume $A \in \mathbb{C}^{n\times n}$ has the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ with multiplicities $m_1, m_2, \ldots, m_p$. Then

$$\exp(A) := r(A),$$

where r is the unique Hermite interpolating polynomial of degree less than $\sum_{i=1}^{p} m_i$ that satisfies the interpolation conditions

$$r^{(j)}(\lambda_i) = \exp(\lambda_i), \;\; j = 0, \ldots, m_i - 1, \;\; i = 1, \ldots, p.$$

The thesis is organized as follows: in Chapter 2 we describe the Krylov subspace methods, giving definitions, properties and we introduce the shift and invert approach proposed in [49] and [68]. In Chapter 3 we describe the basic facts of rational approximations to the exponential, giving particular emphasis to the Chebyshev and Padé approximations. In Chapter 4 we describe some methods for solving linear systems and introduce some concepts related to preconditioning, used for our numerical experiments. In Chapter 5 we present the real valued method of [5] and we describe the resulting acceleration technique; moreover we discuss the relations among rational functions and Krylov subspace methods, giving a unifying perspective for them. In the final chapter, Chapter 6, we list some numerical experiments with realistic situations in which the computation of (1.1.1) is required: we numerically compare several methods currently employed, giving interesting informations about their performance, especially when they are compared with the commonly used Crank-Nicolson integrator.

We would like to stress the fact that we only consider the exponential operator but several concepts and properties naturally extend to other classes of functions.

Most of the analysis reported here was published in [55], written in cooperation with Valeria Simoncini.

1.2 Notations

We describe some notations used throughout the thesis: for any given k, e_k will denote the kth unit vector belonging to $\mathbb{R}^m$ while I will denote the identity matrix whose dimension depends on the context; spec(A) will denote the spectrum of A while we will use λ for its eigenvalues; when referring to the *condition number* of a symmetric matrix A we mean the ratio of the absolute values of the largest and the smallest eigenvalues of A, that is, $\kappa_2(A) := \left|\lambda_{\max}/\lambda_{\min}\right|$ and we will simply write $\kappa(A)$.

In this thesis we assume that the matrix A is real, of large dimension, symmetric negative semi-definite with spec$(A) \subset [\alpha, 0]$, for some $\alpha < 0$ with large modulus. This requirement allows us to simplify the analysis without restricting the applicability of the approach. If, indeed, spec$(A) \subset [\alpha, \beta]$, with $\beta < 0$, then we may shift the matrix towards the origin by defining $A_1 = A - \beta I$; this matrix will have spectrum in $[\alpha - \beta, 0]$ and, once applied our analysis to $\exp(A_1)$, we may express $\exp(A)$ as $\exp(A) = \exp(A_1)\exp(\beta)$.
Also the hypothesis on the negative definiteness of the matrix is not restrictive since, with a suitable scaling and shifting, also indefinite matrices can be considered.
Throughout this thesis we will assume, without loss of generality, that $\|v\| = 1$; in the generic case in which this hypothesis is not satisfied we may define $v_1 = v/\|v\|$, apply our analysis to $y_1 = \exp(A)v_1$ and then compute $\exp(A)v = \|v\|y_1$.
In the following we will use the symbol Π_k to denote the set of algebraic polynomials of degree at most k.

This version slightly differs from the previous one in few grammar corrections.

Chapter 2

Krylov Subspace Methods

In this chapter we analyze the Krylov subspace methods; their application in the context of the matrix exponential was first proposed in [51] in 1983 and from that moment they represent one of the most used methods for computing $\exp(A)v$ when A is large. The ease of implementation is among the main reasons for this success, together with the fact that they simplify the problem, usually by sensibly reducing its dimension.
We start by analyzing the basic facts related to Krylov spaces; we then go into the details of the concepts used in the rest of the thesis; we present the shift and invert method, which will be largely used in the following, together with practical tools for handling problematic situations.

2.1 Krylov subspaces

In 1931 A. N. Krylov published the paper [39] in the context of eigenvalue problems and he introduced what is now called the *Krylov space*; we recall the definition:

Definition 2.1.1. The *Krylov space* of dimension m defined by a matrix $A \in \mathbb{C}^{n\times n}$ and a vector $v \in \mathbb{C}^n$ is

$$K_m(A, v) = \operatorname{span}\{v, Av, \dots, A^{m-1}v\}. \tag{2.1.1}$$

When the context is clear we will only write K_m, without specifying the matrix and the vector involved.

One important property of K_m, which follows directly from its definition, is that

$$K_m(A, v) = \{\ p(A)v \mid p \in \Pi_{m-1}\}.$$

The basic idea of Krylov methods is to project the original (large) matrix A onto a m-dimensional Krylov subspace by constructing a basis V_m of the subspace; then the exponential of the projected and restricted matrix H_m is computed by using a standard technique; finally an approximation to $\exp(A)v$ is recovered from the first column of the matrix $V_m \exp(H_m)$.

In the rest of this thesis we will always consider an orthonormal basis for K_m, unless otherwise specified. The most common approach to build this basis is the *Arnoldi* method, whose name refers to W. E. Arnoldi who presented it in [4]. This method starts from the initial vector $v_1 = v/\|v\|$ and it is an iterative procedure that, at each step, adds a new vector v_i to the basis. A key point of the method is the application of the Gram-Schmidt procedure to orthonormalize this vector with respect to the older ones. In practice, if the vectors $v_1, \ldots, v_j$ have been determined, then v_{j+1} is obtained as $w/\|w\|$ with $w = Av_j - \sum_{i=1}^{j}(v_i^T Av_j)v_i$; unfortunately in finite precision arithmetic this computation leads to perturbed results, which may affect the quality of the final approximation. For this reason it is necessary to resort to a *modified* Gram-Schmidt orthonormalization which is equivalent to the standard one in exact arithmetic but is more reliable in finite precision arithmetic. We sketch the main steps of the global procedure, taking into account our hypothesis $\|v\| = 1$:

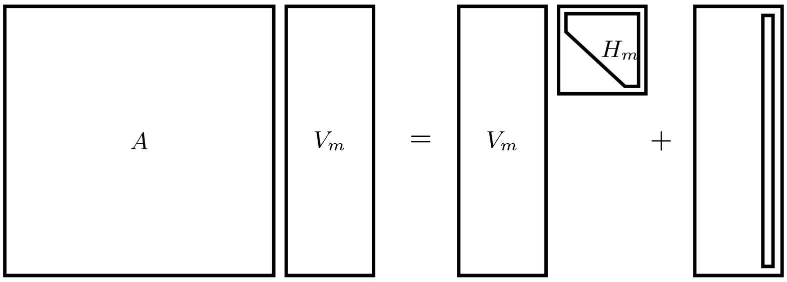
```
**Arnoldi**

1. Set $v_1 = v$;

2. For $j = 1, 2, \ldots, m$:

     Compute $w = Av_j$
     For $i = 1, \ldots, j$:
        $h_{i,j} = w^T v_i$
        $w = w - h_{i,j} v_i$
     End For
     $h_{j+1,j} = \|w\|$
     If $h_{j+1,j} = 0$ then Stop
     $v_{j+1} = \frac{w}{h_{j+1,j}}$
```

```
3. End For
```

If we denote by H_m the $m \times m$ upper Hessenberg matrix consisting of the coefficients $h_{i,j}$ computed from the algorithm and $V_m := [v_1, \dots, v_m]$, then we have the well known relation

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T. \tag{2.1.2}$$

An insightful way for understanding the meaning of (2.1.2) may be offered by its graphical representation, namely



When the matrix A is symmetric, the Arnoldi method simplifies into the *symmetric Lanczos* process, which entails a three-term recurrence. This is due to the additional requirement that the upper Hessenberg matrix $H_m = V_m^T A V_m$ is symmetric and therefore tridiagonal, and so all $h_{i,j} = 0$ for $i = 1, 2, \dots, j-2$. We sketch the algorithm:

Symmetric Lanczos

```
1. Set $v_1 = v$;
2. Set $\beta_1 = 0, v_0 = 0$;
3. For $j = 1, \dots, m$:
       $w = Av_j - \beta_j v_{j-1}$
       $\alpha_j = w^T v_j$
       $w = w - \alpha_j v_j$
       $\beta_{j+1} = ||w||$
```

```
      If $\beta_{j+1} = 0$ then Stop
      $v_{j+1} = \frac{w}{\beta_{j+1}}$
4. End For
```

If we denote with $T_m = (t_{i,j})$ the tridiagonal symmetric matrix with $t_{i,i} = \alpha_i$ and $t_{i,i-1} = \beta_i$, then the *symmetric* Lanczos relation reads

$$AV_m = V_m T_m + t_{m+1,m} v_{m+1} e_m^T. \tag{2.1.3}$$

We shall refer to this approximation as the *Standard Lanczos* method.
The symmetric Lanczos method is advantageous since it leads to savings in computations and in memory requirements; however, as mentioned above, working in finite precision arithmetic perturbs slightly the Lanczos relation. In practice (2.1.3) should be written as

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T + F_m$$

if the columns of F_k represent the rounding errors at each step. Paige [53] and Druskin and coauthors [18] derived small upper bounds for the individual roundoff terms and the conclusion in [18] was that the Lanczos vectors produced in finite precision arithmetic still offer a good approximation to $\exp(A)v$; however there are situations in which the loss of orthogonality among the computed vectors is withering and a double orthogonalization is applied to reduce the errors.
In the following we will assume to work in exact arithmetic.

Another common algorithm to build the projection and the projected matrices is the method proposed by C. Lanczos in [40], particularly useful in the nonsymmetric case and for this reason known as the *nonsymmetric Lanczos* algorithm. The first step is the selection of a vector w such that $w^T v = 1$, while the core of the method is the construction of two biorthogonal bases $\{q_1, \ldots, q_m\}$ and $\{w_1, \ldots, w_m\}$ for the two Krylov subspaces $K_m(A, v)$ and $K_m(A^T, w)$ by means of recursions relating w_i to q_i and q_i to w_{i-1}. If $Q_m = [q_1, \ldots, q_m]$ and $W_m = [w_1, \ldots, w_m]$ then the analogue of the Arnoldi relation (2.1.2) is

$$AQ_m = Q_{m+1} W_m \tag{2.1.4}$$

where W_m is a tridiagonal matrix whose entries are the coefficients of the biorthogonalization while the columns of Q_m are not orthonormal.

In the following we will only use the standard Lanczos method since all the matrices involved in our analysis turn out to be symmetric. We will refer to nonsymmetric Lanczos only for introducing the QMR method in Section 4.2.1, in the context of numerical methods for linear systems.

Relation (2.1.3) represents the core of the Lanczos method since it provides the projection and restriction of A onto K_m, the matrix T_m; indeed, due to the orthogonality of the matrix V_m, $T_m = V_m^T A V_m$. The next step to approximate $\exp(A)v$ involves computing $\exp(T_m)$: this evaluation is no longer problematic since, as we will see later, the dimension of T_m is in general much smaller than that of A. The theoretical justification of this process is given by the following result, presented by Saad in [58] and here reported in our symmetric context.

Theorem 2.1.2 ([58])**.** *Let A be a symmetric matrix and V_m, T_m be the results of m steps of the Lanczos method applied to A. Then for any polynomial p_j of degree $j \leq m-1$ the following equality holds*

$$p_j(A)v = V_m p_j(T_m)e_1.$$

The previous result justifies the crucial approximation

$$\exp(A)v \approx V_m \exp(T_m)e_1 \tag{2.1.5}$$

since, as seen in the Introduction, $\exp(A)$ can always be expressed as a matrix polynomial $p(A)$, where p interpolates exp in the Hermite sense in the eigenvalues of A. If, in the previous theorem, we choose the polynomial p which interpolates exp in the *Ritz* values of A, i.e. in the eigenvalues of T_m, then we have

$$V_m \exp(T_m)e_1 = V_m p(T_m)e_1 = p(A)v$$

from which (2.1.5) follows. Therefore, the approximation (2.1.5) can be seen as an interpolation of exp in the Ritz values of A with respect to K_m and this explains the name *Ritz approximation* sometimes used for denoting it.

2.1.1 Corrected schemes

In [58] Saad observed that in (2.1.5) the only vectors involved are $v_1, \dots, v_m$ even if at the m-th step also the vector v_{m+1} is available; for this reason he suggested to use a *corrected* scheme which takes into account also this vector, resulting in a more accurate approximation. For defining this scheme the function

$$\varphi_1(z) = \frac{\exp(z) - 1}{z} \tag{2.1.6}$$

is needed to write

$$\exp(A)v \;=\; v + A\varphi_1(A)v.$$

By applying Theorem 2.1.2 to the function φ_1 it follows that the vector $V_m\varphi_1(T_m)e_1$ approximates $\varphi_1(A)v$ and, if s_m is the error in the approximation, then

$$\begin{aligned}\exp(A)v &= v + A\varphi_1(A)v\\ &= v + A(V_m\varphi_1(T_m)e_1 + s_m)\\ &= v + (V_mT_m + t_{m+1,m}v_{m+1}e_m^T)\varphi_1(T_m)e_1 + As_m\\ &= V_m[e_1 + T_m\varphi_1(T_m)e_1] + t_{m+1,m}e_m^T\varphi_1(T_m)e_1v_{m+1} + As_m\\ &= V_m\exp(T_m)e_1 + t_{m+1,m}e_m^T\varphi_1(T_m)e_1v_{m+1} + As_m\end{aligned}$$

from which we may deduce the *corrected* approximation

$$\exp(A)v \approx \;\; V_m\exp(T_m)e_1 + t_{m+1,m}e_m^T\varphi_1(T_m)e_1v_{m+1}.$$

To numerically evaluate this approximation one may use the following theoretical equivalence.

Theorem 2.1.3 ([58]). *Define the* $(m+1)\times(m+1)$ *matrix*

$$\widehat{T}_m = \begin{pmatrix} T_m & 0\\ c & 0\end{pmatrix}$$

where c *is any row vector of length* m*. Then*

$$\exp(\widehat{T}_m) = \begin{pmatrix} \exp(T_m) & 0\\ c\varphi_1(T_m) & 1\end{pmatrix}.$$

The choice $c = t_{m+1,m}e_m^T$ allows the straightforward application of the corrected scheme and leads to the approximation

$$\exp(A)v \approx V_{m+1}\exp(\widehat{T}_m)e_1 \qquad (2.1.7)$$

which requires a computational cost similar to that of (2.1.5).
We will use this corrected scheme in Section 2.2.2 for the description of *a posteriori* estimates for the Lanczos process.

2.2 Quality of the Ritz approximation

It is crucial at this point to understand the quality of the approximation (2.1.5). It is always possible to detect the dimension m^* corresponding to an *exact* approximation, that is, such that

$$\exp(A)v = V_{m^*} \exp(T_{m^*})e_1.$$

Obviously the approximation corresponding to the dimension of the matrix A is exact but wasteful since it still requires the exponential of a large matrix. Saad [58] showed that a smaller value m^*, depending also on the vector v, may be found; more precisely m^* is the *grade* of v with respect of A, that is,

$$m^* = \min\{m \in \mathbb{N} \mid \exists p \in \Pi_m : p(A)v = 0\}.$$

However in practical situations exact approximations are too expensive to be computed and the effort is detecting a smaller dimension m such that the corresponding approximation is within a desired tolerance. To this purpose it is crucial to be able to estimate the accuracy of the computed solution in terms of the error with respect to the true result. This is an important topic in numerical analysis and typically two kinds of analysis are performed to provide error estimates: the first one aims to describe the asymptotic behavior of the error; it results in the so called *a priori* error estimates, the main disadvantage being that the knowledge of the true solution is required. The second kind of analysis consists in providing error estimates which depend only on known quantities; they are defined *a posteriori* error estimates and are commonly used to construct stopping criteria for iterative methods.

In this section we list several a priori estimates, with an example showing their performances, and the most common a posteriori error bounds.

2.2.1 A priori error estimates

We report the most relevant a priori error estimates presented over the years; as mentioned above, they are asymptotically accurate, in the sense that they mimic the qualitative behavior of the Lanczos error when the dimension m is sufficiently large; we also report an example in which the numerical performances of the bounds are compared.

Theorem 2.2.1. *Let A be a symmetric negative semidefinite matrix with eigenvalues in the interval $[-4\rho, 0]$. Let ε_m be the error in the Lanczos approximation of $\exp(A)v$, that is,*

$$\varepsilon_m = \| \exp(A)v - V_m \exp(T_m)e_1\|.$$

The following upper bounds for ε_m have been presented and we report them in chronological order:

- *Druskin and Knizhnerman [19]*

$$\varepsilon_m \le \begin{cases} 2[\sqrt{2\pi}+O(\frac{1}{\rho})]\frac{\sqrt{2\rho}}{m}\exp\big[-\frac{m^2}{4\rho}+O(\frac{m^4}{\rho^3})\big] & m \le 2\rho \\ 4\frac{\rho^m}{m!}\exp(\rho^2)\big(1-\frac{\rho}{m+1}\big)^{-1} & m \ge \rho - 1 \end{cases} \tag{2.2.1}$$

- *Tal-Ezer [66]*

$$\varepsilon_m \le \sqrt{\frac{1}{\rho\pi}}\exp\left(-\frac{m^2}{4\rho}\right) \tag{2.2.2}$$

- *Saad [58]*

$$\varepsilon_m \le \frac{2^{m+1}}{m!}\rho^m \tag{2.2.3}$$

- *Stewart and Leyk [65]*

Let $b=\frac{2}{1+\sqrt{5}}$ and $d=\frac{\exp(b)}{2+\sqrt{5}}$ then

$$\varepsilon_m \le 4\exp\left(-\frac{bm^2}{4\rho}\right)\left[1+\sqrt{\frac{\rho\pi}{b}}+\frac{d^{\,4\rho}}{1-d}\right] \tag{2.2.4}$$

- *Hochbruck and Lubich [36]*

$$\varepsilon_m \le \begin{cases} 10\ \exp\big(-\frac{m^2}{5\rho}\big) & \sqrt{4\rho} \le m \le 2\rho \\ \frac{10}{\rho}\exp(-\rho)(\frac{e\rho}{m})^m & m \ge 2\rho. \end{cases} \tag{2.2.5}$$

We now present an example to compare the a priori error estimates listed above.

Example 2.2.2. Let D be a diagonal matrix of dimension 1001 with uniformly distributed entries in the interval $[-40000, 0]$ and let v be the vector of all ones, normalized as to have unit norm. We compare the curve of the *true* Lanczos error with those given by the terms in (2.2.1)-(2.2.5) which depend on m, as m varies.
In Figure 2.1 we actually compare only (2.2.2), (2.2.4) and (2.2.5), since the curve corresponding to (2.2.1) and (2.2.3) were too far from the others and

made the plot difficult to analyze; as soon as the estimates reached the value 10^{-13} we stopped their representation.
From the plot in Figure 2.1 we may appreciate the qualitative similarities among the considered curves, all miming the *superlinear* rate of convergence, and we notice that the estimate (2.2.2) allows to save one hundred iterates with respect to (2.2.4).

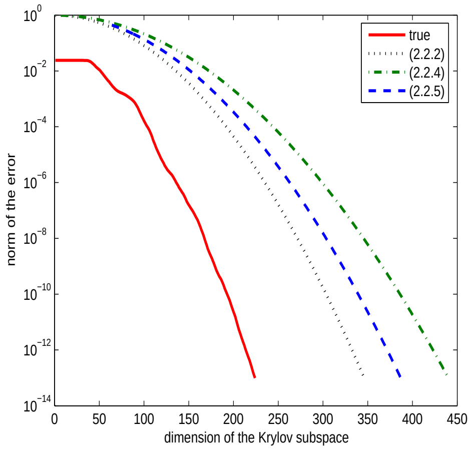


Figure 2.1: Error of Standard Lanczos and different a priori error estimates.

2.2.2 A posteriori error estimates

One of the main disadvantages of Krylov subspace methods is that they do not provide an easy expression for the error; this problem has been addressed by several authors and the most attractive result is due to Saad [58]. The formulation he found, that we report in the following theorem, immediately translates in a posteriori error estimates which are nowadays commonly used as stopping criteria. To present the result we need to introduce a sequence of functions φ_k defined by induction:

$$\varphi_0(z) = \exp(z);$$

$$\varphi_{k+1} = \frac{\varphi_k(z) - \varphi_k(0)}{z}, \quad k \geq 0.$$

Theorem 2.2.3 ([58, Theorem 5.1])**.** *The error produced by the Lanczos approximation (2.1.5) satisfies the following expansion:*

$$\exp(A)v - V_m \exp(T_m)e_1 = t_{m+1,m} \sum_{k=1}^{\infty} \left(e_m^T \varphi_k(T_m)e_1\right) A^{k-1} v_{m+1}.$$

By truncating the series in the previous theorem estimates for the Lanczos error follow, their accuracy depending on the number of terms considered; the estimates proposed in [58] are actually based on this idea: if, for example, only the first term of the series is considered, the corresponding estimate is

$$Er_1 = t_{m+1,m} |e_m^T \varphi_1(T_m)e_1|. \tag{2.2.6}$$

The previous quantity involves the function $\varphi_1(T_m)$ and, if only a rough estimate of the error is required, one may approximate it by $\exp(T_m)$ and the final estimator will be

$$Er_2 = t_{m+1,m} |e_m^T \exp(T_m)e_1|. \tag{2.2.7}$$

In [58] other error estimates were presented which are more accurate but more expensive to evaluate; for our numerical tests we found (2.2.6) and (2.2.7) to be accurate enough. In Figure 6.1 we will show a comparison among (2.2.6) and (2.2.7) in the context of the *Shift and Invert* method we will describe in Section 2.4.

