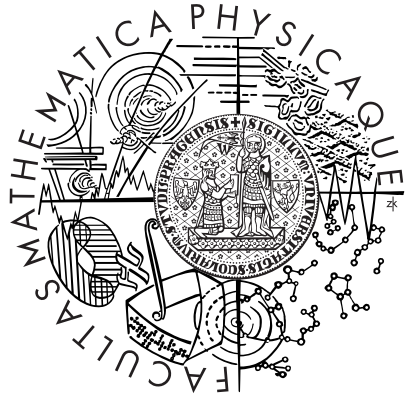
Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Bc. Darja Suchá

Matrix functions and their numerical approximations

Department of Numerical Mathematics

Supervisor of the master thesis: RNDr. Iveta Hnětynková, Ph.D.
Study programme: Mathematics
Specialization: Numerical and Computational Mathematics

Prague 2011

I would like to thank my supervisor RNDr. Iveta Hnětynková, Ph.D. for her help, valuable advice, opinion and suggestions. I also would like to thank Dipl.-Math. Stefan Güttel, Ph.D., M.S. Jie Chen, Ph.D. and Prof. Valeria Simoncini, for the opportunity to use their software and for their valuable advice. I thank Bc. Lukáš Korous, for matrices I used for the numerical experiments in this thesis. And finally, I would like to thank my family and friends for the motivation and support.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.
I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Prague, 5. 12. 2011 Darja Suchá

Contents

Název práce: Maticové funkce a jejich numerické aproximace
Autor: Darja Suchá
Katedra/Ústav: Katedra numerické matematiky
Vedoucí diplomové práce: RNDr. Iveta Hnětynková, Ph.D., katedra numerické matematiky

Abstrakt: V předložené práci studujeme numerické metody pro aproximaci funkce f matice A. Nejprve uvedeme teoretický základ - shrneme možné definice maticových funkcí a jejich vlastnosti. Dále představíme základní numerické metody výpočtu aproximace $f(A)$. V mnoha aplikacích potřebujeme aproximovat maticovou funkci $f(A)$ aplikovanou na předem daný vektor $\boldsymbol{b}$, tj. $f(A)\boldsymbol{b}$. Zejména, pokud A je velká a řídká, výpočet aproximace $f(A)$ a následné přenásobení vektorem $\boldsymbol{b}$ může být výpočetně velmi náročné. Proto se v dalších kapitolách zabýváme numerickými metodami, které počítají přímo aproximaci $f(A)\boldsymbol{b}$. Hlavní důraz je kladen na polynomiální aproximaci ve smyslu nejmenších čtverců a několik modifikací Krylovovských metod. Numerické experimenty ukazují srovnání konvergence a časové náročnosti výpočtu aproximace.

Klíčová slova: maticová funkce, numerické aproximace, ortogonální polynomy, nejmenší čtverce, Krylovovské metody, restart, deflace, rozšířený Krylovův prostor, konvergence

Title: Matrix functions and their numerical approximations
Author: Darja Suchá
Department/Institute: Department of Numerical Mathematics
Supervisor of the master thesis: RNDr. Iveta Hnětynková, Ph.D., department of Numerical Mathematics

Abstract: In the presented work, we study numerical methods for approximation of a function f of a matrix A. First, we give theoretical background - definitions of matrix functions, and their properties. Further, we summarize basic numerical methods for computation of an approximation of matrix functions $f(A)$. In many applications, we need to approximate the matrix function $f(A)$ applied on an apriory given vector $\boldsymbol{b}$, i.e. $f(A)\boldsymbol{b}$. Especially, when A is large and sparse, the computation of approximation to $f(A)$ and subsequent multiplication by the vector $\boldsymbol{b}$ can be computationaly expensive. Therefore we study methods, which compute the approximation of $f(A)\boldsymbol{b}$ directly. Main emphasis is placed on the polynomial approximation in the least squares sense, and several modifications of Krylov subspace methods. Numerical experiments compare convergence and computational time required to obtain reasonable approximation to $f(A)\boldsymbol{b}$.

Keywords: matrix function, numerical approximation, orthogonal polynomials, least squares, Krylov subspace methods, restart, deflation, extended Krylov subspace, convergence

Introduction

'The beginning is the most important part of the work.'
Plato

In this thesis, we are interested in matrix functions and their numerical approximation. Matrix functions arise in many applications, e.g.:

- Systems of linear equations $A\boldsymbol{x} = \boldsymbol{b}$ with the solution $\boldsymbol{y} = f(A)\boldsymbol{b}$, where $f(z) = 1/z$. Such systems come from discretisation of the Heat equation, Maxwell's equations, Wave equation etc.

- Systems of ordinary differential equations of the first order $\boldsymbol{y}' = A\boldsymbol{y}$ with the initial condition $\boldsymbol{y}(0) = \boldsymbol{b}$, with the solution $f(A)\boldsymbol{b}$, where $f(z) = \exp(z)$.

- Nuclear magnetic resonance, see [25], pp. 37, described by the Solomon equations,

$$\frac{dM(t)}{dt} = -M(t)R,$$

  with the given initial condition $M(0) = I$. Here $M(t)$ is the matrix of intensities, and R is a symmetric, diagonally dominant matrix, known as the relaxation matrix. This relation is used in both directions: in simulations and testing to compute $M(t)$ for a given R, $M(t) = \exp(-Rt)$; and in the inverse problem to dermine R from observation identities, $R = -\frac{1}{t}\log M(t)$.

- Systems of ordinary differential equations of the second order, $\boldsymbol{y}''(z) + A\boldsymbol{y}(z) = 0$ with the initial conditions $\boldsymbol{y}(0) = \boldsymbol{b}_1$ and $\boldsymbol{y}'(0) = \boldsymbol{b}_2$, with the solution

$$\boldsymbol{y}(z) = f_1(g_1(A)z)\boldsymbol{b}_1 + g_2(A)f_2(g_1(A)z)\boldsymbol{b}_2,$$

  where $f_1(z) = \cos(z)$, $f_2(z) = \sin(z)$, $g_1(z) = \sqrt{z}$ and $g_2(z) = 1/\sqrt{z}$.

Generally, consider a scalar function $f : \mathbb{C} \to \mathbb{C}$ of a complex variable. We are looking for a generalization of f to a mapping from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{n\times n}$, $n \in \mathbb{N}$, i.e. a correct definition of a function of a matrix $f(A)$ for a given matrix $A \in \mathbb{C}^{n\times n}$. There are several ways how to define such matrix function, [19, 20, 24]. Sometimes it is possible to substitute A for a scalar variable, e.g. when f is a polynomial or rational function, or if it is possible to expand a function f into a convergent serie. Some of the most used definitions, that can be applied to a general function f, are definition via the spectral decomposition, via Hermite interpolation polynomials, or via Cauchy integral representation formula. We

discuss equivalence of these definitions, [25, 29], and summarize some useful properties of matrix functions in the first chapter.

Numerical methods for computation of approximations to matrix functions have been widely studied, [4, 7, 19, 20, 24, 25, 26, 28, 30, 34, 35, 36]. We summarize some of them in the second chapter. Schur decomposition [20, 34], truncate Taylor series [25, 36], rational Padé approximation [19, 23, 30], scaling and squaring method [28] and Chebyshev approximation [7, 44] belong among direct methods. Matrix iterations [24, 25, 27] can be determined only for specific functions. We mention matrix iterations based on the Newton method, for approximation of matrix square root and matrix sign function.

In many applications, only evalution of $f(A)\boldsymbol{b}$ for an apriory given vector $\boldsymbol{b}$ is required. One possibility is to compute an approximation of $f(A)$ using one of the methods above and then multiply by the vector $\boldsymbol{b}$. Especially when A is large and sparse (which is the case in many applications), the computational cost can be very hight. Thus it may be desirable to compute immediately an approximation of the vector $f(A)\boldsymbol{b}$. In the third chapter, we describe classical methods for approximation of $f(A)\boldsymbol{b}$ of non-Krylov type. These include method using contour integral, [9, 25], quadrature rule [25] and polynomial least squares approximation. In the polynomial method, the function f is first approximated by a spline and then the spline is approximated in the sense of least squares using basis of orhonormal polynomials, generated using the three-term Stieltjes reccurence, [6, 50].

Other efficient methods for approximation of $f(A)\boldsymbol{b}$ are Krylov subspace methods, studied in the fourth chapter. We start with the standart Krylov subspace method, [3, 12], where after m iterations, $m << n$, an approximation to $f(A)\boldsymbol{b}$ is computed by projecting the original problem onto a Krylov subspace of dimension m. When A is large, with growing number of iterations, the computation cost of the standart Krylov subspace method may increase. It can be improved by restarting the method after a predefined number of iterations. After each restart it is possible to update the approximation efficiently. This method is called the restarted Krylov subspace method, [2, 15]. The convergence of the method can be accelerated using deflation, [16]. In [1], a special case of the restarted Krylov subspace method was introduced, where the restart length is equal to one. This method is called the method of the steepest descent for matrix functions. The last method studied here is the extended Krylov subspace method, [13, 31], where the approximation is found on an extended Krylov subspace containing information not only about the matrix A, but also about its inverse A^{-1}. In advance, we mention a modification of the Krylov subspace methods based on the rational approximation of the function f, [2, 19].

We conclude this thesis by numerical experiments. In the Bachelor thesis [52], we already compared some of the methods for approximation of $f(A)$ and the standart Krylov subspace method. Here we concentrate on comparision of computational time and convergence behaviour of the methods for evaluation of $f(A)\boldsymbol{b}$.

Chapter 1

Theoretical background

'All theory, dear friend, is gray, but the golden tree of life springs ever green.'
Johann Wolfgang von Goethe

1.1 Notation and auxiliary definitions

$A = (a_{ij})_{i,j=1}^{n} \in \mathbb{C}^{n\times n}$	matrix whose function f we want to compute
$\boldsymbol{b} \in \mathbb{C}^n$	vector
$\Lambda(A) = \{\lambda_1, \ldots, \lambda_n\}$	spectrum of matrix A
$W(A) = \left\{\boldsymbol{v}^H A\boldsymbol{v} : \|\boldsymbol{v}\| = 1, \boldsymbol{v} \in \mathbb{C}^n\right\}$	field of values of matrix A
$J_A = Z^{-1}AZ$	Jordan canonical form of matrix A
$\widetilde{\lambda_1}, \ldots, \widetilde{\lambda_\ell}$	distinct eigenvalues of A with multiplicity j_i, $i = 1, \ldots, \ell$
$\widehat{\lambda_1}, \ldots, \widehat{\lambda_d}$	eigenvalues belonging to a Jordan block $J_i(\widehat{\lambda}_i)$ of dimension m_i
$\psi(A)$	minimal polynomial of matrix A
$\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}$	trace of matrix A
$\rho(A) = \max_{i=1,\ldots,\ell}\{\|\widetilde{\lambda}_i\|\}$	spectral radius of matrix A
$\mathrm{Re}(z)$	real part of a complex number z
$\mathrm{Im}(z)$	imaginary part of a complex number z
$\mathscr{P}_m$	set of all polynomials of degree not exceeding m
$\mathscr{R}_{pq}$	set of all rational functions, with the nominator and the denominator of degrees at most p and q
$\mathscr{K}_m(A, \boldsymbol{b})$	mth Krylov subspace with respect to the matrix A and vector $\boldsymbol{b}$
$\widetilde{\mathscr{K}_{2m}}(A, \boldsymbol{b})$	extended Krylov subspace with respect to the matrix A, its inverse A^{-1} and vector $\boldsymbol{b}$

Definition 1.1.1 *We say, that* ***function is defined on a spectrum of*** $A \in \mathbb{C}^{n\times n}$ *if all the derivatives* $f^{(k)}(\widetilde{\lambda}_i), \quad k = 0, \ldots, j_i, i = 1, \ldots, \ell$ *exists.*

Definition 1.1.2 *Matrix is* ***nonderogatory*** *when it is not derogatory, i.e. one eigenvalue of A belongs to one Jordan block.*

Definition 1.1.3 *We say, that matrix* $A \in \mathbb{C}^{n\times n}$ *is* ***derogatory****, when it has a multiple eigenvalue to which belongs more than one Jordan block.*

Definition 1.1.4 ***Primary matrix function*** *is obtained, when we take the same branches of solution for different Jordan blocks* $J_k(\lambda_k)$*.*

Definition 1.1.5 ***Nonprimary matrix function*** *is a function f, which is not primary. We cannot express this function as a polynomial.*

Definition 1.1.6 *We say, that the vectors* $span\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m\} = \mathscr{K}_m(A, \boldsymbol{b})$ *form an* ***ascending basis*** *of* $\mathscr{K}_m(A, \boldsymbol{b})$ *if and only if* $span\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_j\} = \mathscr{K}_m(A, \boldsymbol{b})$ *for all* $j = 1, \ldots, m$*.*

1.2 Definitions of matrix functions

There are many ways of defining matrix functions. In this section we will focus on the most significant definitions. For a given matrix $A \in \mathbb{C}^{n\times n}$ with the spectrum $\Lambda(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$, we are looking for a generalization of a scalar function $f : \mathbb{C} \to \mathbb{C}$ of complex variable z to a mapping from $\mathbb{C}^{n\times n}$ to $\mathbb{C}^{n\times n}$.

If f is a polynomial of degree n,

$$f = \sum_{k=0}^{n} a_k z^k,$$

then we can simply substitute the matrix A for the scalar variable z, i.e.

$$f(A) := \sum_{k=0}^{n} a_k A^k.$$

Assume that it is possible to expand the function f into an infinite power-series (e.g. Taylor series),

$$f(z) = \sum_{k=0}^{\infty} a_k z^k, \tag{1.1}$$

converging for $|z| < r$, where $r > 0$ is called **radius of convergence**. Then we can define

$$f(A) := \sum_{k=0}^{\infty} a_k A^k \tag{1.2}$$

and the series (1.2) converges if $|\lambda_i| < r, i = 1, \ldots, n$, see [19], pp. 2. It may happen, that the radius of convergence is not large enough. Because of that it is often usefull to use another definitions of matrix functions - via spectral decomposition, Hermite interpolation polynomials or Cauchy integral representation formula.

Spectral decomposition

Let f be defined on the spectrum of A and let A have a Jordan canonical form $J_A = Z^{-1}AZ$ with nonsingular matrix Z, where

$$J_A = \mathrm{diag}(J_1(\widehat{\lambda}_1), \ldots, J_d(\widehat{\lambda}_d)),$$

and the Jordan block

$$J_k(\widehat{\lambda}_k) = \begin{bmatrix} \widehat{\lambda}_k & 1 & \ldots & 0 \\ & \ddots & & \vdots \\ & & \ddots & 1 \\ & & & \widehat{\lambda}_k \end{bmatrix},$$

$k = 1, \ldots, d$, has the dimension m_k, $\sum_{k=1}^{d} m_k = n$. From the property of matrix functions $f(ZJ_AZ^{-1}) = Zf(J_A)Z^{-1}$, we obtain the definition

$$f(A) := Zf(J_A)Z^{-1} = Z\mathrm{diag}(f(J_1(\widehat{\lambda}_1)), \ldots, f(J_d(\widehat{\lambda}_d)))Z^{-1}, \tag{1.3}$$

where

$$f(J_k(\widehat{\lambda}_k)) = \begin{bmatrix} f(\widehat{\lambda}_k) & f'(\widehat{\lambda}_k) & \cdots & \frac{f^{(m_k-1)}(\widehat{\lambda}_k)}{(m_k-1)!} \\ & \ddots & \ddots & \vdots \\ & & f(\widehat{\lambda}_k) & f'(\widehat{\lambda}_k) \\ & & & f(\widehat{\lambda}_k) \end{bmatrix}, \tag{1.4}$$

see [20], pp. 557-559. Note that if A is a derogatory matrix, then the matrices Z and J_A are not uniquely defined. However, the resulting function doesn't depend on the choice of Z and J_A.

Cauchy integral representation formula

Perhaps the most elegant definition of a function of a matrix is a generalization of the Cauchy integral theorem. Let Γ be a closed curve which encloses the spectrum of A, $\Lambda(A)$. Then

$$f(z) = \frac{1}{2\pi i} \int_\Gamma \frac{f(t)}{t-z} dt$$

and the Cauchy integral representation formula for a matrix function is defined as

$$f(A) := \frac{1}{2\pi i} \int_\Gamma f(t)(tI - A)^{-1} dt. \tag{1.5}$$

Hermite interpolation polynomials

We can also define matrix functions using Hermite interpolation polynomials. First, we introduce some theoretical background on minimal polynomial of a matrix $A \in \mathbb{C}^{n\times n}$ and matrix polynomials, see [25], pp. 4-7. **Minimal polynomial** ψ of a matrix A is defined as a polynomial of the lowest degree, for which it holds $\psi(A) = 0$.

Lemma 1.2.1 *Minimal polynomial ψ divides any polynomial p, for which $p(A) = 0$.*

Proof. We will prove this lemma by contradiction. Suppose, that p is not divisible by ψ and so it can be written as $p = \psi q + \tilde{r}$, where the degree of the remainder $\tilde{r}$ is less than that of ψ. Then $0 = p(A) = \psi(A)q(A) + \tilde{r}(A) = \tilde{r}(A)$ and that contradicts the minimality of the degree of ψ unless $\tilde{r} = 0$. Thus $\tilde{r} = 0$ and ψ divides p.

□

Considering the Jordan canonical form of A, we can see, that the minimal polynomial is of the form

$$\psi(t) = \prod_{i=0}^{\ell} (t - \tilde{\lambda}_i)^{n_i}.$$

Lemma 1.2.2 *For two polynomials p and q, $p(A) = q(A)$ if and only if p and q take the same values on the spectrum of A.*

Proof. First, suppose that $p(A) = q(A)$. Then $d := p - q$ is zero at A. It implies that d is divisible by the minimal polynomial of A and so d takes only the value zero on the spectrum of A. That means that p and q take the same values on the spectrum of A.

Conversely, suppose that p and q take the same values on the spectrum of A. Then $d := p - q$ is zero on the spectrum of A and so d must be divisible by the minimal polynomial ψ. In other words, d can be written as a product $d = \psi\tilde{r}$, where $\tilde{r}$ is some polynomial. Then $d(A) = \psi(A)\tilde{r}(A) = 0$, from which follows that $p(A) = q(A)$.

□

We have shown, that matrix polynomials are completely determined by the values of f on the spectrum of A. Now, we can introduce the definition via Hermite polynomials. Let $\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_\ell$ be distinct eigenvalues of A, where n_i is the index of $\widetilde{\lambda}_i$, i.e., the dimension of the largest Jordan block which belongs to the eigenvalue $\widetilde{\lambda}_i,\ i = 1, \ldots, \ell$. Let $r(A)$ be a Hermitian interpolation polynomial of degree less than

$$\sum_{i=1}^{\ell} n_i = \deg \psi$$

satisfying the interpolation conditions

$$r^{(j)}(\widetilde{\lambda}_i) = f^{(j)}(\widetilde{\lambda}_i), j = 0, \ldots, n_i - 1, \quad i = 1, \ldots, \ell. \tag{1.6}$$

If f is defined on the spectrum of A, then we define

$$f(A) := r(A). \tag{1.7}$$

We note, that the required Hermite interpolating polynomial is of the form

$$r(z) = \sum_{i=1}^{\ell} \left[\left(\sum_{j=0}^{n_i-1} \phi_i^{(j)}(\lambda_i)(z - \widetilde{\lambda}_i)^j \right) \prod_{j \neq i} (z - \widetilde{\lambda}_j)^{n_j} \right],$$

where $\phi(z) = f(z) \cdot \left[\prod_{j \neq i} (z - \widetilde{\lambda}_j)^{n_j} \right]^{-1}$, see [25], pp. 6.

Remark 1.2.3 *If polynomial q satisfies interpolation conditions in the sense of (1.6) and some additional interpolation conditions (at the same or different $\widetilde{\lambda}_i$), then r and q take the same values on the spectrum of A. According to Lemma (1.2.2), it holds that $q(A) = r(A) = f(A)$.*

Equivalence of definitions

Definitions (1.3) and (1.7) are equivalent. They are also equivalent with the definition (1.5) if and only if f is analytic in a complex plane containing the spectrum of A. In following, we will prove this equivalence according to [25], pp. 8 and [29], pp. 426-428.

Theorem 1.2.4 *The definition (1.3) and the definition (1.7) are equivalent.*

Proof. From definition (1.7) we have $r(A) = f(A)$ for r satisfying the conditions (1.6). If $A = ZJ_AZ^{-1}$, then

$$\begin{aligned} f(A) &= r(A) = r(ZJ_AZ^{-1}) = Zr(J_A)Z^{-1} \\ &= Z\mathrm{diag}\left(r(J_1(\widehat{\lambda}_1)), \ldots, r(J_d(\widehat{\lambda}_d))\right) Z^{-1}. \end{aligned}$$

Because $r(J_k(\hat{\lambda}_k)),\ k = 1, \ldots, d$ is determined by the values of r on the spectrum of A, and these values form a subset of the values of r on the spectrum of A, from the Remark 1.2.3 and from the form of the Hermite interpolation polynomial it follows, that $r(J_k(\hat{\lambda}_k))$ is equal to (1.4).

□

To complete the proof of equivalence of the definitions, we will first need to formulate an auxiliary proposition, see [29], pp. 425-426:

Proposition 1.2.5 *Let D be a simply connected open subset of $\mathbb{C}$ or an open interval, let the functions f, g be continuous in A, $\Lambda(A) \subset D$. Then $f(A) = g(A)$ for all A, $\Lambda(A) \subset D$ if and only if $f(A) = g(A)$ for all diagonalizable matrices A, $\Lambda(A) \subset D$.*

The proof is based on the fact, that the set of all diagonalizable matrices is dense in the set of all matrices. Now we can prove the last equivalence, see [29], pp. 427-428.

Theorem 1.2.6 *Let f be analytic on a simply connected open subset $D \subset \mathbb{C}$. Then the definition (1.3) is equivalent with the definition (1.5).*

Proof. According to Proposition 1.2.5 it is enough to prove this theorem for diagonalizable matrices. Let the matrix A be diagonalizable, $A = Z\mathrm{diag}\,(\lambda_1, \ldots, \lambda_n)\, Z^{-1}$. Using elementary properties of matrix functions,

$$\begin{aligned} f(A) &= \frac{1}{2\pi i}\int_\Gamma f(t)(tI - A)^{-1}dt \\ &= \frac{1}{2\pi i}\int_\Gamma f(t)\left(tI - Z\mathrm{diag}\,(\lambda_1, \ldots, \lambda_n)\, Z^{-1}\right)^{-1} dt \\ &= Z\mathrm{diag}\left(\frac{1}{2\pi i}\int_\Gamma f(t)\,(tI - \lambda_1)^{-1}\,dt, \ldots, \frac{1}{2\pi i}\int_\Gamma f(t)\,(tI - \lambda_n)^{-1}\,dt\right) Z^{-1} \\ &= Z\mathrm{diag}\,(f(\lambda_1), \ldots, f(\lambda_n))\, Z^{-1}, \end{aligned}$$

and we obtained (1.3).

□

Useful properties of matrix functions

A good definition leads to applicable properties, we summarize the most important of them:

- $f(A)$ commutes with A, $f(A)A = Af(A)$.
- $f(A^T) = f(A)^T$.
- For any nonsingular matrix X, it holds that $f(XAX^{-1}) = Xf(A)X^{-1}$.
- If A is diagonalizable, i.e. $Z^{-1}AZ = D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$, then $f(A) = Z\mathrm{diag}(f(d_1), f(d_2), \ldots, f(d_n))Z^{-1}$.
- $f(\mathrm{diag}(A_{11}, A_{22}, \ldots, A_{nn})) = \mathrm{diag}(f(A_{11}), f(A_{22}), \ldots, f(A_{nn}))$.
- Let f and g be functions defined on the spectrum of A.
  - If $h(z) = f(z) + g(z)$, then $h(A) = f(A) + g(A)$.
  - If $h(z) = f(z)g(z)$, then $h(A) = f(A)g(A)$.
- Let h be defined on the spectrum of A and let the values $g^{(j)}(h(\tilde{\lambda}_i))$, $j = 0, \ldots, n_i - 1$, $i = 1, \ldots, \ell$ exist. Then $f(z) = g(h(z))$ is defined on the spectrum of A and $f(A) = g(h(A))$.
- Let $\Omega \subseteq \mathbb{C}$ be an open subset such that each connected component of Ω is closed under the conjugation, $\mathcal{D} = \{A \in \mathbb{C}^{n\times n} : \Lambda(A) \subseteq \Omega\}$, then $f(A^H) = f(A)^H$ and $f(\overline{A}) = \overline{f(A)}$ for all $A \in \mathcal{D}$.

For more details, see [25], pp. 10-14.

1.3 Examples of elementary matrix functions

Further, we summarize some elementary matrix functions such as matrix exponential, goniometric functions, logarithm, square root and signum. We discuss the ways of definitions and some of their properties.

Matrix exponential

Exponential e^A of a matrix $A \in \mathbb{C}^{n\times n}$ can be defined using the Taylor series:

$$e^A := I + A + \frac{A^2}{2!} + \cdots + \frac{A^k}{k!} + \cdots .$$

This serie always converges and the definition is correct, see [36], pp. 1. As it is shown in [25], pp. 233-238, properties of scalar exponential e^z generally cannot be extended to the matrix function e^A. For example, the equality $e^{(A+B)t} = e^{At}e^{Bt}$ for two complex matrices

A, B holds only if the matrices commute, $AB = BA$. Thus, e.g., $e^A e^{-A} = I$.

Computation of matrix exponential is needed, e.g., while solving systems of time-dependent differential equations in the form

$$\boldsymbol{y}'(z) = A\boldsymbol{y}(z) + \boldsymbol{b}(z), \qquad \boldsymbol{y}(0) = \boldsymbol{y}_0. \tag{1.8}$$

Analytical solution of (1.8) is

$$\boldsymbol{y}(z) = e^{zA}\boldsymbol{y}_0 + \int_0^z e^{(t-z)A}\boldsymbol{b}(t)dt,$$

where A is a negative semidefinite matrix.

Matrix sign function

Scalar function $\mathrm{sign}(z)$ for $z \in \mathbb{C} \setminus \{0\}$ is defined by the formula

$$\mathrm{sign}(z) = \begin{cases} 1 & \mathrm{Re}(z) > 0 \\ -1 & \mathrm{Re}(z) < 0 \end{cases}.$$

Assume for a moment that $J_A = Z^{-1}AZ \in \mathbb{C}^{n\times n}$, where

$$J_A = \begin{bmatrix} J_A^{(1)} & 0 \\ 0 & J_A^{(2)} \end{bmatrix},$$

$J_A^{(1)} \in \mathbb{C}^{p\times p}$, $J_A^{(2)} \in \mathbb{C}^{q\times q}$, all eigenvalues of $J_A^{(1)}$ have negative real part and all eigenvalues of $J_A^{(2)}$ have positive real part, $p + q = n$. Then we can define

$$\mathrm{sign}(A) := Z \begin{bmatrix} -I_p & 0 \\ 0 & I_q \end{bmatrix} Z^{-1}.$$

Matrix sign function can be also defined in other ways. For a scalar function it holds that

$$\mathrm{sign}(z) = z/(z^2)^{1/2},$$

and thus the matrix function can be defined as

$$\mathrm{sign}(A) := A(A^2)^{-1/2}. \tag{1.9}$$

It can also be set

$$\mathrm{sign}(A) := \frac{2}{\pi} A \int_0^\infty \left(t^2 I + A^2\right)^{-1} dt. \tag{1.10}$$

From these representations of a matrix sign function $S :=\mathrm{sign}(A)$ it can be shown, see [25], pp. 107-108, that S is diagonalizable, its eigenvalues are ± 1 and $S^2 = I$. If A is symmetric and positive definite, then $S = I$.

Matrix square root

Square root of a matrix A is a matrix X such that $X^2 = A$. It can be shown, see [25], pp. 20, that if none of the eigenvalues lie in $\mathbb{R}^-$, then there exists a unique matrix X such that $X^2 = A$ and all its eigenvalues have positive real part.

Now, we will derive an integral representation formula for the matrix square root. We customize (1.10) for a certain specific block matrix,

$$\begin{aligned}
\operatorname{sign}\left(\begin{bmatrix} O & A \\ I & O \end{bmatrix}\right) &\overset{(1.10)}{=} \frac{2}{\pi}\begin{bmatrix} O & A \\ I & O \end{bmatrix}\int_0^\infty \left(t^2 I + \begin{bmatrix} O & A \\ I & O \end{bmatrix}\begin{bmatrix} O & A \\ I & O \end{bmatrix}\right)^{-1} dt \\
&= \frac{2}{\pi}\begin{bmatrix} O & A \\ I & O \end{bmatrix}\int_0^\infty \left(\begin{bmatrix} (t^2 I + A)^{-1} & O \\ O & (t^2 I + A)^{-1} \end{bmatrix}\right) dt \\
&= \frac{2}{\pi}\begin{bmatrix} O & A \\ I & O \end{bmatrix}\begin{bmatrix} \int_0^\infty (t^2 I + A)^{-1} dt & O \\ O & \int_0^\infty (t^2 I + A)^{-1} dt \end{bmatrix} \\
&= \frac{2}{\pi}\begin{bmatrix} O & A\int_0^\infty (t^2 I + A)^{-1} dt \\ \int_0^\infty (t^2 I + A)^{-1} dt & O \end{bmatrix}.
\end{aligned} \tag{1.11}$$

Further, from (1.9) we have

$$\operatorname{sign}\left(\begin{bmatrix} O & A \\ I & O \end{bmatrix}\right) = \begin{bmatrix} O & A^{1/2} \\ A^{-1/2} & O \end{bmatrix}. \tag{1.12}$$

Finally, comparing (1.11) with (1.12) we can see that

$$\sqrt{A} = \frac{2}{\pi} A \int_0^\infty (t^2 I + A)^{-1} dt.$$

Matrix square root is not uniquely determined. For example, an identity matrix I_n of dimension $n \times n$ has 2^n diagonal square roots, with elements ± 1 on the diagonal, see [24], pp. 7-8. Only two of them are primary: I_n and $-I_n$. Other, symmetric and nonprimary square roots are unit permutation matrices and so-called Householder matrices in a form $I_n - 2\frac{\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}}$, where $\boldsymbol{v}$ is an arbitrary nonzero vector. Unsymmetric square roots of I_n are in the form XDX^{-1}, where X is nonsingular regular nonorthogonal matrix and D=diag$(\pm 1) \neq \pm I$.

Matrix square root is used for example for computation of a polar decomposition, see [25], pp. 193-220.

Matrix logarithm

Logarithm of a matrix A is a matrix X for which $e^X = A$. If the spectral radius of A satisfies $\rho(A) = \max\limits_{i=1,\ldots,\ell}\{|\tilde{\lambda}_i|\} < 1$, we can define matrix logarithm using the Taylor serie

$$\log(I + A) := A - \frac{A^2}{2} + \cdots + (-1)^{k+1}\frac{A^k}{k} + \cdots.$$

Equivalent definition is via the integral representation formula,

$$\log(A) = \int_0^1 (A - 1)\left[t(A - I) + I\right]^{-1} dt.$$

If any of the eigenvalues of A does not lie on a negative real axis, then there exists a unique matrix logarithm X of the matrix A, whose eigenvalues lie in set $\{z \in \mathbb{C} | -\pi < \mathrm{Im}(z) < \pi\}$, see [25], pp. 269. We call this logarithm **principal logarithm** of A. If the logarithm is defined on the spectrum of A, then $\exp(\log(A)) = A$. But generally $\log(\exp(A)) = A$ does not hold, see [24], pp. 10. Logarithm of a matrix can be used to compute a determinant of A, $\det(A) = \exp(\mathrm{tr}(\log(A))$.

Matrix sine and cosine

Sine and cosine of a matrix A can be defined using Taylor series

$$\sin(A) := A - \frac{A^3}{3!} + \cdots + \frac{(-1)^k A^{2k+1}}{2k+1!} + \cdots,$$

$$\cos(A) := I - \frac{A^2}{2!} + \cdots + \frac{(-1)^k A^{2k}}{2k!} + \cdots.$$

Some formulas, which hold for scalar sine and cosine, hold also for matrix sine and cosine, i.e. formulas for double argument $\cos(2A) = 2\cos^2(A) - I$, $\sin(2A) = 2\sin(A)\cos(A)$ and $\cos^2(A) + \sin^2(A) = I$, see [25], pp. 287-288.

Goniometric matrix functions are used to solve differential equations of the second order

$$\boldsymbol{y}''(z) + A\boldsymbol{y}(z) = 0, \qquad \boldsymbol{y}(0) = \boldsymbol{y}_0, \qquad \boldsymbol{y}'(0) = \boldsymbol{y}_1,$$

whose analytical solution is

$$\boldsymbol{y}(z) = \cos\left(\sqrt{A}z\right)\boldsymbol{y}_0 + \left(\sqrt{A}\right)^{-1}\sin\left(\sqrt{A}z\right)\boldsymbol{y}_1,$$

where $\sqrt{A}$ is an arbitrary square root of A.

Chapter 2

Numerical methods for approximation of $f(A)$

'All exact science is dominated by the idea of approximation.'
Bertrand Russell

In this chapter we summarize basic numerical methods for approximation of matrix functions $f(A)$. These methods are useful usually only for small dense problems, for large problems they can be inefficient and the computational cost may become very large.

These methods were widely studied and compared in the literature, see also the Bachelor thesis, see [52], pp. 23-30.

2.1 Schur decomposition

Schur decomposition of a matrix A has the form

$$A = QTQ^H, \tag{2.1}$$

where $Q \in \mathbb{C}^{n\times n}$ is a unitary matrix and $T \in \mathbb{C}^{n\times n}$ is an upper triangular matrix. Using the basic property of matrix functions, we obtain $f(A) = f(QTQ^H) = Qf(T)Q^H$. The question is, how to compute the function of a triangular matrix T. One possibility is given by the following theorem, see [34], pp. 7-9:

Theorem 2.1.1 *Let $T = (t_{ij})_{i,j=1}^n \in \mathbb{C}^{n\times n}$ be upper triangular matrix, where $\widehat{\lambda}_i = t_{ii}$, $i = 1, \ldots, n$, are its eigenvalues and let f be defined on the spectrum of T. Then $F = f(T) = (f_{ij})_{i,j=1}^n$ is upper triangular, $f_{ij} = 0$ for $i > j$; $f_{ij} = f(\widehat{\lambda}_i)$ for $i = j$; and finally, for $i < j$*

$$f_{ij} = \sum_{(s_0,\cdots,s_k)\in S_{ij}} t_{s_0,s_1} t_{s_1,s_2} \cdots t_{s_{k-1},s_k} f\left[\widehat{\lambda}_{s_0}, \ldots, \widehat{\lambda}_{s_k}\right], \tag{2.2}$$

where S_{ij} is the set of all strictly increasing sequences of integers that start at i and end at j, and $f\left[\widehat{\lambda}_{s_0}, \cdots, \widehat{\lambda}_{s_k}\right]$ is the kth order divided difference of f at $\widehat{\lambda}_{s_0}, \ldots, \widehat{\lambda}_{s_k}$.

Note. We will show later, that

$$f_{ij} = t_{ij}\frac{f_{jj} - f_{ii}}{t_{jj} - t_{ii}} + \sum_{k=i+1}^{j-1} \frac{t_{ik}f_{kj} - f_{ik}t_{kj}}{t_{jj} - t_{ii}}, \quad i \neq j, \quad f_{ii} = t_{ii}, \quad i = j. \tag{2.3}$$

We will use this equality in the proof of Theorem 2.1.1.

Proof. $f_{ij} = 0$ for $i > j$ and $f_{ij} = f(\widehat{\lambda}_i)$ for $i = j$ follows immediately from (2.3). Now we will prove the case (2.2). We first assume that $\lambda_i, i = 1, \ldots, n$ are distinct. Setting $j = i + 1$ in (2.3) we obtain

$$f_{i,i+1} = t_{i,i+1}\frac{f_{i+1,i+1} - f_{i,i}}{\widehat{\lambda}_{i+1} - \widehat{\lambda}_i} = t_{i,i+1} f\left[\widehat{\lambda}_i, \widehat{\lambda}_{i+1}\right].$$

This proves (2.2) for $1 = j - i$. The rest will be proved by induction. We assume, that (2.2) holds for $i, j = 1, \ldots, n$, i.e., $1 \leq j - i \leq n - 1$, $n \geq 2$, and we will show that it holds also for $1 \leq j - i \leq n$. Without loss of generality, it suffices to set $i = 1, j = n$ and show

$$f_{1n} = \sum_{(s_0,\ldots,s_k)\in S_{1n}} t_{s_0 s_1} \cdots t_{s_{k-1}s_k} f\left[\widehat{\lambda}_{s_0}, \ldots, \widehat{\lambda}_{s_k}\right].$$

From (2.3), we have

$$f_{1n} = t_{1n} f\left[\widehat{\lambda}_1, \widehat{\lambda}_n\right] + \sum_{q=2}^{n-1} \frac{f_{1q}t_{qn} - t_{1q}f_{qn}}{\widehat{\lambda}_n - \widehat{\lambda}_1} \tag{2.4}$$

and by the inductive hypotheses, we have for $q = 2, \ldots, n - 1$

$$f_{1q} = \sum_{(s_0,\ldots,s_k)\in S_{1q}} t_{s_0 s_1} \cdots t_{s_{k-1}s_k} f\left[\widehat{\lambda}_{s_0}, \ldots, \widehat{\lambda}_{s_k}\right]$$

and

$$f_{qn} = \sum_{(s_0,\ldots,s_k)\in S_{qn}} t_{s_0 s_1} \cdots t_{s_{k-1}s_k} f\left[\widehat{\lambda}_{s_0}, \ldots, \widehat{\lambda}_{s_k}\right].$$

We customize these expressions. First,

$$\begin{aligned} \sum_{q=2}^{n-1} f_{1q}t_{qn} &= \sum_{q=2}^{n-1} \sum_{(s_0,\ldots,s_k)\in\ S_{1q}} t_{s_0 s_1} \cdots t_{s_{k-1}s_k} t_{qn} f\left[\widehat{\lambda}_{s_0}, \ldots, \widehat{\lambda}_{s_k}\right] \\ &= \sum_{\substack{(s_0,\ldots,s_k)\in\ S_{1n} \\ k>1}} t_{s_0 s_1} \cdots t_{s_{k-1}s_k} f\left[\widehat{\lambda}_{s_0}, \ldots, \widehat{\lambda}_{s_{k-1}}\right] \end{aligned} \tag{2.5}$$

and in a similar way

$$\sum_{q=2}^{n-1} t_{1q}f_{qn} = \cdots = \sum_{\substack{(s_0,\ldots,s_k)\in S_{1n} \\ k>1}} t_{s_0 s_1} \cdots t_{s_{k-1}s_k} f\left[\widehat{\lambda}_{s_1}, \ldots, \widehat{\lambda}_{s_k}\right]. \tag{2.6}$$

Inserting (2.5) and (2.6) into (2.4), we obtain

$$f_{1n} = t_{1n} f\left[\widehat{\lambda}_1, \widehat{\lambda}_n\right] + \sum_{\substack{(s_0,\ldots,s_k)\in S_{1n} \\ k>1}} t_{s_0 s_1} \cdots t_{s_{k-1} s_k} \frac{f\left[\widehat{\lambda}_{s_0}, \ldots, \widehat{\lambda}_{s_{k-1}}\right] - f\left[\widehat{\lambda}_{s_1}, \ldots, \widehat{\lambda}_{s_k}\right]}{\widehat{\lambda}_n - \widehat{\lambda}_1} =$$
$$= \sum_{(s_0,\ldots,s_k)\in S_{1n}} t_{s_0 s_1} \cdots t_{s_{k-1} s_k} f\left[\widehat{\lambda}_{s_0}, \ldots, \widehat{\lambda}_{s_k}\right],$$

and the theorem is proved for distinct eigenvalues.

Now assume that T has multiple eigenvalues. We can write

$$T = \mathrm{diag}(\widehat{\lambda}_1, \ldots, \widehat{\lambda}_n) + N,$$

where

$$N = \begin{bmatrix} 0 & t_{12} & t_{13} & \cdots & t_{1n} \\ 0 & 0 & t_{23} & \cdots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & t_{n-1,n} \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Let us define a sequence of upper triangular matrices

$$T_q = \mathrm{diag}(\widehat{\lambda}_1^{(q)}, \ldots, \widehat{\lambda}_n^{(q)}) + N$$

such that $\lim_{q\to\infty} T_q = T$, and each T_q has distinct eigenvalues $\widehat{\lambda}_1^{(q)}, \ldots, \widehat{\lambda}_n^{(q)}$. We can choose $\widehat{\lambda}_i^{(q)}$, $i = 1, \ldots, n$, to be in the interior of the contour Γ in (1.5). Thus,

$$f(T) = \frac{1}{2\pi i} \int_\Gamma f(t) \left(tI - T\right)^{-1} dt = \lim_{q\to\infty} \frac{1}{2\pi i} \int_\Gamma f(t) \left(tI - T_q\right)^{-1} dt = \lim_{q\to\infty} f(T_q).$$

Further, from the continuity

$$\lim_{q\to\infty} f\left[\widehat{\lambda}_{s_0}^{(q)}, \ldots, \widehat{\lambda}_{s_k}^{(q)}\right] = f\left[\widehat{\lambda}_{s_0}, \ldots, \widehat{\lambda}_{s_k}\right],$$

for $(s_0, \ldots, s_k) \in S_{ij},\ i < j$. Summarizing

$$f_{ij} = \lim_{q\to\infty} f_{ij}^{(q)} = \lim_{q\to\infty} \sum_{s_0,\ldots,s_k\in S_{ij}} t_{s_0 s_1} \cdots t_{s_{k-1} s_k} f\left[\widehat{\lambda}_{s_0}^{(q)}, \ldots, \widehat{\lambda}_{s_k}^{(q)}\right]$$
$$= \sum_{(s_0,\ldots,s_k)\in S_{ij}} t_{s_0 s_1} \cdots t_{s_{k-1} s_k} f\left[\widehat{\lambda}_{s_0}, \ldots, \widehat{\lambda}_{s_k}\right].$$

and that proves (2.2) also for matrix with multiple eigenvalues.

□

Computation of $f(T)$ using this theorem requires $O(2^n)$ elementary operations. More efficient method is the Parlett Method, see [25], pp. 85-86, which is based on the commutativity of matrix and its matrix function, $T = (t_{ij})_{i,j=1}^n$ and $F = f(T) = (f_{ij})_{i,j=1}^n$. From $TF = FT$ we obtain

$$f_{ij} = t_{ij}\frac{f_{jj} - f_{ii}}{t_{jj} - t_{ii}} + \sum_{k=i+1}^{j-1} \frac{t_{ik}f_{kj} - f_{ik}t_{kj}}{t_{jj} - t_{ii}}, \quad i \neq j, \quad f_{ii} = t_{ii}, \quad i = j. \tag{2.7}$$

A problem can occur, when the matrix A has multiple, respectively close eigenvalues. Then the denominator in (2.7) is zero, respectively very small, and the method cannot be used. In [20], pp. 560-561 and [25], pp. 86-87, 221-231 a modification of this method, based on clustering close or multiple eigenvalues in blocks along the diagonal of T is described. Block variant of (2.7), i.e. system of Sylvester's equations, is obtained. This system can be solved using the so-called Bartels-Steward algorithm, that is described in [20], pp. 365-366.

2.2 Truncated Taylor series

If the function has a Taylor expansion

$$f(A) = \sum_{k=0}^{\infty} \frac{f^{(k)}(A)}{k!} A^k,$$

then we can simply approximate $f(A)$ by truncating this serie,

$$f(A) \approx T_q(A) := \sum_{k=0}^{q} \frac{f^{(k)}(A)}{k!} A^k$$

for some parameter $q \in \mathbb{N}$. This parameter is taken as the smallest number for which

$$fl\left[T_q(A)\right] = fl\left[T_{q+1}(A)\right],$$

where $fl\left[T_q(A)\right]$ is the matrix of the floating point numbers obtained by computing $T_q(A)$ in floating point arithmetics, see [36], pp 9. Note that rounding errors and catastrophic cancellation during adding of two numbers with oposite signs in finite precise arithmetics can occur, and thus aproximating $f(A)$ using truncated Taylor series can be efficient only near origin.

2.3 Rational Padé approximation

The function f can be approximated by the so-called $[p/q]$ Padé approximation

$$f(z) \approx R_{pq} := \frac{N_{pq}(z)}{D_{pq}(z)} = \frac{n_0 + n_1 z + n_2 z^2 + \cdots + n_p z^p}{d_0 + d_1 z + d_2 z^2 + \cdots + d_q z^q},$$

where N_{pq} is a polynomial of degree p and D_{pq} is a polynomial of degree q. Similarly, we can approximate

$$f(A) \approx N_{pq}(A) \cdot D_{pq}(A)^{-1}.$$

Assume that the function can be expanded into a serie (1.1). The coefficients $\{n_i\}_{i=0}^{p}$ and $\{d_i\}_{i=0}^{q}$ can be found by considering the equality

$$D_{pq}(z) \cdot \left(a_0 + a_1 z + a_2 z^2 + \cdots + a_{p+q} z^{p+q}\right) = N_{pq}(z)$$

and equating coefficients of the same powers of z up to $p+q$. For example, in case of the matrix exponential e^A, the polynomials N_{pq} and D_{pq} have the form, see [19], pp. 13,

$$N_{pq}(z) = \sum_{i=0}^{p} \frac{(p+q-i)!p!}{(p+q)!(p-i)!i!} z^i$$
$$D_{pq}(z) = \sum_{i=0}^{p} \frac{(p+q-i)!q!}{(p+q)!(q-i)!i!} (-z)^i,$$

where

$$\begin{aligned} \lim_{p\to\infty} N_{pp} &= e^{A/2}, \\ \lim_{p\to\infty} D_{pp} &= e^{-A/2}. \end{aligned}$$

Diagonal Padé approximation (i.e., the case when $p = q$) is prefered, due to the fact that computation of R_{pq} is not cheaper then computing $R_{p^*p^*}$, where $p^* = \max\{p, q\}$, see [36], pp. 9-10. Because of their stability properties, Padé $[p+1, p]$ and $[p, p]$ approximations are used in the numerical solution of initial value problems with one-step methods.

2.4 Scaling and squaring method and its modifications

Some of the previous methods applied to the matrix exponential e^A can be improved by using scaling and squaring method. Scaling and squaring method is usually applied to rational Padé approximation resp. to truncated Taylor series near to the origin, i.e., when $\|A\|$ is small. The scaling and squaring method for the matrix exponential uses its property

$$e^A = \left(e^{A/2^s}\right)^{2^s}$$

for some $s \in \mathbb{N}$. The approximant of $e^{A/2^s}$ is determined and then the final approximation of e^A is obtained by repeated squaring. Parameter s is chosen such that $e^{A/2^s}$ can be reliably and efficiently computed. It is chosen as small as possible such that it satisfies the condition $\|A\|/2^s < 1$, for more details, see [36], pp. 31-33.