The error estimates (2.2.6) and (2.2.7) turn out to be a key part of the numerical implementation of the Krylov subspace methods; indeed they offer good results with irrelevant computational costs since the quantities involved are available at each step.

2.3 Memory requirements and computational costs

The computational cost of Krylov subspace methods is essentially determined by three parts: the construction of the space $K_m(A, v)$, the evaluation of $\exp(T_m)$ and the product $V_m \exp(T_m)e_1$. For the former, following the algorithm sketched in Section 2.1, at each step the most expensive part is the matrix-vector product with A, requiring $2n^2$ operations, while we omit the cheaper operations. Thus, if m iterates are performed, the cost is $2mn^2$ operations.

The cost to evaluate $\exp(T_m)$ depends on the chosen method; for our numerical tests we used the Matlab routine `expm` implementing the Padé approximation

with scaling and squaring, which we will describe in Section 3.2. However, for a fair analysis, we consider the computational cost of a revisited implementation of it presented by Higham in [34]. The resulting computational cost of this improved procedure consists of the cost of one matrix equation and of a number of matrix products which depends on the degree of the approximation and on the norm of T_m; thus, if we only look at the order of magnitude, the cost for $\exp(T_m)$ is $O(m^3)$.
For the product $V_m \exp(T_m)e_1$ $2mn$ operations are required.
Thus to get the approximation (2.1.5) the dominating cost is the term $2mn^2$. If we take into account that the direct computation of $\exp(A)v$ would require more than n^3 operations, if the revisited Padé method is used, then the saving produced by (2.1.5) is evident.

The main *drawback* of the Krylov approach is represented by the storage requirement, especially when the matrix A is very large; indeed, to compute the approximation corresponding to a the dimension m one needs to store the matrix V_m of dimension $n \times (m+1)$, the matrix T_m of dimension $m \times m$ and the approximate vector. To overcome the storage problem an alternative approach could be applied which consists in resorting to a *two-pass strategy*: in the first pass the Lanczos method is applied and at each step the older columns of V_m are discarded and T_m is built column by column; once $\exp(T_m)$ has been generated we compute the vector $\exp(T_m)e_1$; the second pass consists in reapplying Lanczos to recompute the columns of V_m and use them one at a time to sum up $V_m \exp(T_m)e_1$. In this way the computing time increases with respect to the standard process and this may make this strategy less appealing with respect to other available in literature. Bergamaschi and Vianello [11], for example, showed that in these situations the Chebyshev series method proposed in [19] and [66] can offer a good performance, even though it requires an initial approximation of the extremal eigenvalues of A. Another alternative in the case in which the storage is of primary importance is represented by the recently proposed *Restarted Krylov* method, as well described in [22] and [1]. Let m be the largest dimension allowed for Krylov subspaces; consider the Lanczos process to build $K_m(A, v)$ by starting from $v_1^{(1)} = v$ and let $V_{m+1}^{(1)}$ and $T_m^{(1)}$ be the resulting matrices. The idea is to apply the Lanczos method by starting from the last column of $V_{m+1}^{(1)}$, say $v_1^{(2)} = v_{m+1}^{(1)}$, to get $V_{m+1}^{(2)}$ and

$T_m^{(2)}$. This process is repeated and the matrix

$$\widehat{T_k} = \begin{bmatrix} T_1 & & & \\ E_2 & T_2 & & \\ & \ddots & \ddots & \\ & & E_k & T_k \end{bmatrix} \in \mathbb{C}^{mk\times mk}, \;\; E_j = (T_j)_{m+1,m} e_1 e_m^T,$$

is updated. The resulting approximation is

$$\exp(A)v \approx [V_1 \; V_2 \; \dots \; V_k] \exp(\widehat{T_k}) e_1$$

and the process is repeated until the desired accuracy on the final solution is reached.

2.4 Shift and Invert

Recently four authors, Moret-Novati in [48] and van den Eshof-Hochbruck in [68], proposed the same technique, independently of each other, for accelerating the convergence of the numerical approximation to $\exp(A)v$. The starting points of the two papers are different and we start by describing the approach in the older paper, i.e. [48].
Moret and Novati considered in [48] the *Restricted Denominator* (*RD*) rational forms which were introduced by Nørsett in [52], with the following definition:

Definition 2.4.1. For any $\sigma \in \mathbb{R}$ and $x \in \mathbb{C}$ a form of the type

$$R_{j,k}(x,\sigma) = \frac{q_j(x)}{(1+\sigma x)^k}, \;\; q_j \in \Pi_j, \;\; k \geq 0, \tag{2.4.1}$$

is called an *RD(j,k)*-rational form.

An important property of these functions is that

$$\lim_{k\to\infty} R_{k,k}(-x,\sigma) = \exp(x)$$

for any $x \in \mathbb{C}$ having $\Re(x) < 0$, as shown in [52]. This relation suggests to use the *RD* forms to approximate the exponential, one task being to determine a good parameter σ such that $R_{k,k}(-A,\sigma)$ approximates $\exp(A)v$ for a relatively small degree k.

The technique proposed by Moret and Novati is valid for *sectorial operators*, that is for matrices A such that

$$W(A) \subset \left\{ \lambda : |\arg(\lambda - \gamma)| < \theta, 0 < \theta < \frac{\pi}{2}, \gamma \geq 0 \right\}$$

where $W(A)$ denotes the *field of values* of A, i.e.

$$W(A) = \{x^T A x | \ x \in \mathbb{C}^n, \ x^T x = 1\}.$$

Symmetric negative semidefinite matrices are sectorial operators and then in our case the analysis in [48] may be considered.
The crux of the approach proposed in [48] is to approximate $\exp(A)v$ with vectors of the form $R_{m-1,m-1}(-A, \sigma)$; they may be viewed as elements of $K_m((I - \sigma A)^{-1}, v)$ for which the Lanczos relation

$$(I - \sigma A)^{-1} V_m = V_m T_m + t_{m+1,m} v_{m+1} e_m^T \tag{2.4.2}$$

holds. Moreover, if we define $f_\sigma(z) = \exp(\frac{1}{\sigma}(1 - z^{-1}))$ then

$$\exp(A)v = f_\sigma((I - \sigma A)^{-1})v \approx V_m f_\sigma(T_m) e_1;$$

thus the resulting approximation to $\exp(A)v$ is

$$y_m := V_m \exp\left(\frac{1}{\sigma}(I - T_m^{-1})\right) e_1. \tag{2.4.3}$$

As discussed in Section 2.3, the advantage of Krylov subspace methods is that only matrix-vector products are required. However to build the space $K_m((I - \sigma A)^{-1}, v)$ any matrix-vector product is actually a linear system to solve. The positive aspect is that all these linear systems have the same symmetric shifted coefficient matrix $I - \sigma A$ and then the great advantage is that factorization, as well as preconditioners, may be computed once for all.

In [68] van den Eshof and Hochbruck proposed exactly the same approximation (2.4.3) but in the context of spectral transformations and we recall here their derivation.
When approximating $\exp(A)v$, with A symmetric negative semidefinite, only its largest eigenvalues play a role since the exponential function decays rapidly. However there are situations in which the Lanczos method has difficulties in detecting these eigenvalues; then a remedy could be to transform the problem in such a way that the Lanczos method can approximate faster the leading eigenvalues. A spectral transformation performs well in such a situation and

results in the auxiliary matrix $(I - \sigma A)^{-1}$, as in [48], for a suitable shift parameter σ. This method is well known in the context of eigenvalue problems as the *Shift and Invert* method and we will refer to it with this name.

2.4.1 Choice of the shift parameter

In [68] the authors gave an a priori estimate for the error of the approximate solution (2.4.3); this estimate in practice represents the starting point for choosing the shift parameter. Before stating the main result we introduce

$$\mathcal{R}_i^j = \{p(t)(1 - \sigma t)^{-i} | p \in \Pi_j\} \tag{2.4.4}$$

and

$$E_i^j(\sigma) = \inf_{r \in \mathcal{R}_i^j} \sup_{t \le 0} |r(t) - \exp(t)|.$$

Theorem 2.4.2 ([68])**.** *Let μ be such that $A - \mu I$ is negative semidefinite. Then*

$$\left\| V_m \exp\left(\frac{1}{\sigma}(I - T_m^{-1})\right) e_1 - \exp(A)v \right\| \le \exp(\mu) E_{m-1}^{m-1}(\tilde{\sigma}),$$

with $\tilde{\sigma} = \frac{\sigma}{1-\sigma\mu}$.

In [2] Andersson analyzed the asymptotic behavior of $E_j^j(\sigma)$ and detected for it the optimal value of σ; we report this result in the following theorem:

Theorem 2.4.3 ([2])**.** *Asymptotically the optimal value for σ is given by $\sqrt{2}/j$, for which we have*

$$\lim_{j \to \infty} \left(E_j^j(\sqrt{2}/j) \right)^{1/j} = \frac{1}{\sqrt{2} + 1}.$$

In practice one is interested only in approximations corresponding to small degree j, for which the asymptotic analysis is meaningless. However in this case minimizing $E_j^j(\sigma)$ is very challenging; for this reason in [68] the search for the optimal shift parameter was carried out numerically, by using the Remez algorithm to find the optimal polynomial approximation to $\exp\left(\frac{1}{\sigma}(1 - t^{-1})\right)$ on the interval $[0, 1]$. In this way the authors collected a table of suggested shift values corresponding to the dimension of the Krylov space considered; we will report these values in Table 5.3, when dealing with our derivation of the optimal shift parameters.
We refer to Chapter 6 for the implementation details of this technique.

In the context of iterative algorithms the idea of applying a spectral transformation for accelerating the convergence has been extensively used, see e.g. [38] published in 1972. However, as far as we know, the first work combining Arnoldi and inverse iterations is due to Ericsson and Ruhe [23] dating back to 1980. In this work the authors fix a parameter μ and compute the eigenvalues of the matrix $(A-\mu I)^{-1}$ that, for a suitable choice of the shift parameter μ, approximate some of the eigenvalues of A. For approximating all eigenvalues of A it is then necessary using different shift parameters and in [23] their selection was essentially based on heuristics. Ruhe extended this approach in [57], where he introduced the name *Rational Krylov* method, by considering arbitrary rational functions, that is, he considered functions belonging to the space

$$\tilde{R}_i^j = \{p(t)\Pi_{k=1}^{i}(1+\gamma_k t)^{-1} \mid p \in \Pi_j\}, \;\; \gamma_k \in \mathbb{R}.$$

However in [13] it was shown that the optimal approximation from $\tilde{R}_i^j$ is also contained in R_i^j. This result plays in favor of the shift and invert approach, which clearly may be viewed as a special case of the method proposed in [23] when just one shift parameter is used. Also the *Restricted Denominator* method introduced in [52] and reported in (2.4.1) deals exactly with functions in R_i^j.

2.5 Krylov Plus Inverted Krylov

In the context of numerical approximations to functions of symmetric matrices Druskin and Knizhnerman [21] proposed an *extended Krylov method* which may be interpreted as a particular rational Krylov method. The rationale was the observation that for some important problems, e.g. Maxwell's system and acoustic equations, the matrix A^{-1} may be easily used, for example by a Fast Fourier Transform, at low computational cost. This suggested the idea of considering a Krylov space method which contains information on both A and its inverse, namely by working with the space

$$K_{k,m}(A,v) = \text{span}\{A^{-k+1}v, \ldots, A^{-1}v, v, Av, \ldots, A^{m-1}v\}$$

for $m \geq 1$ and $k \geq 1$.
When building $K_{k,m}(A,v)$ two starting vectors are considered, v and $A^{-1}v$, and then at each step two vectors are added to the basis $\mathcal{V}_m$; thus $\mathcal{V}_m$ may be described as

$$\mathcal{V}_m = [V_1, \ldots, V_m] \in \mathbb{R}^{n\times 2m}$$

where $V_i \in \mathbb{R}^{n\times 2}$ contains two vectors, one multiplied by A and one by A^{-1}, orthonormalized with respect to the columns of $\mathcal{V}_{i-1}$.
By defining $\mathcal{T}_m := \mathcal{V}_m^T A \mathcal{V}_m \in \mathbb{R}^{2m\times 2m}$ the analogue of the approximation (2.1.5) reads

$$\exp(A)v \approx \mathcal{V}_m \exp(\mathcal{T}_m)e_1. \tag{2.5.1}$$

Recently Simoncini [62], dealing with Krylov methods for solving the Lyapunov matrix equations, used the extended Krylov method in a more general setting with dissipative matrices, that is, matrices M such that $M + M^T$ is negative definite. Moreover she suggested a strategy in which the user may arbitrarily increase both m and k to reach the required accuracy. This was a great improvement with respect to the method proposed in [21] in which a small value k was fixed a priori and only m was allowed to increase.
In the rest of the thesis we will use the name *KPIK* for denoting this approach, as introduced in [62] as the acronym for *Krylov Plus Inverted Krylov*.

We conclude with an example in which we compare the KPIK method with the Standard Lanczos method; for the implementation of the former we followed the algorithm sketched in [62] with direct solvers for the systems with A; in Section 6.5 additional implementation details are presented.

Example 2.5.1. We consider the diagonal matrix D of dimension 10000×10000 with entries uniformly distributed in $[-10000, -0.1]$ and the vector v of all ones, normalized as to have unit norm. We compare the convergence of the Standard Lanczos method with that of the Krylov Plus Inverted Krylov method, by plotting the error curves versus the space dimension; for the error we measured the distance among the true solution and the approximation and stopped the iteration as soon as this quantity dropped below 10^{-13}.
In Figure 2.2 we may appreciate the great saving obtained when using the KPIK method; indeed, for reaching the same accuracy the former needs a space of dimension 50, while the Lanczos method needs the 1000% more.

In Chapter 6 we will present tests comparing the elapsed time for the Krylov Plus Inverted Krylov method and other approaches described in the following chapter.

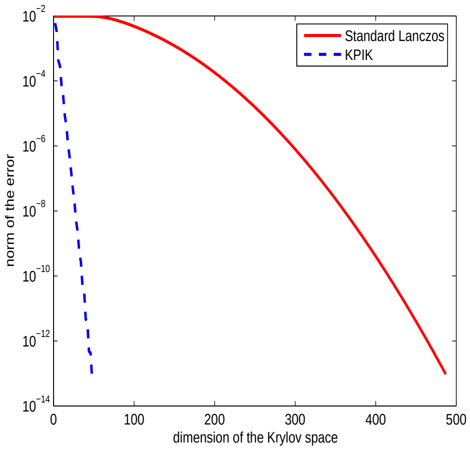


Figure 2.2: Dimension of the Krylov space for Standard Lanczos and the KPIK method

Chapter 3

Rational Approximations

Rational functions represent an important tool in approximation theory; they are largely used to approximate several classes of functions and they are among the most commonly used instruments to approximate the exponential.
One of the goals of this thesis is to offer a unifying framework for rational approximations to the exponential and Krylov subspace methods.

In this chapter we introduce definitions and concepts which will be used in the following: we start with a theoretical result stating that any continuous function may be approximated by a rational form; we then go into the details of the approximation of the exponential, with special emphasis for the Padé and the Chebyshev functions. For the practical use of rational functions we describe the partial fraction expansion, which will be a key instrument of our analysis.

3.1 Definitions and basic facts

In this section we consider the problem of approximating a generic continuous function, from which specific results for the exponential operator follow.
We introduce the fundamental definition:

Definition 3.1.1. A *rational function* $\mathcal{R}_{\mu,\nu}$ is a ratio of polynomials, i.e.

$$\mathcal{R}_{\mu,\nu}(z) = \frac{\pi_\mu z^\mu + \pi_{\mu-1} z^{\mu-1} + \ldots + \pi_0}{\rho_\nu z^\nu + \rho_{\nu-1} z^{\nu-1} + \ldots + \rho_0} = \frac{\mathcal{N}_\mu(z)}{\mathcal{D}_\nu(z)} \tag{3.1.1}$$

where the terms π_i's, ρ_j's and z are complex, $\mathcal{D}_\nu(z) \neq 0$ and the numerator and the denominator have no common roots; moreover we assume $\pi_\mu \neq 0$ and $\rho_\nu \neq 0$.

We denote by $R(\mu, \nu)$ the set of rational functions of the form (3.1.1). To define the rational form (3.1.1) $\mu + \nu + 2$ parameters have to be fixed; however it is possible to reduce this number to $\mu + \nu + 1$ also in the case $\rho_0 \neq 0$ since we may assume, without loss of generality, that $\rho_0 = 1$; indeed, if this is not the case, we may resort to it by multiplying numerator and denominator by the same suitable factor.

We introduce some notation: we denote with $C[a, b]$ the set of all continuous functions in the closed finite interval $[a, b]$ and we consider the so-called uniform norm

$$\|f\|_{C[a,b]} = \max\{|f(x)| : x \in [a, b]\};$$

moreover we define the best rational approximation in $C[a, b]$ of order (μ, ν) as

$$E_{\mu,\nu}(f)_{C[a,b]} = \inf\{\|f - \mathcal{R}_{\mu,\nu}\|_{C[a,b]} : \mathcal{R}_{\mu,\nu} \in R(\mu, \nu)\}.$$

It is always possible to detect the rational function attaining this value, as stated in the following theorem.

Theorem 3.1.2 ([54]). *Let $f \in C[a, b]$; then there exists a rational function $\mathcal{R}_{\mu,\nu} \in R(\mu, \nu)$ such that*

$$\|f - \mathcal{R}_{\mu,\nu}\|_{C[a,b]} = E_{\mu,\nu}(f)_{C[a,b]}.$$

$\mathcal{R}_{\mu,\nu}$ is called a rational function of best rational approximation to f in $C[a, b]$, or of *best uniform approximation* to f.

We restrict our analysis to rational functions having numerator and denominator with the same degree ν, the *diagonal rational functions*, and we simply write $\mathcal{R}_\nu$ in place of $\mathcal{R}_{\nu,\nu}$.
We address now the problem of practically evaluating $\mathcal{R}_\nu(A)v$.

3.1.1 Partial Fraction Expansion

When the dimension of the matrix A is small the vector $w = \mathcal{R}_\nu(A)v$ may be computed directly by solving the linear system

$$\mathcal{D}_\nu(A)w = \mathcal{N}_\nu(A)v, \tag{3.1.2}$$

once the powers of A have been computed to evaluate $\mathcal{N}_\nu(A)$ and $\mathcal{D}_\nu(A)$.
This strategy is not feasible when A is a large matrix, as we are assuming; in this case a useful tool when handling $\mathcal{R}_\nu(A)$ is its partial fraction expansion. We stress that we only consider the case of simple poles since this is the case

for the rational approximations to exp we will use.
Let $\xi_1, \dots, \xi_\nu$ be the poles and $\omega_0, \dots, \omega_\nu$ the residues of $\mathcal{R}_\nu$, that is

$$\omega_0 = \lim_{z\to\infty} \frac{\mathcal{N}_\nu(z)}{\mathcal{D}_\nu(z)}, \;\; \omega_j = \frac{\mathcal{N}_\nu(\xi_j)}{\mathcal{D}'_\nu(\xi_j)}, \;\; j = 1, \dots, \nu,$$

then $\mathcal{R}_\nu$ has the following *partial fraction expansion*:

$$\mathcal{R}_\nu(z) = \omega_0 + \sum_{j=1}^{\nu} \frac{\omega_j}{z - \xi_j}. \tag{3.1.3}$$

For $\mathcal{R}_\nu(A)v$ expression (3.1.3) reads

$$\mathcal{R}_\nu(A)v = \omega_0 v + \sum_{j=1}^{\nu} \omega_j (A - \xi_j I)^{-1} v \tag{3.1.4}$$

which requires the solution of the shifted linear systems appearing in the sum; in the next chapter we will consider practical methods for solving them.
Since we assume A to be real the poles in (3.1.3) come in complex conjugates, therefore we can almost halve the number of terms in the sum, resulting in the expression

$$\mathcal{R}_\nu(A)v = \omega_0 v + \sum_{\substack{j=1 \\ j\,\mathrm{odd}}}^{\nu-1} 2\Re\Big(\omega_j (A - \xi_j I)^{-1} v\Big) + \omega_\nu (A - \xi_\nu I)^{-1} v \tag{3.1.5}$$

where ξ_ν denotes the real pole if ν is odd.

We now address the problem of determining good rational approximations for the exponential.

3.2 Padé approximation

Padé approximations are among the most used rational forms in the context of approximation theory; in this section, by using the notation in [54], we recall their definition as stated by Baker in 1973:

Definition 3.2.1. We say that the *Padé approximant* of order (μ, ν) exists if there exist two polynomials $\mathcal{N}_{\mu\nu}$ of degree μ and $\mathcal{D}_{\mu\nu}$ of degree ν such that

i) $f(z) - \mathcal{N}_{\mu\nu}(z)/\mathcal{D}_{\mu\nu}(z) = O(z^{\mu+\nu+1})$ as $z \to 0$,

ii) $\mathcal{D}_{\mu\nu}(0) = 1$.

Usually the Padé approximants of order (μ, ν) are displayed in a table, called the *Padé table*, whose form is reported in Table 3.1.

$\mu\backslash\nu$	0	1	2	$\dots$
0	(0,0)	(1,0)	(2,0)	$\dots$
1	(0,1)	(1,1)	(2,1)	$\dots$
2	(0,2)	(1,2)	(2,2)	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Table 3.1: Padé table

In the survey [45], Moler and van Loan listed some of the reasons for preferring diagonal Padé approximants when considering the exponential operator: one of them is related to the computational costs, since the same computational effort is needed for evaluating $\mathcal{R}_{\mu,\nu}$ for any couple (μ, ν) but the largest accuracy is attained in the diagonal case. In the case in which the spectrum of A is in the left half plane another important reason for choosing $\mu = \nu$ is related to the errors: indeed for $\mu \neq \nu$ rounding errors may significantly affect the computation [45]. For these reasons we refer only to diagonal Padé functions.

A direct consequence of condition (i) in Definition 3.2.1 is that the Padé approximation has a *local* behavior. This represents the main drawback of the method since the quality of the approximation rapidly deteriorates when we move away from the origin. In terms of matrices this condition forces to use the Padé approximation only for matrices with small norm; however a fundamental property unique to the exponential allows to enlarge the range of applicability of the Padé approximation. Indeed, for any scalar z and any integer m

$$\exp(z) = \big(\exp(z/m)\big)^m. \tag{3.2.1}$$

This means that if z has large modulus we may always find a scaling factor m such that z/m is small enough to compute its exponential by means of Padé approximants; finally the m-th power of this computed value is determined to approximate $\exp(z)$.

For matrices usually an effective technique is to look for a scaling factor of the form 2^s; in this way, once the approximation M to $\exp(A/2^s)$ has been

determined, to recover $\exp(A)$ the matrix M is squared s times: this procedure is known as *scaling and squaring* and goes back at least to [41], dating back to 1967.
As declared in [45], if properly implemented this procedure is one of the most effective, at least for moderate size matrices; a confirmation of this is the fact that the Matlab routine `expm` implements exactly this procedure.
In [34] Higham presented a revisited implementation of scaling and squaring Padé, resulting in computational savings; one element of this improvement was the justification of a smaller scaling factor leading to less squaring steps, which are expensive and represent a great source of rounding errors.

3.3 Chebyshev approximation

Theorem 3.1.2 gives information about the existence of the best uniform approximation; for practical implementations the second fundamental ingredient is to characterize this approximant. In this direction P. Chebyshev detected a property, now known as *Chebyshev alternation*, which characterizes the polynomial of best uniform approximation. From that property it is also possible to characterize the best rational approximation, for which the name *Chebyshev* approximation is extended.

R. Varga [71] was among the first to study the Chebyshev approximation to the exponential. Some years later in [16] Cody, Meinardus and Varga considered the problem of finding the rational function $\mathcal{R}_\nu$ such that

$$||\mathcal{R}_\nu(z) - \exp(-z)||_{C[0,+\infty)} = \min_{r\in\mathcal{R}_\nu} \max_{z\in[0,+\infty)} |r(z) - \exp(-z)| =: \zeta_\nu.$$

They solved this problem and listed the coefficients of the numerator and denominator of the best approximants up to $\nu = 14$; fifteen years later in [14] the coefficients were listed up to $\nu = 30$. For our experiments, when the Chebyshev rational forms were required, we used these coefficients.
Moreover in [14] the authors showed that $\zeta_\nu \to 0$ geometrically; the concluding result was presented in [32]:

$$\lim_{\nu\to 0} \zeta_\nu^{1/\nu} = \frac{1}{9.2890\ldots}.$$

In practice we may use the approximation

$$||\exp(A) - \mathcal{R}_\nu(A)|| \lesssim 10^{-\nu} \tag{3.3.1}$$

when A is symmetric negative semidefinite, as in our assumption.

Recently Trefethen and coauthors [67], dealing with the numerical computation of contour integrals, offered an interesting connection among quadrature formulas and rational functions. In particular their analysis perfectly fits the computation of $\exp(A)v$; moreover they proposed to use the *Carathéodory-Fejér* method for recovering the coefficients of the best rational approximation to the exponential.

Chapter 4

Linear Systems and Preconditioning

4.1 Overview

In Section 2.4 we described the *shift and invert* method, proposed in [48] and [68], for which an essential element is the Krylov space $K_m((I-\sigma A)^{-1}, v)$, for a suitable real parameter σ. To build this space linear systems with the coefficient matrix $(I-\sigma A)$ need to be solved. Using the partial fraction expansion of a rational function, as in (3.1.5), requires to solve linear systems with coefficient matrices of the form $(A-\zeta I)$, with ζ complex. Thus solving linear systems of the shifted form

$$(I-\omega A) \tag{4.1.1}$$

turns out to be a key aspect of our problem; indeed the accuracy on the final approximation to $\exp(A)v$ strongly depends also on the accuracy reached when solving these linear systems.

When needed we use the symbol Z to denote the shifted matrix (4.1.1); we stress that this matrix is complex symmetric, that is, if $z_{i,j}$ is the generic (i,j) entry of Z then the generic (i,j) entry of Z^T is $z_{j,i}$. We address the problem of solving linear systems of the form

$$Zx = b. \tag{4.1.2}$$

To this purpose we restrict our attention to the solvers and the preconditioners used in the implementations presented in this thesis; moreover we stress that they represent a minor part of the amount of methods for linear systems currently available, see e.g. [59], [8], [33] and [64] for a recent survey.

4.2 Solving linear systems

Direct methods represent the simplest way for solving linear systems. They are based on a factorization of the coefficient matrix and the Gaussian elimination method is an example of them: it consists in factorizing the matrix Z into two triangular factors in such a way that the solution to (4.1.2) may be computed by solving two triangular systems; in particular, when the coefficient matrix is symmetric positive definite its Cholesky factor represents the coefficient matrix of the consequent systems. However, when the coefficient matrix (4.1.1) is very large direct methods may become prohibitively expensive, or even impossible to apply. A viable alternative in this situation is represented by *iterative* methods: they start from an initial guess and find successive approximations to obtain more accurate solutions at each step.
We restrict our attention to iterative methods based on Krylov space approximations.

4.2.1 Krylov Methods

Let x_0 be an initial guess to the solution, let $r_0 = b - Zx_0$ be the corresponding residual, $\beta = ||r_0||$, and $K_m = K_m(Z, r_0)$ be the Krylov space of dimension m defined by Z and r_0; thus the standard Lanczos relation (2.1.3) reads

$$ZV_m = V_m T_m + t_{m+1,m} v_{m+1} e_m^T. \tag{4.2.1}$$

At the m-th step an approximation x_m to x is constructed which belongs to the space $x_0 + K_m$; thus $x_m = x_0 + V_m y_m$ for a certain vector y_m obtained by fulfilling some specific condition.
We report the two conditions we will use later on for which we need to introduce the vector $r_m := b - Zx_m$, i.e. the residual at the m-th iterate: the conditions are

i) *Galerkin* condition

$$r_m \perp K_m \tag{4.2.2}$$

ii) *Minimal Residual* condition

$$||r_m|| = \min_{x \in x_0 + K_m} ||b - Zx||.$$

We now list the methods belonging to this class which we will use in the following:

- **FOM.** The Full Orthogonalization Method (FOM) consists in imposing the Galerkin condition on the residual; for the properties of Krylov spaces the condition (4.2.2) may be written as

  $$V_m^T(b - Zx_m) = 0$$

  that, taking into account the Lanczos relation (4.2.1), reduces to the tridiagonal system

  $$T_m y_m = \beta e_1. \tag{4.2.3}$$

- **CG.** When the coefficient matrix Z is real symmetric positive definite then T_m is also symmetric positive definite, and the whole procedure can be simplified so as to derive a coupled-two term recurrence; this method is known as the Conjugate Gradients (CG) method, see e.g. [59].
  An important feature of this method is that the error may be bounded in terms of the spectral condition number of Z, indeed

  $$||x - x_m||_Z \le 2\left(\frac{\sqrt{\kappa(Z)} - 1}{\sqrt{\kappa(Z)} + 1}\right)^m ||x - x_0||_Z$$

  and the symbol $\| \cdot \|_Z$ denotes the Z norm, that is $\|x\|_Z^2 = x^T Z x$. This relation highlights that the number of iterations to reach a certain accuracy depends on $\kappa(Z)$.

- **Simplified QMR.** The *Quasi Minimal Residual (QMR)* method applies to nonsymmetric systems, say $Bx = b$. Let Q_m and W_m be the matrices obtained by applying the nonsymmetric Lanczos method, as described in Section 2.1, to B and $r_0 = b - Bx_0$, $\beta = \|r_0\|$; thus at the m-th step the vector $x_m = x_0 + Q_m y_m$ is the approximate solution with y_m obtained by imposing certain conditions we define next. For (2.1.4), the residual r_m may be written as

  $$r_m = Q_{m+1}(\beta e_1 - W_m y_m).$$

  If Q_{m+1} had orthonormal columns, as in the symmetric case, then $\|r_m\| = \|\beta e_1 - W_m y_m\|$ and one could select the vector y_m such that

  $$||\beta e_1 - W_m y_m|| = \min_{y \in \mathbb{R}^m} ||\beta e_1 - W_m y||; \tag{4.2.4}$$

this is the criterion of the *MINRES* method which ensures that $\|r_m\|$ is minimized. As discussed in Section 2.1, Q_{m+1} does not have orthonormal columns; nonetheless, the *QMR* method imposes the same condition (4.2.4), thus imposing only a *quasi*-minimization condition of the residual, from which the name of the method follows.

In [26] Freund and Nachtigal proposed a *simplified* version of the QMR method for J-symmetric and J-Hermitian matrices. In the following we will consider only J-symmetric matrices and we recall their definition

Definition 4.2.1. Given a nonsingular matrix $J \in \mathbb{C}^{n\times n}$ a matrix A is J-symmetric if

$$A^T J = JA.$$

When the coefficient matrix is J-symmetric then the nonsymmetric Lanczos method may be applied by starting from a vector v_1 and the vector w_1 defined as $w_1 = \varsigma_1 J v_1$ for some $\varsigma_1 \in \mathbb{C}, \varsigma_1 \neq 0$, as suggested in [26]. Starting from these vectors it follows that $w_n = \varsigma_n J v_n$ for any $n \geq 1$, with a consequent simplification of the Lanczos process since we only need to generate the vectors $v_1, \ldots, v_n$, without matrix multiplications with A^T or complicated formulas to compute the w_i's.

In our implementation we used a slightly different formulation of simplified QMR and we will present it in Section 4.3, when dealing with preconditioning.

4.2.2 Solving $A^2x = b$

In [70] van der Vorst proposed a useful method for solving equations of the form $f(A)x = v$, when A is symmetric positive definite, by exploiting information obtained by applying the Conjugate Gradient method to $Ax = b$.

A specific example is the case in which $f(A) = A^2$; we recall here the basic facts related to this case, which is the one we will encounter in our implementation, stressing that the basic idea is to solve $Ay = b$ and $Ax = y$.

Let $y_0 = 0$ be the initial guess for applying CG to the system $Ay = b$ and consider the space $K_{i+1}(A, r_0)$, r_0 being the initial residual. If $R_i := [r_0, \ldots, r_i]$ and $\{r_0, \ldots, r_i\}$ is a set of orthogonal vectors spanning $K_{i+1}(A, r_0)$, then the following relation holds:

$$AR_i = R_i T_i - t_{i+1,i} r_{i+1} e_i^T ,$$

with T_i a tridiagonal matrix.

The $(i + 1)$st approximate solution obtained by applying CG to $Ay = b$

is $y_{i+1} = R_i T_i^{-1} e_1$. Let x_{i+1} be the approximate solution to $Ax = y_{i+1}$ obtained by applying the Conjugate Gradient with respect to $K_{i+1}(A, r_0)$; thus $x_{i+1} = R_i z_{i+1}$ and the Galerkin condition $R_i^T A x_{i+1} = R_i^T y_{i+1}$ becomes $R_i^T y_{i+1} = R_i^T A R_i z_{i+1} = R_i^T R_i T_i z_{i+1}$ from which we get the expression $T_i z_{i+1} = (R_i^T R_i)^{-1} R_i^T y_{i+1}$ and, taking into account the form of y_{i+1},

$$\begin{aligned} x_{i+1} &= R_i T_i^{-1} (R_i^T R_i)^{-1} R_i^T y_{i+1} = R_i T_i^{-1} (R_i^T R_i)^{-1} R_i^T R_i T_i^{-1} e_1 \\ &= R_i T_i^{-1} T_i^{-1} e_1 = R_i T_i^{-2} e_1. \end{aligned}$$

With suitable algebraic manipulations a recursive formula for computing x_{i+1} is derived; the important facts are that to compute x_{i+1} is not necessary to store all the r_i's and only four additional flops per iterations are required with respect to the computation of y_{i+1}; moreover y_{i+1} needs not be computed explicitly if only $A^2 x = b$ has to be solved.

In [70] a convergence analysis is reported showing the validity of the approach. This technique is very effective when solving systems with A and A^2 as coefficient matrices, since in the same run of the Conjugate Gradient method the two approximate solutions are available; however preconditioning is not applicable to this method, as we face in Section 6.2.

4.2.3 Complex systems

In general any solver available in literature works also when applied to complex systems, the only difference being that the computations involve complex arithmetic. However this may be completely avoided by resorting to an equivalent real formulation of the original complex system: assume indeed that $A = R + \imath S$ with $R = (A + A^*)/2$ and $S = (A - A^*)/2\imath$ and let $b = b_R + \imath b_I$. Then the solution $x = x_R + \imath x_I$ of (4.1.1) may be computed by solving the real system

$$\begin{bmatrix} R & -S \\ S & R \end{bmatrix} \begin{bmatrix} x_R \\ x_I \end{bmatrix} = \begin{bmatrix} b_R \\ b_I \end{bmatrix}. \qquad (4.2.5)$$

There are also other possible real formulations, e.g. when A is symmetric the formulation

$$\begin{bmatrix} R & S \\ S & -R \end{bmatrix} \begin{bmatrix} x_R \\ -x_I \end{bmatrix} = \begin{bmatrix} b_R \\ b_I \end{bmatrix}$$

has the same symmetry property than the original system.

To preserve the structure of A Day and Heroux [17] defined a real equivalent formulation, called the K-formulation, obtained by rewriting each entry in A

as a 2×2 real matrix and Benzi and Bertaccini in [10] considered block preconditioners for this formulation.

In general the advantage in solving (4.2.5), that in any case has a doubled dimension with respect to the original one, is that it requires only real arithmetic although there is no guarantee that it is simpler to solve than (4.1.1).

The main motivation for avoiding complex arithmetic when dealing with large systems is the scarcity of preconditioning software currently available, as noticed in next section.

Implementing an iterative method in complex formulation may be wasteful also when dealing with matrices in which most of the entries are real and the complex ones are localized, as often occurs in practice: in this case if, for example, the matrix is needed only for matrix-vector products, then one could keep the complex entries separate from the real, thus achieving computational savings, as shown in [12].

4.3 Preconditioning linear systems

A common property of Krylov subspace methods is that their convergence rate depends on the spectral properties of the coefficient matrix, with the consequence that for some systems the convergence may be very slow. In these situations a transformation of the coefficient matrix may help and it is usually achieved by multiplying for a suitable matrix, called *preconditioner*.

In general preconditioning transforms the system $Ax = b$ into the equivalent one

$$M_1^{-1} A M_2^{-1} \hat{x} = M_1^{-1} b, \;\; \hat{x} = M_2 x. \tag{4.3.1}$$

The main purpose of this transformation is to make the system in (4.3.1) converge faster than the original one; moreover the matrices M_1 and M_2 should be chosen so that the improvement in the convergence overcomes the additional costs due to their computation and application.

As special case the identity matrix may be chosen, leading to *ad hoc* definitions: the choice $M_2 = I$ defines the *left* preconditioning which results in the simple formulation $M_1^{-1} A x = M_1^{-1} b$, while $M_1 = I$ corresponds to *right* preconditioning $A M_2^{-1} \hat{x} = b, \;\; \hat{x} = M_2 x$.

Usually when A is symmetric the matrices M_1 and M_2 are chosen such that $M_1^{-1} A M_2^{-1}$ is symmetric and $\kappa(M_1^{-1} A M_2^{-1}) \ll \kappa(A)$, to ensure a faster convergence than that expected for $Ax = b$. Clearly a good choice is to select M_1 and M_2 such that $M_1^{-1} A M_2^{-1}$ approximates the identity matrix.

When the matrix A is large and sparse effective preconditioners are often defined by using its *incomplete* factorization; for the LU factorization, for example, the crux is to compute a sparse lower triangular matrix L and a sparse upper triangular matrix U so that the residual matrix $R = LU - A$ satisfies certain conditions, such as having zero entries in fixed positions; the same is true if the incomplete Cholesky factorization is desired, for which the residual is $R = LL^T - A$, with L lower triangular matrix.
A common requirement on R is to have a predefined sparsity pattern and then entries in fixed positions are set to zero; another common criterion is setting to zero the elements in the factors which are smaller than a fixed value, usually called *drop tolerance*; more precisely, the comparison is carried out with this drop tolerance multiplied by the norm of the inspected row. In our experiments we applied the latter strategy, both for the incomplete LU and for the incomplete Cholesky factorizations, with a common drop tolerance of 10^{-2}.

In some numerical experiments in Chapter 6 we used the *Preconditioned Conjugate Gradients* (PCG): it may be applied when the coefficient matrix is symmetric positive definite and its basic idea is to apply CG to the transformed system $M^{-1}AM^{-1}\tilde{x} = M^{-1}b$, with M symmetric positive definite and $\tilde{x} = Mx$, see e.g. ([31], Section 10.3).

The systems deriving from the partial fraction expansion (3.1.4) have a coefficient matrix with a shifted form, complex symmetric but indefinite. For their solution we used the simplified QMR procedure, as suggested in [26] by Freund and Nachtigal: for indefinite symmetric systems of the form $Ky = c$ consider a symmetric matrix $M \in \mathbb{C}^{n\times n}$ chosen as preconditioner; in this way by writing M as $M = M_1M_2$ the preconditioned system to be solved is $Ax = b$ with $A = M_1^{-1}KM_2^{-1}, x = M_2y$ and $b = M_1^{-1}c$. It follows that A is J-symmetric for $J = M_1^TM_2^{-1}$ and then the simplified QMR method applies. In our implementation M was chosen as an incomplete factorization of the coefficient matrix.

4.3.1 Preconditioning complex systems

As mentioned above, preconditioning complex systems is much harder than preconditioning the real ones, one reason being the limited amount of methods available, with respect to that for real systems; indeed, as discussed in [10], every preconditioner for the complex formulation $(R + \imath S)x = b$ has a real equivalent formulation, while there are infinitely many choices of the preconditioner for the real equivalent formulation that do not have any complex equivalent.

In Section 5.4 we describe the method proposed by Axelsson and Kucherov [5] to precondition complex systems by only using real arithmetic, from which an acceleration technique to approximate $\exp(A)v$ derives.

Our complex systems have a coefficient matrix with the very special form $A - \gamma I$. Bertaccini in [12] dealt with the problem of solving sequences of systems of this form, namely $A - \gamma_j I$. He proposed a method tailored to effectively preconditioning all of them: the idea is to consider an effective preconditioner for A and then, for each shift value, it is updated to ensure effectiveness.

Chapter 5

Krylov Space Methods and Rational Approximations

In Chapter 2 and Chapter 3 we described two distinct classes of approaches for approximating $\exp(A)v$: the Krylov space methods and the rational approximations. Connections between these two sets of methods have been investigated, among the others, by Lopez and Simoncini [42], that analyzed in depth approximations which combine the Krylov subspace methods and rational approximations to the exponential function.
The main result of this chapter is that these two categories are strictly related; behind the relevance of this unifying view in its own right, this new perspective allows us to better understand both classes of methods.

5.1 Rational function approximations and Krylov subspaces

The first interaction among the Krylov space methods and the rational approximations occurs when we think to rational approximations as a way to evaluate $\exp(T_m)e_1$, with T_m stemming from the Lanczos recurrence (2.1.3) applied to A and v.
Another strong relation between the two approaches is detected when dealing with the practical evaluation of $\mathcal{R}_\nu(A)v$, for a certain rational function $\mathcal{R}_\nu$ approximating exp. As discussed in Chapter 3, we assume that $\mathcal{R}_\nu$ has ν distinct poles, in such a way that its partial fraction expansion may be written

as (3.1.4). To use this expression, the shifted systems $(A - \xi_j I)x = v$ need to be solved; for small matrices direct methods represent a viable approach while Krylov subspace methods represent a good choice for large problems; indeed the shift invariance property

$$K_m(A - \delta I, v) \;=\; K_m(A, v), \quad \forall \delta$$

allows one, for any j, to solve the system $(A - \xi_j I)x = v$ in the same space $K_m(A, v)$. The result is an approximate solution $x_m^{(j)}$ and, once solved all systems, the resulting approximation is

$$\mathcal{R}_\nu(A)v \;\;\approx\;\; \omega_0 v + \sum_{j=1}^{\nu} \omega_j x_m^{(j)}.$$

Although for our numerical tests we followed another strategy, for our analysis we assume to solve the shifted systems by imposing the Galerkin condition (4.2.2) on the residual; then $x_m^{(j)}$ may be computed as $V_m y_m^{(j)}$ with $y_m^{(j)}$ solution to the tridiagonal system (4.2.3). The resulting approximation is

$$\exp(A)v \approx V_m \mathcal{R}_\nu(T_m)e_1 =: x_m^K. \tag{5.1.1}$$

The approximation (5.1.1) was obtained by combining the use of the partial fraction expansion of $\mathcal{R}_\nu$ with a Galerkin method to solve the resulting linear systems; however, Krylov methods could apply also when computing $\mathcal{R}_\nu(A)v$ by means of (3.1.2). If, in particular, the Galerkin method is used to solve these systems, then the orthogonality condition is

$$V_m^T(\mathcal{D}_\nu(A)V_m y_m - \mathcal{N}_\nu(A)v) = 0$$

and the approximation to $\exp(A)v$ will be

$$x_m^G := V_m y_m = V_m(V_m^T \mathcal{D}_\nu(A)V_m)^{-1}V_m^T \mathcal{N}_\nu(A)v.$$

In [42] the authors showed that the two approximations x_m^K and x_m^G tend to coalesce as convergence takes place.