Analogeous method can be applied to another functions - matrix logarithm, matrix sine and cosine function. For matrix logarithm inverse scaling and squaring method can be used as it is described in [25], pp. 273-274. It is based on the identity

$$\log(A) = 2^s \log(A^{1/2^s}).$$

For matrix sine and cosine the following modification is advantageous, see [20], pp. 567. Using the identities

$$\begin{aligned} \cos(2A) &= 2\cos^2(A) - I, \\ \sin(2A) &= 2\sin(A)\cos(A), \end{aligned}$$

we set the initial approximation as $S_0 \approx \sin(A/2^s)$, $C_0 \approx \cos(A/2^s)$ and then for $j = 1, \ldots, s$, we compute the iterations as

$$\begin{aligned} S_j &= 2S_{j-1}C_{j-1}, \\ C_j &= 2C_{j-1}^2 - I. \end{aligned}$$

2.5 Chebyshev approximation

The Chebyshev (or best $L_\infty([a,b])$) approximation to f on an interval $[a,b]$ is a rational function R^* satisfying

$$\max_{z\in[a,b]} |R^*(z) - f(z)| = \min_{R\in\mathscr{R}_{pq}} \max_{z\in[a,b]} |R(z) - f(z)|,$$

where $\mathscr{R}_{pq}$ is a set of all rational functions with the nominator and the denominator of degrees at most p and q, respectively. The best Chebyshev approximation can be constructed using the Remez algorithm, for details see [44], pp. 72-84.

The result can be directly translated for Hermitian matrices with eigenvalues on the negative real axis. The coefficients of R^* were determined for various values of p, q in the work of Cody, Meinardus and Varga, see [7], pp. 50-65.

2.6 Matrix iterations

Matrix iteration is a process

$$X_{k+1} = g(X_k), \quad k = 0, 1, 2, \ldots,$$

where g is usually a polynomial or a rational function. The initial approximation X_0 is usually taken as the identity matrix I or the matrix A itself. Matrix iterations are usually used for matrix sign and matrix square root functions.

Matrix iterations can be based for example on a **Newton method**, that is described in [25], pp. 139. We describe it now for the matrix square root. Suppose that Y is an approximate solution of the matrix equation $X^2 = A$. Let E be a matrix such that $X = Y + E$. Then

$$A = (Y + E)^2 = Y^2 + YE + EY + E^2. \tag{2.8}$$

Setting $X_k = Y, E_k = E$ and $X_{k+1} = X$, $k = 0, 1, 2, \ldots$, we obtain the process

$$\begin{aligned} X_k E_k + E_k X_k &= A - X_k^2 \\ X_{k+1} &= X_k + E_k, \quad k = 0, 1, 2, \ldots \end{aligned} \tag{2.9}$$

For a different choice of X_0 and adding some more presumptions, different methods can be obtained. For example, if we set $X_0 = A$ and suppose, that E_k and X_k commute, we obtain **Newton iteration**,

$$X_{k+1} = \frac{1}{2}\left(X_k + X_k^{-1} A\right), \quad k = 0, 1, 2, \ldots.$$

Newton iteration is quadratically convergent, but it is numerically stable only if A is well conditioned, see [26], pp. 537-549. Another disadvantage is, that at each iteration we need to compute an inverse of the matrix X_k. Further methods, mentioned in [24], pp. 18-19, can be obtained by modifying (2.9).

More stable method is, e.g., the double step **Denman-Beavers iteration**, [27], pp. 227-242,

$$\begin{aligned} X_{k+1} &= \tfrac{1}{2}(X_k + Y_k^{-1}), \quad X_0 = A, \\ Y_{k+1} &= \tfrac{1}{2}(Y_k + X_k^{-1}), \quad Y_0 = I, \quad k = 0, 1, 2, \ldots, \end{aligned}$$

for which

$$\lim_{k\to\infty} X_k = A^{1/2}, \quad \lim_{k\to\infty} Y_k = A^{-1/2}.$$

Another option is the double-step **Meini iteration** [35], pp. 362-376,

$$\begin{aligned} Y_{k+1} &= -Y_k Z_k^{-1} Y_k, \quad Y_0 = I - A, \\ Z_{k+1} &= Z_k + 2Y_{k+1}, \quad Z_0 = 2(I + A), \quad k = 0, 1, 2, \ldots, \end{aligned}$$

for which

$$\lim_{k\to\infty} Y_k = 0, \quad \lim_{k\to\infty} Z_k = 4A^{1/2}.$$

Both Denman-Beavers iteration and Meini iteration are quadratically convergent, but we have to compute inverse matrix in each step. If we want to avoid this, we can use **Schulz iteration** [27], pp. 227-242,

$$\begin{aligned} Y_{k+1} &= \tfrac{1}{2} Y_k (3I - Z_k Y_k), \quad Y_0 = A, \\ Z_{k+1} &= \tfrac{1}{2} (3I - Z_k Y_k) Z_k, \quad Z_0 = I, \quad k = 0, 1, 2, \ldots, \end{aligned}$$

for which

$$\lim_{k\to\infty} X_k = A^{1/2}, \quad \lim_{k\to\infty} Y_k = A^{-1/2},$$

is convergent only if $\|\mathrm{diag}(A - I, A - I)\| < 1$.

For the matrix sign function, iterative methods can be determinded in a similar way as for the matrix square root using the identity $\mathrm{sign}^2(A) = I$, i.e., applying Newton's method on the matrix equation $X^2 = I$. Then the **Newton iteration** for matrix sign function is given by

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}), \quad X_0 = A, \quad k = 0, 1, 2, \ldots.$$

It converges quadratically to $\mathrm{sign}(A)$ if A has no imaginary eigenvalues. Number of iterations can be reduced using the **Newton-scaled iteration**

$$X_{k+1} = \frac{1}{2}(\mu_k X_k + \mu_k^{-1} X_k^{-1}), \quad X_0 = A, \quad k = 0, 1, 2, \ldots$$

for a properly chosen scale parameters μ_k, which is discussed in [4], pp. 127-140. To avoid computing matrix inverses, we can use **Newton-Schulz iteration**

$$X_{k+1} = \frac{1}{2} X_k (3I - X_k^2), \quad X_0 = A,$$

which converges only for $\|I - A^2\| < 1$ we can also use **Padé family of iterations**, which is described and its convergence is discussed in [30], pp. 273-291.

Chapter 3

Methods for approximation of $f(A)\boldsymbol{b}$ of non-Krylov type

'Far better an approximate answer to the right question, than the exact answer to the wrong question, which can always be made precise.'
John Tukey

In this chapter, we describe methods, that compute an approximation of $f(A)\boldsymbol{b}$ of non-Krylov type. We shortly mention the quadrature rule and contour integral and the last method, polynomial least squares approximation, will be described in detail. This method will be compared with Krylov subspace methods in numerical experiments.

3.1 Quadrature rule

Suppose that $f(A)$ has integral representation,

$$f(A) = \int_0^a g(A,t)dt,$$

where $a \in \mathbb{R}$ and g is a rational function. Applying quadrature formula

$$\int_0^a g(A,t)dt \approx \sum_{k=1}^{p} c_k g(A,t_k),$$

we obtain an approximation

$$f(A)\boldsymbol{b} \approx \sum_{k=1}^{p} c_k g(A,t_k)\boldsymbol{b}, \tag{3.1}$$

see [25], pp. 306-308. The quadrature formula might be the Gauss rule, repeated rule such as the repeated trapezium or repeated Simpson rule etc. Matrix sign function, matrix square root and matrix logarithm have integral representations with rational function g, thus for these functions this method can be applied.

In case that a Hessenberg decomposition $A = V_k H_k V_k^H$, with upper Hessenberg matrix

$H_k \in \mathbb{C}^{k\times k}$ and a matrix with orthonormal columns $V_k \in \mathbb{C}^{n\times k}$, $k << n$, can be computed (this decomposition will be described in detail in Chapter 4), then (3.1) transforms to

$$f(A)\boldsymbol{b} \approx V \sum_{j=1}^{m} g(H_k, t_j) V_k^H \boldsymbol{b}.$$

Using such Hessenberg reduction, the problem can be solved in $O(n^2)$ flops, opposed to $O(n^3)$ flops for a dense system. Another efficient possibility is to apply Schur decomposition (2.1) on (3.1).

3.2 Contour integral

If f is analytic on and inside a closed contour Γ that encloses the spectrum of A, we can represent $f(A)\boldsymbol{b}$ by Cauchy integral formula (1.5), see [9], pp. 6-9,

$$f(A)\boldsymbol{b} = \frac{1}{2\pi i} \int_\Gamma f(t) \left(tI - A\right)^{-1} \boldsymbol{b} dt. \tag{3.2}$$

Assume that the contour Γ is a circle with a center α and a radius β,

$$\Gamma = \left\{ t : t - \alpha = \beta e^{i\theta},\ 0 \le \theta \le 2\pi \right\}.$$

Then $dt = i\beta e^{i\theta} d\theta = i(t(\theta) - \alpha) d\theta$. Denoting

$$w(t) = f(t)(tI - A)^{-1}\boldsymbol{b}$$

we obtain

$$\int_\Gamma w(t) dt = i \int_0^{2\pi} (t(\theta) - \alpha) w(t(\theta)) d\theta. \tag{3.3}$$

On this integral, we apply the m-point repeated trapezium rule and obtain the approximation

$$f(A)\boldsymbol{b} \approx \frac{2\pi i}{m} \sum_{j=0}^{m-1} (t_j - \alpha) w(t_j),$$

where $t_j - \alpha = \beta e^{2j\pi/m}$, i.e. $t_0, \ldots, t_m$ are equally spread points on the contour Γ (since Γ is a circle, we have $t_0 = t_m$). If A is a real matrix and if the center α is taken real, then it suffices use just the points t_j in the upper half plane and then take the real part of the result.

The attractivity of this approximation is in that it is exponentially accurate when applied to a periodic function. But in general, it is very unefficient unless A is well conditioned.

3.3 Polynomial least squares approximations for $f(A)\boldsymbol{b}$

In this section we will consider a polynomial method, which approximates $f(A)\boldsymbol{b}$ in the sense of least squares as it is described in [6]. For simplicity, suppose that A is real symmetric matrix (this approach can also be extended to the case, when A is nonsymmetric

with real eigenvalues). First, we approximate function f by a spline function s. Then we approximate this spline by a polynomial, see [6], pp. 5.

Suppose, that the spectrum $\Lambda(A)$ of A is included in some interval $[\alpha, \beta]$, $\alpha < \beta \in \mathbb{R}$. We define inner product of two functions g, h associated to a weight function w

$$\langle g, h \rangle_{[\alpha,\beta]} = \int_\alpha^\beta g(t)h(t)w(t)dt,$$

and the corresponding norm

$$\|g(t)\|_{[\alpha,\beta]} = \langle g(t), g(t) \rangle_{[\alpha,\beta]}^{1/2}. \tag{3.4}$$

Using this notation, we will construct an orthonormal basis of polynomials $\mathbb{P}_{k+1}(t) = \{P_j(t) | j = 1, 2, \ldots, k+1\}$ by the so called Stieltjes procedure. We put $P_0(t) = 0, P_1(t) = \frac{\mathbf{1}}{\|\mathbf{1}\|}$, where $\mathbf{1}$ is constant function with value one. Stieltjes procedure generates the basis using the three-term reccurence

$$\beta_{j+1}P_{j+1}(t) = tP_j(t) - \alpha_j P_j(t) - \beta_j P_{j-1}(t), \quad j = 1, \ldots, k, \tag{3.5}$$

where $\alpha_j = \langle tP_j(t), P_j(t) \rangle_{[\alpha,\beta]}$, $\beta_{j+1} = \left\| \widetilde{P}_{j+1} \right\|_{[\alpha,\beta]}^{-1}$ and $P_{j+1}(t) = \beta_{j+1}\widetilde{P}_{j+1}(t)$, see [6], pp. 3-4. Define $\boldsymbol{v}_j := P_j(A)\boldsymbol{b}$. With the assumption $\Lambda(A) \subset [\alpha, \beta]$, the approximation of $f(A)\boldsymbol{b}$ is given by

$$f(t) \approx \sum_{j=1}^{k+1} \gamma_j P_j(A)\boldsymbol{b} = \sum_{j=1}^{k+1} \gamma_j \boldsymbol{v}_j =: \boldsymbol{z}_{k+1}(t),$$

where $\gamma_j(t) = \langle f(t), P_j(t) \rangle_{[\alpha,\beta]}$ and from (3.5) we obtain the Stieltjes reccurence for $\boldsymbol{v}_j$,

$$\beta_{j+1}\boldsymbol{v}_{j+1} = A\boldsymbol{v}_j - \alpha_j \boldsymbol{v}_j - \beta_j \boldsymbol{v}_{j-1}.$$

It is unlikely, that for arbitrary function f the numerical integration can be avoided in computation of $\gamma_j(t) = \langle f(t), P_j(t) \rangle_{[\alpha,\beta]}$. Thus, it is often advantageous to first approximate f using a piecewise cubic spline. The advantage is, that on each subinterval the inner products needs to be computed only for polynomials. In addition, splines can be adjusted at the areas, where the function f has 'stiff region' by placing more knots in the places with high derivatives. For this, a form of (exact) Gauss-Chebyshev quadrature will allow us to completely bypass numerical integration. Cubic spline $s(t)$ is defined as a piecewise cubic polynomial on the knots

$$\alpha = t_0 < t_1 < \cdots < t_{n-1} < t_n = \beta,$$

$$s(t) := \sum_{i=0}^{n-1} s_i(t), \quad t \in [\alpha, \beta],$$

where for $i = 0, \ldots n-1$, the polynomial piece is

$$s_i(t) = \begin{cases} a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3 & \text{if } t \in [t_i, t_{i+1}], \\ 0 & \text{otherwise.} \end{cases}$$

Since now, we will consider

$$s(t) \approx \sum_{j=1}^{k+1} \gamma_j \boldsymbol{v}_j$$

and $\gamma_j = \langle s(t), P_j(t) \rangle_{[\alpha,\beta]}$.

Transformation of variable

Now we need to define inner product and the corresponding norm on each subinterval $[t_i, t_{i+1}]$, $i = 1, \ldots, n-1$. First, consider Chebyshev polynomials of the first kind $T_p(x) = \cos(p \cos^{-1} x)$ defined on the interval $[-1, 1]$. These polynomials also satisfy the three-term reccurence

$$T_{p+1}(x) = 2xT_p(x) - T_{p-1}(x), \quad T_0(x) = 1, \quad T_1(x) = x$$

and they constitute a sequence of orthogonal polynomials on $[-1, 1]$ with respect to the weight function $\frac{1}{\sqrt{1-x^2}}$. On the subintervals $[t_i, t_{i+1}]$, consider the transformation of variable

$$x^{(i)}(t) = \frac{2}{t_{i+1} - t_i} t - \frac{t_{i+1} + t_i}{t_{i+1} - t_i}, \quad i = 1, \ldots, n, \quad x^{(i)}(t) \in [-1, 1]\,.$$

Then, we can define

$$C_p^{(i)}(t) = T_p(x^{(i)}(t)), \quad t \in [t_i, t_{i+1}]\,, \quad i = 1, \ldots, n-1,$$

the corresponding inner product

$$\langle g(t), h(t) \rangle_{[t_i, t_{i+1}]} := \int_{t_i}^{t_{i+1}} \frac{g(t)h(t)}{\sqrt{(t - t_i)\,(t_{i+1} - t)}} dt \tag{3.6}$$

and the corresponding norm

$$\|g(t)\|_{[t_i, t_{i+1}]} := \langle g(t), g(t) \rangle_{[t_i, t_{i+1}]}\,. \tag{3.7}$$

Polynomials $C_p^{(i)}(t)$ are orthogonal with respect to the weight function $\dfrac{1}{\sqrt{(t - t_i)(t_{i+1} - t)}}$, i.e.,

$$\left\langle C_p^{(i)}(t), C_q^{(i)}(t) \right\rangle_{[t_i, t_{i+1}]} = \frac{\pi}{2} \left[\delta_{p-q} + \delta_{p+q}\right]. \tag{3.8}$$

Using (3.6) we can define an inner product on the whole interval $[\alpha, \beta]$ as

$$\langle g(t), h(t) \rangle_{[\alpha, \beta]} := \sum_{i=0}^{n-1} \langle g(t), h(t) \rangle_{[t_i, t_{i+1}]} \tag{3.9}$$

and the corresponding norm satisfies

$$\|g(t)\|^2_{[\alpha, \beta]} = \sum_{i=0}^{n-1} \|g(t)\|^2_{[t_i, t_{i+1}]}\,.$$

Computing coefficients α_j, β_{j+1}, γ_{j+1}

With the definition of an inner product on each subinterval, we can exploit the orthogonality of the basis to efficiently compute the coefficients for the Stieltjes reccurence, see [6], pp. 6-8.

First we describe computation of α_j. Polynomial $P_j(t)$ from the orthonormal basis $\mathbb{P}_{k+1}$ can be expressed on the subinterval $[t_i, t_{i+1}]$ as

$$P_j(t) = \sum_{p=0}^{j-1} \mu_{pj}^{(i)} C_p^{(i)}(t). \tag{3.10}$$

Just for now we assume, that the coefficients $\mu_{pj}^{(i)}$ are known. Rewriting the formula for the three-term recurrence for Chebyshev polynomials on the subinterval $[t_i, t_{i+1}]$, we obtain

$$\begin{aligned} tC_p^{(i)}(t) &= \frac{t_{i+1}-t_i}{4} C_{p+1}^{(i)}(t) + \frac{t_{i+1}+t_i}{2} C_p^{(i)}(t) + \frac{t_{i+1}-t_i}{4} C_{p-1}^{(i)}(t), \quad p \geq 1, \\ tC_0^{(i)}(t) &= \frac{t_{i+1}-t_i}{2} C_1^{(i)}(t) + \frac{t_{i+1}+t_i}{2} C_0^{(i)}(t) \end{aligned} \tag{3.11}$$

We use the convections $\mu_{-1,j}^{(i)} = 0$ and $\mu_{p,j}^{(i)}$ for $p \geq j$. Inserting (3.11) into (3.10) multiplied by t, we obtain

$$tP_j(t) = \frac{t_{i+1}-t_i}{4} \mu_{0j}^{(i)} C_1^{(i)}(t) + \sum_{p=0}^{j} \left(\frac{t_{i+1}-t_i}{4} \left(\mu_{p-1,j}^{(i)} + \mu_{p+1,j}^{(i)} \right) + \frac{t_{i+1}+t_i}{2} \right) C_p^{(i)}(t).$$

Defining

$$\sigma_{pj}^{(i)} := \frac{t_{i+1}-t_i}{4} \left(\mu_{p-1,j}^{(i)} + \mu_{p+1,j}^{(i)} \right) + \frac{t_{i+1}+t_i}{2} \mu_{pj}^{(i)}, \quad p = 0, \ldots, j, \tag{3.12}$$

gives

$$tP_j(t) = \frac{t_{i+1}-t_i}{4} \mu_{0j}^{(i)} C_1^{(i)}(t) + \sum_{p=0}^{j} \sigma_{pj} C_p^{(i)}(t).$$

According to the definition of the inner product (3.9), we have

$$\alpha_j = \langle tP_j(t), P_j(t) \rangle_{[\alpha,\beta]} = \sum_{i=0}^{n-1} \langle tP_j(t), P_j(t) \rangle_{[t_i, t_{i+1}]}.$$

Finally

$$\alpha_j = \pi \sum_{i=0}^{n-1} \left(\sigma_{0j}^{(i)} \mu_{0j}^{(i)} + \frac{t_{i+1}-t_i}{8} \mu_{0j}^{(i)} \mu_{1j}^{(i)} + \sum_{p=1}^{j} \sigma_{pj}^{(i)} \mu_{pj}^{(i)} \right). \tag{3.13}$$

Further we consider the computation of β_{j+1}. We denote the right-hand side of (3.5) by $S_j(t) = tP_j(t) - \alpha_j P_j(t) - \beta_j P_{j-1}(t)$. For $j = 0$, we have

$$\beta_1 = \|S_0(t)\|_{[\alpha,\beta]} = \sqrt{\sum_{i=0}^{n-1} \|C_0(t)\|_{[t_i,t_{i+1}]}^2} = \sqrt{n\pi}$$

and for $j \geq 1$, we have by some manipulations

$$\beta_{j+1}^2 = \|S_j(t)\|_{[\alpha,\beta]}^2 = \sum_{i=0}^{n-1} \left\| \frac{t_{i+1}-t_i}{4} \mu_{0j}^{(i)} C_1^{(i)}(t) + \sum_{p=0}^{j} \left(\sigma_{pj}^{(i)} - \alpha_j \mu_{pj}^{(i)} - \beta_j \mu_{p,j-1}^{(i)} \right) C_p^{(i)}(t) \right\|_{[t_i,t_{i+1}]}^2.$$

Defining

$$\eta_{pj}^{(i)} := \sigma_{pj}^{(i)} - \alpha_j \mu_{pj}^{(i)} - \beta_j \mu_{p,j-1}^{(i)}, \quad p = 0, \ldots, j \tag{3.14}$$

yields the formula for β_{j+1}

$$\beta_{j+1} = \sqrt{\pi \sum_{i=0}^{n-1} \left[\eta_{0j}^{(i)^2} + \frac{1}{2} \left(\eta_{1j}^{(i)} + \frac{t_{i+1} - t_i}{4} \mu_{0j}^{(i)} \right)^2 + \frac{1}{2} \sum_{p=2}^{j} \eta_{pj}^{(i)^2} \right]}. \tag{3.15}$$

Now, we can find an update formula for $\mu_{p,j+1}^{(i)}$. Since $P_{j+1}(t) = S_j(t)/\beta_{j+1}$, $j = 0, \ldots, k$,

$$\begin{array}{rcl} \mu_{01}^{(i)} & = & 1/\beta_1, \\ \mu_{1,j+1}^{(i)} & = & \left[\eta_{1j}^{(i)} + \frac{t_{i+1} - t_i}{4} \mu_{0j}^{(i)} \right] / \beta_{j+1} \\ \mu_{p,j+1}^{(i)} & = & \eta_{pj}^{(i)} / \beta_{j+1} \text{ for } p = 0, 2, 3, \ldots, j. \end{array} \tag{3.16}$$

Finally, consider computing γ_{j+1}. We have

$$\gamma_{j+1} = \langle s(t), P_{j+1}(t) \rangle_{[\alpha,\beta]} = \sum_{i=0}^{n-1} \left\langle s_i(t), \sum_{p=0}^{j} \mu_{p,j+1}^{(i)} C_p^{(i)}(t) \right\rangle_{[t_i, t_{i+1}]}, \tag{3.17}$$

where $s_i(t)$ is a cubic polynomial of the form

$$s_i(t) = \xi_0^{(i)} C_0^{(i)}(t) + \xi_1^{(i)} C_1^{(i)}(t) + \xi_2^{(i)} C_2^{(i)}(t) + \xi_3^{(i)} C_3^{(i)}(t),$$

with $h_i = \dfrac{t_{i+1} - t_i}{2}$, and

$$\begin{array}{rcl} \xi_0^{(i)} & = & \frac{5}{2} d_i h_i^3 + \frac{3}{2} c_i h_i^2 + b_i h_i + a_i \\ \xi_1^{(i)} & = & \frac{15}{4} d_i h_i^3 + 2 c_i h_i^2 + b_i h_i \\ \xi_2^{(i)} & = & \frac{3}{2} d_i h_i^3 + \frac{1}{2} c_i h_i^2 \\ \xi_3^{(i)} & = & \frac{1}{4} d_i h_i^3. \end{array} \quad . \tag{3.18}$$

Previous derivations can be summarized in the following algorithm for computation of an approximation to $f(A)\boldsymbol{b}$ using polynomial least squares method:

Algorithm 1

Given $t_0, t_1, \ldots t_n$, where $t_0 = \alpha$, $t_n = \beta$ and $t_i < t_{i+1} \;\; i = 1, \ldots, n-1$.