5.2 Preconditioning linear systems

We restrict our analysis to the approximation to $\exp(A)v$ based on the partial fraction expansion of $\mathcal{R}_\nu$ and address the problem of preconditioning the

linear systems involved.
When using (3.1.4) it is necessary to solve linear shifted systems with coefficient matrices of the form $A - \xi_j I$ for different values of ξ_j. As discussed in Section 4.3, an effective way for accelerating the solution of linear systems is to *precondition* the systems. In our context one possible choice could be to select a different preconditioner for each pole, in such a way that each system is solved faster, with a consequent acceleration of the global procedure. However, to keep under control computational costs and memory requirements, we only consider using one preconditioner for all poles. Moreover, to preserve the shifted form of the coefficient matrix, we consider a preconditioner of the form

$$(A - \tau I)^{-1}$$

for a suitable parameter τ which in general is complex, or real positive. Then, in place of the linear system $(A - \xi_j I)x = v$ we solve the system

$$(A - \tau I)^{-1}(A - \xi_j I)x = (A - \tau I)^{-1}v. \tag{5.2.1}$$

At this point the crucial part of the approach is to select a parameter τ which makes the preconditioning effective for all poles.
Before moving in this direction we state one of the main results of our analysis: we show that if $\mathcal{R}_\nu$ is a rational approximation to exp then the shift and invert technique applied to $\mathcal{R}_\nu$ is equivalent to preconditioning all systems stemming from the partial fraction expansion of $\mathcal{R}_\nu$.

Proposition 5.2.1 ([55])**.** *Let $\mathcal{R}_\nu$ be a rational function with distinct poles and partial fraction expansion $\mathcal{R}_\nu(z) = \omega_0 + \sum_{j=1}^{\nu} \omega_j/(z - \xi_j)$. For a chosen $\sigma > 0$, let y_{SI} be the approximation to $\mathcal{R}_\nu(A)v$ obtained by applying the shift and invert approach with shift parameter σ.*
Let $y_{prec} = \omega_0 v + \sum_{j=1}^{\nu} \omega_j x_m^{(j)}$, where for each j, $x_m^{(j)}$ is the Galerkin approximation to $x^{(j)} = (A - \xi_j I)^{-1}v$ in $K_m((A - \frac{1}{\sigma}I)^{-1}(A - \xi_j I), v)$.
Then $y_{SI} = y_{prec}$.

Proof. Let V_m and T_m be defined by the relation

$$(I - \sigma A)^{-1}V_m = V_m T_m + t_{m+1,m} v_{m+1} e_m^T; \tag{5.2.2}$$

thus applying the shift and invert procedure for approximating $\mathcal{R}_\nu(A)v$ results in the relation

$$\mathcal{R}_\nu(A)v \approx y_{SI} := V_m \mathcal{R}_\nu\left(\frac{1}{\sigma}(I - T_m^{-1})\right)e_1.$$

Then, by resorting to the partial fraction expansion of $\mathcal{R}_\nu$, y_{SI} can be written as

$$y_{SI} \quad = \quad V_m \left(\omega_0 e_1 + \sum_{j=1}^{\nu} \omega_j \left(-\frac{1}{\sigma} T_m^{-1} + (\frac{1}{\sigma} - \xi_j) I \right)^{-1} e_1 \right). \quad (5.2.3)$$

On the other hand, y_{prec} is obtained as $y_{prec} = \omega_0 v + \sum_{j=1}^{\nu} \omega_j x_m^{(j)}$, where each $x_m^{(j)}$ approximates $x^{(j)} = (A - \xi_j I)^{-1} v$. For $j = 1, \ldots, \nu$, we multiply by $(A - \frac{1}{\sigma} I)^{-1}$ the system $(A - \xi_j I) x^{(j)} = v$ from the left getting

$$\left(A - \frac{1}{\sigma} I \right)^{-1} (A - \xi_j I) x^{(j)} = \left(A - \frac{1}{\sigma} I \right)^{-1} v; \qquad (5.2.4)$$

we solve it in the space $K_m((A - \frac{1}{\sigma} I)^{-1} (A - \xi_j I), v)$, even if the most natural approximation space would be $K_m(\left(A - \frac{1}{\sigma} I\right)^{-1} (A - \xi_j I), \left(A - \frac{1}{\sigma} I\right)^{-1} v))$. Thanks to the equality

$$\left(A - \frac{1}{\sigma} I \right)^{-1} \left(A - \xi_j I \right) = I + \left(\frac{1}{\sigma} - \xi_j \right) \left(A - \frac{1}{\sigma} I \right)^{-1} \qquad (5.2.5)$$

and for the shift and scaling invariance property of Krylov spaces, we get

$$K_m \left(\left(A - \frac{1}{\sigma} I \right)^{-1} \left(A - \xi_j I \right), v \right) = K_m \left(\left(A - \frac{1}{\sigma} I \right)^{-1}, v \right).$$

Moreover, relation (5.2.2) can be scaled as

$$\left(A - \frac{1}{\sigma} I \right)^{-1} V_m = -V_m \sigma T_m - \sigma t_{m+1,m} v_{m+1} e_m^T. \qquad (5.2.6)$$

Therefore, let $x^{(j)} \approx x_m^{(j)} \in K_m((A - \frac{1}{\sigma} I)^{-1}, v)$ with $x_m^{(j)} = V_m z_m^{(j)}$. Imposing the Galerkin condition on the residual vector and using (5.2.5) yield the equation

$$V_m^T \left(I + \left(\frac{1}{\sigma} - \xi_j \right) \left(A - \frac{1}{\sigma} I \right)^{-1} \right) V_m z_m^{(j)} = V_m^T \left(A - \frac{1}{\sigma} I \right)^{-1} V_m e_1.$$

Taking into account (5.2.6), we get for $z_m^{(j)}$ the system

$$\left(I - \left(\frac{1}{\sigma} - \xi_j \right) \sigma T_m \right) z_m^{(j)} = -\sigma T_m e_1,$$

or, equivalently, $(-\frac{1}{\sigma}T_m^{-1}+(\frac{1}{\sigma}-\xi_j)I)z_m^{(j)}=e_1$. We have thus shown that

$$y_{prec} \;=\; V_m\left(\omega_0 e_1+\sum_{j=1}^{\nu}\omega_j\left(-\frac{1}{\sigma}T_m^{-1}+(\frac{1}{\sigma}-\xi_j)I\right)^{-1}e_1\right),$$

which is the same as (5.2.3). □

The previous result shows that, when applied to a rational function, the shift and invert technique may be viewed as a particular way of preconditioning the linear systems stemming from the partial fraction expansion.
An additional relevant consequence of the previous result is that we may look at the shift parameter as the reciprocal of the parameter defining the preconditioner. We will exploit this view point to select a good shift parameter in the framework of preconditioning, resulting in a larger amount of useful informations than what is available in the context of spectral transformations.

5.3 Selecting the acceleration parameter

In this section we address the problem of selecting an effective preconditioning parameter $\tau := 1/\sigma$ for systems (5.2.1).
For simplicity we omit the index for the poles and for the solution to the corresponding shifted linear systems.

We start by analyzing spectral information about the matrices occurring in the approach; the preconditioned system corresponding to a pole ξ is

$$\left(I+(\tau-\xi)(A-\tau I)^{-1}\right)x=(A-\tau I)^{-1}v; \tag{5.3.1}$$

if λ is an eigenvalue of A then an eigenvalue of the coefficient matrix in (5.3.1) may be written as $\hat{\lambda}=1+(\tau-\xi)/(\lambda-\tau)$, implying that the whole spectrum lies on a curve of the complex plane.
The special case $\tau=\xi$ would reduce the system corresponding to ξ to the unscaled one, for which the selection of the shift parameter is meaningless; we may then assume $\tau\neq\xi$. Dividing the system (5.3.1) by $(\tau-\xi)$ yields

$$\left((A-\tau I)^{-1}-\chi I\right)x=\tilde{v}, \quad \text{with} \quad \chi=\frac{1}{\xi-\tau}, \tag{5.3.2}$$

and $\tilde{v}$ defined accordingly. The coefficient matrix in (5.3.2) is given by a real negative definite symmetric matrix shifted by a complex multiple of the identity. When Krylov methods are applied to systems having this kind of coefficient matrix their performance may be fully characterized by using spectral informations of the coefficient matrix, as shown in [25] and [42]. Before stating these bounds we notice that eigenvalues of the coefficient matrix in (5.3.2) lie on the horizontal line

$$y = \frac{\Im(\xi)}{|\tau - \xi|^2}, \quad \text{with} \quad x \in \left[-\frac{1}{\tau} + \frac{\tau - \Re(\xi)}{|\tau - \xi|^2}, \frac{1}{\alpha - \tau} + \frac{\tau - \Re(\xi)}{|\tau - \xi|^2} \right].$$

Proposition 5.3.1 ([42, Lemma 5.2]). *Given the linear system $(\widetilde{A} - \chi I)x = \tilde{v}$ with $\widetilde{A}$ symmetric and semidefinite and $\chi \in \mathbb{C}$, let x_m be the Galerkin approximate solution to x in $K_m(\widetilde{A}, \tilde{v})$. Let $\lambda_{\max}, \lambda_{\min}$ be the largest and smallest eigenvalues of $\widetilde{A} - \Re(\chi)I$ in absolute value, respectively. Then the error satisfies*

$$||x - x_m|| < g(\lambda_{\min}, \lambda_{\max}, \tilde{v}, \chi) \ \frac{1}{\rho^m + 1/\rho^m}$$

where g is a function of the spectrum of $\widetilde{A}, \tilde{v}$ and of χ only, while $\rho = \gamma + \sqrt{\gamma^2 - 1}$ and

$$\gamma = \frac{|\lambda_{\min} - i\Im(\chi)| + |\lambda_{\max} - i\Im(\chi)|}{|\lambda_{\min} - \lambda_{\max}|}.$$

The previous result shows that the decay rate of the error in $K_m(\widetilde{A}, \tilde{v})$ varies inversely with the quantity γ and then the larger γ, the smaller the subspace dimension m, that is, the faster the convergence; this suggests us that γ is the quantity to "manipulate" for accelerating the convergence. This of course makes sense in the case in which the problem depends upon a parameter we *may* vary, as in the case of shift and invert.

In our context, we can apply the result above both to the original partial fraction expansion approximation, having coefficient matrix $A - \xi_j I$, as well as to the preconditioned system (5.3.1). In the former case, setting $\widetilde{A} = A$ and $\chi = \xi$, we obtain

$$\gamma(\xi) = \frac{|\alpha - \xi| + |\xi|}{-\alpha}. \tag{5.3.3}$$

Clearly $\gamma(\xi) \geq 1$ and $\gamma \approx 1$ when $|\alpha| \gg |\xi|$; then when $|\alpha|$ is very large, as we are assuming, the error bound predicts very slow convergence of the linear system.

In the preconditioned case, setting $\widetilde{A} = (A - \frac{1}{\sigma}I)^{-1}$ and $\chi = 1/(\xi - \tau)$, after simple algebraic manipulations we get

$$\gamma^{prec}(\xi, \tau) = \frac{(\tau - \alpha)|\xi| + \tau|\alpha - \xi|}{-\alpha|\tau - \xi|}. \tag{5.3.4}$$

This suggests, as criterion for selecting τ, making $\gamma^{prec}(\xi, \tau)$ much larger than $\gamma(\xi)$, to accelerate the decay rate of the errors, and then the convergence of the procedure.

An ideal value τ should accelerate the convergence of (5.2.4) for any pole; this requirement would be attained if $\gamma^{prec}(\xi_j, \tau) > \gamma(\xi_j)$ for any j or, by using a simpler but more stringent condition, if

$$\min_{\xi} \gamma^{prec}(\xi, \tau) \geq \max_{\xi} \gamma(\xi);$$

unfortunately, this inequality turns out to be hard to analyze and another criterion needs to be detected.

The two parameters $\gamma^{prec}(\xi, \tau)$ and $\gamma(\xi)$ may be related in the following way

$$\gamma^{prec}(\xi, \tau) = F(\alpha, \xi, \tau)\gamma(\xi) \tag{5.3.5}$$

where

$$F(\alpha, \xi, \tau) = \frac{\tau}{|\tau - \xi|} - \frac{\alpha|\xi|}{(|\alpha - \xi| + |\xi|)|\tau - \xi|} = \frac{\tau - c}{|\tau - \xi|}, \tag{5.3.6}$$

with $c = \alpha|\xi|/(|\alpha - \xi| + |\xi|)$. Acceleration on the convergence would then be attained by choosing the parameter τ which maximizes the function $F(\alpha, \xi, \tau)$. In the following proposition we analyze this function, stating relations useful for our analysis. Also the value τ attaining the maximum for $F(\alpha, \xi, \tau)$ is characterized.

Proposition 5.3.2 ([55]). *Given α and ξ, let $F(\tau) = F(\alpha, \xi, \tau)$ be defined in (5.3.6) and assume that $\Re(\xi) > \alpha/2$ and $\Im(\xi) \neq 0$. Then*

i) $F(\tau) \geq 1$ for $\tau \geq \tau_0$ with $\tau_0 = \frac{1}{2}\frac{|\xi|^2 - c^2}{\Re(\xi) - c}$ and $\Re(\xi) > c$;

ii) $F(\tau_{\max}) \geq F(\tau)$, for every τ, where

$$\tau_{\max} = \tau_{\max}(\xi) = \frac{\Re(\xi)c - |\xi|^2}{c - \Re(\xi)} \quad \textit{and} \quad F(\tau_{\max}) = \frac{|c - \xi|}{|\Im(\xi)|} \geq 1;$$

iii) $\gamma^{prec}(\xi, \tau_0) = \gamma(\xi)$ *and* $\lim_{\tau \to \infty} \gamma^{prec}(\xi, \tau) = \gamma(\xi)$.

Proof. Let $\xi = \xi_R + \imath \xi_I$. We first show that $\xi_R > c$. Since $c < 0$ then clearly $\xi_R > c$ when $\xi_R \geq 0$. For $\xi_R < 0$, using $\alpha < 2\xi_R$ we obtain $\alpha|\xi| < 2\xi_R|\xi| \leq \xi_R|\xi| \leq \xi_R(|\xi| + |\alpha - \xi|)$, from which

$$\xi_R > \frac{\alpha|\xi|}{|\xi| + |\alpha - \xi|} = c.$$

To prove (i), we observe that

$$F(\tau) \geq 1 \qquad \Leftrightarrow \qquad 2(\xi_R - c)\tau \geq |\xi|^2 - c^2. \tag{5.3.7}$$

Using $\xi_R > c$, the previous requirement corresponds to imposing $\tau \geq \tau_0$.

To prove (ii) we explicitly write

$$F'(\tau) = -\frac{(\tau - \xi_R)}{|\tau - \xi|^3}(\tau - c) + \frac{1}{|\tau - \xi|} = 0 \quad \Leftrightarrow \quad -(\tau - \xi_R)(\tau - c) + |\tau - \xi|^2 = 0$$

from which the expression for $\tau_{\max}$ follows. Moreover, F is an increasing function for $\tau \leq \tau_{\max}$ and a decreasing one otherwise, so that $F(\tau_{\max})$ is a maximum.

To prove that $F(\tau_{\max}) \geq 1$ we notice that $F(\tau_{\max})^2 = 1 + (c - \xi_R)^2/\xi_I^2$, from which we obtain that $(F(\tau_{\max}) - 1)(F(\tau_{\max}) + 1) = \frac{(c-\xi_R)^2}{\xi_I^2}$. The result follows by taking into account that

$$F(\tau_{\max}) + 1 = \frac{|\xi_I| + |c - \xi|}{|\xi_I|}.$$

Finally, the first equality in (iii) follows from $F(\tau_0) = 1$ in (5.3.7), while it can be readily verified that $\lim_{\tau \to \infty} F(\tau) = 1$. □

We add some remarks to the results of the previous proposition: first of all we observe that the hypotheses are fulfilled in our situation since we are assuming $|\alpha|$ large, say $|\alpha| \gg |\xi|$, which in particular guarantees that $\Re(\xi) > \alpha/2$ and implies that $\tau_{max} \approx |\xi|$.

The point 5.3.2(ii) offers the explicit expression for the optimal value τ_{max} ensuring the best acceleration for the system corresponding to ξ. In view of 5.3.2(iii) we may restrict the choice of the parameter τ to the interval $[\tau_0, \infty[$, avoiding values of the parameter that are too close to the extremes since they do not improve the convergence.

Proposition 5.3.2 offers a good theoretical analysis of the ratio $\frac{\gamma^{prec}(\xi,\tau)}{\gamma(\xi)}$, although of little use for our purposes; indeed it results in a parameter τ_{max} whose effectiveness is restricted to the solution of the system corresponding to the fixed pole whereas we are interested in a value of τ which accelerates the convergence of (5.2.4) for any j; moreover the parameter τ_{max} depends on the spectrum of A, through the presence of α.
In the following we determine a condition on τ which is independent of A and guarantees the acceleration of the general procedure.

In expression (5.3.5) we may exploit our initial assumption $|\alpha| \gg |\xi|$ allowing the approximations $\gamma(\xi) \approx 1$ and $c \approx -|\xi|$; the final result is

$$\gamma^{prec}(\tau,\xi) = \frac{\tau}{|\tau-\xi|}\gamma(\xi) + \frac{|\xi|}{|\tau-\xi|} \geq \frac{\tau}{|\tau-\xi|} + \frac{|\xi|}{|\tau-\xi|} =: \mathcal{H}(\tau,\xi). \quad (5.3.8)$$

The quantity $\mathcal{H}(\tau,\xi) = \frac{\tau+|\xi|}{|\tau-\xi|} \geq 1$ also represents an upper bound for $F(\xi,\tau)$ since $-c \leq |\xi|$. Interestingly, the quantity $\mathcal{H}(\tau,\xi)$ has all features we were looking for: its maximum corresponds to a lower bound for $\gamma^{prec}(\tau,\xi)$, its expression does not depend on any eigenvalue of A and maximizing it is a simple task: we now proceed with detecting its maximum in the following lemma.

Lemma 5.3.3 ([55]). *Let $\mathcal{H}(\tau,\xi)$ be defined in (5.3.8). Then*

$$\mathcal{H}(|\xi_i|,\xi_i) = \max_{\tau>0} \mathcal{H}(\tau,\xi_i).$$

Proof. We have

$$\begin{aligned}\mathcal{H}(\tau,\xi)^2 = \ & \left(\tfrac{\tau+|\xi|}{|\tau-\xi|}\right)^2 = \tfrac{\tau^2+2\tau|\xi|+|\xi|^2}{|\tau-\xi|^2} \\ = \ & \tfrac{|\tau-\xi|^2-2\tau\xi_R+2\tau|\xi|}{|\tau-\xi|^2} = 1 + 2(|\xi| + \Re(\xi))\tfrac{\tau}{|\tau-\xi|^2}.\end{aligned}$$

Since we assume that $\tau > 0$, the value $\tau = |\xi|$ is the only critical point for $\tau/|\tau-\xi|^2$, its derivative being $\tau^2 - |\xi|^2$. Moreover $\tau = |\xi|$ represents a maximum for $\tau/|\tau-\xi|^2$ and consequently for $\mathcal{H}(\tau,\xi)^2$. Then

$$\mathcal{H}(\tau,\xi)^2 \leq 1 + 2(|\xi| + \Re(\xi))\frac{|\xi|}{||\xi|-\xi|^2} = \frac{4|\xi|^2}{||\xi|-\xi|^2} = \mathcal{H}(|\xi_i|,\xi_i)^2$$

from which the result follows. □

From the previous result we may conclude that, as resulting from Proposition 5.3.2, the value $\tau_i := |\xi_i|$ defines an effective preconditioner for the system corresponding to the i-th pole; however our practical task is selecting just one preconditioning parameter, among $\tau_1, \dots, \tau_\nu$, which improves the convergence of all systems and this condition may be expressed in terms of the following *max-min* problem

$$\max_{\tau_1,\dots,\tau_\nu} \min_{\xi_1,\dots,\xi_\nu} \mathcal{H}(\tau_i, \xi_j). \tag{5.3.9}$$

The formulation (5.3.9) has several nice features: it is completely independent of the matrix A; it depends only on the degree and on the poles of the chosen rational function; moreover we do not need to solve it analytically, since we are interested only in a finite, usually small, number of terms. For its practical use, once the rational function has been chosen and its poles are available, we can explicitly list all values of $\mathcal{H}(\tau_i, \xi_j)$ and decide once for all the optimal value of τ we can pick up. Moreover, we stress that the function $\mathcal{H}$ assumes the same value on complex conjugate numbers; thus the number of terms in (5.3.9) is halved, since we are assuming A to be real and then the poles come in complex conjugate pairs.

We consider the Chebyshev function $\mathcal{R}_{14}$, described in Section 3.3, with the poles as listed in [14]. The degree 14 is one of the largest used in practice since, thanks to relation (3.3.1), it corresponds to an approximation with accuracy 10^{-14}; however we have selected this value to show that also for delicate problems the technique described in the previous section is quite simple.

$\tau_i = \|\xi_i\|$	ξ_1	ξ_3	ξ_5	ξ_7	ξ_9	ξ_{11}	ξ_{13}
18.8616	1.1657	1.2516	1.3564	1.4831	1.6260	1.7628	1.8515
14.1496	1.1615	1.2590	1.3905	1.5708	1.8105	2.0910	2.3111
10.9932	1.1515	1.2533	1.4010	1.6254	1.9739	2.4925	3.0433
8.7609	1.1387	1.2391	1.3924	1.6430	2.0832	2.9193	4.3218
7.2115	1.1261	1.2219	1.3727	1.6300	2.1170	3.2233	6.5221
6.2274	1.1160	1.2068	1.3520	1.6045	2.0975	3.3081	8.9488
5.7485	1.1105	1.1981	1.3391	1.5859	2.0716	3.2821	9.5758

Table 5.1: Values of $\mathcal{H}(\tau_i, \xi_j)$, $i, j = 1, \dots \nu$, for Chebyshev with $\nu = 14$.

In Table 5.1 we list the data $\mathcal{H}(\tau_i, \xi_j)$, with the poles sorted with decreasing imaginary parts. The optimal value of τ for problem (5.3.9) is given by $\tau_1 =$

18.8616, ensuring that $\gamma^{prec}(\xi,\tau) \geq \mathcal{H}(\tau_1,\xi_1) = 1.1657$.
In several numerical examples it turned out that for all degrees the best value of τ is always associated with ξ_1. Therefore we propose to use the parameter

$$\tau_{\text{opt}} := |\xi_1| \quad \Leftrightarrow \quad \sigma_{\text{opt}} = \frac{1}{|\xi_1|}. \tag{5.3.10}$$

The corresponding values associated with Chebyshev rational poles are listed in Table 5.2 for $\nu \leq 20$. The entries in the table can be used as follows: if a final tolerance *tol* for the approximation of $\exp(A)v$ is requested, then the shift and invert approach may be used with a shift value corresponding to $\nu \geq -\log_{10}(tol)$ (e.g., tol=10^{-8} yields $\nu \geq 8$ so that $\sigma = 0.1062$ or a smaller value in the table may be used).

ν	1	2	3	4	5	6	7	8	9	10
σ_{opt}	1.7271	0.7565	0.4134	0.2720	0.1988	0.1551	0.1264	0.1062	0.0914	0.0801
ν	11	12	13	14	15	16	17	18	19	20
σ_{opt}	0.0711	0.0639	0.0580	0.0530	0.0488	0.0452	0.0421	0.0394	0.0369	0.0348

Table 5.2: Optimal values of the parameter, cf. (5.3.10), for various rational function degrees.

For comparison purposes we report here the shift parameters derived by van den Eshof and Hochbruck in [68], with a completely different strategy, see Section 2.4.

m	$E_m^m(\sigma_{opt})$	σ_{opt}	m	$E_m^m(\sigma_{opt})$	σ_{opt}
1	$6.7 \cdot 10^{-2}$	$1.73 \cdot 10^{0}$	11	$4.0 \cdot 10^{-6}$	$9.90 \cdot 10^{-2}$
2	$2.0 \cdot 10^{-2}$	$4.93 \cdot 10^{-1}$	12	$1.6 \cdot 10^{-6}$	$1.19 \cdot 10^{-1}$
3	$7.3 \cdot 10^{-3}$	$2.64 \cdot 10^{-1}$	13	$6.1 \cdot 10^{-7}$	$1.00 \cdot 10^{-1}$
4	$3.1 \cdot 10^{-3}$	$1.75 \cdot 10^{-1}$	14	$2.5 \cdot 10^{-7}$	$8.64 \cdot 10^{-2}$
5	$1.4 \cdot 10^{-3}$	$1.30 \cdot 10^{-1}$	15	$1.0 \cdot 10^{-7}$	$7.54 \cdot 10^{-2}$
6	$4.0 \cdot 10^{-4}$	$1.91 \cdot 10^{-1}$	16	$4.0 \cdot 10^{-8}$	$8.67 \cdot 10^{-2}$
7	$1.6 \cdot 10^{-4}$	$1.44 \cdot 10^{-1}$	17	$1.6 \cdot 10^{-8}$	$7.63 \cdot 10^{-2}$
8	$6.5 \cdot 10^{-5}$	$1.90 \cdot 10^{-1}$	18	$6.6 \cdot 10^{-9}$	$6.78 \cdot 10^{-2}$
9	$2.4 \cdot 10^{-5}$	$1.47 \cdot 10^{-1}$	19	$2.7 \cdot 10^{-9}$	$7.62 \cdot 10^{-2}$
10	$9.7 \cdot 10^{-6}$	$1.19 \cdot 10^{-1}$	20	$1.1 \cdot 10^{-9}$	$6.82 \cdot 10^{-2}$

Table 5.3: The tabulated values in [68] of the shift and invert parameter. m is the number of shift and invert Lanczos iterations.

In [68] the authors suggested to detect the shift parameter from Table 5.3 which corresponds to the desired accuracy, i.e. to $E_m^m(\sigma_{opt})$; thus, if one is interested in the accuracy 10^{-5} then he may arbitrarily choose among 0.19 and 0.147.

The values of Tables 5.2 and 5.3 are similar, even if we determined them in a fully algebraic way while in [68] numerical procedures were applied for their selection and there is an explicit dependence on the number of shift and invert Lanczos iterations. However, this similarity may be viewed as an additional motivation for the reliability of our approach.

5.3.1 Asymptotic behavior

In selecting the optimal shift parameter we avoided the use of spectral information on the matrix A; the reason for this was the desire of offering shift parameters tabulated once for all, depending only on the rational approximation to the exponential considered. However, if with this strategy we obtained a parameter selection of immediate application, on the other side we lost all additional information that would improve the strategy. We refer in particular to spectral information on A that, in some cases, may be useful for recovering the *superlinear* convergence, as described in Section 2.2. In practice the only information we used was the extrema eigenvalues of A; in this sense our analysis was based on asymptotic arguments and not on the actual eigenvalue distribution. For better describing this “loss” of information we consider the problem of approximating the exponential of two matrices having the same extremal eigenvalues but with different eigenvalue distributions: we consider the matrix $\widetilde{A}$ of size $n = 3375$ stemming from the discretization of the 3D Laplace operator, whose extreme eigenvalues are $\lambda_{\min} \approx -2329.4$ and $\lambda_{\max} \approx -22.597$; we define the singular matrix $A = \widetilde{A} - \lambda_{\max} I$ and we consider the diagonal matrix D with nonzero entries *uniformly distributed* in the same spectral interval as A. The vector v is taken as a normalized vector of all ones. We study the performance of the accelerated process with the optimal parameter $\sigma_{\text{opt}} = 0.053$ and with another possible candidate, $\sigma_{\min} = 1/\max_j |\Re(\xi_j)| = 0.1124$, taken for $\nu = 14$ poles. In the left plot of Figure 5.1 the convergence histories are plotted for both matrices and both shift parameters. For the matrix A the shift and invert method with shift parameter σ_{min} is faster than that corresponding to σ_{opt}. For the matrix D the fastest convergence corresponds to σ_{opt}, confirming the theory.

In the same plot also the asymptotic estimates $\mathcal{H}(1/\sigma, \xi_1)^j$, $j = 1, \dots, m$ are reported for $\sigma = \sigma_{\text{opt}}$ (filled squares) and $\sigma = \sigma_{\min}$ (circles); both curves well represent the initial convergence phase of the shift-invert procedure with D, with a slightly better performance for $\mathcal{H}(1/\sigma_{\text{opt}}, \xi_1)$.

In the right plot of Figure 5.1 we report a graphical representation of how the shift parameter influences the performance of the whole procedure. We consider D as before and plot the number of iterations required by the shift and invert method for reaching the accuracy of 10^{-14} when the shift parameter varies in $[10^{-2}, 10^{4}]$. The symbol "*" refers to the choice $\sigma = \sigma_{\rm opt}$. The total number of iterations does not grow up sensibly for $\sigma > \sigma_{\rm opt}$ while it reaches very large values for $\sigma < \sigma_{\rm opt}$. It is interesting to notice that in [68, section 6] the value $\sigma = 0.01$ was often used in practical situations but it corresponds, in our plot, to a much larger number of iterations than that corresponding to $\sigma_{\rm opt}$.

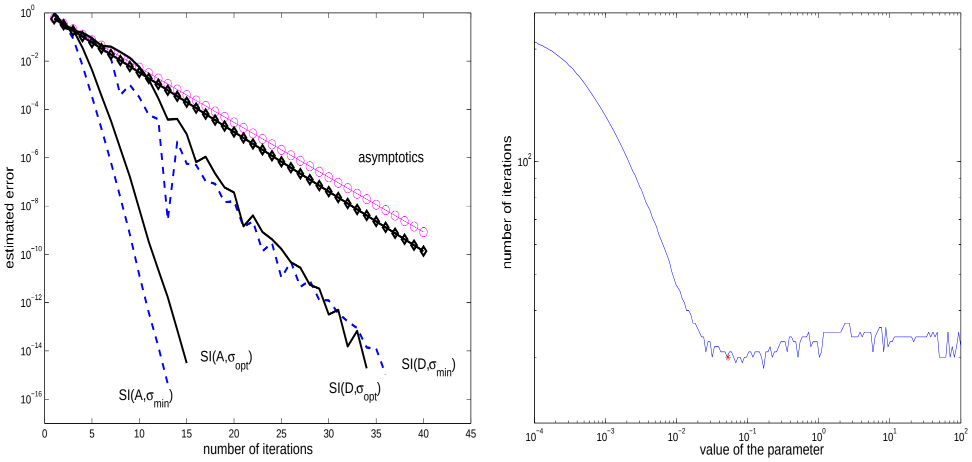


Figure 5.1: Left: Convergence history of shift and invert Lanczos for a matrix A stemming from a shifted Laplace operator, and for a diagonal matrix D with uniformly distributed eigenvalues in spec(A). Here $\sigma_{\rm opt} = 0.053$ and $\sigma_{\rm min} = 0.1124$. Reported are also the asymptotic values $\mathcal{H}(1/\sigma, \xi_1)^j$, $j = 1, \ldots, m$, for $\sigma_{\rm min}$ (circles) and $\sigma_{\rm opt}$ (filled squares). Right: Number of iterations of shift and invert Lanczos applied to the diagonal matrix D versus value of the shift σ; the symbol "*" refers to the choice $\sigma = \sigma_{\rm opt}$.

5.4 Real valued method

In this section we describe the method proposed by Axelsson and Kucherov in [5] to precondition complex systems by only using real arithmetic; by applying this approach to the approximation of $\exp(A)v$ an acceleration technique derives.

Assume we need to solve the complex system

$$(R + \imath S)u = b, \tag{5.4.1}$$

with $R = R^T,\ S = S^T,\ u = u_R + \imath\, u_I$ and $b = b_R + \imath\, b_I$; then fix a parameter $\eta > 0$ and consider the equivalent real form

$$\begin{pmatrix} R - \eta S & \sqrt{1+\eta^2} S \\ \sqrt{1+\eta^2} S & -R - \eta S \end{pmatrix} \begin{pmatrix} u_R \\ z \end{pmatrix} = \begin{pmatrix} b_R \\ (b_I - \eta b_R)/(\sqrt{1+\eta^2}) \end{pmatrix},$$

where $z = (\eta u_R - u_I)/\sqrt{1+\eta^2}$. If $R + \eta S$ is nonsingular then, by applying the Schur complement reduction, the real part of the solution, that is u_R, may be recovered by solving the real symmetric system

$$C u_R = w, \tag{5.4.2}$$

with $C = R - \eta S + (1+\eta^2) S (R + \eta S)^{-1} S$ and $w = b_R + S(R + \eta S)^{-1}(b_I - \eta b_R)$. The imaginary part, u_I, will be characterized as the solution to the system

$$R\, u_R - S\, u_I = b_R. \tag{5.4.3}$$

The most costly part of this technique is the solution of (5.4.2) and in [5] the key idea is to precondition it through the matrix $B = R + \eta S$, by getting the equivalent system

$$M u_R = \tilde{w} \tag{5.4.4}$$

with $M = B^{-1}C$ and $\tilde{w}$ defined accordingly. The matrix M is shown to be symmetric positive definite in [5, Remark 1]; nevertheless at this point the parameter η becomes fundamental for guaranteeing that the matrix M in (5.4.4) is well conditioned so that (5.4.4) is easily solvable.
We report the result in [5] which detects the optimal value of η when R and S have certain properties:

Theorem 5.4.1 ([5, Theorem 2.1])**.** *With the notation above, assume that R is symmetric positive definite and S is symmetric positive semidefinite; then the condition number of M is minimized when*

$$\eta = \frac{\hat{\lambda}}{1 + \sqrt{1 + \hat{\lambda}^2}}$$

and $\hat{\lambda}$ is the maximal eigenvalue of $R^{-1}S$.

5.4.1 Real valued method and the partial fraction expansion

In our context we face complex systems when computing the partial fraction expansion of a rational approximation to the exponential. Indeed, if $\mathcal{R}_\nu$ is such a function, the evaluation of $\mathcal{R}_\nu(A)v$ by means of the expansion (3.1.4) entails solving complex systems. The crucial aspect is that they all have the shifted symmetric form $(A-\xi I)u = v$ for a fixed pole $\xi = \xi_R + \imath\xi_I$. Rewriting our problem in the notation of the method of [5] leads to $R = A - \xi_R I$ and $S = -\xi_I I$. Substituting in the coefficient matrix of the system (5.4.2) we obtain

$$C = -B + 2\eta\,\xi_I\,I - (1+\eta^2)\xi_I^2 B^{-1},$$

where the preconditioner becomes $B = -(R+\eta S) = (\xi_R + \eta\,\xi_I)I - A$; the resulting preconditioned system reads

$$Mu_R = B^{-1}w, \tag{5.4.5}$$

with $M = B^{-1}C = -I + 2\eta\xi_I B^{-1} - \xi_I^2(1+\eta^2)B^{-2}$ and $w = v - \eta\,\xi_I B^{-1}v$.

Moreover for the imaginary part of the solution the system (5.4.3) simplifies into the expression $u_I = \frac{1}{\xi_I}(-A + \xi_R I)u_R + \frac{1}{\xi_I}v$, which only involves matrix-vector multiplications.

The crux of the procedure is then represented by the system (5.4.5) for which iterative methods are necessary for the presence of B^{-1}. However it is real and we will show in the next proposition that the matrix $-M$ is symmetric positive definite for any choice of $\eta > 0$ and for all poles, and thus the Conjugate Gradient method can be used.

Proposition 5.4.2 ([55]). *Let u_R be the solution to $Mu_R = B^{-1}w$ (cf. (5.4.5)) and for $\tau = \xi_R + \eta\,\xi_I$, $\eta > 0$, consider the (preconditioned) linear system*

$$(\tau I - A)^{-1}(A - \xi I)u = (\tau I - A)^{-1}v, \tag{5.4.6}$$

and set $K = (\tau I - A)^{-1}(A - \xi I)$. Then $M = -K^*K \in \mathbb{R}^{n\times n}$. Moreover, u_R is the real part of the solution of $K^*Ku = K^*(\tau I - A)^{-1}v$.

Proof. Let $R = A - \xi_R I$ and $S = -\xi_I I$, and note that R and S commute, so that

$$\begin{aligned} K^*K &= (R+\eta S)^{-2}(R - \imath S)(R + \imath S) = (R+\eta S)^{-2}(R^2 + S^2) \\ &= I + 2\eta\xi_I(R+\eta S)^{-1} + \xi_I^2(1+\eta^2)(R+\eta S)^{-2} = -M. \end{aligned}$$

Therefore, K^*K is real symmetric and M is negative definite. Analogously, we can write $K^*(\tau I - A)^{-1}v = (R+\eta S)^{-2}(R - \imath S)v$ whose real part is given by $\Re(K^*(\tau I - A)^{-1}v) = (R + \eta S)^{-2}Rv = (R + \eta S)^{-1}w$. Therefore, the real part of the equation $K^*Ku = K^*(\tau I - A)^{-1}v$ is given by $-M\Re(u) = -B^{-1}w$, from which it follows that $u_R = \Re(u)$. □

The connection among (5.4.5) and (5.4.6) established in the above proposition is very important; indeed it shows that (5.4.5) is nothing but the normal equation of (5.4.6) for a special choice of the acceleration parameter. The matter of importance is, at this point, the fact that also (5.4.5) is equivalent to the preconditioned system (5.2.1) described in the context of accelerating the computation of the partial fraction expansion of $\mathcal{R}_\nu(A)$ by means of Krylov methods. The previous result thus allows looking at the method proposed in [5] in the framework of preconditioning the linear systems stemming from the partial fraction expansion of a certain rational function.

5.4.2 Selecting the acceleration parameter

Theorem 5.4.1 taken from [5] suggests the optimal value η to consider to reach the fastest convergence for (5.4.5); however this theorem only applies when the matrix R is symmetric positive definite and S is symmetric positive semidefinite. In our case $R = A - \xi_R I$ which in general is not definite. Moreover Theorem 5.4.1 would detect the optimal value η corresponding to a fixed pole ξ; rather our aim is to define just one preconditioner to be applied to all systems to save computational costs and memory storage, since the construction of a preconditioner is generally expensive; on the other hand this single preconditioner must be effective for each system to be solved. In this section we devise a parameter which satisfies these conditions.

To fulfill our purposes we define $B = \tau I - A$ as the single preconditioner to be applied to $(A - \xi_j I)x = v$ for any pole ξ_j; moreover, to take into account the features of each system, we choose $\tau = \Re(\xi_j) + \eta_j \Im(\xi_j)$, thus allowing η_j to change accordingly to each pole. We will show that in this way we reach our goal of defining a single preconditioner and reaching a good global acceleration.

By allowing η to change also the matrix M in (5.4.5) will change. Next proposition provides sharp bounds for the condition number of M that do not depend on the spectrum of A nor require further hypotheses on $A - \xi I$; these bounds depend upon τ and then, by minimizing them, we will reach the optimal τ.

Proposition 5.4.3 ([55])**.** *Assume that the hypotheses of Proposition 5.4.2 hold and that $\tau > \max\{0, \xi_R\}$. Then*

$$\kappa(M) \leq \max\left\{\frac{|\xi|^2}{\tau^2}, \frac{|\alpha-\xi|^2}{(\alpha-\tau)^2}\right\}\frac{|\tau-\xi|^2}{\xi_I^2}. \tag{5.4.7}$$

Moreover, if it also holds that $\tau \leq |\xi|$, then $\frac{|\alpha-\xi|^2}{(\alpha-\tau)^2} \leq \frac{|\xi|^2}{\tau^2}$ and hence

$$\kappa(M) \leq \frac{|\xi|^2}{\xi_I^2}\frac{|\tau-\xi|^2}{\tau^2}. \tag{5.4.8}$$

Proof. Writing $-M = (R+\eta S)^{-2}(R^2+S^2) = (R-\eta\xi_I I)^{-2}(R^2+\xi_I^2 I)$ we get

$$\operatorname{spec}(-M) = \left\{\frac{(\lambda-\xi_R)^2+\xi_I^2}{(\lambda-\tau)^2}\middle|\lambda \in \operatorname{spec}(A)\right\}.$$

For $\lambda \in [\alpha, 0]$, let $\mu \in \operatorname{spec}(-M)$, $\mu = g(\lambda) = \frac{\lambda^2 - 2\lambda\xi_R + |\xi|^2}{(\lambda-\tau)^2}$. We have

$$g'(\lambda) = 2\frac{\lambda(\xi_R-\tau)+\tau\xi_R-|\xi|^2}{(\lambda-\tau)^3} = 0 \quad \Leftrightarrow \quad \hat{\lambda} := \frac{\tau\xi_R-|\xi|^2}{\tau-\xi_R}.$$

Since $\tau > \xi_R$, it holds that $g'(\lambda) > 0$ only for $\lambda > \hat{\lambda}$, hence

$$g(\hat{\lambda}) = \frac{\xi_I^2}{|\tau-\xi|^2} \leq \mu \quad \forall \mu \in \operatorname{spec}(-M). \tag{5.4.9}$$

To derive an upper bound, we notice that since $\hat{\lambda}$ is the only critical point and it is associated with a minimum, the maximum of g in $[\alpha, 0]$ is given by $\max\{g(\alpha), g(0)\}$. Collecting this bound and (5.4.9), the bound (5.4.7) for $\kappa(M)$ follows.