1. Compute a cubic spline $s(t)$ which interpolates $f(t)$ in points $(t_i, f(t_i))$, $i = 1, \ldots, n$
2. $\beta_1 = \sqrt{n\pi}$
3. $\boldsymbol{v}_1 = \boldsymbol{b}/\beta_1$
4. Compute $\xi_p^{(i)}$ for $i = 0, \ldots, n-1$ and $p = 0, \ldots, 3$, using (3.18)
5. $\mu_{01}^{(i)} = 1/\beta_1$, for $i = 0, \ldots, n-1$
6. $\gamma_1 = \pi \sum_{i=0}^{n-1} \xi_0^{(i)} \mu_{01}^{(i)}$
7. $\boldsymbol{z}_1 = \gamma_1 \boldsymbol{v}_1$
8. **for** $j = 1, \ldots, k$ **do**
9. Compute $\sigma_{pj}^{(i)}$ for $i = 0, \ldots, n-1$ and $p = 0, \ldots, j$ using (3.12)
10. Compute α_j using (3.13)
11. Compute $\eta_{pj}^{(i)}$ for $i = 0, \ldots, n-1$ and $p = 0 \ldots, j$ using (3.14)
12. Compute β_{j+1} using (3.15)
13. Compute $\mu_{p,j+1}^{(i)}$ for $i = 0, \ldots, n-1$ and $p = 0, \ldots, j$ using (3.16)
14. $\boldsymbol{v}_{j+1} = (A\boldsymbol{v}_j - \alpha_j \boldsymbol{v}_j - \beta_j \boldsymbol{v}_{j-1})/\beta_{j+1}$
15. Compute γ_{j+1} using (3.17)
16. $\boldsymbol{z}_{j+1} = \boldsymbol{z}_j + \gamma_{j+1} \boldsymbol{v}_{j+1}$
17. **enddo**

Convergence analysis

We approximated $f(t)$ on the interval $[\alpha, \beta]$ by the spline $s(t)$, and we projected $s(t)$ onto the polynomial space $\mathbb{P}_{k+1}$. Denote

$$\Phi_{k+1} = \sum_{j=1}^{k+1} \gamma_j P_j(t) \approx s(t)$$

thus

$$\boldsymbol{z}_{k+1} = \Phi_{k+1}(A)\boldsymbol{b} \approx s(A)\boldsymbol{b}.$$

For the norm (3.4), $\Phi_{k+1}(t)$ approximates $s(t)$ in the least squares sense, [6], pp. 9, i.e.

$$\Phi_{k+1}(t) = \arg \min_{\Phi \in \mathbb{P}_{k+1}} \|\Phi(t) - s(t)\|_{[\alpha,\beta]}.$$

Suppose that the matrix A is symmetric. Then

$$\|\boldsymbol{z}_{k+1} - f(A)\boldsymbol{b}\|_{[\alpha,\beta]} \leq \max_{t \in [\alpha,\beta]} |\Phi_{k+1}(t) - f(t)| \, \|b\|_2 .$$

To bound the difference between $\Phi_{k+1}(t)$ and $f(t)$, we use the triangle inequality,

$$|\Phi_{k+1}(t) - f(t)| \leq |\Phi_{k+1}(t) - s(t)| + |s(t) - f(t)| .$$

For the difference between $s(t)$ and $f(t)$ on the interval $[\alpha, \beta]$, we first suppose that $f(t)$ is fourth order differentiable on the interval $[\alpha, \beta]$, and $s(t)$ is the unique cubic spline that interpolates $f(t)$ on the knots

$$\alpha = t_0 < t_1 < \ldots < t_{n-1} < t_n = \beta$$

with the boundary condition

$$s'(t_0) = f'(t_0) \qquad \text{and} \qquad s'(t_n) = f'(t_n).$$

Then

$$\max_{t\in[\alpha,\beta]} |s(t) - f(t)| \leq \frac{5M}{384} \max_{0\leq i\leq n-1} (t_{i+1} - t_i)^4,$$

where $M = \max_{t\in[\alpha,\beta]} \left|f^{(4)}(t)\right|$. This result was shown in [48].

We define the **modulus of continuity** of a function $g(t)$ on the interval $[\alpha, \beta]$ as

$$\omega(g; [\alpha, \beta]; \delta) := \sup_{\substack{t_1,t_2\in[\alpha,\beta] \\ |t_1-t_2|<\delta}} |g(t_1) - g(t_2)|$$

for $\delta > 0$. In case the context is clear, we will use shorthand notation $\omega(\delta)$. Before we estimate $|\Phi_{k+1}(t) - s(t)|$, we need the following lemmas, see [47] and [45], pp. 22.

Lemma 3.3.1 *For the norm defined in this section, (3.4), it holds that*

$$\|g(t)\|_{[\alpha,\beta]} \leq \sqrt{n\pi} \max_{t\in[\alpha,\beta]} |g(t)|.$$

Lemma 3.3.2 *Let $g_{k+1}(t) \in \mathbb{P}_{k+1}$ be any polynomial of degree not exceeding k. Then using the notation of this section, we have*

$$\max_{t\in[\alpha,\beta]} |g_{k+1}(t)| \leq \sqrt{\frac{2(k+1)}{\pi}} \|g_{k+1}(t)\|_{[\alpha,\beta]}.$$

By the property of uniform norm, for any continuous function $g(t)$ there exist a polynomial $g^*_{k+1}(t)$ of degree k, such that

$$\max_{t\in[\alpha,\beta]} \left|g^*_{k+1}(t) - g(t)\right| \leq \max_{t\in[\alpha,\beta]} |\Phi_{k+1}(t) - g(t)|.$$

Lemma 3.3.3 *If a function g is continous on the interval $[\alpha, \beta]$, then*

$$\max_{t\in[\alpha,\beta]} \left|g^*_{k+1}(t) - g(t)\right| \leq 6\omega\left(\frac{\beta-\alpha}{2k}\right).$$

These lemmas can be used to prove the following theorem presented in [6], pp. 10-11., which gives the upper bound for $|\Phi_{k+1}(t) - s(t)|$.

Theorem 3.3.4 *The uniform norm of the residual polynomial admits the bound*

$$\max_{t\in[\alpha,\beta]} |\Phi_{k+1}(t) - s(t)| \leq \left(6\sqrt{2n(k+1)} + 1\right)\omega\left(\frac{\beta-\alpha}{2k}\right).$$

Proof. Using triangle inequality, there holds

$$\max_{t\in[\alpha,\beta]} |\Phi_{k+1}(t) - s(t)| \leq \max_{t\in[\alpha,\beta]} \left|\Phi_{k+1}(t) - s^*_{k+1}(t)\right| + \max_{t\in[\alpha,\beta]} \left|s^*_{k+1}(t) - s(t)\right|. \tag{3.19}$$

According to Lemma 3.3.2, we have

$$\max_{t\in[\alpha,\beta]} \left|\Phi_{k+1}(t) - s^*_{k+1}(t)\right| \leq \sqrt{\frac{2(k+1)}{\pi}} \left\|\Phi_{k+1}(t) - s^*_{k+1}(t)\right\|_{[\alpha,\beta]}, \tag{3.20}$$

Since Φ_{k+1} is an approximation in the sence of least squares,

$$\begin{aligned} \left\|\Phi_{k+1}(t) - s^*_{k+1}(t)\right\|_{[\alpha,\beta]} &\leq \|\Phi_{k+1}(t) - s(t)\|_{[\alpha,\beta]} + \left\|s(t) - s^*_{k+1}(t)\right\|_{[\alpha,\beta]} \\ &\leq \left\|s^*_{k+1}(t) - s(t)\right\|_{[\alpha,\beta]} + \left\|s(t) - s^*_{k+1}(t)\right\|_{[\alpha,\beta]} \\ &= 2\left\|s(t) - s^*_{k+1}(t)\right\|_{[\alpha,\beta]}, \end{aligned} \tag{3.21}$$

and (3.20) becomes

$$\max_{t\in[\alpha,\beta]} \left|\Phi_{k+1}(t) - s^*_{k+1}(t)\right| \leq 2\sqrt{\frac{2(k+1)}{\pi}} \left\|s^*_{k+1}(t) - s(t)\right\|_{[\alpha,\beta]}.$$

and from Lemma 3.3.1 we have

$$\max_{t\in[\alpha,\beta]} \left|\Phi_{k+1}(t) - s^*_{k+1}(t)\right| \leq 2\sqrt{2n(k+1)} \max_{t\in[\alpha,\beta]} \left|s^*_{k+1}(t) - s(t)\right|.$$

Thus, (3.19) becomes

$$\max_{t\in[\alpha,\beta]} |\Phi_{k+1}(t) - s(t)| \leq \left(2\sqrt{2n(k+1)} + 1\right) \max_{t\in[\alpha,\beta]} \left|s^*_{k+1}(t) - s(t)\right|.$$

The proof is finished by applying Lemma 3.3.3.

□

Finally, we can formulate the following theorem for the upper bound of $\|\Phi_{k+1}(t) - s(t)\|$, see [6], pp. 12:

Theorem 3.3.5 *The norm of the residual polynomial admits the bound:*

$$\|\Phi_{k+1}(t) - s(t)\|_{[\alpha,\beta]} \leq 18\sqrt{n\pi}\omega\left(\frac{\beta-\alpha}{2k}\right).$$

Proof. Using (3.21), Lemma 3.3.2 and 3.3.3, gives

$$\begin{aligned} \|\Phi_{k+1}(t) - s(t)\|_{[\alpha,\beta]} &\leq \left\|\Phi_{k+1}(t) - s^*_{k+1}(t)\right\|_{[\alpha,\beta]} + \left\|s^*_{k+1}(t) - s(t)\right\|_{[\alpha,\beta]} \\ &\leq 3\left\|s^*_{k+1}(t) - s(t)\right\|_{[\alpha,\beta]} \\ &\leq 3\sqrt{n\pi} \max_{t\in[\alpha,\beta]} \left|s^*_{k+1}(t) - s(t)\right| \\ &\leq 18\sqrt{n\pi}\omega\left(\frac{\beta-\alpha}{2k}\right). \end{aligned}$$

□

Summarizing, the estimate of the error is

$$\|\boldsymbol{z}_{k+1} - f(A)\boldsymbol{b}\|_{[\alpha,\beta]} \leq \|b\|_2 \left(\left(6\sqrt{2n(k+1)} + 1\right)\omega\left(\frac{\beta-\alpha}{2k}\right) + \frac{5}{24}M \max_{0\leq i\leq n-1} h_i^4\right).$$

Chapter 4

Krylov subspace methods for approximation of $f(A)\boldsymbol{b}$

'There is no method but to be very intelligent.'
Thomas Stearns Eliot

Krylov subspace methods, first introduced in [32], are iterative methods that play a key-role in a large and sparse matrix-vector problems; not only in solving systems of linear equations but also in problems of actions of a vector on matrix functions, $f(A)\boldsymbol{b}$. In Krylov subspace methods an approximate solution of $f(A)\boldsymbol{b}$ can be found in a finite number of iterations, usually significantly smaller than n, and thus using Krylov subspace methods can be a good and efficient choice for our problem.

Suppose that $A \in \mathbb{C}^{n\times n}$ and $0 \neq \boldsymbol{b} \in \mathbb{C}^n$ are given. We define the mth Krylov subspace as

$$\mathscr{K}_m(A, \boldsymbol{b}) = \operatorname{span}\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}\} = \{q(A)\boldsymbol{b} : q \in \mathscr{P}_{m-1}\}.$$

We need an orthonormal basis of this subspace (because the vectors $\boldsymbol{b}, A\boldsymbol{b}, A^2\boldsymbol{b}, \ldots$ can easily become 'numericaly' dependend). This basis can be constructed using the so called Arnoldi algorithm, [3], with the starting vector $\boldsymbol{v}_1 = \boldsymbol{b}/\|\boldsymbol{b}\|$. Assume that the algorithm does not terminate before step m. Then we obtain an **Arnoldi decomposition** of A with respect to $\mathscr{K}_m(A, \boldsymbol{b})$,

$$AV_m = V_{m+1}H_{m+1,m} = V_mH_m + \eta_{m+1,m}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^T, \tag{4.1}$$

where the columns of $V_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m]$ form an orthonormal basis of $\mathscr{K}_m(A, \boldsymbol{b})$, the matrix $H_{m+1,m} \in \mathbb{C}^{(m+1)\times m}$ is unreduced upper Hessenberg of the form

$$H_{m+1,m} = \begin{bmatrix} \eta_{1,1} & \eta_{1,2} & \cdots & \eta_{1,m} \\ \eta_{2,1} & \eta_{2,2} & \cdots & \eta_{2,m} \\ & \eta_{3,2} & \ddots & \vdots \\ & & \ddots & \vdots \\ & & \eta_{m,m-1} & \eta_{m,m} \\ & & & \eta_{m+1,m} \end{bmatrix}$$

and $H_m = [I_m, 0]\, H_{m+1,m} \in \mathbb{C}^{m\times m}$. For the Krylov subspaces, it holds that

$$\mathscr{K}_1(A, \boldsymbol{b}) \subset \mathscr{K}_2(A, \boldsymbol{b}) \subset \ldots \subset \mathscr{K}_m(A, \boldsymbol{b}), \quad m = 1, \ldots, L,$$

where L is the smallest index for which

$$\mathscr{K}_L(A, \boldsymbol{b}) = \mathscr{K}_{L+1}(A, \boldsymbol{b}) = \mathscr{K}_{L+2}(A, \boldsymbol{b}) = \ldots.$$

The Arnoldi approximation $\boldsymbol{f}_m$ to $f(A)\boldsymbol{b}$ is then defined as

$$\boldsymbol{f}_m := \beta V_m f(H_m)\boldsymbol{e}_1, \quad \text{where } \beta = \|\boldsymbol{b}\|, \quad \text{for } m = 1, \ldots, L, \tag{4.2}$$

see Algorithm 2:

Algorithm 2

Given A, $\boldsymbol{b}$
1. $\beta = \|\boldsymbol{b}\|$, $\boldsymbol{v}_1 = \boldsymbol{b}/\beta$
2. **for** $j = 1, \ldots, m$ do
3. $\quad \boldsymbol{w} = A\boldsymbol{v}_j$
4. $\quad$ **for** $i = 1, \ldots, j$ do
5. $\quad\quad \eta_{i,j} = (\boldsymbol{w}, A\boldsymbol{v}_j)$
6. $\quad\quad \boldsymbol{w} = \boldsymbol{w} - \eta_{i,j}\boldsymbol{v}_i$
7. $\quad$ **enddo**
8. $\quad \eta_{j+1,j} = \|\boldsymbol{w}\|$
9. $\quad$ **if** $\eta_{j+1,j} \neq 0$, then $\boldsymbol{v}_j = \boldsymbol{w}/\eta_{j+1,j}$ **else** stop
10. **enddo**
11. compute approximation $\boldsymbol{f}_m = \beta V_m f(H_m)\boldsymbol{e}_1$

Note that the Arnoldi algorithm simplifies into a three-term reccurence in case that A is a Hermitian matrix. Then the process is called the Lanczos algorithm, see [33], and the Hessenberg matrix H_m is tridiagonal. The Hermitian Lanczos algorithm is nothing but the matrix formulation of the Stieltjes algorithm for computing the basis of orthonormal polynomials $\mathbb{P}_{k+1}$, see [50].
For a general matrix we can also construct a basis of $\mathscr{K}_m(A, \boldsymbol{b})$ using three-term reccurence. However, the basis is no longer orthogonal. This process is called the two-sided Lanczos algorithm, see [33]. We construct two sequences of vectors, $\{\boldsymbol{v}\}_{j=1}^m, \{\widehat{\boldsymbol{v}}_j\}_{j=1}^m$, such that

$$AV_m = V_m T_m + \beta_{m+1}\boldsymbol{v}_{m+1}\boldsymbol{e}_m^T,$$
$$A^H\widehat{V}_m = \widehat{V}_m T_m^H + \gamma_{m+1}\widehat{\boldsymbol{v}}_{m+1}\boldsymbol{e}_m^T,$$

where $\alpha_j = \widehat{\boldsymbol{v}}_j^H A\boldsymbol{v}_j$, matrices $\widehat{V}_m = [\widehat{\boldsymbol{v}}_1, \ldots, \widehat{\boldsymbol{v}}_m]$ and $V_m = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m]$ satisfy

$$\widehat{V}_m^H A V_m = T_m, \quad \widehat{V}_m^H V_m = I$$

and T_m is of the form

$$T_m = \begin{bmatrix} \alpha_1 & \gamma_2 & & & \\ \beta_2 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & & \\ & & & & \gamma_m \\ & & & \beta_m & \alpha_m \end{bmatrix}.$$

Then we can approximate $f(A)\boldsymbol{b}$ by

$$\boldsymbol{f}_m = \beta V_m f(T_m)\boldsymbol{e}_1.$$

The following lemma, [46], pp. 215, shows, that for any matrix polynomial $p_j(A) \in \mathscr{P}_j$, $j \leq m-1$, the Arnoldi approximation is exact.

Lemma 4.0.6 *Let $A \in \mathbb{C}^{n\times n}, \boldsymbol{b} \in \mathbb{C}^n$ and let V_m, H_m be the matrices obtained after m steps of the Arnoldi process applied to A and $\boldsymbol{b}$. Then for any polynomial p_j of degree $j \leq m-1$ the following equality holds,*

$$p_j(A)\boldsymbol{b} = \beta V_m p_j(H_m)\boldsymbol{e}_1.$$

The lemma can be easily proved using mathematical induction. According to this lemma, the approximation $\boldsymbol{f}_m$ can be expressed using interpolation polynomials, as it is summarized in the following theorem, see [46], pp. 215.

Theorem 4.0.7 *Suppose, that $q \in \mathscr{P}_{m-1}$ is a polynomial that interpolates f in a Hermite sense on the spectrum of H_m, then*

$$\boldsymbol{f}_m = \beta V_m f(H_m)\boldsymbol{e}_1 = \beta V_m q(H_m)\boldsymbol{e}_1 = q(A)\boldsymbol{b}. \tag{4.3}$$

Proof. For $m \geq L$, let $q(A)$ be a Hermite interpolation polynomial. Then $q(A) = f(A) = V_m f(H_m)V_m^H$ and thus $q(A)\boldsymbol{b} = V_m f(H_m)V_m^H\boldsymbol{b} = \beta V_m f(H_m)\boldsymbol{e}_1$.

For $m < L$, there exists a Hermite interpolation polynomial $\tilde{q}(H_m)$ to $f(H_m)$. Then from the definition (1.7), $\tilde{q}(H_m)\boldsymbol{b} = f(H_m)\boldsymbol{b}$ giving

$$\beta V_m f(H_m)\boldsymbol{e}_1 = \beta V_m \tilde{p}(H_m)\boldsymbol{e}_1 = \beta\tilde{q}(A)V_m\boldsymbol{e}_1 = \tilde{q}(A)\boldsymbol{b}.$$

□

In the following sections, we will refer to the method, that gives an approximation (4.2) based on the Arnoldi decomposition (4.1), as the **standart Krylov subspace method**.

4.1 Restarted Krylov subspace method

Computational cost and storage requirements of the standart Krylov subspace method increase with growing number of iterations, due to the fact that the algorithm uses long recurrences. This problem can be reduced by regular restarting of the Arnoldi process after a given number of steps. In this section we show how the approximation of $f(A)\boldsymbol{b}$ is updated at the end of the current Arnoldi process using the value of the previous approximation and matrices incurred while computing current Arnoldi decomposition.

Before we briefly describe the method itself, based on [15], we give some theoretical background in order to properly understand the idea of restarting the Krylov subspace method.

Similar results as were shown for the Arnoldi decomposition can be described for more

general decompositions of $\mathscr{K}_m(A, \boldsymbol{b})$. Consider $\{\boldsymbol{w}_m\}_{m=1}^{L}$, a sequence of ascending (not necessarily orthonormal) basis vectors such that

$$\mathscr{K}_m(A, \boldsymbol{b}) = \operatorname{span}\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m\}, \quad m = 1, \ldots, L. \tag{4.4}$$

Then there exists the unique, so called **Arnoldi-like decomposition**,

$$AW_m = W_{m+1}H_{m+1,m} = W_m H_m + \eta_{m+1,m}\boldsymbol{w}_{m+1}\boldsymbol{e}_m^T, \tag{4.5}$$

where $W_m = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m] \in \mathbb{C}^{n\times m}$, $H_{m+1,m} \in \mathbb{C}^{(m+1)\times m}$ is an unreduced upper Hessenberg matrix and $H_m = [I_m, 0]H_{m+1,m} \in \mathbb{C}^{m\times m}$. The following lemma, [15], pp. 2484, is a simple generalization of the corresponding result for Arnoldi decompositions.

Lemma 4.1.1 *For any polynomial $q_m(z) = a_m z^m + a_{m-1}z^{m-1} + \ldots + a_1 z + a_0 \in \mathscr{P}_m$ the vector $q_m(A)\boldsymbol{b}$ can be represented as*

$$q_m(A)\boldsymbol{b} = \begin{cases} \beta\left[W_m q_m(H_m)\boldsymbol{e}_1 + \alpha_m\gamma_m\boldsymbol{w}_{m+1}\right], & m < L, \\ \beta W_L q_m(H_L)\boldsymbol{e}_1, & m \geq L, \end{cases} \tag{4.6}$$

where

$$\gamma_m := \prod_{j=1}^{m} \eta_{j+1,j}$$

and $\beta\boldsymbol{w}_1 = \boldsymbol{b}$. In particular, for any $q \in \mathscr{P}_{m-1}$

$$q(A)\boldsymbol{b} = \beta W_m q(H_m)\boldsymbol{e}_1.$$

The proof of this lemma can be found in [18]. It is based on verifying the assertion for monomials, taking into account the sparsity pattern of a Hessenberg matrix.

We next introduce the polynomial notation of vectors from the Arnoldi-like decomposition (4.5). To each vector $\boldsymbol{w}_m$, there corresponds a unique polynomial

$$w_{m-1} \in \mathscr{P}_{m-1} \tag{4.7}$$

such that $\boldsymbol{w}_m = w_{m-1}(A)\boldsymbol{b}$. Arnoldi-like recurrence (4.5) rewritten for each polynomials becomes

$$\lambda\left[w_0(\lambda), \ldots, w_{m-1}(\lambda)\right] = \left[w_0(\lambda), \ldots, w_{m-1}(\lambda)\right] H_m + \eta_{m+1,m}\left[0, \ldots, 0, w_m(\lambda)\right]. \tag{4.8}$$

Obviously, each zero of w_m is an eigenvalue of H_m. Moreover, the zeros of multiplicity ℓ are the eigenvalues of H_m corresponding to a Jordan block of dimension ℓ. Further, for an approximation using the Arnoldi-like decomposition, we have the following theorem, see [46], pp. 2485:

Theorem 4.1.2 *Let $q_{m-1} \in \mathscr{P}_{m-1}$ be a polynomial and let $f(H_m)$ be defined. Then*

$$q_{m-1}(H_m) = f(H_m) \tag{4.9}$$

if and only if q_{m-1} interpolates f in Hermite sense at the eigenvalues of H_m. Moreover,

$$\boldsymbol{f}_m := \beta W_m f(H_m)\boldsymbol{e}_1 = \beta W_m q(H_m)\boldsymbol{e}_1 = q(A)\boldsymbol{b}. \tag{4.10}$$

Approximation (4.10) is called the Krylov subspace approximation to $f(A)\boldsymbol{b}$ associated with the Arnoldi-like decomposition. Shortly, we refer to this as the **Arnoldi-like approximation**.