We next assume that $\tau \leq |\xi|$ holds for all poles ξ. We write

$$g(\alpha) - g(0) = \frac{\alpha^2(\tau^2-|\xi|^2) - 2\alpha\tau(\xi_R\tau - |\xi|^2)}{\tau^2(\tau-\alpha)^2}.$$

For $\tau \leq |\xi|$ the first addend in the numerator of the last expression is negative. For the second addend, we separately treat the cases of positive and negative pole's real part. If $\xi_R < 0$, the second addend gives $-2\alpha\tau(\xi_R\tau - |\xi|^2) \leq 0$. If $\xi_R > 0$, we can get $-2\alpha\tau(\xi_R\tau - |\xi|^2) \leq -2\alpha\tau(\tau^2 - |\xi|^2) \leq 0$. We have thus shown that $g(\alpha) - g(0) \leq 0$, which completes the proof. □

The bound in (5.4.7) may be rather sharp, its sharpness depending on whether the extremes of the function g defined in the proof are attained.
For giving an illustration of the previous bound we consider the following example:

Example 5.4.4. We consider the Chebyshev rational form of degree $\nu = 14$ as approximation to the exponential; we choose A as the 125×125 matrix obtained by the discretization of the 3D Laplacian with homogeneous boundary conditions; the matrix is shifted so as to have zero largest eigenvalue. Table 5.4 reports the bound in (5.4.7) corresponding to these data, when only the poles with positive imaginary part are considered. We used $\tau_{\text{opt}} = \min_{j=1,\dots,\nu} |\xi_j| = 5.7485$; the reason of this choice will be clarified in the next.

ξ_j $(\nu = 14)$	$\kappa(M(\xi_j))$	estimate (5.4.7)
-8.8978 + 16.631i	19.115	19.115
-3.7033 + 13.656i	8.9609	8.9609
-0.2087 + 10.991	4.7174	4.7315
2.2698 + 8.4617i	2.7012	2.7152
3.9934 + 6.0048i	1.7001	1.7082
5.0893 + 3.5888i	1.2122	1.2132
5.6231 + 1.1941i	1.0097	1.0110

Table 5.4: Condition number of M and its upper bound, as the poles vary.

Our aim is now to consider a single parameter τ and to this purpose we will start from the bounds in (5.4.8). Let $W_\xi(\tau) = \frac{|\tau-\xi|^2}{\tau^2}$ be the part of the upper bound in (5.4.8) that depends on τ; then

$$W_\xi(\tau)' = \frac{2(\tau-\xi_R)\tau^2 - 2\tau((\tau-\xi_R)^2+\xi_I^2)}{\tau^4} = \frac{2}{\tau^3}(\tau\Re(\xi) - |\xi|^2)$$

so that

$$W_\xi(\tau)' = 0 \quad \Leftrightarrow \quad \tau_*(\xi) = \frac{|\xi|^2}{\Re(\xi)}.$$

If $\Re(\xi) < 0$, then $W_\xi(\tau_*)$ is a maximum and $\tau_*(\xi)$ is negative. We thus restrict our attention to the poles with positive real parts. Moreover, we observe that for $\tau > \tau_*$ and $\Re(\xi) < 0$, the function W_ξ is decreasing, so that the larger τ the smaller the bound for $\Re(\xi) < 0$. We then recall that for (5.4.8) to hold the selected parameter τ must satisfy

$$\Re(\xi) \leq \tau \leq |\xi|, \quad \forall \xi.$$

Let the poles be sorted as $\Re(\xi_1) \leq \cdots \leq \Re(\xi_\nu)$. Then $\tau_*(\xi_\nu) \geq \Re(\xi_j)$ for $j \leq \nu$ and we define

$$\tau_{\text{opt}} := \min \left\{ \min_{j=1,\ldots,\nu} |\xi_j|, \tau_*(\xi_\nu) \right\} \tag{5.4.10}$$

as the parameter to use for accelerating the convergence of this real valued procedure.

Remark 5.4.5. For the Chebyshev poles of Table 5.4 it holds that $\min_{j=1,\ldots,\nu} |\xi_j| = |\xi_\nu|$ so that

$$\tau_{\text{opt}} = |\xi_\nu|.$$

To have an idea of the effectiveness of our choice of τ we consider the following example:

Example 5.4.6. We consider the 125×125 matrix stemming from the finite difference discretization of the 3D Laplace operator on the unit cube and Dirichlet homogeneous boundary conditions scaled by a factor $t = 0.1$; v is a normalized normally distributed random vector; the Chebyshev rational function of degree $\nu = 8$ is considered. In Figure 5.2 we report the total number of Conjugate Gradient iterations required by the method to solve all systems $Mu_R = \hat{w}$ with accuracy 10^{-10} (see Algorithm AK in Section 6.2), for different values of the parameter $\tau \in [0, 7]$. The symbol "*" indicates the number of iterations for the choice $\tau = \tau_{\text{opt}}$, showing the high quality of the a-priori selected parameter.

The analysis above conforms with the multiple choice in [5], although in our case extremely fast convergence cannot be achieved for *all* shifted systems.

In Section 6.2 we will present the implementation details of this procedure and numerical tests comparing it with other approaches.

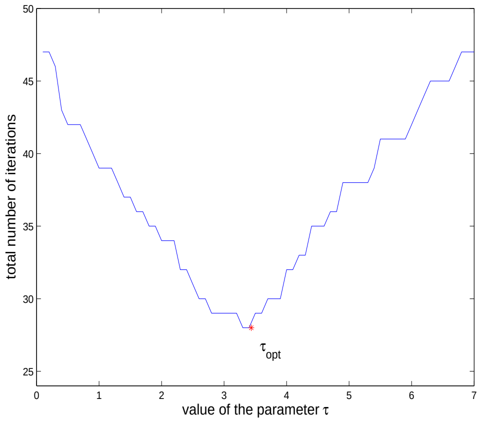


Figure 5.2: Total number of iterations for the variant of the Axelsson-Kucherov method applied to exp, as a function of the parameter τ. The symbol "*" refers to the number of iterations for $\tau = \tau_{\text{opt}}$.

Chapter 6

Numerical Experiments

In this chapter we describe the implementation of the methods described so far and we present their numerical performance:

- *Partial Fraction Expansion* (PFE). Computation of (3.1.5) by explicitly solving each complex shifted symmetric system;

- *Standard Lanczos.* Classical Lanczos approach described in Section 2.1;

- *AK.* Variant of method by Axelsson and Kucherov described in Section 4.3.1;

- *Shift-Invert Lanczos* (SI). Acceleration procedure described in Section 2.4;

- *Krylov Plus Inverted Krylov* (KPIK). Method described in Section 2.5.

We test these methods on the computation of vectors of the form $\exp(A)v$ with matrices A stemming from realistic applications; we compare their performance, by separately handling the case when all linear systems involved are solved by direct methods from the one in which iterative methods are used; we also discuss and test possible strategies for improving PFE and SI.
In the last part of this chapter we address the important task of integrating PDEs for which the computation of vectors of the form $\exp(A)v$ is crucial; to this purpose we compare some of the methods listed above with the Crank-Nicolson approach, which is largely used for problems of this kind. The results are interesting and very promising for PFE and SI.

Most of the numerical tests presented in this chapter have been published in [55].

For PFE and AK the degree ν of the chosen rational approximation determines the number of linear systems to be solved; moreover for all methods we set the final accuracy to $10^{-\nu}$. Thus the value ν plays an important role in the numerical examples, and we present tests with different values of ν, ranging from 4 to 13. Once fixed ν we selected the shift parameter as the value in Table 5.2 corresponding to it; however a single parameter could be used, for example the one corresponding to a large degree, say $\nu = 14$.
Interestingly, we observed that when implementing PFE, AK and SI the solution of linear systems required over 95% of the total computational efforts, representing the only bottleneck of these methods.

We focus on matrices which have eigenvalues in a large interval of the negative real axis to stress the need of accelerating; indeed, as described in Theorem 2.2.1, when the spectrum is large the convergence is expected to be slow. However the presence of very small eigenvalues often makes the norm of the final solution $\exp(A)v$ become very small, forcing the use of very small stopping tolerances; to avoid this problems we introduced a scaling factor t and worked on the computation of the vector $\exp(tA)v$. The value t may be actually interpreted as a *time* variable, as actually is in the practical integration of differential equations. We used the value $t = 0.1$ which ensures in any example a solution with norm larger than 10^{-4}.

The numerical experiments of the first sections were performed in Matlab [44], version 7.0.1 (R14-SP1) while for the others Matlab 7.4 was used. CPU timings were obtained with the function `cputime`.

6.1 Implementation details for SI

As discussed in Section 2.4, when implementing the shift and invert technique at each step the construction of the new basis vector requires the solution of a linear system with coefficient matrix $I - \sigma A$. When using an iterative method as inner solver, a stopping tolerance tol_{inn} needs to be set. Such tolerance should be in accordance with the required shift and invert final accuracy, say tol_{fin}. In the following we present numerical experiments in which the inner tolerance is fixed a priori, namely $\mathrm{tol}_{inn} = 0.01 \cdot \mathrm{tol}_{fin}$, and others in which a tuning technique is used, as in [68], by taking advantage of the *relaxation* strategy [63] we will describe in Section 6.6.
In Section 2.2 a posteriori error estimates were defined to figure out the reached accuracy on the Standard Lanczos approximation. When implementing shift and invert one could directly apply one of the a posteriori estimates (2.2.6) or

(2.2.7), the only difference being the argument of the functions involved; the estimate (2.2.7), for example, in this context reads

$$t_{m+1,m}|e_m^T \exp((I - T_m^{-1})/\sigma)e_1|, \tag{6.1.1}$$

while the more accurate estimate (2.2.6) becomes

$$t_{m+1,m}|e_m^T \varphi_1((I - T_m^{-1})/\sigma)e_1|. \tag{6.1.2}$$

A well known problem of estimate (6.1.1) is that in the very first iterates it may highly underestimate the true error, with the unpleasant effect of stopping the process when only an inadequate approximation is given. To limit this problem we thought to an alternative estimate for the first iterates; we found effective monitoring the relative quantity

$$t_{m+1,m}|e_m^T \exp((I - T_m^{-1})/\sigma)e_1|/|e_1^T \exp((I - T_m^{-1})/\sigma)e_1| \tag{6.1.3}$$

and as soon as it becomes smaller than a selected parameter, say τ, we go back to (6.1.1). Numerical experiments suggested us that the value $\tau = 10^{-1}$ is a good choice. This safeguard procedure should in practice replace (6.1.1) until the components of the approximation vector $\exp((I - T_m^{-1})/\sigma)e_1$ take the expected exponential pattern.

Example 6.1.1. In this example we compare the effectiveness of the a posteriori estimates (6.1.1) -(6.1.3). We consider the approximation of $\exp(0.1\,A)v$, where A is the 4900×4900 matrix stemming from the 2D Laplace operator with Dirichlet homogeneous boundary conditions, v is the normalized vector of all ones, and the safeguard parameter is equal to 0.1.

In Figure 6.1 we observe that the estimate (6.1.2) catches perfectly the pattern of the curve for the true error, even if there is a distance among the two curves of about two orders of magnitude.

The estimate (6.1.1) has a completely wrong behavior for almost 20 iterates but then it results in a good accuracy. The safeguard strategy (6.1.3) gives a right estimate of the true error, even at the beginning of the process. In conclusion, (6.1.3) and (6.1.2) seem to be accurate estimates of the true error; however, due to the lower effort in evaluating (6.1.3), we use it in our numerical experiments.

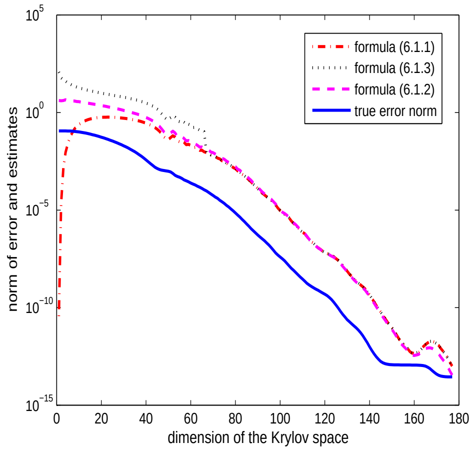


Figure 6.1: Convergence history of Shift and Invert and different error estimates. Safeguard parameter for (6.1.3) equal to 0.1 (see text).

6.2 Implementation details for AK

We recall the framework we are in: we are interested in evaluating the term $\mathcal{R}_\nu(A)v$ used to approximate $\exp(A)v$. For simplicity and without loss of generality, we assume ν to be even. For odd degree rational approximation, the real shifted system corresponding to the real pole can be solved explicitly without resorting to the method discussed above.

We sketch the algorithm to implement the procedure described in Section 5.4 obtained by considering our modification of the real valued method proposed in [5]. We recall that $\xi_1, \ldots, \xi_\nu$ are the poles and $\omega_0, \omega_1, \ldots, \omega_\nu$ the residuals of $\mathcal{R}_\nu$, while for the definition of M and the other matrices we refer to (5.4.5) and the related expressions:

Algorithm AK.

```
Given $A$, $v$, $\xi_1, \ldots, \xi_\nu$, $\omega_0, \omega_1, \ldots, \omega_\nu$

  i) Choose a parameter $\tau > 0$

 ii) Set $B = \tau I - A$; $w_1 = B^{-1}v$; $w_2 = B^{-1}w_1$

iii) For each pole $\xi_j = \xi_R + \imath\xi_I,\ j = 1, 3, 5, \ldots, \nu - 1$:
```

- Solve $Mu_R = \widehat{w}$ with $M = -I + 2(\tau - \xi_R)B^{-1} - |\tau - \xi|^2 B^{-2}$ and $\widehat{w} = +w_1 - (\tau - \xi_R)w_2$
- Compute $u_I = \frac{1}{\xi_I}(-Au_R + \xi_R u_R + v)$
- Set $x_j = u_R + \imath u_I$

iv) **Compute** $y_{AK} = \omega_0 v + 2 \sum_{\substack{j=1 \\ j \text{ odd}}}^{\nu-1} \Re(\omega_j x_j)$

As already mentioned, since the expression for M involves B^{-1}, the solution of $Mu_R = \widehat{w}$ is performed iteratively. It becomes thus necessary to fix the accuracy to impose on these inner systems, since it will influence the quality of the final approximation to $\exp(A)v$. In particular, the stopping tolerance needs to be smaller than the accuracy requested.
In our experiments the required accuracy on the final approximation is $tol_{fin} = 10^{-\nu}$ and the inner tolerance $tol_{inn} = 10^{-\nu-2}$ delivers a sufficiently accurate final solution to the exponential. No further study was attempted to refine this value.

Each matrix-vector multiply with M requires solving two systems with $B = \tau I - A$, and this is related to the fact that M is the coefficient matrix of the normal equation, as shown in Proposition 5.4.2. We consider solving systems with B both with direct and iterative methods.

Example 6.2.1. In this example we compare the original method in [5] with our variant on the data of Example 5.4.6 with three different dimensions for the matrix A. Once computed the Cholesky factor of B the systems are solved by resorting to the Matlab command "\" applied to the resulting triangular systems.
We compare the original method, where an optimal $B = B(\tau)$ is determined and factorized for each pole, with Algorithm AK, where a single suboptimal B is computed and factorized at step (ii) of the algorithm.

The numbers in Table 6.1 show that the new strategy improves performance, especially on large problems, while the total number of iterations does not significantly grow, compared to the optimal shift selection in [5].

In the case an iterative solver is used, one is faced with the problem of efficiently solving two systems with B at each iteration of the solver with M. By exploiting the positive definiteness of B, we consider the following alternatives: a) Two calls to the Conjugate Gradients in sequence; b) Two

		original method	Algorithm AK
n	tol	time (# its)	time (# its)
125	10^{-5}	0.02 (10)	0.02 (11)
	10^{-8}	0.04 (23)	0.23 (32)
	10^{-11}	0.05 (40)	0.05 (61)
	10^{-14}	0.07 (69)	0.09(105)
3375	10^{-5}	1.59 (11)	1.31 (11)
	10^{-8}	2.92 (25)	2.23 (28)
	10^{-11}	4.80 (46)	4.08 (53)
	10^{-14}	6.87 (72)	5.97 (85)
15625	10^{-5}	30.42 (11)	22.14 (11)
	10^{-8}	55.08 (25)	31.77 (27)
	10^{-11}	84.32 (46)	54.90 (51)
	10^{-14}	119.97 (74)	77.73 (84)

Table 6.1: Comparison of the original method in [5] and Algorithm AK for Example 5.4.6. Direct methods are used to solve the linear systems with B.

calls to Preconditioned Conjugate Gradients in sequence; c) One call to the method due to van der Vorst and described in Section 4.2. We consider the problems of Example 5.4.6 and Example 6.2.2 described next.

Example 6.2.2. In this example we approximate $\exp(tA)v$, $t = 0.1$, where the $n \times n$ matrix A stems from the finite difference discretization of the 2D operator

$$\mathcal{L}(u) = (a(x,y)u_x)_x + (b(x,y)u_y)_y, \qquad a(x,y) = 1+y-x, \quad b(x,y) = 1+x+x^2$$

on the unit square, with Dirichlet homogeneous boundary conditions [68]. Two grid refinements are considered; v is a normalized normally distributed random vector.

The numbers in Table 6.2 show that the variant that simultaneously approximates the systems with B and B^2 is faster than both the standard CG method and its preconditioned version. It is important to notice that in the approach proposed in [70] preconditioning is not applicable, nonetheless, its performance is superior to that of standard PCG applied twice. We should mention that when using the approach in [70], one could employ a different (optimal) B for each shifted system at no additional cost. We decided to maintain Algorithm AK for consistency with the case of the direct solves.

n	tol	AK + Variant	AK + CG	AK + PCG
		Example 5.4.6		
125	10^{-5}	0.02	0.04	0.05
	10^{-8}	0.04	0.07	0.08
	10^{-11}	0.08	0.15	0.17
	10^{-14}	0.15	0.29	0.32
3375	10^{-5}	0.42	0.65	1.22
	10^{-8}	0.77	1.75	2.91
	10^{-11}	1.73	3.88	6.07
	10^{-14}	2.81	6.69	11.11
15625	10^{-5}	3.20	4.57	8.61
	10^{-8}	5.88	13.31	21.21
	10^{-11}	13.42	28.07	44.51
	10^{-14}	22.10	52.51	83.22
		Example 6.2.2		
2500	10^{-5}	0.68	1.38	1.10
	10^{-8}	1.69	4.02	3.01
	10^{-11}	3.43	8.32	8.46
	10^{-14}	5.86	15.70	12.58
10000	10^{-5}	3.67	9.38	7.69
	10^{-8}	8.60	28.54	22.34
	10^{-11}	17.50	61.85	47.12
	10^{-14}	29.54	122.99	89.27

Table 6.2: CPU time of Algorithm AK when different iterative schemes are used to solve with $B = \tau I - A$.

6.3 Using direct methods

In this section we compare the performances of Standard Lanczos, SI, PFE and AK when the linear systems in AK, PFE and SI are solved with a direct method.

Reordering the entries of the coefficient matrix may result advantageous; thus we always applied the *minimum degree reordering*, see e.g. [9]. More precisely, to any coefficient matrix we applied the Matlab routine `symamd` and, once computed the Cholesky factorization of the reordered matrix, we solved the resulting triangular systems by resorting to the Matlab command "\".

In Table 6.3 we show the effectiveness of the minimum degree reordering applied to SI and AK; we consider the matrix $\sigma I - A$ with A stemming from the 3D discretization of the Laplace operator on the unit cube and Dirichlet homogeneous boundary conditions and $\sigma = 0.053$ corresponding to $\nu = 14$ in Table 5.2. The data in the table are the numbers of nonzero entries of the Cholesky factor of $\sigma I - A$ before and after the minimum degree reordering. Numerical experiments with other shift parameters showed similar results.

n	lexicographic	minimum degree
5^3	2,729	1,344
15^3	715,289	195,219
25^3	9,405,649	2,400,509

Table 6.3: Example 6.3.1. Number of nonzero entries for the Cholesky factor of $\sigma I - A$, $\sigma = 0.053$

The results of Table 6.3 clearly shows the advantage of reordering the matrix for the storage requirement; moreover, in all numerical experiments we also experienced computational saving in the elapsed time to solve the consequent systems.

The coefficient matrices for PFE have the form $A - \xi_j I$, where we consider the poles ξ_j of the Chebyshev approximation to the exponential. In general these matrices are indefinite, preventing the use of the Cholesky factorization. However we reordered the matrices with the minimum degree ordering and we solved the resulting systems by means of the backslash Matlab command "\". In these cases we observed that the reordering accelerates the solution of the consequent systems.

For PFE we also considered a different strategy: once computed the LU factorization of $A - \xi_j I$ we solved the systems with the computed factors. This strategy was more time consuming than the one described before and then we ignored it.

In all tables of this section we list the CPU time corresponding to Standard Lanczos, SI, PFE and AK, including the factorization time when needed; moreover for Lanczos and SI we report, in parentheses, the dimension of the final approximation space. For AK we report the global number of CG iterates needed for the systems with the matrix $M = M(\xi_j)$ defined in Section 5.4.1.

For SI and Lanczos the final stopping tolerance is $10^{-\nu}$; for AK and PFE ν represents the degree of the diagonal Chebyshev rational function while for CG in AK the tolerance $10^{-\nu-2}$ is used.

Example 6.3.1. We consider the $n \times n$ matrix stemming from the finite difference discretization of the 3D Laplace operator on the unit cube and Dirichlet homogeneous boundary conditions, with eigenvalues in $[-179.14, -12.862]$ for $n = 125$. We approximate the vector $\exp(tA)v$, with $t = 0.1$, v a normalized normally distributed random vector and for three different discretization refinements.

n	tol	Standard Lanczos	Part.Fract. Expansion	AK	Shift-Invert Lanczos
125	10^{-5}	0.01 (13)	0.01	0.02 (11)	0.01 (7)
	10^{-8}	0.01 (18)	0.01	0.03 (32)	0.01 (11)
	10^{-11}	0.01 (22)	0.03	0.05 (61)	0.01 (14)
	10^{-14}	0.01 (24)	0.03	0.08(105)	0.01 (17)
3375	10^{-5}	0.14 (47)	1.32	1.33 (11)	0.48 (8)
	10^{-8}	0.21 (55)	2.13	2.23 (28)	0.65 (13)
	10^{-11}	0.35 (67)	2.88	4.07 (53)	0.85 (19)
	10^{-14}	0.52 (77)	3.70	5.94 (85)	1.06 (25)
15625	10^{-5}	2.69 (89)	30.35	22.05 (11)	11.49 (10)
	10^{-8}	2.95 (93)	51.61	31.60 (27)	11.88 (11)
	10^{-11}	4.76 (113)	69.03	54.68 (51)	14.22 (17)
	10^{-14}	7.25 (130)	90.20	77.31 (84)	16.96 (24)

Table 6.4: Example 6.3.1. CPU time (and number of iterations in parenthesis when appropriate) when systems with shifted matrices are solved with a direct method. Different dimension problems and various stopping tolerances are reported.

Firstly, we note that the CPU time required by all methods sensibly grows up when the dimension of the problem increases. In particular, the ranking among the methods is the same independently of the dimension of the problem, even if the differences are less evident for $n = 125$ since the problem is very small.

In terms of CPU time we notice that standard Lanczos is always the most efficient, while the second best performance is offered by shift and invert. Nevertheless, the memory requirements for Lanczos considerably increase with the dimension of the problem. For handling the largest case one could resort to a *two pass* strategy, as described in Section 2.3; it would result in an almost

doubled CPU time but in the storage of few vectors. In this way Lanczos would employ a CPU time similar to that of SI.
For SI we highlight that the dimension of the approximation space does not grow when the problem dimension increases, as pointed out in [68].
The Axelsson-Kucherov method performs worse than both SI and Standard Lanczos with a significant difference in CPU time. The positive aspect of this method are the limited storage requirements, since only the Cholesky factor and the vectors resulting from CG need to be stored. Moreover for loose tolerances the difference with standard Lanczos and SI is less evident; this is coherent with the original context in which the method was proposed [5].
Furthermore we stress that for the largest problem AK performs better than PFE, although it is not comparable with SI and standard Lanczos, suggesting its use for other matrix functions for which the standard Lanczos method does not show superlinear convergence.

In summary, in this example the Standard Lanczos method performs better than the considered acceleration strategies; the explanation of this phenomenon is in the spectra of the matrices involved; for $n = 125$, indeed, the smallest eigenvalue has modulus less than 200, which still allows standard Lanczos to converge fast. In the following example we look at a different problem:

Example 6.3.2. We consider the vector v and the matrices A defined in Example 6.2.2.
We stress that these matrices have large spectra; indeed, for the coarser discretization, the spectrum is contained in the interval $[-35424, -25.256]$ and we may expect a slow convergence for Standard Lanczos in view of Theorem 2.2.1, as confirmed by the numerical experience.

In Table 6.5 we notice a completely different scenario with respect to the previous one; the shift and invert method always offers the best performance, both for the elapsed time and the memory requirements. The standard Lanczos does not work satisfactorily on these problems, especially if memory constraints are imposed. The reason of this behavior lies in the spectral properties of the considered matrices: indeed, the smallest eigenvalues of the two matrices have very large modulus, thus making the standard Lanczos converge very slowly, as described in the bounds of Theorem 2.2.1.

		Standard	Part.Fract.	AK	Shift-Invert
n	tol	Lanczos	Expansion		Lanczos
2500	10^{-5}	16 (194)	0.22	0.29 (11)	0.12 (10)
	10^{-8}	18 (200)	0.33	0.50 (27)	0.13 (11)
	10^{-11}	53 (242)	0.44	0.92 (51)	0.20 (19)
	10^{-14}	111 (280)	0.53	1.39 (84)	0.24 (24)
10000	10^{-5}	615 (406)	1.24	1.39 (9)	0.67 (11)
	10^{-8}	610 (406)	1.87	2.53 (25)	0.66 (11)
	10^{-11}	1221 (484)	2.55	4.71 (47)	0.94 (17)
	10^{-14}	- (> 500)	3.20	7.49 (82)	1.24 (23)

Table 6.5: Example 6.3.2. CPU time (and number of iterations in parenthesis when appropriate) when systems with shifted matrices are solved with a direct method. Different dimension problems and various stopping tolerances are reported.

6.4 Using iterative methods

In this section we compare the methods AK, PFE and SI when all systems are solved by an iterative method. For AK and SI the result is an inner-outer procedure, requiring the definition of a suitable inner tolerance.
For comparison purposes we report the data from the previous tables, namely Table 6.4 and Table 6.5, for the standard Lanczos method.

We list the methods considered and their main features:

- PFE+QMR. Partial Fraction Expansion where each complex shifted system is solved by a preconditioned simplified QMR method, as described in Section 4.2. The preconditioner is a complex symmetric LDL^T incomplete factorization of the shifted matrix, obtained by a simple modification of the factors computed with the Matlab `luinc` factorization with dropping tolerance equal to 10^{-2}. The system stopping threshold is $10^{-\nu}$.

- SI+PCG. Shift-Invert Lanczos where systems with $I - \sigma A$ are solved with PCG. The Matlab `cholinc` function with dropping tolerance 10^{-2} is used to generate the preconditioner. The inner system stopping threshold is $10^{-\nu}$.

- AK+Variant. We report the results of Table 6.2 of the variant of the Axelsson-Kucherov method, which solves systems with $B = \tau I - A$ and

> B^2 with a single iterative method. For odd degrees the system with the real pole is solved with Preconditioned Conjugate Gradients as in SI+PCG. The inner system stopping threshold is $10^{-\nu-3}$.

In SI+PCG and AK+Variant, the shifted matrix was reordered with a Cuthill-McKee permutation (Matlab function `symrcm`) before building the preconditioner, whereas minimum degree reordering was used for PFE+QMR.

The CPU time for the two test problems are reported in Table 6.6. For SI+PCG and AK+Variant, the total number of outer iterations and the average number of inner iterations is shown. For PFE+QMR, the average number of iterations is shown in parenthesis.

The first difference with the results of Table 6.4 and Table 6.5 is the sensible reduction of the elapsed times when iterative solvers are applied, except for SI. For the first problem, the 3D Laplace operator of Example 6.3.1, standard Lanczos outperforms all methods, even after a two-step procedure, although its benefits are less evident than those in the direct solver case. For the second problem, the 2D elliptic operator of Example 6.2.2, PFE+QMR requires the smallest CPU time, in contrast with the direct case in which SI was the method of choice. However the differences between SI+PCG and PFE+QMR are much less evident than the differences between SI and PFE in the direct case; this similarity fully recovers the equivalence of Theorem 5.2.1 between these preconditioned methods.

6.5 Implementation details for KPIK

As mentioned in Section 2.5, for the numerical implementation of KPIK we followed the algorithm presented by Simoncini [62]: it consists of a first part in which the Krylov space $K_{k,m}(A, v)$ is built and a second in which the Lyapunov equation for the projected and restricted problem is solved. For our tests we used exactly the same steps to get $K_{k,m}(A, v)$; then, once determined the matrices $\mathcal{V}_m$ and $\mathcal{T}_m$, the approximation (2.5.1) was computed by resorting to the Matlab command `expm` for $\mathcal{T}_m$.

At the i-th step of the construction of the space $K_{k,m}(A, v)$ the $n \times 2$ block V_i is required: its evaluation entails a linear system with A as coefficient matrix. We present two kinds of experiments, one employing direct solvers and the other based on iterative ones; in the latter case the implementation was stopped as soon as the error with respect to the true solution was below a fixed tolerance.

We compare the performances of KPIK, Standard Lanczos, SI and PFE. We

		Standard	PFE+	SI+	AK+
		Lanczos	QMR	PCG	Variant
n	tol		(avg its.)	(out/avg in)	(out/avg in)
			Example 6.3.1		
125	10^{-5}	0.01	0.01 (3)	0.01 (7/3)	0.02 (15/5)
	10^{-8}	0.01	0.01 (4)	0.03 (11/5)	0.04 (35/9)
	10^{-11}	0.01	0.02 (5)	0.14 (14/6)	0.08 (71/12)
	10^{-14}	0.01	0.05 (7)	0.04 (17/7)	0.15(111/16)
3375	10^{-5}	0.14	0.67 (8)	0.44 (8/7)	0.42 (20/6)
	10^{-8}	0.21	1.15 (11)	0.81 (13/9)	0.77 (30/7)
	10^{-11}	0.35	1.75 (14)	1.27 (19/10)	1.73 (69/11)
	10^{-14}	0.52	2.30 (16)	1.94 (25/12)	2.81(89/126)
15625	10^{-5}	2.69	5.29 (11)	4.05 (10/10)	3.20 (23/7)
	10^{-8}	2.95	9.36 (17)	5.37 (11/13)	5.88 (29/7)
	10^{-11}	4.76	14.29 (22)	8.87 (17/15)	13.42 (74/12)
	10^{-14}	7.25	19.52 (27)	14.39 (24/18)	22.10 (86/12)
			Example 6.2.2		
2500	10^{-5}	16	0.36 (13)	0.54 (10/12)	0.68 (25/8)
	10^{-8}	18	0.68 (18)	0.75 (11/16)	1.69 (29/7)
	10^{-11}	53	1.09 (22)	1.46 (19/18)	3.43 (76/13)
	10^{-14}	111	1.54 (26)	2.12 (24/21)	5.86 (87/12)
10000	10^{-5}	615	2.46 (24)	4.4 (11/21)	3.6 (32/10)
	10^{-8}	610	4.92 (35)	5.5 (11/27)	8.6 (27/ 7)
	10^{-11}	1221	8.17 (43)	9.8 (17/32)	17.5 (92/15)
	10^{-14}	-	11.74 (51)	15.4 (13/37)	29.5 (95/13)

Table 6.6: Approximation when shifted systems are solved with iterative methods.

omit AK because from the tests performed so far it turned out to be less effective than the others.
To solve the systems with A we use the same devices described in the previous sections: for direct methods we resort to the `symamd` reordering, the Cholesky factorization and the backslash operator; for iterative solves we use the `symrcm` reordering, the incomplete Cholesky factorization and PCG method. In this latter case the inner tolerance needs to be fixed: the numerical tests in [62] show that imposing a very loose tolerance for the Lyapunov equation gives accurate results; in our case we experimented that to reach the accuracy tol_{fin} on the final solution the largest admissible value for the inner tolerance is $tol_{inn} = tol_{fin} * 10$; for our tests we used this value.

All experiments in this section and the next were carried out on one processor of a Sun Fire V40z with 2390.895 MHz and 16 GB RAM, running Matlab 7.4.

Example 6.5.1. We consider the matrix A and the vector v described in Example 6.2.2 for three different space discretizations.
We compare the elapsed time of KPIK, Standard Lanczos, SI and PFE. For the iterative methods the final accuracy required was $10^{-\nu}$ while, when required, the drop tolerance for the Incomplete Cholesky factorization was 10^{-2}.
When referring to iterative methods we mean SI+PCG, PFE+QMR,
KPIK+PCG whereas for Standard Lanczos we just report the same column presented in the context of direct methods.
For SI+PCG and KPIK+PCG, the total number of outer iterations and the average number of inner iterations is shown, whereas for PFE+QMR we report in parenthesis the average number of iterations.

From the data of Table 6.7 we notice that the KPIK method is very effective, although PFE and SI perform always better; however KPIK is always much faster than the Standard Lanczos method, thus offering always a smaller elapsed time, other than smaller memory requirements, as shown in Figure 2.2.

6.6 Enhancements for PFE and SI

In this section we describe a possible enhancement for both shift and invert and PFE, with sensible computational savings.

n	ν	Standard Lanczos	KPIK	SI	PFE
			Direct methods		
		time (its.)	time (its.)	time (its.)	time
50^2	5	6.53 (193)	0.09 (10)	0.08 (12)	0.14
	8	7.65 (200)	0.12 (18)	0.09 (11)	0.21
	11	18.83 (241)	0.17 (26)	0.12 (19)	0.30
	14	42.18 (280)	0.25 (36)	0.16 (24)	0.37
100^2	5	294.49 (406)	0.48 (6)	0.47 (12)	0.74
	8	293.68 (406)	0.72 (16)	0.50 (13)	1.12
	11	648.65 (485)	1.05 (26)	0.62 (17)	1.59
	14	- (>500)	1.41 (36)	0.83 (23)	1.98
			Iterative methods		
		time(its.)	time(out/avg in)	time(out/avg in)	time(avg its.)
50^2	5	6.53 (193)	0.19 (10/12)	0.23 (9/11)	0.17 (12)
	8	7.65 (200)	0.41 (18/19)	0.32 (11/15)	0.28 (16)
	11	18.83 (241)	0.70 (26/24)	0.62 (19/17)	0.49 (20)
	14	42.18 (280)	1.11 (36/29)	0.92 (24/21)	0.64 (24)
100^2	5	294.49 (406)	1.05 (6/21)	1.90 (9/21)	1.22 (22)
	8	293.68 (406)	3.27 (16/35)	2.87 (11/27)	2.08 (31)
	11	648.65 (485)	6.26 (26/44)	5.09 (17/31)	3.72 (39)
	14	- (>500)	10.03 (36/53)	7.96 (23/37)	4.85 (45)

Table 6.7: Example 6.2.2. CPU times of Standard Lanczos, KPIK, SI, PFE.

6.6.1 Relaxation strategy for SI

In this section we describe the *relaxation strategy* proposed by Simoncini and Szyld in [63] and van den Eshof and coauthors in [69] and used in [68] to improve the performance of the shift and invert method described in Section 2.4.

As mentioned at the beginning of Section 6.1, at each step of the shift and invert method a linear system of the form $Zv_{j+1} = v_j$ needs to be solved, for j varying from 0 to the value m corresponding to the sought space. In [63] a theoretical justification was presented to use a decreasing accuracy when solving these systems. More precisely, they considered the situation in which, for building the Krylov space $K_m(Z^{-1}, v)$, in place of the exact systems $Zv_{j+1} = v_j$ one performs $(Z + E_j)v_{j+1} = v_j$, with E_j error matrices which changes at every step. It is shown in [63] that $||E_j||$ can be allowed to

grow as convergence takes place; this means that the accuracy in solving the linear systems can be *relaxed*, thus leading to computational savings.
We use the relaxed tolerance set in [68]: given a fixed tolerance $\epsilon > 0$, the stopping tolerance η_j for the j-th inner system is

$$\eta_j = \frac{\epsilon}{\|e_{j-1}\| + \epsilon},$$

where e_{j-1} is the error in the approximation of the exponential operator at the previous iteration. In practice, $\|e_{j-1}\|$ is replaced by the estimate

$$\|e_{j-1}\| \lesssim \frac{\delta_{j-1}}{1-\delta_{j-1}} \|y_{j-1}\|$$

and

$$\delta_{j-1} = \frac{\|y_{j-1} - y_{j-2}\|}{\|y_{j-1}\|}.$$

We fixed ϵ to be equal to the initial inner tolerance.

We have some numerical evidence that a similar relaxation strategy may be applied also to the KPIK method and this will be investigated in a forthcoming work.

6.6.2 Enhancement for PFE

In the implementation of PFE we used a single preconditioner for all systems stemming from the partial fraction expansion. In this section we present numerical experiments obtained by using the same preconditioner for all systems, from which the name *PFE+QMR mono* derives. We did not consider in detail the choice of the optimal preconditioner but numerical tests showed that the shifted complex symmetric matrix $A - \xi_1 I$ corresponding to the pole with largest imaginary part worked better than the other shifted matrices of the form $A - \xi_j I$; in practice we reordered the matrix with the `symamd` function and we computed an Incomplete Cholesky factorization (Matlab 7.4 function `cholinc`) with dropping tolerance 10^{-2}.

This strategy makes the SI+PCG and the PFE+QMR methods even closer to each other, since we have shown that SI+PCG may be viewed as a special way of preconditioning the PFE systems with a single, parameter dependent, matrix.

		PFE+QMR	PFE+QMR	SI+PCG	SI+PCG
		mono		relax	
n	tol	(avg its.)	(avg its.)	(out/avg in)	(out/avg in)
2500	10^{-5}	0.20 (12)	0.18 (12)	**0.15** (9/ 7)	0.29 (10/11)
	10^{-8}	**0.27** (16)	0.32 (16)	**0.27** (15/ 9)	0.32 (11/15)
	10^{-11}	0.47 (20)	0.53 (20)	**0.42** (21/10)	0.61 (19/17)
	10^{-14}	0.63 (24)	0.73 (23)	**0.58** (26/12)	0.90 (24/21)
10000	10^{-5}	1.31 (22)	1.37 (22)	**1.18** (8/14)	2.50 (11/21)
	10^{-8}	**2.19** (31)	2.67 (32)	2.22 (14/16)	3.05 (11/27)
	10^{-11}	3.99 (39)	4.32 (39)	**3.65** (20/18)	5.44 (17/31)
	10^{-14}	**5.29** (45)	6.14 (46)	5.33 (26/20)	8.49 (23/37)

Table 6.8: Example 6.2.2. CPU Time and number of iterations for the original PFE+QMR and SI+PCG methods, and for their enhanced versions.

6.6.3 Comparisons between the original and enhanced versions

In this section we compare the performance of SI+PCG with the enhanced version SI+PCG+relax and PFE+QMR with PFE+QMR mono for the problem of Example 6.2.2. For SI+PCG+relax the initial inner tolerance was equal to the outer tolerance.

In Table 6.8 we notice the important improvement obtained with the enhanced versions, especially for SI, with instances reaching an improvement of almost 50%. For PFE it is interesting that the average number of iterations is in practice the same of the original one, with improvement of the elapsed time in all cases. Best timings are in boldface. The two different enhanced preconditioned techniques PFE+QMR+mono and SI+PCG+relax behave quite similarly and, taking into account the Matlab timings fluctuation, it is difficult to depict a clear winner.

In PFE the common preconditioner is computed once for all, whereas each shifted system is solved separately. This is the major remaining drawback of the enhanced PFE+QMR method when a few time steps are performed, since many systems need to be solved. On the other hand, SI precisely avoids this step, since it constructs a single preconditioner that, in the case of a rational function, still allows one to keep the shifted form of the systems, so that all systems can be solved simultaneously with a single SI-Lanczos iteration as in (5.3.1).

6.7 Integrating PDEs

We next consider the discretization of the following parabolic equation in two spatial dimensions ([68]):

$$\frac{\partial}{\partial t} u = \mathcal{L}(u), \quad (x, y) \in (0, 1)^2, \quad 0 \le t \le \mathbf{T}, \tag{6.7.1}$$

where the solution $u = u(t, x, y)$ is subject to the initial condition $u(0, x, y) = g(x, y)$ and to mixed boundary conditions (b.c.): homogeneous Dirichlet b.c. on the western and eastern boundaries, and homogeneous Neumann b.c. on the northern and southern boundaries of the domain. The operator $\mathcal{L}$ is as in Example 6.2.2 while the function g is defined in the following.

After standard centered finite difference space discretization the problem (6.7.1) is equivalent to

$$\begin{array}{r} \frac{d}{dt} u = Au, \quad 0 \le t \le \mathbf{T} \\ u(0) = u_0 \end{array} \tag{6.7.2}$$

and the function g is chosen such that u_0 is the normalized vector of all ones.

Assume we fix nx and ny nodes in the x and y directions, respectively; then, for the presence of Dirichlet homogeneous b.c. for $x = 0$ and $x = 1$, the size of A is simply $(nx - 2)ny$; indeed the finite difference scheme deletes the rows corresponding to the extrema of the x side.

We assume that $nx = ny$, thus the distance among the nodes in the x direction and in the y direction is the same, namely $\Delta x = \Delta y$.

In Figure 6.2 the Arnoldi approximate solution to $\exp(\mathbf{T}A)u_0$ for $m = 500$ is plotted, for the fine discretization corresponding to $nx = ny = 100$ and $\mathbf{T} = 0.1$.

We compare two methods for solving the ordinary differential equation (6.7.2): the *Ideal one-step method*, as defined in [29], and the *Crank-Nicolson* method.

We recall here two important definitions for generic numerical methods for differential equations in the two space variables x and y [30]:

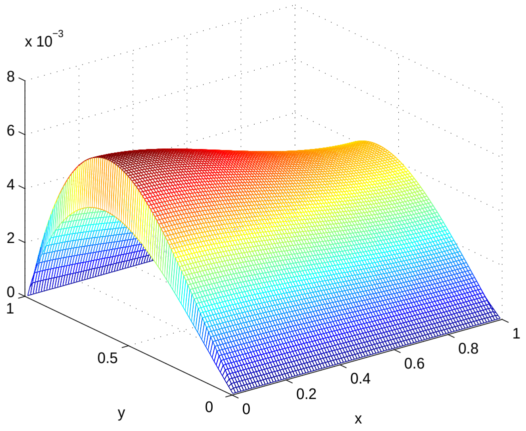


Figure 6.2: Solution of (6.7.2) for $nx = ny = 100$ and $\mathbf{T} = 0.1$.