Remark 4.1.3 *We briefly describe one possibility of choice of the basis vectors* $\boldsymbol{w}_m = w_{m-1}(A)\boldsymbol{b}$*, [15], pp. 2485-2486. Let*

$$\begin{array}{ccc} \vartheta_1^{(1)} & & \\ \vartheta_1^{(2)} & \vartheta_2^{(2)} & \\ \vartheta_1^{(3)} & \vartheta_2^{(3)} & \vartheta_3^{(3)} \\ \vdots & \vdots & \vdots \end{array}$$

be a fixed sequence of nodes. Then we can choose the basis vectors as

$$w_{m-1}(z) = \omega_{m-1}(z - \vartheta_1^{(m-1)})(z - \vartheta_2^{(m-1)}) \cdot \ldots \cdot (z - \vartheta_{m-1}^{(m-1)}),$$

where $\omega_{m-1} \neq 0$ *is a parameter. The nodes can be chosen, e.g., as zeros of Chebyshev polynomials. Other choices are described in [38], [39], [41].*

Since now, we will use the following notation. We define the **nodal polynomial** associated with the nodes $\vartheta_1, \ldots, \vartheta_m$ as

$$p(z) := (z - \vartheta_1)(z - \vartheta_2) \cdots (z - \vartheta_m). \tag{4.11}$$

We denote the unique polynomial which interpolates f in the Hermite sense by $I_p f$. We define the m**th order divided difference of** f **with respect to the nodes** $\{\vartheta_j\}_{j=1}^m$ by

$$\Delta_p f := \frac{f - I_p f}{p}. \tag{4.12}$$

The following theorem, [15], pp. 2486, gives an expression of the error for the Arnoldi-like approximation.

Theorem 4.1.4 *Given* $A \in \mathbb{C}^{n\times n}, \boldsymbol{b} \in \mathbb{C}^n$ *and a function* f*, let (4.5) be an Arnoldi-like decomposition, and let* $w_m \in \mathscr{P}_{m-1}$ *be the associated polynomial (4.7). Then*

$$f(A)\boldsymbol{b} - \underbrace{\beta W_m f(H_m)\boldsymbol{e}_1}_{f_m} = \beta\gamma_m \left[\Delta_{w_m} f\right](A)\boldsymbol{w}_{m+1}. \tag{4.13}$$

Proof. First consider an arbitrary set of nodes $\vartheta_1, \ldots, \vartheta_m$ with associated nodal polynomial (4.11). From the definition (4.12), it holds that $f(z) = [I_p f](z) + [\Delta_p f](z)p(z)$. Inserting A for z in this identity and multiplying by $\boldsymbol{b}$, we obtain

$$f(A)\boldsymbol{b} = [I_p f](A)\boldsymbol{b} + [\Delta_p f](A)p(A)\boldsymbol{b}. \tag{4.14}$$

Now we apply Lemma 4.1.1. Since $I_p f \in \mathscr{P}_{m-1}$, we have

$$[I_p f](A)\boldsymbol{b} = \beta W_m [I_p f](H_m)\boldsymbol{e}_1 \tag{4.15}$$

and since $p \in \mathscr{P}_m$ is monic,

$$p(A)\boldsymbol{b} = \beta W_m p(H_m)\boldsymbol{e}_1 + \beta\gamma_m \boldsymbol{w}_{m+1}. \tag{4.16}$$

Substituting (4.15) and (4.16) into (4.14) gives

$$f(A)\boldsymbol{b} - \beta W_m \left[I_p f\right](H_m)\boldsymbol{e}_1 = \beta \left[\Delta_p f\right](A)\left(W_m p(H_m)\boldsymbol{e}_1 + \gamma_m \boldsymbol{w}_{m+1}\right).$$

Choosing p as the characteristic polynomial w_m of H_m, it follows that $w_m(H_m) = O$ by the Cayley-Hamilton theorem. Since $I_{w_m} f$ interpolates f at the eigenvalues of H_m, $\left[I_{w_m}\right] f(H_m) = f(H_m)$ by (4.9), and (4.13) is proved.

□

Krylov approximation after Arnoldi restart

Consider two subsequent restarts of the Arnoldi process. We turn to the question, how to compute an approximation of $f(A)\boldsymbol{b}$ after a restart.

Starting with the given matrix $A \in \mathbb{C}^{n\times n}$ and the vector $\boldsymbol{b} \in \mathbb{C}^n$, we compute the first Arnoldi decomposition of $\mathscr{K}_m(A, \boldsymbol{b})$ after m iterations,

$$AV_m^{(1)} = V_m^{(1)} H_m^{(1)} + \eta_{m+1,m}^{(1)} \boldsymbol{v}_{m+1}^{(1)} \boldsymbol{e}_m^T, \quad \boldsymbol{v}_1^{(1)} = \boldsymbol{b}/\left\|\boldsymbol{b}\right\|. \tag{4.17}$$

Using the last vector $\boldsymbol{v}_{m+1}^{(1)}$ from the first Arnoldi decomposition (4.17), we compute the Arnoldi decomposition of the next Krylov subspace $\mathscr{K}_m(A, \boldsymbol{v}_{m+1}^{(1)})$,

$$AV_m^{(2)} = V_m^{(2)} H_m^{(2)} + \eta_{m+1,m}^{(2)} \boldsymbol{v}_{m+1}^{(2)} \boldsymbol{e}_m^T, \quad \boldsymbol{v}_1^{(2)} = \boldsymbol{v}_{m+1}^{(1)}. \tag{4.18}$$

Then the columns of $W_{2m} = \left[V_m^{(1)}, V_m^{(2)}\right]$ form a basis of $\mathscr{K}_{2m}(A, \boldsymbol{b})$. We can combine the Arnoldi decompositions (4.17) and (4.18) to an **Arnoldi-like decomposition**

$$AW_{2m} = W_{2m} H_{2m} + \eta_{m+1,m}^{(2)} \boldsymbol{v}_{m+1}^{(2)} \boldsymbol{e}_{2m}^T, \tag{4.19}$$

where the Hessenberg matrix H_{2m} is of the form

$$H_{2m} = \begin{bmatrix} H_m^{(1)} & O \\ \eta_{m+1,m}^{(1)} \boldsymbol{e}_1 \boldsymbol{e}_m^T & H_m^{(2)} \end{bmatrix}.$$

Remark 4.1.5 *We restarted the Arnoldi process with $\boldsymbol{v}_{m+1}^{(1)}$, which is a natural choice. However, we could restart it with any vector of the form*

$$\widehat{\boldsymbol{v}}_{m+1} = V_m^{(1)} \boldsymbol{y} + y_{m+1} \boldsymbol{v}_{m+1}^{(1)} \in \mathscr{K}_{m+1}(A, \boldsymbol{b}) \setminus \mathscr{K}_m(A, \boldsymbol{b}),$$

where $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]^T \in \mathbb{C}^m$ is a coefficient vector. In this case, $H_m^{(1)}$ is replaced by its rank-one modification $H_m^{(1)} - (\eta_{m+1,m}^{(1)}/y_{m+1})\boldsymbol{y}\boldsymbol{e}_m^T$, and $\eta_{m+1,m}^{(1)}$ is replaced by $\eta_{m+1,m}^{(1)}/y_{m+1}$, see [15], pp. 2488.

Now, the goal is to compute the approximation of $f(A)\boldsymbol{b}$ without using $V_m^{(1)}$. The former is defined as

$$\boldsymbol{f}_{2m} = [I_{w_{2m}} f]\,(A)\boldsymbol{b} = \beta W_{2m}\,[I_{w_{2m}} f]\,(H_{2m})\boldsymbol{e}_1 = \beta W_{2m} f(H_{2m})\boldsymbol{e}_1, \tag{4.20}$$

where w_{2m} is the nodal polynomial with zeros $\Lambda(H_m^{(1)}) \cup \Lambda(H_m^{(2)})$, including their multiplicity, see (4.10). Function $f(H_{2m})$ is of the form

$$f(H_{2m}) = \begin{bmatrix} f(H_m^{(1)}) & O \\ X_{2,1} & f(H_m^{(2)}) \end{bmatrix}, \quad X_{2,1} \in \mathbb{C}^{m\times m}, \tag{4.21}$$

and thus (4.20) becomes

$$\boldsymbol{f}_{2m} = \beta V_m^{(1)} f(H_m^{(1)})\boldsymbol{e}_1 + \beta V_m^{(2)} X_{2,1}\boldsymbol{e}_1.$$

From the commutativity of a matrix and its matrix function, $H_{2m}f(H_{2m}) = f(H_{2m})H_{2m}$, we obtain a Sylvester equation

$$H_m^{(2)} X_{2,1} - X_{2,1} H_m^{(1)} = \eta_{m+1,m}^{(1)} \left[f(H_m^{(2)})\boldsymbol{e}_1\boldsymbol{e}_m^T - \boldsymbol{e}_1\boldsymbol{e}_m^T f(H_m^{(1)}) \right]$$

with an unknown $X_{2,1}$. This problem is well conditioned only if the spectra of $H_m^{(1)}$ and $H_m^{(2)}$ are well separated, see [22], pp. 292-294. Instead of that, we can derive a computable expression by way of interpolation, see [15], pp. 2489.

Lemma 4.1.6 *Consider two successive Arnoldi decompositions (4.17) and (4.18). Let $w_m^{(1)}$, $w_m^{(2)}$ and w_{2m} denote the monic nodal polynomials associated with $\Lambda(H_m^{(1)}), \Lambda(H_m^{(2)})$ and $\Lambda(H_{2m}) = \Lambda(H_m^{(1)}) \cup \Lambda(H_m^{(2)})$, respectively with H_{2m} being the upper Hessenberg matrix of the combined Arnoldi-like decomposition (4.19). Then*

$$[I_{w_{2m}} f]\,(H_{2m})\boldsymbol{e}_1 = \begin{bmatrix} \left[I_{w_m^{(1)}} f\right](H_m^{(1)})\boldsymbol{e}_1 \\ \gamma_m^{(1)} \left[I_{w_m^{(2)}} (\Delta_{w_m^{(1)}} f)\right](H_m^{(2)})\boldsymbol{e}_1 \end{bmatrix}, \tag{4.22}$$

where $\gamma_m^{(1)} = \prod_{j=1}^m \eta_{j+1,j}^{(1)}$.

Proof. Due to the block triangular structure of H_{2m} as (4.21)

$$[I_{w_{2m}} f]\left(\begin{bmatrix} H_m^{(1)} & O \\ \eta_{m+1,m}^{(1)} & H_m^{(2)} \end{bmatrix} \right) = \begin{bmatrix} [I_{w_{2m}} f]\,(H_m^{(1)}) & O \\ X_{2,1} & [I_{w_{2m}} f]\,(H_m^{(2)}) \end{bmatrix}. \tag{4.23}$$

First, we prove the polynomial identity

$$[I_{w_{2m}} f] = I_{w_{2m}^{(1)}} f + I_{w_{2m}^{(2)}} \left(\Delta_{w_m^{(1)}} f \right) w_m^{(1)}, \tag{4.24}$$

by showing that polynomials on a both sides of (4.24) have the same degree $2m-1$ and interpolate f in the Hermite sense at the nodes $\Lambda(H_m^{(1)}) \cup \Lambda(H_m^{(2)})$. For the nodes $\vartheta \in \Lambda(H_m^{(1)})$, we have $w_m^{(1)}(\vartheta) = 0$, and therefore

$$\left[I_{w_m^{(1)}} f\right](\vartheta) + \left[I_{w_m^{(2)}} \left(\Delta_{w_m^{(1)}} f \right)\right](\vartheta) w_m^{(1)}(\vartheta) = \left[I_{w_m^{(1)}} f\right](\vartheta) = f(\vartheta) = [I_{w_{2m}} f]\,(\vartheta).$$

For the nodes $\vartheta \in \Lambda(H_m^{(2)})$, we have

$$\begin{aligned}\left[I_{w_m^{(1)}} f\right](\vartheta) + \left[I_{w_m^{(2)}}\left(\Delta_{w_m^{(1)}} f\right)\right](\vartheta) w_m^{(1)}(\vartheta) &= \left[I_{w_m^{(1)}} f\right](\vartheta) + \left[\Delta_{w_m^{(1)}} f\right](\vartheta) w_m^{(1)}(\vartheta) \\ &= f(\vartheta) = [I_{w_{2m}} f](\vartheta),\end{aligned}$$

with the second equality following from the definition (1.7). Thus (4.24) is proved. Inserting the matrix $H_m^{(1)}$ into the polynomials on both sides of (4.24), noting that $w_m^{(1)}(H_m^{(1)}) = O$, we obtain the first block of (4.23).

To verify the second block of the vector (4.22), the identity (4.24) will be written as

$$[I_{w_{2m}} f](H_{2m}) = M^{(1)} + M^{(2)} M^{(3)},$$

where the matrices

$$M^{(1)} := \left[I_{w_m^{(1)}} f\right](H_{2m}), \quad M^{(2)} := \left[I_{w_{2m}^{(2)}}(\Delta_{w_m^{(1)}} f)\right](H_{2m}), \quad M^{(3)} := w_m^{(1)}(H_{2m})$$

have a block lower triangular structure

$$M^{(i)} = \begin{bmatrix} M_{1,1}^{(i)} & O \\ M_{2,1}^{(i)} & M_{2,2}^{(i)} \end{bmatrix}, \quad i = 1, 2, 3.$$

In addititon $M_{1,1}^{(3)} = w_m^{(1)}(H_m^{(1)}) = O$. Using this notation, the second block of (4.22) is given by

$$X_{2,1}\boldsymbol{e}_1 = M_{2,1}^{(1)}\boldsymbol{e}_1 + M_{2,2}^{(2)} M_{2,1}^{(3)}\boldsymbol{e}_1. \tag{4.25}$$

For the first term on the right of (4.25), we have $M_{2,1}^{(1)}\boldsymbol{e}_1 = \mathbf{0}$ because, as the (2,1)-block of $M^{(1)} = \left[I_{w_m^{(2)}} f\right](H_{2m})$, a polynomial of degree $m-1$ in the Hessenberg matrix H_{2m}, $M_{2,1}^{(1)}$ has a zero first column. Next, again by the block lower triangular structure of H_{2m}, it holds $M_{2,2}^{(2)} = \left[I_{w_{2m}^{(2)}}(\Delta_{w_m^{(1)}} f)\right](H_m^{(2)})$. Finally, we note that $M_{2,1}^{(3)}\boldsymbol{e}_1 = \gamma_m^{(1)}\boldsymbol{e}_1$. This follows in a similar way as the evaluation of $M_{2,1}^{(1)}\boldsymbol{e}_1$, but there $M^{(3)} = w_m^{(1)}(H_{2m})$ is a polynomial of degree m in the $2m \times 2m$ upper Hessenberg matrix H_{2m}. Again by the sparsity structure of powers of Hessenberg matrices, the first column of $M_{2,1}^{(3)}$ is a multiple of $\boldsymbol{e}_1$. Comparing coefficients reveals this multiple to be $\gamma_m^{(1)}$. Inserting these quantities in (4.25) establishes the second block of identity (4.22), and the proof is complete.

□

Thanks to Lemma 4.1.6, we can find an expression for $X_{2,1}\boldsymbol{e}_1$, i.e., comparing (4.20) and (4.22) reveals that

$$X_{2,1}\boldsymbol{e}_1 = \gamma_m^{(1)}\left[\Delta_{w_m^{(1)}} f\right](H_m^{(2)})\boldsymbol{e}_1.$$

Then the approximation (4.20) based on the Arnoldi-like decomposition (4.5) can be written as

$$\boldsymbol{f}_{2m} = \beta V_m^{(1)} f(H_m^{(1)})\boldsymbol{e}_1 + \beta\gamma_m^{(1)} V_m^{(2)}\left[\Delta_{w_m^{(1)}} f\right](H_m^{(2)})\boldsymbol{e}_1. \tag{4.26}$$

Thus, approximations (4.26) for the restarted Krylov subspace method are computed subsequently in the form

$$\boldsymbol{f}^{(1)} = \beta V_m^{(1)} f(H_m^{(1)})\boldsymbol{e}_1$$

$$\boldsymbol{f}^{(2)} = \boldsymbol{f}^{(1)} + \beta\gamma_m^{(1)} V_m^{(2)} \left[\Delta_{w_m^{(1)}} f\right](H_m^{(2)})\boldsymbol{e}_1,$$

and after k steps

$$\boldsymbol{f}^{(k)} = \boldsymbol{f}^{(k-1)} + \gamma_m^{(k-1)} V_m^{(k)} f^{(k-1)}(H_m^{(k)})\boldsymbol{e}_1,$$

where $\gamma_m^{(k)} = \gamma_m^{(k-1)} \prod_{j=1}^m \eta_{j+1,j}^{(k)}$, $\gamma_m^{(0)} = \beta$, $f^{(k)} = \Delta_{w_m}^{(k)} f^{(k-1)}$ and $f^{(0)} = f$. We summarize this into the following algorithm:

Algorithm 3

Given $A, \boldsymbol{b}, f$

1. $\beta = \|\boldsymbol{b}\|, f^{(0)} = f, \boldsymbol{f}^{(0)} = \boldsymbol{0}, \gamma^{(0)} = 1, \boldsymbol{v}_{m+1}^{(0)} = \boldsymbol{b}/\beta$
2. **for** $i = 1, 2, \ldots, k$
3. Compute the decomposition $AV_m^{(i)} = V_m^{(i)} H_m^{(i)} + \eta_{m+1,m}^{(i)} \boldsymbol{v}_{m+1}^{(i)} \boldsymbol{e}_m^T$ of $\mathscr{K}_m(A, \boldsymbol{v}_{m+1}^{(i-1)})$.
4. Update the approximation $\boldsymbol{f}^{(i)} = \boldsymbol{f}^{(i-1)} + \gamma_m^{(i-1)} V_m^{(i)} f^{(i-1)}(H_m^{(i)})\boldsymbol{e}_1$.
5. $\gamma_m^{(i)} = \gamma_m^{(i-1)} \prod_{j=1}^m \eta_{j+1,j}^{(i)}$
6. $f^{(i)} = \Delta_{w_m^{(i)}} f^{(i-1)}$, where $w_m^{(i)}$ is the characteristic polynomial of $H_m^{(i)}$.
7. **enddo**

This algorithm seems to be very attractive from a computational point of view, but it can be affected by several stability problems, caused by the difficulty of numerical computing of interpolation polynomials of high degree. Therefore the following less efficient variant, which is free from numerical problems, was derived in [15], pp. 2491-2492.

After $k-1$ restarts of the Arnoldi algorithm, we can collect the Arnoldi decompositions into the $(k-1)$-fold Arnoldi-like decomposition

$$AW_{(k-1)m} = W_{(k-1)m} H_{(k-1)m} + \eta_{m+1,m}^{(k-1)} \boldsymbol{v}_{m+1}^{(k-1)} \boldsymbol{e}_{(k-1)m}^T,$$

where $W_{(k-1)m} = \left[V_m^{(1)} V_m^{(2)}, \ldots, V_m^{(k-1)}\right]$. Combining this with the Arnoldi decomposition

$$AV_m^{(k)} = V_m^{(k)} H_m^{(k)} + \eta_{m+1,m}^{(k)} \boldsymbol{v}_{m+1}^{(k)} \boldsymbol{e}_m^T$$

of the following Krylov space $\mathscr{K}_m(A, \boldsymbol{v}_{m+1}^{(k-1)})$, we obtain the following Arnoldi-like decomposition

$$AW_{km} = W_{km} H_{km} + \eta_{m+1,m}^{(k)} \boldsymbol{v}_{m+1,m}^{(k)} \boldsymbol{e}_{km}^T$$

with $W_{km} = \left[W_{(k-1)m}, V_m^{(k)}\right]$ and

$$H_{km} = \begin{bmatrix} H_{(k-1)m} & O \\ \eta_{m+1,m}^{(k-1)} \boldsymbol{e}_1 \boldsymbol{e}_{(k-1)m}^T & H_m^{(k)} \end{bmatrix}.$$

Than the required approximation is

$$\boldsymbol{f}^{(k)} = \beta W_{km} f(H_{km})\boldsymbol{e}_1 = \boldsymbol{f}^{(k-1)} + \beta V_m^{(k)} \left[f(H_{km})\boldsymbol{e}_1\right]_{(k-1)m+1:km}, \tag{4.27}$$

giving the following algorithm:

Algorithm 4

Given $A, \boldsymbol{b}, f$

1. $\beta = \|\boldsymbol{b}\|, \boldsymbol{f}^{(0)} = \boldsymbol{0}, \boldsymbol{v}_{m+1}^{(0)} = \boldsymbol{b}/\beta$
2. **for** $i = 1, 2, \ldots, k$
3. Compute the decomposition $AV_m^{(i)} = V_m^{(i)} H_m^{(i)} + \eta_{m+1,m}^{(i)} \boldsymbol{v}_{m+1}^{(i)} \boldsymbol{e}_m^T$ of $\mathscr{K}_m(A, \boldsymbol{v}_{m+1}^{(i-1)})$.
4. **if** $i = 1$ **then**
5. $H_{im} = H_m^{(1)}$
6. **else**
7. $H_{im} = \begin{bmatrix} H_{(i-1)m} & O \\ \eta_{m+1,m}^{(i-1)} \boldsymbol{e}_1 \boldsymbol{e}_{(i-1)m}^T & H_m^{(i)} \end{bmatrix}$
8. **endif**
9. **enddo**
10. Update the approximation $\boldsymbol{f}^{(i)} = \boldsymbol{f}^{(i-1)} + \beta V_m^{(i)} \left[f(H_{im}) \boldsymbol{e}_1 \right]_{(i-1)m+1:im}$.

This allows us to discard the basis vectors of previous cycles, but, on the other hand, it requires the evaluation of f for a matrix of the size km in the k-th cycle. This can become a substantial computational problem as k gets large. Moreover, we need just the first m entries of the first column of $f(H_{km})$, but we compute the whole matrix. An alternative approach, which promises less work per cycle is to use commutativity of a function and its matrix function, is described in [2], pp. 5-6. By comparing the blocks in the identity

$$f(H_{km}) H_{km} = H_{km} f(H_{km}),$$

we obtain a system of Sylvester equations

$$X_{k,k-j} H_m^{(k-j)} - H_m^{(k)} X_{k,k-j} = \eta_{m+1,m}^{(k-1)} \boldsymbol{e}_1 \boldsymbol{e}_m^T X_{k-1,k-j} - X_{k,k-j+1} \eta_{m+1,m}^{(k-j)} \boldsymbol{e}_1 \boldsymbol{e}_m^T,$$

for $j = 1, 2, \ldots, k-1$, where

$$H_{km} = \begin{bmatrix} H_m^{(1)} & & & \\ \eta_{m+1,m}^{(1)} \boldsymbol{e}_1 \boldsymbol{e}_m^T & H_m^{(2)} & & \\ & \ddots & \ddots & \\ & & \eta_{m+1,m}^{(k-1)} \boldsymbol{e}_1 \boldsymbol{e}_m^T & H_m^{(k)} \end{bmatrix} \in \mathbb{C}^{km \times km},$$

$$f(H_{km}) = \begin{bmatrix} X_{1,1} & & & \\ X_{2,1} & X_{2,2} & & \\ \vdots & \vdots & \ddots & \\ X_{k,1} & X_{k,2} & & X_{k,k} \end{bmatrix} \in \mathbb{C}^{km \times km}.$$

The matrices $H_m^{(k-j)}, H_m^{(k)}$ are upper Hessenberg and thus the Sylvester equations are easy to solve. We still, however, have to store the whole matrix H_{km}, i.e. the matrices $H_1, \ldots, H_k$ and the elements $\eta_{m+1,m}^{(1)}, \ldots, \eta_{m+1,m}^{(k)}$. Moreover, the Sylvester equation tends to be severely ill-conditioned since H_j and H_{j-k} represent compressions of the same matrix A and thus their spectra are by no means well separated, see [2], pp. 6.

Error of the restarted Krylov subspace approximation

Now, we study the error of the restarted Krylov subspace approximation (4.27), using an extension of an idea described in [46], pp. 223-224 and [43], pp. 95-119. Given $\widetilde{m}$ complex nodes $\widetilde{\vartheta}_1, \widetilde{\vartheta}_2, \cdots, \widetilde{\vartheta}_{\widetilde{m}}$ such that $f(\widetilde{\vartheta}_j)$ is defined for each j, we define the sequence of associated nodal polynomials

$$p_0(z) := 1, \quad p_j(z) := (z - \widetilde{\vartheta}_1)(z - \widetilde{\vartheta}_2) \dots (z - \widetilde{\vartheta}_j), \quad j = 1, 2, \dots, \widetilde{m}.$$

We denote the associated divided differences of f by

$$\Phi_0(z) := f(z), \quad \Phi_j(z) := \left[\Delta_{w_j} f\right](z).$$

From the interpolation identity (see (4.24))

$$I_{w_{j+1}} f = I_{w_j} f + w_j \Delta_{w_j} f, \quad j = 0, 1, \dots, \widetilde{m} - 1,$$

we see that these obey the recursion

$$\Phi_j(z) = \frac{\Phi_{j-1}(z) - \Phi_{j-1}(\widetilde{\vartheta}_j)}{z - \widetilde{\vartheta}_j}, \quad j = 1, 2, \dots, \widetilde{m}.$$

For $0 \le \ell \le \widetilde{m}$ and $0 \le j \le \widetilde{m} - \ell$, we define by

$$\Delta_\ell^j f := \frac{1}{2\pi i} \int_\Gamma \frac{f(t)}{(t - \widetilde{\vartheta}_\ell) \cdots (t - \widetilde{\vartheta}_{\ell+j})} dt$$

the j**th order divided difference with respect to the nodes** $\widetilde{\vartheta}_\ell, \widetilde{\vartheta}_{\ell+1}, \dots, \widetilde{\vartheta}_{\ell+j}$. Let us now consider the matrix

$$\widetilde{W}_{\widetilde{m}} := \left[p_0(A)\boldsymbol{v}_{m+1}^{(k)}, p_1(A)\boldsymbol{v}_{m+1}^{(k)}, \dots, p_{\widetilde{m}-1}(A)\boldsymbol{v}_{m+1}^{(k)}\right] \in \mathbb{C}^{n \times \widetilde{m}} \tag{4.28}$$

and the bidiagonal matrix

$$\widetilde{B}_{\widetilde{m}} = \begin{bmatrix} \widetilde{\vartheta}_1 & & & \\ 1 & \widetilde{\vartheta}_2 & & \\ & \ddots & \ddots & \\ & & 1 & \widetilde{\vartheta}_{\widetilde{m}} \end{bmatrix} \in \mathbb{C}^{\widetilde{m} \times \widetilde{m}},$$

for which

$$A\widetilde{W}_{\widetilde{m}} = \widetilde{W}_{\widetilde{m}} \widetilde{B}_{\widetilde{m}} + \left[0, \dots, 0, p_{\widetilde{m}}(A)\boldsymbol{v}_{m+1}^{(k)}\right]. \tag{4.29}$$

Extending (4.5) by (4.29), we obtain the Arnoldi-like decomposition

$$A\left[W_{km}, \widetilde{W}_{\widetilde{m}}\right] = \left[W_{km}, \widetilde{W}_{\widetilde{m}}\right] \widetilde{H}_{km+\widetilde{m}} + p_{\widetilde{m}}(A)\boldsymbol{v}_{m+1}^{(k)} \boldsymbol{e}_{km+\widetilde{m}}^T,$$

where

$$\widetilde{H}_{km+\widetilde{m}} = \begin{bmatrix} H_{km} & O \\ \eta_{m+1,m}^{(k-1)} \boldsymbol{e}_1 \boldsymbol{e}_{km}^T & \widetilde{B}_{\widetilde{m}} \end{bmatrix} \in \mathbb{C}^{(km+\widetilde{m}) \times (km+\widetilde{m})}.$$

If we approximate $f(A)\boldsymbol{b} \approx \widetilde{\boldsymbol{f}}_k := [W_{km} W_{\widetilde{m}}]\, f(\widetilde{H}_{km+\widetilde{m}})\boldsymbol{e}_1$, then the associated error may by represented as

$$f(A)\boldsymbol{b} - \widetilde{\boldsymbol{f}}_k = \widetilde{f}(A) p_{\widetilde{m}}(A) \boldsymbol{v}_{m+1}^{(k)}, \tag{4.30}$$

where $\widetilde{f} := \widetilde{\gamma}_{km+\widetilde{m}} \Delta_{\widetilde{w}} f$, $\widetilde{w} \in \mathscr{P}_{km+\widetilde{m}}$ is the characteristic polynomial of $\widetilde{H}_{km+\widetilde{m}}$, $\widetilde{\gamma}_{km+\widetilde{m}}$ is the product of the subdiagonal entries of $\widetilde{H}_{km+\widetilde{m}}$.

Lemma 4.1.7 *In terms of the notation introduced above,*

$$f(\widetilde{H}_{km+\widetilde{m}}) = \begin{bmatrix} f(H_{km}) & O \\ \widetilde{F}_{k,\widetilde{m}} & f(\widetilde{B}_{\widetilde{m}}) \end{bmatrix},$$

$$f(\widetilde{B}_{\widetilde{m}}) = \begin{bmatrix} f(\vartheta_1) & & & \\ \Delta_1^1 & f(\vartheta_2) & & \\ \vdots & \vdots & \ddots & \\ \Delta_1^{\widetilde{m}-1} & \Delta_2^{\widetilde{m}-2} & \cdots & f(\vartheta_{\widetilde{m}}) \end{bmatrix}$$

and

$$\boldsymbol{e}_j^T \widetilde{F}_{k,\widetilde{m}} = \eta_{m+1,m}^{(k-1)} \boldsymbol{e}_{km}^T \Phi_j(H_{km}).$$

Proof of this lemma, based on the result of Opitz, [42], can be found in [2], pp. 10-11.

From Lemma 4.1.7 and from the definition of $\widetilde{W}_{\widetilde{m}}$ in (4.28), we can see that the approximation is

$$\widetilde{\boldsymbol{f}}_k = W_{km} f(H_{km})\boldsymbol{e}_1 + \eta_{m+1,m}^{(k-1)} \sum_{j=1}^{\widetilde{m}} \left[\boldsymbol{e}_{km}^T \Phi_j(H_{km})\boldsymbol{e}_1\right] p_{j-1}(A)\boldsymbol{v}_{m+1}^{(k)}.$$

This result, together with the error representation (4.30) gives us the error of the restarted Krylov subspace approximation (4.27), see [2], pp. 11,

$$f(A)\boldsymbol{b} - W_{km} f(H_{km})\boldsymbol{e}_1 = \eta_{m+1,m}^{(k-1)} \sum_{j=1}^{\widetilde{m}} \left[e_{km}^T \Phi_j(H_{km})\boldsymbol{e}_1\right] p_{j-1}(A)\boldsymbol{v}_{m+1}^{(k)} + \widetilde{f}(A) p_{\widetilde{m}}(A)\boldsymbol{v}_{m+1}^{(k)}. \tag{4.31}$$

In [43] it was shown, that the remainder term in (4.31) may be written as

$$\widetilde{f}(A) p_{\widetilde{m}}(A)\boldsymbol{v}_{m+1}^{(k)} = p_{\widetilde{m}}(A)\left[\Phi_{\widetilde{m}}(A)\boldsymbol{b} - W_{km}\Phi_{\widetilde{m}}(H_{km})\boldsymbol{e}_1\right].$$

4.2 Modification of the standart and the restarted Krylov subspace method based on rational approximation to f

When we approximate matrix functions using the standart or the restarted Krylov subspace method, the following modification can be used, see [2], pp. 7-9, and [19], pp. 8-10. First, a function f is approximated by a partial fraction,

$$f(z) \approx r(z) = \frac{n_{pq}(z)}{d_{pq}(z)} = h(z) + \sum_{\ell=1}^{N} \frac{\alpha_\ell}{\omega_\ell - z}, \quad N \in \mathbb{N},$$

where $n_{pq}(z)$ is a polynomial of degree p and $d_{pq}(z)$ is a polynomial of degree q and thus $h(z)$ is a polynomial of degree $p-q$ for $p \geq q$ and $h \equiv 0$ for $p < q$. Then

$$\begin{aligned} f(A)\boldsymbol{b} \approx r(A)\boldsymbol{b} &= h(A)\boldsymbol{b} + \sum_{\ell=1}^{N} \alpha_\ell (\omega_\ell I - A)^{-1}\boldsymbol{b} \\ &= \beta W_{km} h(H_{km})\boldsymbol{e}_1 + \beta W_{km} \sum_{\ell=1}^{N} \alpha_\ell (\omega_\ell I - H_{km})^{-1}\boldsymbol{e}_1. \end{aligned}$$

That means, that for the restarted Krylov subspace method and also for the standart Krylov subspace method (set $k = 1$ in the following derivations), we need to find

$$r(H_{km})\boldsymbol{e}_1 := h(H_{km})\boldsymbol{e}_1 + \sum_{\ell=1}^{N} \alpha_\ell (\omega_\ell I - H_{km})^{-1}\boldsymbol{e}_1. \tag{4.32}$$

Denote $\widehat{r}_0 = h(H_{km})\boldsymbol{e}_1$ and $\widehat{r}_\ell = (\omega_\ell I - H_{km})^{-1}\boldsymbol{e}_1$. Then (4.32) becomes

$$r(H_{km})\boldsymbol{e}_1 = \widehat{r}_0 + \sum_{\ell=1}^{N} \alpha_\ell \widehat{r}_\ell.$$

In case that h is a polynomial of a low degree, evaluation of $\widehat{r}_0$ is straightforward. E.g., for $h(z) = a_1 z + a_0$ we have

$$\widehat{r}_0 = h(H_{km})\boldsymbol{e}_1 = \left[(a_1 H_m^{(1)} + a_0 I)\boldsymbol{e}_1, a_2 \eta_{m+1,m}^{(1)} \boldsymbol{e}_1 \boldsymbol{e}_m^T \boldsymbol{e}_1, 0, \ldots, 0\right]^T.$$

Evaluating $\widehat{r}_\ell$ is done via solving the linear system of equations

$$(\omega_\ell I - H_{km})\widehat{r}_\ell = \boldsymbol{e}_1.$$

Due to the sparsity pattern of the right hand side $\boldsymbol{e}_1$ and the block lower triangular form of H_{km}, this can be computed recursively. Let

$$\widehat{r}_\ell = \left[\boldsymbol{r}_{\ell,1}^T, \boldsymbol{r}_{\ell,2}^T, \ldots, \boldsymbol{r}_{\ell,k}^T\right]^T$$

be partioned comformingly with H_{km}. Then the reccurence is

$$(\omega_\ell I - H_1)\boldsymbol{r}_{\ell,1} = \boldsymbol{e}_1, \quad (\omega_\ell I - H_j)\boldsymbol{r}_{\ell,j} = \eta_{m+1,m}^{(j-1)} \boldsymbol{e}_1 \boldsymbol{e}_m^T \boldsymbol{r}_{\ell,j-1}, \quad j = 2, \ldots, k$$

Moreover, for evaluation of (4.27) only the last block of $r(H_{km})\boldsymbol{e}_1$ is required, which can be obtained as

$$[O, \ldots, O, I]\, r(H_{km})\boldsymbol{e}_1 = \boldsymbol{r}_{0,k} + \sum_{\ell=1}^{N} \alpha_\ell \boldsymbol{r}_{\ell,k}.$$

Note that the rational approximation to f is usually real for real arguments, but its poles ω_ℓ and α_ℓ appear in complex conjugate pairs, $\omega_{\ell+1} = \overline{\omega}_\ell$ and $\alpha_{\ell+1} = \overline{\alpha}_\ell$. Since all other quantities in the equations $(\omega_\ell I - \widehat{H}_k)\widehat{r}_\ell = \boldsymbol{e}_1$ are real, we have $\widehat{r}_{\ell+1} = \overline{\widehat{r}}_\ell$ and therefore $r_{\ell+1,j} = \overline{r}_{\ell,j}$. Thus $\alpha_\ell \boldsymbol{r}_{\ell,k} + \alpha_{\ell+1}\boldsymbol{r}_{\ell+1,k} = 2\mathrm{Re}(\alpha_\ell \boldsymbol{r}_{\ell,k}) = 2\left[\mathrm{Re}(\alpha_\ell)\boldsymbol{r}_{\ell,k}^{(R)} - \mathrm{Im}(\alpha_\ell)\boldsymbol{r}_{\ell,k}^{(I)}\right]$. Setting $\boldsymbol{r}_{\ell,j} = \boldsymbol{r}_{\ell,j}^{(R)} + i\boldsymbol{r}_{\ell,j}^{(I)}$ and $\omega_\ell = \omega_\ell^{(R)} + i\omega_\ell^{(I)}$, straightforward computation shows that

$$\begin{aligned} \left(|\omega_\ell|^2 I - 2\omega_\ell^{(R)} H_m^{(1)} + H_m^{(1)^2}\right)\boldsymbol{r}_{\ell,1}^{(R)} &= \left(\omega_\ell^{(R)} I - H_m^{(1)}\right)\boldsymbol{e}_1 \\ \boldsymbol{r}_{\ell,1}^{(I)} &= \tfrac{1}{\omega_\ell^{(I)}}\left(\left[\omega_\ell^{(R)} I - H_m^{(1)}\right]\boldsymbol{r}_{\ell,1}^{(R)} - \boldsymbol{e}_1\right). \end{aligned}$$

For $j = 2, 3, \ldots, k$, we have

$$\begin{aligned}\left(|\omega_\ell|^2 I - 2\omega_\ell^{(R)} H_m^{(j)} + H_m^{(j)^2}\right) \boldsymbol{r}_{\ell,j}^{(R)} &= \omega_\ell^{(I)} \eta_{m+1,m}^{(j-1)} \boldsymbol{e}_1 \boldsymbol{e}_m^T \boldsymbol{r}_{\ell,j-1}^{(I)} + \left(\omega_\ell^{(R)} I - H_m^{(j)}\right) \eta_{m+1,m}^{(j-1)} \boldsymbol{e}_1 \boldsymbol{e}_m^T \boldsymbol{r}_{\ell,j-1}^{(R)} \\ \boldsymbol{r}_{\ell,j}^{(I)} &= \frac{1}{\omega_\ell^{(I)}} \left(\left[\omega_\ell^{(R)} I - H_m^{(j)}\right] \boldsymbol{r}_{\ell,j}^{(R)} - \eta_{m+1,m}^{(j-1)} \boldsymbol{e}_1 \boldsymbol{e}_m^T \boldsymbol{r}_{\ell,j-1}^{(R)}\right)\end{aligned}$$

which avoids complex arithmetic.

Now we can give the algorithm for approximating $f(A)\boldsymbol{b}$ using rational approximation of f, according to [2], pp. 8:

Algorithm 5

Given $A, \boldsymbol{b}, f$, coefficients and poles $(\alpha_\ell, \omega_\ell)$ of a rational function $r \approx f$

1. $\beta = \|\boldsymbol{b}\|, \boldsymbol{f}^{(0)} = \boldsymbol{0}, \boldsymbol{v}_{m+1}^{(0)} = \boldsymbol{b}/\beta$
2. **for** $i = 1, 2, \ldots, k$ **do**
3. Compute the decomposition $AV_m^{(i)} = V_m^{(i)} H_m^{(i)} + \eta_{m+1,m}^{(i)} \boldsymbol{v}_{m+1}^{(i)} \boldsymbol{e}_m^T$ of $\mathscr{K}_m(A, \boldsymbol{v}_{m+1}^{(i-1)})$.
4. **if** $i = 1$ **then**
5. **for** $\ell = 1, 2, \ldots, N$ **do**
6. Solve $(\omega_\ell I - H_1)\boldsymbol{r}_{\ell,1} = \boldsymbol{e}_1$.
7. **enddo**
8. **else**
9. **for** $\ell = 1, 2, \ldots, N$ **do**
10. Solve $(\omega_\ell I - H_i)\boldsymbol{r}_{\ell,i} = \eta_{m+1,m}^{(i-1)} (\boldsymbol{e}_m^T \boldsymbol{r}_{\ell,i-1}) \boldsymbol{e}_1$.
11. **enddo**
12. $\boldsymbol{h}_i = \sum_{\ell=1}^N \alpha_\ell \boldsymbol{r}_{\ell,i}$
13. Update approximation $\boldsymbol{f}^{(i)} = \boldsymbol{f}^{(i-1)} + V_m^{(i)} \boldsymbol{h}_i$.
14. **enddo**

4.3 Generalisation of the steepest descent method for matrix functions

In the work of M. Afanasjew, M. Eiermann, O. G. Ernst and S. Güttel [1] a special case of the restarted Krylov subspace method is considered, where the restart length is set to one. In that special case, the algorithm of the restarted method after k restarts gives a decomposition

$$AW_k = W_{k+1} B_{k+1,k} = W_k B_k + \sigma_{k+1} \boldsymbol{v}_{k+1} \boldsymbol{e}_k^T, \tag{4.33}$$

where

$$B_{k+1,k} = \begin{bmatrix} \rho_1 & & & \\ \sigma_2 & \rho_2 & & \\ & \sigma_3 & \ddots & \\ & & \ddots & \rho_k \\ & & & \sigma_{k+1} \end{bmatrix} \in \mathbb{C}^{(k+1)\times k},$$

$B_k = [I_k \mathbf{0}]\, B_{k+1,k} \in \mathbb{C}^{k\times k}$ and $W_k = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k] \in \mathbb{C}^{n\times k}$. Resulting approximation of $f(A)\boldsymbol{b}$ is then defined as

$$\boldsymbol{f}^{(k)} := \sigma_1 W_k f(B_k)\boldsymbol{e}_1. \tag{4.34}$$

The algorithm of the restarted Krylov subspace method in this special case, where the restart length m is set to one, is:

Algorithm 6

Given $\boldsymbol{b}, A$, set $\sigma_1 := \|\boldsymbol{b}\|\,,\ \boldsymbol{v}_1 := \boldsymbol{b}/\sigma_1$.
1 **for** $i = 1, 2, \ldots, k$ **do**
2. $\boldsymbol{w} = A\boldsymbol{v}_i$
3. $\rho_i = \boldsymbol{v}_i^H \boldsymbol{w}$
4. $\boldsymbol{w} = \boldsymbol{w} - \rho_i \boldsymbol{v}_i$
5. $\sigma_{i+1} = \|\boldsymbol{w}\|$
6. $\boldsymbol{v}_{i+1} = \boldsymbol{w}/\sigma_{i+1}$
7. **enddo**

Approximants $\boldsymbol{f}^{(k)}$ can be explicitly represented, see [1], pp. 209, as

$$\boldsymbol{f}^{(k)} = \sum_{r=1}^{k} \left(\prod_{j=1}^{r} \sigma_j \right) \left(\Delta_1^{r-1} f \right) \boldsymbol{v}_r = \boldsymbol{f}^{(k-1)} + \left(\prod_{j=1}^{k} \sigma_j \right) \left(\Delta_1^{k-1} f \right) \boldsymbol{v}_k.$$

The following theorem, see [1], pp. 209-210, gives a convergence result for $\boldsymbol{f}_k$, connection to a certain interpolation processes is given in [11], pp. 81-83.

Theorem 4.3.1 *Let* $W(A) := \left\{ \boldsymbol{v}^H A \boldsymbol{v} : \|\boldsymbol{v}\| = 1 \right\}$ *denote the field of values of* A *and let*

$$\delta := \max_{\zeta,\eta \in W(A)} |\zeta - \eta|$$

be its diameter. Let f *be analytic in (a neighborhood of)* $W(A)$ *and let* $\rho > 0$ *be maximal such that* f *can be continued analytically to*

$$W_\rho := \left\{ \lambda \in \mathbb{C} : \min_{\eta \in W(A)} |\lambda - \zeta| < \rho \right\}.$$

If f *is entire, we set* $\rho = \infty$. *If* $\rho > \delta$, *then*

$$\lim_{k\to\infty} \boldsymbol{f}^{(k)} = f(A)\boldsymbol{b}$$

and the convergence is at least linear.

We note that this theorem holds also for Arnoldi approximation of arbitrary restart length and also for its unrestarted variant. We also have superlinear convergence if f is an entire function, see [15], pp. 2495-2496.

We will reffer to the restarted Krylov subspace method with restart length one as a **steepest descent method for matrix functions**. This follows from the special case $f(A) = A^{-1}$, where A is Hermitian, positive definite matrix, i.e. f is a function we need for solving a system of linear equations $A\boldsymbol{x} = \boldsymbol{b}$. Here, the method is equivalent to FOM

with restart length 1 as well as to IOM with the truncation parameter equal to 1. If we choose $\boldsymbol{f}^{(0)} = 0$ as the initial approximation, then

$$\boldsymbol{f}^{(k)} = \boldsymbol{f}^{(k-1)} + (\sigma_1\sigma_2\cdots\sigma_k)\left(\Delta_1^{k-1}f\right)\boldsymbol{v}_k = \boldsymbol{f}^{(k-1)} + \alpha_k\boldsymbol{r}_{k-1},$$

where $\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{f}^{(k)}$ and

$$\alpha_k = \frac{1}{\rho_k} = \frac{1}{\boldsymbol{v}_k^H A\boldsymbol{v}_k} = \frac{\boldsymbol{r}_{k-1}^H\boldsymbol{r}_{k-1}}{\boldsymbol{r}_{k-1}^H A\boldsymbol{r}_{k-1}},$$

which is known as the method of the steepest descent.

Asymptotic behaviour

Now we show, based on [1], pp. 210-214, that the method with restart length $m = 1$ yields asymptoticaly two periodical behaviour.

First we consider a special case, where the vector $\boldsymbol{b}$ (and therefore $\boldsymbol{v}_1$) is a linear combination of two first orthonormal eigenvectors of a Hermitian matrix $A \in \mathbb{C}^{n\times n}$. Here

$$\boldsymbol{v}_1 = \frac{1}{\sqrt{1+|\gamma|^2}}\boldsymbol{z}_1 + \frac{\gamma}{\sqrt{1+|\gamma|^2}}\boldsymbol{z}_2,$$

where $\boldsymbol{z}_j, j = 1, 2$, are normalized eigenvectors associated with the eigenvalues λ_j of A. Then for $k = 1, 2, \ldots,$

$$\boldsymbol{v}_{2k-1} = \boldsymbol{v}_1 = \frac{1}{\sqrt{1+|\gamma|^2}}\boldsymbol{z}_1 + \frac{\gamma}{\sqrt{1+|\gamma|^2}}\boldsymbol{z}_2, \tag{4.35}$$

$$\boldsymbol{v}_{2k} = \boldsymbol{v}_2 = -\frac{|\gamma|}{\sqrt{1+|\gamma|^2}}\boldsymbol{z}_1 + \frac{\gamma}{|\gamma|\sqrt{1+|\gamma|^2}}\boldsymbol{z}_2. \tag{4.36}$$

This can be easily verifed by using straightforward calculation, see [6], pp. 210. Under the assumptions above, the entries of the bidiagonal matrices B_k are given by

$$\begin{aligned} \rho_{2k-1} &= \theta\lambda_1 + (1-\theta)\lambda_2, \\ \rho_{2k} &= (1-\theta)\lambda_1 + \theta\lambda_2, \\ \sigma_{k+1} &= \sqrt{\theta(1-\theta)}(\lambda_2 - \lambda_1), \end{aligned}$$

with $\theta := 1/(1+|\gamma|^2)$, see [1], pp. 211.