Definition 6.7.1.

- By *convergence* we mean that the results of the method approach the analytical values as Δt, Δx and Δy approach zero;
- by *stability* we mean that errors made at one stage of the calculations do not cause increasingly large errors as the computations are continued, but rather will eventually damp out.

6.7.1 Ideal one-step method

The solution of (6.7.2) at $t = \mathbf{T}$ is $\exp(\mathbf{T}A)u_0$; in real applications this vector is not computed directly but a time-stepping strategy is applied, due in general to stability and accuracy requirements [28]. This means that $\exp(\mathbf{T}A)u_0$ is approximated by a sequence of $\texttt{round}(\mathbf{T}/\Delta t)$ applications of $\exp(\Delta tA)$ as

$$\exp(\Delta tA)\cdots\exp(\Delta tA)u_0, \tag{6.7.3}$$

where Δt is the time step length and `round` is the Matlab function which approximates any real number to the nearest integer; this procedure was analyzed by Gallopoulos and Saad in [29].

The methods described so far for approximating vectors of the form $\exp(A)v$ apply to (6.7.3) since it translates in the iterative formula

$$w_{i+1} = \exp(\Delta tA)w_i,\ i = 0, \dots, \texttt{ceil}(\mathbf{T}/\Delta t)$$

with $w_0 = u_0$; thus about $\mathbf{T}/\Delta t$ vectors of the form $\exp(\Delta t A)w$ need to be subsequently computed.

6.7.2 The Crank-Nicolson method

The Crank-Nicolson method is a procedure to approximately solve discretized partial differential equations; one of its qualities is that it is second order accurate in both spatial and time variables, that is, the error between the term $\partial^2 u/\partial x^2$ and its approximation is $O(\Delta x^2)$, and the same is true for $\partial^2 u/\partial y^2$ and for the time derivatives.
When applied to (6.7.2) the Crank-Nicolson method yields the recursive formula

$$\left(I - \frac{\Delta t}{2} A\right) u_{n+1} = \left(I + \frac{\Delta t}{2} A\right) u_n; \qquad (6.7.4)$$

it is clearly an *implicit* method which, at each step, requires to solve a system with the same coefficient matrix $(I - \frac{\Delta t}{2} A)$, which is shifted symmetric positive definite. As well known, this method is unconditionally stable, that is, it is stable independently of Δx, Δy and Δt, see e.g. ([30], Section 8.4, pp. 406).
The convergence analysis presented in ([30], Section 8.4, pp. 406) for the Crank-Nicolson method was sketched for the 1D heat flow equation, resulting in the conclusion that the method is convergent only when the ratio $\Delta t/(\Delta x)^2$ is finite, preferably small; a similar restriction for $\Delta t/[(\Delta x)^2 + (\Delta y)^2]$ was said to ensure the convergence for 2D problems. We found similar qualitative restrictions on the ratio $\Delta t/[(\Delta x)^2 + (\Delta y)^2]$ in several books, but no precise expression for its value.

In the following example the linear systems in (6.7.4) are solved with the Preconditioned Conjugate Gradients method applied to the matrix reordered with `symrcm` and an incomplete Cholesky factor is used as preconditioner; for the inner tolerance we use $tol_{inn} = tol_{fin}/1000$.

6.7.3 Numerical tests

We consider different possible space and time discretizations so as to approximate the exact solution at $\mathbf{T} = 0.1$.

We analyze the methods that in previous examples showed the best performances, namely the standard Lanczos and the enhanced versions of SI and PFE methods and compare them with the Crank-Nicolson technique.

In Table 6.9 we present the errors as Δt varies; 50 nodes are considered for the discretization of the domain $(0,1)^2$ both in the x and y direction; the final accuracy is 10^{-6}.

Δt	Standard Lanczos	Crank Nicolson	SI +PCG relax	PFE+QMR mono
1e-04	6.1e-06	2.7e-08	9.2e-07	3.6e-04
5e-04	1.8e-06	7.0e-07	3.8e-06	6.0e-05
1e-03	1.6e-07	2.8e-06	3.2e-06	2.2e-05
5e-03	1.2e-07	1.6e-02	1.0e-07	4.1e-06
1e-02	4.4e-08	5.3e-02	1.0e-07	4.1e-06
5e-02	1.1e-08	2.1e-01	1.5e-07	1.1e-06
1e-01	1.1e-08	3.1e-01	3.2e-07	9.6e-07

Table 6.9: Errors of Standard Lanczos, of Crank-Nicolson and of enhanced accelerated methods SI and PFE to approximate the solution at $\mathbf{T} = 0.1$; Δt is the time step.

From the tabulated values it is evident that the Crank-Nicolson method reaches a good accuracy only for a very small step size, namely, not larger than 0.001; this seems to be coherent with the restrictions on the ratio $\Delta t/[(\Delta x)^2 + (\Delta y)^2]$ to ensure the convergence; the jump of the error when passing from $\Delta t = 1e - 03$ to $\Delta t = 5e - 03$ would suggest that the upper bound for the ratio $\Delta t/2(\Delta x)^2$ in this case could be 10^5; indeed all larger ratios do not lead to convergence. For this reason, when comparing the elapsed time in Tables 6.10 and 6.11 we start from very small values for Δt.

Moreover it results that the accuracy for the Crank-Nicolson method deteriorates as Δt gets larger, while for all other methods the behavior is the opposite, depending on cancellation errors. From Table 6.9 it is also evident the different accuracy for PFE and SI; for both methods the inner tolerance is set to $tol_{inn} = tol_{fin}/100$ but for SI this value changes dinamically during the iterative process, thanks to the relaxation strategy applied.

In the next experiment we compare the elapsed time for the considered methods, for two different space discretizations and different final accuracies which are of interest in the context of evolution problems.

By taking into account the data in Table 6.9 we assume that for $\Delta t > 0.005$ the Crank-Nicolson method does not reach the required accuracy and we put the symbol "$\dagger''$ to highlight this phenomenon; we do the same for PFE for the three smallest time steps.

The crucial result of this example is that the enhanced version of PFE reaches always the best CPU time and it corresponds to the largest step size; in practice in a single step ($\Delta t = 0.1$) the required accuracy is reached in the lowest time. We stress here that also the enhanced SI version works very well, with results which could be considered almost equal to those of PFE+QMR mono, if taking into account the Matlab fluctuations.
The Crank-Nicolson method does not perform satisfactorily, with CPU time that are even always worse than the Standard Lanczos procedure.
We thus conclude that for all practical situations in which a time stepping procedure is needed the enhanced versions of PFE and SI may offer a good result in just one iteration, representing a welcome event; moreover they sensibly outperform the commonly used Crank-Nicolson method.

In addition, we explicitly observe that the the costs of the acceleration procedures and of standard Lanczos have an opposite behavior; indeed for the former it decreases as the number of time steps decreases, whereas for Standard Lanczos it becomes unacceptably large due to the increasing value of $\|\Delta t A\|$.

In conclusion these numerical tests showed that the enhanced acceleration techniques based on shift and invert and on the explicit solution of the complex shifted systems stemming from a partial fraction expansion perform very well. In particular, for the common problem of integrating PDEs they allow a large time step and offer very good results, as concern CPU time and memory requirements, even better than those offered by the common Crank-Nicolson integrator.

grid	final		Standard	Crank	SI +PCG	PFE+QMR
(nx,ny)	accuracy	Δt	Lanczos	Nicolson	relax	mono
(50,50)	10^{-4}	1e-04	1.02	6.67	18.5	†
		5e-04	0.40	1.75	5.14	†
		1e-03	0.30	0.91	2.70	†
		5e-03	0.23	†	0.78	1.53
		1e-02	0.27	†	0.56	0.86
		5e-02	1.19	†	0.25	0.26
		1e-01	2.77	†	0.18	**0.15**
	10^{-6}	1e-04	1.34	8.11	25.15	†
		5e-04	0.60	2.06	6.53	†
		1e-03	0.42	1.09	3.62	†
		5e-03	0.32	†	1.08	3.08
		1e-02	0.44	†	0.76	1.61
		5e-02	2.41	†	0.38	0.41
		1e-01	6.41	†	0.28	**0.23**
(90,90)	10^{-4}	1e-04	4.31	32.69	115.34	†
		5e-04	2.07	9.62	26.82	†
		1e-03	1.63	6.41	16.31	†
		5e-03	1.79	†	5.59	8.81
		1e-02	2.53	†	4.00	5.49
		5e-02	19.48	†	1.88	1.73
		1e-01	66.26	†	1.38	**0.99**
	10^{-6}	1e-04	5.70	45.15	135.12	†
		5e-04	3.04	12.29	33.67	†
		1e-03	2.44	7.86	20.15	†
		5e-03	2.58	†	7.42	16.16
		1e-02	3.94	†	5.41	9.56
		5e-02	52.85	†	2.81	2.79
		1e-01	187.31	†	2.10	**1.62**

Table 6.10: Parabolic problem (cf. (6.7.1)). CPU times of Crank-Nicolson, Standard Lanczos and enhanced accelerated methods to approximate the solution at $\mathbf{T} = 0.1$, for different time step lengths Δt and different number of nodes nx, ny in the discretization of the domain $(0, 1)^2$.

grid (nx,ny)	final accuracy	Δt	Standard Lanczos	Crank Nicolson	SI +PCG relax	PFE+QMR mono
(120,120)	10^{-4}	1e-04	10.51	55.99	270.57	†
		5e-04	5.41	18.99	65.34	†
		1e-03	4.73	14.28	36.74	†
		5e-03	5.70	†	13.40	21.83
		1e-02	10.02	†	9.51	14.05
		5e-02	238.48	†	4.48	4.10
		1e-01	901.58	†	3.43	**2.29**
	10^{-6}	1e-04	14.16	78.63	339.99	†
		5e-04	7.03	25.92	80.39	†
		1e-03	6.10	18.90	46.49	†
		5e-03	7.53	†	17.80	38.71
		1e-02	12.96	†	13.06	23.35
		5e-02	266.67	†	6.69	7.12
		1e-01	902.46	†	4.80	**3.64**

Table 6.11: Parabolic problem (cf. (6.7.1)). CPU times of Crank-Nicolson, Standard Lanczos and enhanced accelerated methods to approximate the solution at $\mathbf{T} = 0.1$, for different time step lengths Δt and $nx = ny = 120$ in the discretization of the domain $(0,1)^2$.

Bibliography

[1] M. Afanasjew, M. Eiermann, O. G. Ernst, and S. Guettel, *Implementation of a restarted Krylov subspace method for the evaluation of matrix functions*, Submitted to LAA special edition.

[2] J. E. Andersson, *Approximation of e^{-x} by rational functions with concentrated negative poles*, J. Approx. Theory, 32 (1981), pp. 85–95.

[3] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems (Advances in Design and Control) (Advances in Design and Control)*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005.

[4] W. E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[5] O. Axelsson and A. Kucherov, *Real valued iterative methods for solving complex symmetric linear systems*, Numer. Linear Algebra Appl., 7 (2000), pp. 197–218.

[6] G. A. Baker and P. Graves-Morris, *Padé Approximants*, Encyclopedia of Mathematics and its applications, Cambridge University Press, Cambridge, 1996.

[7] C. Baldwin, R. Freund, and E. Gallopoulos, *A parallel iterative method for exponential propagation*, in Proc. Seventh SIAM Conference on Parallel Processing for Scientific Computing, D. Bailey et al., ed., SIAM Philadelphia, 1995, pp. 534–539.

[8] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *TEMPLATES for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, 1993.

[9] M. Benzi, *Preconditioning techniques for large linear systems: A survey*, Journal of Computational Physics, 182 (2002), pp. 418–477.

[10] M. Benzi and D. Bertaccini, *Real-valued iterative algorithms for complex symmetric linear systems*, tech. report, Department of Mathematics and Computer Science at Emory University, October 2006. To appear in IMA Journal of Numerical Analysis.

[11] L. Bergamaschi and M. Vianello, *Efficient computation of the exponential operator for large, sparse, symmetric matrices*, Numer. Linear Algebra Appl., 7 (2000), pp. 27–45.

[12] D. Bertaccini, *Efficient solvers for sequences of complex symmetric linear systems*, ETNA, 18 (2004), pp. 49–64.

[13] P. B. Borwein, *Rational approximations with real poles to e^{-x} and x^n*, J. Approx. Theory, 38 (1983), pp. 279–283.

[14] A. J. Carpenter, A. Ruttan, and R. S. Varga, *Extended numerical computations on the 1/9 conjecture in rational approximation theory*, in Rational Approximation and Interpolation, P. R. Graves-Morris, E. B. Saff, and R. S. Varga, eds., vol. 1105 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1984, pp. 383–411.

[15] P. Castillo and Y. Saad, *Preconditioning the matrix exponential operator with applications*, J. Scientific Computing, 13 (1999), pp. 225–302.

[16] W. J. Cody, G. Meinardus, and R. S. Varga, *Chebyshev rational approximations to e^{-x} in $[0, +\infty)$ and applications to heat-conduction problems*, J. Approx. Theory, 2 (March 1969), pp. 50–65.

[17] D. Day and M. A. Heroux, *Solving complex-valued linear systems via equivalent real formulations*, SIAM J. Sci. Comput., 23 (2001), pp. 480–498.

[18] V. Druskin, A. Greenbaum, and L. Knizhnerman, *Using nonorthogonal Lanczos vectors in the computation of matrix functions*, SIAM J. Sci. Comput., 19 (1998), pp. 38–54.

[19] V. Druskin and L. Knizhnerman, *Two polynomial methods of calculating functions of symmetric matrices*, U.S.S.R. Comput. Math. Math. Phys., 29 (1989), pp. 112–121.

[20] ——, *Krylov subspace approximation of eigenpairs and matrix functions in exact and computer arithmetic*, Numerical Linear Algebra with Appl., 2 (1995), pp. 205–217.

[21] ——, *Extended Krylov subspaces: approximation of the matrix square root and related functions*, SIAM J. Matrix Analysis and Applications, 19 (1998), pp. 755–771.

[22] M. Eiermann and O. Ernst, *A restarted Krylov subspace method for the evaluation of matrix functions*, SIAM J. Numer. Anal., 44 (2006), pp. 2481–2504.

[23] T. Ericsson and A. Ruhe, *The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems*, Math. Comp, 35 (1980), pp. 1251–1268.

[24] G. F. Franklin, M. L. Workman, and D. Powell, *Digital Control of Dynamic Systems*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[25] R. W. Freund, *On Conjugate Gradient Type Methods and Polynomial Preconditioners for a Class of Complex Non-Hermitian Matrices*, Numer. Math., 57 (1990), pp. 285–312.

[26] R. W. Freund and N. M. Nachtigal, *Software for simplified Lanczos and QMR algorithms*, Applied Numerical Mathematics, 19 (1995), pp. 319–341.

[27] A. Frommer and V. Simoncini, *Stopping criteria for rational matrix functions of Hermitian and symmetric matrices.* To appear in SIAM J. Scient. Computing.

[28] E. Gallopoulos and Y. Saad, *On the parallel solution of parabolic equations*, Proc. 3d ACM Int'l. Conf. Supercomputing, (1989), pp. 17–28.

[29] E. Gallopoulos and Y. Saad, *Efficient solution of parabolic equations by Krylov approximation methods*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 1236–1264.

[30] C. F. Gerald, *Applied Numerical Analysis*, Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 1978.

[31] G. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 3rd ed., 1996.

[32] A. A. Gonchar and E. A. Rakhmanov, *Equilibrium distributions and degree of rational approximation of analytic functions*, Math. Sbornik, 134(176) (1987), pp. 306–352. English transl. in *Math. USSR Sb.* 62(2):305-348, 1989.

[33] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Frontiers in Applied Mathematics, No. 17, SIAM, Philadelphia, 1997.

[34] N. J. Higham, *The Scaling and Squaring Method for the Matrix Exponential Revisited*, SIAM J. Matrix Analysis Appl., 26 (2005), pp. 1179–1193.

[35] M. Hochbruck and M. E. Hochstenbach, *Subspace extraction for matrix functions*, tech. report, Dept. of Math., Case Western Reserve University, September 2005. Submitted.

[36] M. Hochbruck and C. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 34 (1997), pp. 1911–1925.

[37] M. Hochbruck and C. Lubich, *Exponential integrators for quantum-classical molecular dynamics*, BIT, Numerical Mathematics, 39 (1999), pp. 620–645.

[38] P. S. Jensen, *The solution of large symmetric eigenproblems by sectioning*, SIAM J. Numer. Anal., 9 (1972), pp. 534–545.

[39] A. N. Krylov, *O Čislennom rešenii uravnenija, kotorym v techničeskih voprasah opredeljajutsja častoy malyh kolebaniĭ material'nyh*, Izv. Adad. Nauk SSSR otd. Mat. Estest., (1931), pp. 491–539.

[40] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, 45 (1950), pp. 255–282.

[41] J. D. Lawson, *Generalized Runge-Kutta processes for stable systems with large Lipschitz constants*, SIAM J. Numer. Anal., 4 (1967), pp. 372–380.

[42] L. Lopez and V. Simoncini, *Analysis of projection methods for rational function approximation to the matrix exponential*, SIAM J. Numer. Anal., 44 (2006), pp. 613–635.

[43] Y. Y. Lu, *Exponentials of symmetric matrices through tridiagonal reductions*, Linear Algebra and its Applications, 279 (1998), pp. 317–324.

[44] The MathWorks, Inc., *MATLAB 7*, September 2004.

[45] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Rev., 20 (October 1978), pp. 801–836.

[46] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review, 45 (2003), pp. 3–49.

[47] I. Moret and P. Novati, *An interpolatory approximation of the matrix exponential based on Faber polynomials*, Journal of Computational and Applied Mathematics, 131 (2001), pp. 361–380.

[48] ——, *RD-rational approximations of the matrix exponential*, BIT, Numerical Mathematics, 44 (2004), pp. 595–615.

[49] ——, *Interpolating functions of matrices on zeros of quasi-kernel polynomials*, Numer. Linear Algebra Appl., 12 (2005), pp. 337–353.

[50] I. Najfeld and T. F. Havel, *Derivatives of the matrix exponential and their computation*, Adv. Appl. Math., 16 (1995), pp. 321–375.

[51] A. Nauts and R. Wyatt, *New approach to many state quantum dynamics: The recurisive residue generation method*, Phys. Rev. Lett., 51 (1983), pp. 2238–2241.

[52] S. P. Nørsett, *Restricted Padé approximations to the exponential function*, SIAM J. Numer. Anal., 15 (Oct. 1978), pp. 1008–1029.

[53] C. C. Paige, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341–349.

[54] P. Petrushev and V. Popov, *Rational approximation of real function*, Cambridge University Press, Cambridge, 1987.

[55] M. Popolizio and V. Simoncini, *Acceleration techniques for approximating the matrix exponential*, tech. report, Dipartimento di Matematica, Bari, I.

[56] R. F. Rinehart, *The equivalence of definitions of a matrix function*, American Mathematical Monthly, 62 (1955), pp. 395–414.

[57] A. Ruhe, *Rational Krylov sequence methods for eigenvalue computation*, Lin. Alg. Appl., 58 (1984), pp. 391–405.

[58] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., 29 (1992), pp. 209–228.

[59] ——, *Iterative methods for sparse linear systems*, SIAM, Society for Industrial and Applied Mathematics, 2nd ed., 2003.

[60] R. B. Sidje, *Expokit: A Software Package for Computing Matrix Exponentials*, ACM Transactions on Math. Software, 24 (1998), pp. 130–156.

[61] R. B. Sidje and W. J. Stewart, *A numerical study of large sparse matrix exponentials arising in Markov chains*, Comput. Stat. Data Anal., 29 (1999), pp. 345–368.

[62] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM J. Sci. Comput., 29 (2007), pp. 1268–1288.

[63] V. Simoncini and D. B. Szyld, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM J. Sci. Comput., 25 (2003), pp. 454–477.

[64] V. Simoncini and D. B. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, J. Numerical Linear Algebra with Appl, 14 (2007), pp. 1–59.

[65] D. E. Stewart and T. S. Leyk, *Error estimates for Krylov subspace approximations of matrix exponentials*, J. Computational and Applied Mathematics, 72 (1996), pp. 359–369.

[66] H. Tal-Ezer, *Spectral methods in time for parabolic problems*, SIAM J. Numer. Anal., 26 (1989), pp. 1–11.

[67] L. N. Trefethen, J. A. C. Weideman, and T. Schmelzer, *Talbot quadratures and rational approximations*, BIT, Numerical Mathematics, 46 (2006), pp. 653–670.

[68] J. van den Eshof and M. Hochbruck, *Preconditioning Lanczos approximations to the matrix exponential*, SIAM J. Sci. Comput., 27 (2006), pp. 1438–1457.

[69] J. van den Eshof and G. L. G. Sleijpen, *Inexact Krylov subspace methods for linear systems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 125–153.

[70] H. A. van der Vorst, *An iterative solution method for solving $f(A)x = b$ using Krylov subspace information obtained for the symmetric positive definite matrix A*, J. Comput. Appl. Math., 18 (1987), pp. 249–263.

[71] R. S. Varga, *On higher order stable implicit methods for solving parabolic partial differential equations*, J. of Mathematics and Physics, XL (1961), pp. 220–231.