Now, we turn to a general case. Assume, that A has only simple eigenvalues,

$$\lambda_{\min} = \lambda_1 < \lambda_2 < \ldots < \lambda_{n-1} < \lambda_n = \lambda_{\max}, \quad n \geq 2.$$

By $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ we denote the corresponding normalized eigenvectors. Assume, that $\boldsymbol{b}$ (and therefore $\boldsymbol{v}_1$) has nonzero components in all eigenvectors, $\boldsymbol{z}_j^H\boldsymbol{b} \neq 0$, for $j = 1, 2\ldots, n$. Suppose, that A is diagonal (otherwise replace A by Q^HAQ and $\boldsymbol{b}$ by $Q^H\boldsymbol{b}$) and $\boldsymbol{b}$ is real (otherwise replace $\boldsymbol{b}$ by $Q^H\boldsymbol{b}$, where $Q = \mathrm{diag}(b_1/|b_1|, \ldots, b_n/|b_n|)$). In summary, we can

assume, that A is Hermitian matrix with pairwise distinct diagonal entries and that $\boldsymbol{b}$ is a real vector with nonzeros entries. Now, we introduce theoretical background, necessary to formulate the key theorem, which describes the asymptotic behaviour of the diagonal and subdiagonal elements of the matrix B_k.

The sequence $\{\sigma_{k+1}\}_{k\in\mathbb{N}}$ is bounded,

$$0 \leq \sigma_{k+1} = \|(A-\rho_k I)\boldsymbol{v}_k\| \leq \|A-\rho_k I\| \leq \|A\| + |\rho_k| \leq 2\,\|A\|\,,$$

and nondecreasing,

$$\begin{aligned}
\sigma_{k+1} &= \|(A-\rho_k I)\boldsymbol{v}_k\| \\
&= \|\boldsymbol{v}_{k+1}\|\,\|(A-\rho_k I)\boldsymbol{v}_k\| && \text{since } \|\boldsymbol{v}_{k+1}\| = 1 \\
&= \left|\boldsymbol{v}_{k+1}^H(A-\rho_k I)\boldsymbol{v}_k\right| && \text{since } \sigma_{k+1}\boldsymbol{v}_{k+1} = (A-\rho_k I)\boldsymbol{v}_k \\
&= \left|\boldsymbol{v}_{k+1}^H(A-\rho_k I)^H\boldsymbol{v}_k\right| && \text{since } A \text{ is Hermitian} \\
&= \left|\boldsymbol{v}_{k+1}^H(A-\rho_{k+1}I+(\rho_{k+1}-\rho_k)I)^H\boldsymbol{v}_k\right| \\
&= \left|\boldsymbol{v}_{k+1}^H(A-\rho_{k+1}I)^H\boldsymbol{v}_k + (\rho_{k+1}-\rho_k)\boldsymbol{v}_{k+1}^H\boldsymbol{v}_k\right| \\
&= \left|\boldsymbol{v}_{k+1}^H(A-\rho_{k+1}I)^H\boldsymbol{v}_k\right| && \text{since } \boldsymbol{v}_k \perp \boldsymbol{v}_{k+1} \\
&\leq \|(A-\rho_{k+1}I)\boldsymbol{v}_{k+1}\|\,\|\boldsymbol{v}_k\| && \text{by the Cauchy-Schwarz inequality} \\
&= \|(A-\rho_{k+1}I)\boldsymbol{v}_{k+1}\| \\
&= \sigma_{k+2}.
\end{aligned}$$

Consequently, the sequence $\{\sigma_{k+1}\}_{k\in\mathbb{N}}$ is also convergent. Moreover, $\sigma_{k+1} = \sigma_{k+2}$ if and only if the vectors $\boldsymbol{v}_k$ and $(A-\rho_{k+1}I)\boldsymbol{v}_{k+1} = \sigma_{k+2}\boldsymbol{v}_{k+2}$ are linearly dependent, i.e. $\boldsymbol{v}_k$ and $\boldsymbol{v}_{k+2}$ are linearly dependent. The following lemmas, see [1], pp. 212-214, describe the behaviour of $\{\boldsymbol{v}_k\}_{k\in\mathbb{N}}$.

Lemma 4.3.2 *Each accumulation point of the vector sequence $\{\boldsymbol{v}_k\}_{k\in\mathbb{N}}$ is contained in the linear hull of the eigenvectors of A associated with its extremal eigenvalues.*

Proof can be found in [1] pp. 212-213.

Lemma 4.3.3 *For the vector sequence $\{\boldsymbol{v}_k\}_{k\in\mathbb{N}}$ there exist nonzero real numbers α and β, $\alpha^2+\beta^2=1$, which depend on the spectrum of A and on $\boldsymbol{b}$, such that*

$$\lim_{k\to\infty}\boldsymbol{v}_{2k-1} = \alpha\boldsymbol{z}_1 + \beta\boldsymbol{z}_n \text{ and } \lim_{k\to\infty}\boldsymbol{v}_{2k} = \operatorname{sign}(\alpha\beta)\,[-\beta\boldsymbol{z}_1 + \alpha\boldsymbol{z}_n]\,.$$

Proof. We will count the candidates for accumulation points of the sequence $\{\boldsymbol{v}_k\}$, denote them by $\boldsymbol{u}$. By Lemma 4.3.2, $\boldsymbol{u} \in \operatorname{span}\{\boldsymbol{z}_1, \boldsymbol{z}_n\}$. Since $\|\boldsymbol{u}\| = 1$, every accumulation point can be written as $\boldsymbol{u} = \alpha\boldsymbol{z}_1 + \beta\boldsymbol{z}_n$ with $\alpha^2+\beta^2=1$. For every vector of this form,

$$\left\|A\boldsymbol{u} - (\boldsymbol{u}^HA\boldsymbol{u})\boldsymbol{u}\right\|^2 = \alpha^2\beta^2(\lambda_n-\lambda_1)^2 = \alpha^2(1-\alpha^2)(\lambda_n-\lambda_1)^2.$$

Each accumulation point is a limit of a subsequence $\{\boldsymbol{v}_{k_\nu+1}\}$, for which the associated sequence $\{\sigma_{k_\nu+1}\}$ converges. Denote its limit by σ^*. Then $\left\|A\boldsymbol{u} - (\boldsymbol{u}^HA\boldsymbol{u})\boldsymbol{u}\right\| = \sigma^*$ and

$$\alpha^2(1-\alpha^2) = \left(\frac{\sigma^*}{\lambda_n-\lambda_1}\right)^2. \tag{4.37}$$

This equation has at most four solutions α, which shows that there are at most eight points of accumulation.

Assume now that $\boldsymbol{v}_k$ is sufficiently close to an accumulation point $\boldsymbol{u} = \boldsymbol{u}_1 = \alpha\, \boldsymbol{z}_1 + \beta \boldsymbol{z}_n$. Since all operations in Algorithm 6 are continuous, $\boldsymbol{v}_{k+1}$ for k sufficiently large, is arbitrarily close to

$$\boldsymbol{u}_2 = \frac{A - (\boldsymbol{u}_1^H A \boldsymbol{u}_1)\boldsymbol{u}_1}{\|A - (\boldsymbol{u}_1^H A \boldsymbol{u}_1)\boldsymbol{u}_1\|} = \operatorname{sign}(\alpha\beta)\,[-\beta \boldsymbol{z}_1 + \alpha \boldsymbol{z}_n]$$

(which is also an accumulation point of $\{\boldsymbol{v}_k\}$ different from $\boldsymbol{u}_1$ since $\alpha\beta \neq 0$). Moreover, $\boldsymbol{v}_{k+2}$ must be then close to

$$\boldsymbol{u}_3 = \frac{A - (\boldsymbol{u}_2^H A \boldsymbol{u}_2)\boldsymbol{u}_2}{\|A - (\boldsymbol{u}_2^H A \boldsymbol{u}_2)\boldsymbol{u}_2\|} = \alpha \boldsymbol{z}_1 + \beta \boldsymbol{z}_n = \boldsymbol{u}_1.$$

Since we already know there are only finitely many accumulation points of $\{\boldsymbol{v}_k\}$, we conclude that the sequence $\{\boldsymbol{v}_k\}$ must asymptotically alternate between $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$.

□

We conclude this section by the following theorem, see [1], pp. 211.

Theorem 4.3.4 *If A is Hermitian with extremal eigenvalues $\lambda_{\min}$ and $\lambda_{\max}$ and if the vector $\boldsymbol{b}$ has nonzero components in the associated eigenvectors, then there exists a real number $\theta \in (0,1)$, which depends on the spectrum of A and $\boldsymbol{b}$, such that the entries ρ_k and σ_{k+1}, $k = 1, 2, \ldots,$ of the bidiagonal matrices B_k in (4.33) satisfy*

$$\begin{aligned} \lim_{k\to\infty} \rho_{2k-1} &= \theta\lambda_{\min} + (1-\theta)\lambda_{\max} =: \rho_1^*, \\ \lim_{k\to\infty} \rho_{2k} &= (1-\theta)\lambda_{\min} + \theta\lambda_{\max} =: \rho_2^*, \\ \lim_{k\to\infty} \sigma_{k+1} &= \sqrt{\theta(1-\theta)}\,(\lambda_{\max} - \lambda_{\min}) =: \sigma^*, \end{aligned} \tag{4.38}$$

where $\theta \in (0,1)$.

Proof. By elementary calculations it folows, that

$$\begin{aligned} \lim_{k\to\infty} \rho_{2k-1} &= \lim_{k\to\infty} \boldsymbol{v}_{2k-1}^H A \boldsymbol{v}_{2k-1} = \boldsymbol{u}_1^H A \boldsymbol{u}_1 = (\alpha \boldsymbol{z}_1 + \beta \boldsymbol{z}_n)^H A (\alpha \boldsymbol{z}_1 + \beta \boldsymbol{z}_n) \\ &= \alpha^2 \lambda_{\min} + \beta^2 \lambda_{\max} = \theta\lambda_{\min} + (1-\theta)\lambda_{\max}, \end{aligned}$$

where $\theta := \alpha^2$ and in similar way

$$\begin{aligned} \lim_{k\to\infty} \rho_{2k} &= \lim_{k\to\infty} \boldsymbol{v}_{2k}^H A \boldsymbol{v}_{2k} = \boldsymbol{u}_2^H A \boldsymbol{u}_2 = \operatorname{sign}(\alpha\beta)(-\beta \boldsymbol{z}_1 + \alpha \boldsymbol{z}_n)^H A \operatorname{sign}(\alpha\beta)(-\beta \boldsymbol{z}_1 + \alpha \boldsymbol{z}_n) \\ &= \beta^2 \lambda_{\min} + \alpha^2 \lambda_{\max} = (1-\theta)\lambda_{\min} + \theta\lambda_{\max}. \end{aligned}$$

The result for $\lim_{k\to\infty} \sigma_{k+1}$ follows from (4.37).

□

We can conclude, that asymptotically the restart Krylov subspace method with restart length $m = 1$ is equivalent to an interpolation f in just two points ρ_1^* and ρ_2^* with increasing multiplicity.

Remark 4.3.5 *Based on (4.38), we know, that ρ_1^* and ρ_2^* lie in an open interval $(\lambda_{\min}, \lambda_{\max})$ symmetricaly according to $\frac{1}{2}(\lambda_{\min} + \lambda_{\max})$. If $\frac{1}{2}(\lambda_{\min} + \lambda_{\max})$ is an eigenvalue of the matrix A, then*

$$|\rho_1^* - \rho_2^*| \leq \frac{\sqrt{2}}{2}(\lambda_{\min} - \lambda_{\max}),$$

see [1], pp. 219. There is no more information about the general case. More precise information can be available for a special case. Consider, that $\Lambda(A)$ is symmetric with respect to $\frac{1}{2}(\lambda_{\min} + \lambda_{\max})$. Then

$$\left|\lambda_j - \frac{1}{2}(\lambda_{\min} + \lambda_{\max})\right| = \left|\lambda_{n+1-j} - \frac{1}{2}(\lambda_{\min} + \lambda_{\max})\right|, \quad j = 1, 2, \ldots, \left\lfloor \frac{n}{2} \right\rfloor.$$

Moreover, if the coefficients of $\boldsymbol{v}_1 = \sum_{j=1}^{n} c_{j,1}\boldsymbol{z}_j$ are symmetric as well, then it is easy to see, that $\rho_k = \frac{1}{2}(\lambda_{\min} + \lambda_{\max})$ for every k and thus $\rho_1^* = \rho_2^* = \frac{1}{2}(\lambda_{\min} + \lambda_{\max})$.

4.4 Deflated restarting for matrix functions

Restarting the standart Krylov subspace method causes slowing down of convergence. This section describes a scheme, derived in [16], which is based on further generalization of decompositions of Arnoldi type. This modification of the Krylov subspace method - deflated restarting - can accelerate the convergence after a restart. Consider a more general decomposition than Arnoldi or Arnoldi-like,

$$AW_{m+\ell} = W_{m+\ell}K_{m+\ell} + \boldsymbol{w}\boldsymbol{k}_{m+\ell}^T, \tag{4.39}$$

where $K_{m+\ell} \in \mathbb{C}^{(m+\ell)\times(m+\ell)}, W_{m+\ell} \in \mathbb{C}^{n\times(m+\ell)}$ with $\operatorname{range}(W_{m+\ell}) = \mathscr{K}_m(A, \boldsymbol{b})$, $\boldsymbol{w} \in \mathscr{K}_{m+1}(A, \boldsymbol{b}) \setminus \mathscr{K}_m(A, \boldsymbol{b})$ and $\boldsymbol{k}_{m+\ell} \in \mathbb{C}^{m+\ell}$. The columns of $W_{m+\ell}$ are linearly dependent if and only if $\ell > 0$. We will refer to decomposition (4.39) as a **Krylov-like decomposition** of A with respect to $\mathscr{K}_m(A, \boldsymbol{b})$. Under some additional assumptions, Krylov-like decompositon simplifies into various special cases of decompositions, i.e. (4.39) becomes:

- a *Krylov decomposition* if $\ell = 0$ and thus the columns of W_m are linearly independent, see [51],
- an *Arnoldi-like decomposition* (4.5) if $\ell = 0$ and the columns of W_m form an ascending basis of $\mathscr{K}_m(A, \boldsymbol{b})$, in which case K_m is an unreduced upper Hessenberg matrix, see [15],
- an *Arnoldi decomposition* (4.1) if $\ell = 0$ and the columns of W_m are orthonormal and form an ascending basis of $\mathscr{K}_m(A, \boldsymbol{b})$, in which case K_m is also unreduced upper Hessenberg and which constitutes the most familiar situation, see, e.g. [46].

Let f be a function such that $f(K_{m+\ell})$ is defined. We then define the Krylov-like approximation $f(A)\boldsymbol{b}$ associated with (4.39) as

$$\boldsymbol{f}_{m+\ell} := W_{m+\ell} f(K_{m+\ell})\widehat{\boldsymbol{b}},$$

where $\widehat{\boldsymbol{b}} \in \mathbb{C}^{m+\ell}$ is any vector such that $W_{m+\ell}\widehat{\boldsymbol{b}} = \boldsymbol{b}$.

Next we give several lemmas to show some properties of Krylov-like approximations, see [16], pp. 4-6.

Lemma 4.4.1 *For any polynomial $q(z) = a_m z^m + \ldots + a_0 \in \mathscr{P}_m$,*

$$q(A)\boldsymbol{b} = W_{m+\ell} q(K_{m+\ell})\hat{\boldsymbol{b}} + a_m(\boldsymbol{k}_{m+\ell}^T K_{m+\ell}^{m-1}\hat{\boldsymbol{b}})\boldsymbol{w}, \tag{4.40}$$

using the notation of (4.39). In particular, for $q \in \mathscr{P}_{m-1}$ this simplifies to

$$q(A)\boldsymbol{b} = W_{m+\ell} q(K_{m+\ell})\hat{\boldsymbol{b}}.$$

Proof. Due to linearity of matrix polynomials it is sufficient to verify (4.40) for the monomials $q(z) = z^j$, $j = 0, \ldots, m$. This can be proved using mathematical induction. For $j = 0$, we have

$$A^0\boldsymbol{b} = W_{m+\ell} K_{m+\ell}^0 \hat{\boldsymbol{b}} = W_{m+\ell}\hat{\boldsymbol{b}} = \boldsymbol{b}.$$

For $j = 1, 2, \ldots, m$, it holds that

$$A^j\boldsymbol{b} = A(A^{j-1}\boldsymbol{b}) = A(W_{m+\ell}K_{m+\ell}^{j-1}\hat{\boldsymbol{b}}) \overset{(4.39)}{=} W_{m+\ell}K_{m+\ell}^j\hat{\boldsymbol{b}} + (\boldsymbol{k}_{m+\ell}^T K_{m+\ell}^{j-1}\hat{\boldsymbol{b}})\boldsymbol{w}.$$

The vector $A^j\boldsymbol{b}$ is contained in $\mathscr{K}_{j+1}(A,\boldsymbol{b}) \setminus \mathscr{K}_j(A,\boldsymbol{b})$. Since $\boldsymbol{w} \in \mathscr{K}_{m+1}(A,\boldsymbol{b}) \setminus \mathscr{K}_m(A,\boldsymbol{b})$, the rightmost vector must be $\boldsymbol{k}_{m+\ell}^T K_{m+\ell}^{j-1}\hat{\boldsymbol{b}} = 0$ for $1 \le j \le m-1$. For $j = m$, we obtain the identity (4.40).

□

The vector $\boldsymbol{w}$ lies in $\mathscr{K}_{m+1}(A,\boldsymbol{b}) \setminus \mathscr{K}_m(A,\boldsymbol{b})$ and can be therefore expressed as $\boldsymbol{w} = p_m(A)\boldsymbol{b}$ with a unique polynomial p_m of exact degree m. This gives the following result, see [16], pp. 5.

Lemma 4.4.2 *For the polynomial p_m defined by $\boldsymbol{w} = p_m(A)\boldsymbol{b}$,*

$$W_{m+\ell} p_m(K_{m+\ell})\hat{\boldsymbol{b}} = \boldsymbol{0}. \tag{4.41}$$

More generally, for any polynomial q

$$W_{m+\ell} q(K_{m+\ell}) p_m(K_{m+\ell})\hat{\boldsymbol{b}} = \boldsymbol{0}. \tag{4.42}$$

Proof. Writing $p_m = a_m z^m + \ldots + a_0$ and substituing $\boldsymbol{w} = p_m(A)\boldsymbol{b}$ in (4.40) yields

$$p_m(A)\boldsymbol{b} = W_{m+\ell} p_m(K_{m+\ell})\hat{\boldsymbol{b}} + a_m(\boldsymbol{k}_{m+\ell}^T K_{m+\ell}^{m-1}\hat{\boldsymbol{b}}) p_m(A)\boldsymbol{b},$$

or equivalently,

$$(1 - a_m(\boldsymbol{k}_{m+\ell}^T K_{m+\ell}^{m-1}\hat{\boldsymbol{b}}))p_m(A)\boldsymbol{b} = W_{m+\ell}p_m(K_{m+\ell})\hat{\boldsymbol{b}}. \tag{4.43}$$

The vector $\boldsymbol{w} = p_m(A)\boldsymbol{b}$ is an element of $\mathscr{K}_{m+1}(A, \boldsymbol{b}) \setminus \mathscr{K}_m(A, \boldsymbol{b})$ and the right-hand side of (4.43) is an element of $\mathscr{K}_m(A, \boldsymbol{b})$. Thus, the equality in (4.43) holds only if both sides of (4.43) vanish, i.e., if $W_{m+\ell}p_m(K_{m+\ell})\hat{\boldsymbol{b}} = \boldsymbol{0}$.

Further, we prove (4.42) by mathematical induction. From linearity of matrix polynomials it is sufficient to prove it for monomials $K_{m+\ell}^j, j = 1, \ldots, m$. For $j = 0$ we have (4.41). For $j \geq 1$, we assume that $W_{m+\ell}K_{m+\ell}^{j-1}p_m(K_{m+\ell})\hat{\boldsymbol{b}} = \boldsymbol{0}$. Then, using (4.39),

$$\begin{aligned} W_{m+\ell}K_{m+\ell}^j p_m(K_{m+\ell})\hat{\boldsymbol{b}} &= (W_{m+\ell}K_{m+\ell})K_{m+\ell}^{j-1}p_m(K_{m+\ell})\hat{\boldsymbol{b}} \\ &= (AW_{m+\ell} - \boldsymbol{w}\boldsymbol{k}_{m+\ell}^T)K_{m+\ell}^{j-1}p_m(K_{m+\ell})\hat{\boldsymbol{b}} \\ &= AW_{m+\ell}K_{m+\ell}^{j-1}p_m(K_{m+\ell})\hat{\boldsymbol{b}} - (\boldsymbol{k}_{m+\ell}^T K_{m+\ell}^{j-1}p_m(K_{m+\ell})\hat{\boldsymbol{b}})\boldsymbol{w} \\ &= -(\boldsymbol{k}_{m+\ell}^T K_{m+\ell}^{j-1}p_m(K_{m+\ell})\hat{\boldsymbol{b}})\boldsymbol{w}. \end{aligned}$$

Since $\boldsymbol{w} \in \mathscr{K}_{m+1}(A, \boldsymbol{b}) \setminus \mathscr{K}_m(A, \boldsymbol{b})$ and $W_{m+\ell}K_{m+\ell}^j p_m(K_{m+\ell})\hat{\boldsymbol{b}} \in \mathscr{K}_m(A, \boldsymbol{b})$ we have

$$W_{m+\ell}K_{m+\ell}^j p_m(K_{m+\ell})\hat{\boldsymbol{b}} = \boldsymbol{0}.$$

□

The following lemma describes the property of the zeros of p_m, see [16], pp. 6.

Lemma 4.4.3 *The zeros of p_m, where $\boldsymbol{w} = p_m(A)\boldsymbol{b}$, are contained in the spectrum of K_{m+1}. More precisely, p_m divides the characteristic polynomial of $K_{m+\ell}$.*

Proof. The jth column of the matrix $W_{m+\ell}$ can be expressed as $p^{(j)}(A)\boldsymbol{b}$ for some polynomial $p^{(j)}$ of degree at most $m-1$. From (4.39) we can see, that these polynomials satisfy the recurrence

$$z\left[p^{(1)}(z), \ldots, p^{(m+\ell)}(z)\right] = \left[p^{(1)}(z), \ldots, p^{(m+\ell)}(z)\right] K_{m+\ell} + p_m(z)\boldsymbol{k}_{m+\ell}^T. \tag{4.44}$$

If z_0 is a zero of p_m, then $p_m(z_0)\boldsymbol{k}_{m+\ell}^T$ vanishes and z_0 must be an eigenvalue of $K_{m+\ell}$. If we differentiate (4.44), we obtain

$$\begin{aligned} z\left[\frac{dp^{(1)}(z)}{dz}, \ldots, \frac{dp^{(m+\ell)}(z)}{dz}\right] + \left[p^{(1)}(z), \ldots, p^{(m+\ell)}(z)\right] = \\ = \left[\frac{dp^{(1)}(z)}{dz}, \ldots, \frac{dp^{(m+\ell)}(z)}{dz}\right] K_{m+\ell} + \frac{dp_m(z)}{dz}\boldsymbol{k}_{m+\ell}^T. \end{aligned}$$

If z_0 is a double zero of p_m, than $dp_m(z_0)/dz = 0$ and z_0 is an eigenvalue of $K_{m+\ell}$ associated with the eigenvector $\left[p^{(1)}(z_0), \ldots, p^{(m+\ell)}(z_0)\right]$ and the principal vector $\left[dp^{(1)}(z_0)/dz, \ldots, dp^{(m+\ell)}(z_0)/dz\right]$. Consequently, the eigenvalue z_0 of $K_{m+\ell}$ has algebraic multiplicity at least two. For zeros of higher order the result follows from further differentiation.

□

Now we are ready to prove the main theorem, given in [16], pp. 6-7.

Theorem 4.4.4 *The Krylov-like approximation to $f(A)\boldsymbol{b}$ introduced as*

$$\boldsymbol{f}_{m+\ell} = W_{m+\ell} f(K_{m+\ell})\widehat{\boldsymbol{b}}$$

can be characterised as

$$\boldsymbol{f}_{m+\ell} = q_{m-1}(A)\boldsymbol{b}, \tag{4.45}$$

where q_{m-1} interpolates f in the Hermite sense at the zeros of p_m, i.e. at some, but not at all eigenvalues of $K_{m+\ell}$.

If Γ is a Jordan curve which contains the eigenvalues of A and $K_{m+\ell}$ in its interior, and f is analytic in and on Γ, then

$$\boldsymbol{f}_{m+\ell} = \frac{1}{2\pi i}\int_{\Gamma} f(t)\boldsymbol{x}_{m+\ell}(t)dt, \tag{4.46}$$

where $\boldsymbol{x}_{m+\ell}(t) = W_{m+\ell}(tI_{m+\ell} - K_{m+\ell})^{-1}\widehat{\boldsymbol{b}}$ is the Krylov-like approximation to $(tI_n - A)^{-1}\boldsymbol{b}$ associated with (4.39).

Proof. Let $r \in \mathscr{P}_{m+\ell-1}$ be the Hermite interpolation polynomial of f at the eigenvalues of $K_{m+\ell}$, then $f(K_{m+\ell}) = r(K_{m+\ell})$. Thus the Krylov-like approximation can be expressed as $\boldsymbol{f}_{m+\ell} = W_{m+\ell} r(K_{m+\ell})\hat{\boldsymbol{b}}$, and it suffices to show that

$$W_{m+\ell} r(K_{m+\ell})\hat{\boldsymbol{b}} = q_{m-1}(A)\boldsymbol{b}.$$

Since, by Lemma 4.4.3, the zeros of p_m are contained in the spectrum of $K_{m+\ell}$, the polynomial q_{m-1} also interpolates r at the zeros p_m, and therefore $r - q_{m-1}$ must be divisible by p_m, i.e., $r = sp_m + q_{m-1}$ for some polynomial s. Thus,

$$W_{m+\ell} r(K_{m+\ell})\hat{\boldsymbol{b}} = W_{m+\ell} s(K_{m+\ell}) p_m(K_{m+\ell})\hat{\boldsymbol{b}} + W_{m+\ell} q_{m-1}(K_{m+\ell})\hat{\boldsymbol{b}}.$$

By Lemma 4.4.2, $W_{m+\ell} s(K_{m+\ell}) p_m(K_{m+\ell})\hat{\boldsymbol{b}} = \boldsymbol{0}$, and, by Lemma 4.4.1, $W_{m+\ell} q_{m-1}(K_{m+\ell})\hat{\boldsymbol{b}} = q_{m-1}(A)\boldsymbol{b}$, which gives (4.45).

The characterization (4.46) is an immediate consequence of the representation of a matrix function as a Cauchy integral. Under the given assumptions,

$$f(K_{m+\ell}) = \frac{1}{2\pi i}\int_{\Gamma} f(t)(tI_{m+\ell} - K_{m+\ell})^{-1}dt.$$

□

As a consequence, the approximation $\boldsymbol{f}_{m+\ell} = W_{m+\ell} f(K_{m+\ell})\widehat{\boldsymbol{b}}$ is determined by the zeros of p_m but it is independent of the specific choice of $\widehat{\boldsymbol{b}}$, as long as $W_{m+\ell}\widehat{\boldsymbol{b}} = \boldsymbol{b}$.

The restarted Krylov approximation with deflation

Now we describe the Arnoldi method with deflated restarting presented in [16], pp. 7-11. Let $0 \le \ell \le m$ be given. In the first restart a standard Arnoldi decomposition of A with respect to the subspace $\mathscr{K}_m(A, \boldsymbol{b})$ is computed,

$$AV_m^{(1)} = V_m^{(1)} H_m^{(1)} + \eta_{m+1,m}^{(1)} \boldsymbol{v}_{m+1}^{(1)} \boldsymbol{e}_m^T.$$

Then ℓ eigenvalues of $H_m^{(1)}$ are extracted using the partial Schur decomposition

$$H_m^{(1)} U_{m,\ell}^{(1)} = U_{m,\ell}^{(1)} T_\ell^{(1)},$$

where $T_\ell^{(1)} \in \mathbb{C}^{\ell\times\ell}$ is an upper triangular matrix and the columns of $U_{m,\ell}^{(1)} \in \mathbb{C}^{m\times\ell}$ are orthonormal. Putting $Y_{m,\ell}^{(1)} = V_m^{(1)} U_{m,\ell}^{(1)}$, gives

$$AY_{m,\ell}^{(1)} = Y_{m,\ell}^{(1)} T_\ell^{(1)} + \eta_{m,m+1}^{(1)} \boldsymbol{v}_{m+1}^{(1)} \boldsymbol{u}^{(1)}, \tag{4.47}$$

where we denote the row vector $\boldsymbol{u}^{(1)} = \boldsymbol{e}_m^T U_{m,\ell}^{(1)}$. We extend the factorization (4.47) by m Arnodi steps and obtain

$$A\left[Y_{m,\ell}^{(1)} V_m^{(2)}\right] = \left[Y_{m,\ell}^{(1)} V_m^{(2)}\right] \begin{bmatrix} T_\ell^{(1)} & S_{m,\ell}^{(1)} \\ \eta_{m+1,m}^{(1)} \boldsymbol{e}_1 \boldsymbol{u}^{(1)} & H_m^{(2)} \end{bmatrix} + \eta_{m+1,m}^{(2)} \boldsymbol{v}_{m+1}^{(2)} e_{\ell+m}^T,$$

where

$$G^{(2)} := \begin{bmatrix} T_\ell^{(1)} & S_{m,\ell}^{(1)} \\ \eta_{m+1,m}^{(1)} \boldsymbol{e}_1 \boldsymbol{u}^{(1)} & H_m^{(2)} \end{bmatrix} \in \mathbb{C}^{(m+\ell)\times(m+\ell)}.$$

The matrix $\left[Y_{m,\ell}^{(1)} V_m^{(2)} \boldsymbol{v}_{m+1}^{(2)}\right] \in \mathbb{C}^{n\times(m+\ell-1)}$ has orthonormal columns and $S_{m,\ell}^{(1)} = \left[Y_{m,\ell}^{(1)}\right]^H A V_m^{(2)} \in \mathbb{C}^{\ell\times m}$ is in general dense matrix. Repeating the process, for $j = 2, \ldots, k$ we obtain in the jth cycle

$$A\left[Y_{m,\ell}^{(j-1)} V_m^{(j)}\right] = \left[Y_{m,\ell}^{(j-1)} V_m^{(j)}\right] G^{(j)} + \eta_{m+1,m}^{(j)} \boldsymbol{v}_{m+1}^{(j)} e_{\ell+m}^T,$$

where

$$G^{(j)} = \begin{bmatrix} T_\ell^{(j-1)} & S_{m,\ell}^{(j-1)} \\ \eta_{m+1,m}^{(j-1)} \boldsymbol{e}_1 \boldsymbol{u}^{(j-1)} & H_m^{(j)} \end{bmatrix} \in \mathbb{C}^{(\ell+m)\times(\ell+m)}.$$

and the matrix $H_m^{(j)} \in \mathbb{C}^{m\times m}$ is upper Hessenberg, $T_\ell^{(j-1)} \in \mathbb{C}^{\ell\times\ell}$ is upper triangular, $Y_{m,\ell}^{(j-1)} = \left[Y_{m,\ell}^{(j-2)} V_m^{(j-1)}\right] U_{m,\ell}^{(j-1)} \in \mathbb{C}^{n\times\ell}$, $S_{m,\ell}^{(j-1)} = \left[Y_{m,\ell}^{(j-1)}\right]^H A V_m^{(j)}$ and $\boldsymbol{u}^{(j-1)} = \boldsymbol{e}_{m+\ell}^T U_{m,\ell}^{(j-1)}$. Then ℓ eigenvalues of $G^{(j-1)}$ are computed using partial Schur decomposition

$$G^{(j-1)} U_{m,\ell}^{(j-1)} = U_{m,\ell}^{(j-1)} T_\ell^{(j-1)},$$

and set $\boldsymbol{u}^{(j-1)} = \boldsymbol{e}_{\ell+m}^T U_{m,\ell}^{(j-1)}$.

Putting everything together, we obtain a Krylov-like decomposition

$$AW^{(k)} = W^{(k)} K^{(k)} + \eta_{m+1,m}^{(k)} \boldsymbol{v}_{m+1}^{(k)} \boldsymbol{e}_{km+(k-1)\ell}^T, \tag{4.48}$$

where $W^{(k)} = \left[V_m^{(1)} Y_{m,\ell}^{(1)} \ldots Y_{m,\ell}^{(k-1)} V_m^{(k)}\right] \in \mathbb{C}^{n\times(km+(k-1)\ell)}$,

$$K^{(k)} = \begin{bmatrix} G^{(1)} & & & \\ F^{(1)} & G^{(2)} & & \\ & \ddots & \ddots & \\ & & F^{(k-1)} & G^{(k)} \end{bmatrix} \in \mathbb{C}^{(km+(k-1)\ell)\times(km+(k-1)\ell)},$$

with $G^{(1)} = H_m^{(1)} \in \mathbb{C}^{m\times m}$,

$$F^{(1)} = \eta_{m+1,m}^{(1)} \boldsymbol{e}_{\ell+1}\boldsymbol{e}_m^T \in \mathbb{R}^{(\ell+m)\times m},$$

$$F^{(j)} = \eta_{m+1,m}^{(j)} \boldsymbol{e}_{\ell+1}\boldsymbol{e}_{\ell+m}^T \in \mathbb{R}^{(\ell+m)\times(\ell+m)}, \quad j = 2, 3, \ldots, k-1.$$

The Krylov-like approximation associated with (4.48) is then

$$\boldsymbol{f}^{(k)} := \beta W^{(k)} f(K^{(k)})\boldsymbol{e}_1, \tag{4.49}$$

where $\beta = \|\boldsymbol{b}\|$. Taking into account the block-triangular structure of $K^{(k)}$, we obtained an update scheme for $\boldsymbol{f}^{(k)}$,

$$\boldsymbol{f}^{(k)} = \boldsymbol{f}^{(k-1)} + \beta \left[Y_{m,\ell}^{(k-1)} V_m^{(k)}\right] \left[f(K^{(k)})\boldsymbol{e}_1\right]_{(k-1)m+(k-2)\ell+1:km+(k-1)\ell}.$$

By Theorem 4.4.4 the approximation $\boldsymbol{f}^{(k)}$ in (4.49) can be represented as $q_{km-1}(A)\boldsymbol{b}$, where q_{km-1}, a polynomial of degree $km-1$, interpolates f in the Hermite sense at km of the $km + (k-1)\ell$ eigenvalues of $K^{(k)}$.These interpolation nodes can be characterized as described in the following theorem, for the proof, see [16], pp. 9-10.

Theorem 4.4.5 *The approximation to $f(A)\boldsymbol{b}$ (4.49) can be characterised as*

$$\boldsymbol{f}^{(k)} = q_{km-1}(A)\boldsymbol{b},$$

where q_{km-1} interpolates f in the Hermite sense at the zeros of p_{km}. Let $\theta_1^{(j)}, \theta_2^{(j)}, \ldots \theta_\ell^{(j)}$ be the eigenvalues of $T_\ell^{(j)}$, including their multiplicities. The zeros of q_{km-1} are given by

$$\bigcup_{j=1}^{k-1} \left(\Lambda\left(G^{(j)}\right) \setminus \left\{\theta_1^{(j)}, \ldots, \theta_\ell^{(j)}\right\}\right) \cup \Lambda\left(G^{(k)}\right).$$

If Γ is a Jordan curve which contains the eigenvalues of A and $K^{(k)}$ in its interior, such that f is analytic in and on Γ, then

$$\boldsymbol{f}^{(k)} = \frac{1}{2\pi i}\int_\Gamma f(t)\boldsymbol{x}^{(k)}(t)dt,$$

where $\boldsymbol{x}^{(k)}(t) = \|\boldsymbol{b}\|\,(tI - K^{(k)})^{-1}\boldsymbol{e}_1$ is the approximation to the solution of $(tI-A)\boldsymbol{x}(t) = \boldsymbol{b}$ after k cycles of the restarted FOM method with restart length m and ℓ deflated eigenvalues beginning with $\boldsymbol{x}_0(t) = 0$.

Reorthogonalization

In finite-precision arithmetic the orthogonality of the approximate eigenvectors $Y_{m,\ell}^{(k-1)}$ can be lost. This can be overcame if $Y_{m,\ell}^{(k-1)}$ is reorthogonalised before the Arnoldi decomposition is extended by the vectors $V_m^{(k)}$, see [16] pp. 10. Suppose, we have computed the decomposition in the kth cycle

$$A\widetilde{Y}_{m,\ell}^{(k-1)} = \widetilde{Y}_{m,\ell}^{(k-1)} T_\ell^{(k-1)} + \eta_{m+1,m}^{(k-1)} \boldsymbol{v}_{m+1}^{(k-1)} \boldsymbol{u}^{(k-1)},$$

where the columns of $\widetilde{Y}_{m,\ell}^{(k-1)}$ have lost orthogonality due to the rounding errors, but are still linearly independent. We compute a QL decomposition

$$\left[\widetilde{Y}_{m,\ell}^{(k-1)}\boldsymbol{v}_{m+1}^{(k-1)}\right] = QL := \left[Y_{m,\ell}^{(k-1)}\boldsymbol{v}_{m+1}^{(k-1)}\right]\begin{bmatrix}\widehat{L} & O\\ * & 1\end{bmatrix},$$

where the matrix $Q \in \mathbb{C}^{n\times(\ell+1)}$ has orthonormal columns. Then $Y_{m,\ell}^{(k-1)}$ has orthonormal columns and $\widehat{L} \in \mathbb{C}^{\ell\times\ell}$ is nonsingular lower triangular matrix. By elementary computations we obtain

$$\begin{aligned} AY_{m,\ell}^{(k-1)} &= \left[Y_{m,\ell}^{(k-1)}\boldsymbol{v}_{m+1}^{(k-1)}\right]\begin{bmatrix}\widehat{L} & O\\ * & 1\end{bmatrix}\begin{bmatrix}T_{\ell}^{(k-1)}\\ \eta_{m+1,m}^{(k-1)}\boldsymbol{u}^{(k-1)}\end{bmatrix}\widehat{L}^{-1}\\ &=: Y_{m,\ell}^{(k-1)}T_{\ell,new}^{(k-1)} + \eta_{m+1,m}^{(k-1)}\boldsymbol{v}_{m+1}^{(k-1)}\boldsymbol{u}_{new}^{(k-1)}. \end{aligned}$$

In the restarted Krylov subspace method with deflation, only $Y_{m,\ell}^{(k-1)}$ and $V_m^{(k)}$ need to be stored to update $\boldsymbol{f}^{(k-1)}$. However, the method requires evaluation of $f(K^{(k)})$, i.e. a function of a matrix, which dimension grows with number of restarts.

The previous derivations can be summarized to the following algorithm, which computes the Krylov-like approximation of $f(A)\boldsymbol{b}$.

Algorithm 7

Given $A, \boldsymbol{b}, f$

1. $\beta = \|\boldsymbol{b}\|, \boldsymbol{v}_1 = \boldsymbol{b}/\beta$, $Y^{(0)} = []$
2. Compute the decomposition $AV_m^{(1)} = V_m^{(1)}H_m^{(1)} + \eta_{m+1,m}^{(1)}\boldsymbol{v}_{m+1}^{(1)}\boldsymbol{e}_m^T$.
3. $F^{(1)} = \eta_{m+1,m}^{(1)}\boldsymbol{e}_{\ell+1}\boldsymbol{e}_m^T \in \mathbb{R}^{(\ell+m)\times m}$, $f^{(1)} = \beta V_m^{(1)} f(H_m^{(1)})\boldsymbol{e}_1$.
4. **for** $i = 2, 3, \ldots, k$ **do**
5. Compute partial Schur decomposition $H_m^{(i-1)}U_{m,\ell}^{(i-1)} = U_{m,\ell}^{(i-1)}T_{\ell}^{(i-1)}$.
6. Set $Y_{m,\ell}^{(i-1)} = \left[Y_{m,\ell}^{(i-2)}V_m^{(i-1)}\right]U_{m,\ell}^{(i-1)}$ and reorthogonalize.
7. Compute $A\left[Y_{m,\ell}^{(i-1)}V_m^{(i)}\right] = \left[Y_{m,\ell}^{(i-1)}V_m^{(i)}\right]G^{(i)} + \eta_{m+1,m}^{(i)}\boldsymbol{v}_{m+1}^{(i)}\boldsymbol{e}_{\ell+m}^T$ by m further Arnoldi steps.
8. Set $K^{(i)} = \begin{bmatrix} K^{(i-1)} & O\\ O\cdots OF^{(i-1)} & G^{(i)}\end{bmatrix}$.
9. Set $F^{(i)} = \eta_{m+1,m}^{(i)}\boldsymbol{e}_{\ell+1}\boldsymbol{e}_{\ell+m}^T \in \mathbb{R}^{(\ell+m)\times(\ell+m)}$.
10. Set $\boldsymbol{f}^{(i)} = \boldsymbol{f}^{(i-1)} + \beta\left[Y_{m,\ell}^{(i-1)}V_m^{(i)}\right]\left[f(K^{(i)})\boldsymbol{e}_1\right]_{(i-1)m+(i-2)\ell+1:im+(i-1)\ell}$.
11. **enddo**

Note that it is possible to modify the algorithm using a rational approximation of a function f, in a similar way as it was described earlier for the standart and the restarted Krylov subspace method.

4.5 Extended Krylov subspace method

The last method from the family of the Krylov subspace methods we give in our summary is the **extended Krylov subspace method**, first proposed in [13]. The extended Krylov subspace methods contains information not only about A, but also about A^{-1}, i.e.

$$\widetilde{\mathscr{K}_{2m}}(A, \boldsymbol{b}) = \text{span}\left\{\boldsymbol{b}, A^{-1}\boldsymbol{b}, A\boldsymbol{b}, A^{-2}\boldsymbol{b}, A^2\boldsymbol{b}, \ldots, A^{m-1}\boldsymbol{b}, A^{-m}\boldsymbol{b}\right\},$$

see the work of V. Simoncini and L. Knizherman, [31]. In [49] the following implementation of the extended Krylov subspace method was proposed. We start with the pair $\{\boldsymbol{b}, A^{-1}\boldsymbol{b}\}$ and generate an orthonormal basis of the extended subspace with a block Arnoldi-type method, by adding two vectors at the same time, one multiplied by A and one multiplied by A^{-1}. The described process generates an Arnoldi-like reccurence

$$A\widetilde{V}_{2m} = \widetilde{V}_{2m}\widetilde{H}_{2m} + \mathbf{v}_{2m+2}\tilde{\eta}_{2m+2,2m}E_{2m}^T,$$

where $\widetilde{V}_{2m} = [\mathbf{v}_2, \mathbf{v}_4, \ldots, \mathbf{v}_{2m}] \in \mathbb{C}^{2m\times 2m}$ has orhonormal columns, $\mathbf{v}_{2j} = \left[\boldsymbol{v}_j^{(1)}, \boldsymbol{v}_j^{(2)}\right]$, $j = 1, \ldots, m+1$, $\widetilde{H}_{2m} = \widetilde{V}_{2m}^H A\widetilde{V}_{2m} \in \mathbb{C}^{2m\times 2m}$, $\tilde{\eta}_{2m+2,2m} = \mathbf{v}_{2m+2}^H A\widetilde{V}_{2m}$ and E_{2m}^T is a matrix, that contains the last two columns of the identity matrix. The Arnoldi-like approximation is then computed as

$$\widetilde{\boldsymbol{f}}_{2m} = \beta\widetilde{V}_{2m}f(\widetilde{H}_{2m})\boldsymbol{e}_1, \tag{4.50}$$

for more details see [31], pp. 3. An algorithm for the extended Krylov subspace method has the form

---

Algorithm 8

Given $\boldsymbol{b}, A, f$

1. $\beta = \|\boldsymbol{b}\|$, set $\boldsymbol{v}_1^{(1)} = \boldsymbol{b}$, $\boldsymbol{v}_1^{(2)} = A\boldsymbol{v}_1^{(1)}$
2. Set $\mathbf{v}_2 = GS([\boldsymbol{v}_1^{(1)}\boldsymbol{v}_1^{(2)}])$, $\widetilde{V}_0 = []$.
3. **for** $i = 1, 2, \ldots, m$ **do**
4. $\quad\widetilde{V}_{2i} = \left[\widetilde{V}_{2i-2}, \mathbf{v}_{2i}\right]$
5. $\quad$Set $\widetilde{H}_{2i} = \widetilde{V}_{2i}^H A\widetilde{V}_{2i}$.
6. $\quad$Compute $\widetilde{\boldsymbol{x}}_{2i} = f(\widetilde{H}_{2i})\boldsymbol{e}_1$.
7. $\quad\mathbf{v}'_{2i+2} = [A\mathbf{v}_{2i}\boldsymbol{e}_1, A^{-1}\mathbf{v}_{2i}\boldsymbol{e}_1]$
8. $\quad$Orthogonalize $\mathbf{v}'_{2i+2}$ with respect to $\widetilde{V}_{2i}$, obtain $\widehat{\mathbf{v}}_{2i+2}$.
9. $\quad\mathbf{v}_{2i+2} = GS(\widehat{\mathbf{v}}_{2i+2})$
10. **enddo**
11. Compute approximation $\widetilde{\boldsymbol{f}} = \beta\widetilde{V}_{2m}f(\widetilde{H}_{2m})\boldsymbol{e}_1$.

Here GS denotes the Gram-Schmidt procedure to orthogonalize the columns of the given matrix. The following proposition was given in [31], pp. 4.

Proposition 4.5.1 *Let* $\mathbf{v}_{2j} = \left[\boldsymbol{v}_j^{(1)}, \boldsymbol{v}_j^{(2)}\right]$, $j = 1, \ldots, m$. *With the previous notation, if* $\dim(\widetilde{\mathcal{K}}_{2m}(A, \boldsymbol{b})) = 2m$, *then for* $1 \le j \le m$

$$\boldsymbol{v}_j^{(1)} = p_{j-1}(A)\boldsymbol{b} + q_{j-1}(A^{-1})\boldsymbol{b}, \quad \boldsymbol{v}_j^{(2)} = r_{j-1}(A^{-1})A^{-1}\boldsymbol{b} + s_{j-1}(A)\boldsymbol{b},$$

where $\deg p_{j-1} = \deg r_{j-1} = j-1$ *and* $\deg q_{j-1} \le j-1,\ \deg s_{j-1} \le j-1$.

Proof. We will use mathematical induction to proof this proposition. Without loss of generality suppose that $\|\boldsymbol{b}\| = 1$. For $m = 1$, we have

$$\boldsymbol{v}_1^{(1)} = \boldsymbol{b} = p_0(A)\boldsymbol{b}$$

and

$$\boldsymbol{v}_1^{(2)} = c_1 A^{-1}\boldsymbol{b} + c_2 A^0 \boldsymbol{b} = r_0(A^{-1})A^{-1}\boldsymbol{b} + s_0(A)\boldsymbol{b},$$

where $r_0, s_0 \in \mathscr{P}_0$ and $c_1, c_2 \neq 0$.

We proceed with $m+1$. We have

$$A\boldsymbol{v}_m^{(1)} = Ap_{m-1}(A)\boldsymbol{b} + Aq_{m-1}(A^{-1})\boldsymbol{b} = cA^m\boldsymbol{b} + \sum_{i=-m+2}^{m-1} c_i A^i \boldsymbol{b},$$

with $c \neq 0$. Orthogonalization with respect to the previous vectors $\boldsymbol{v}_j^{(1)}, \boldsymbol{v}_j^{(2)}, j = 1, \ldots, m$ gives

$$cA^m\boldsymbol{b} + \sum_{i=-m+2}^{m-1} c_i A^i \boldsymbol{b} + \sum_{i=1}^{m} \left(c_i^{(1)} \boldsymbol{v}_i^{(1)} + c_i^{(2)} \boldsymbol{v}_i^{(2)} \right) = cA^m\boldsymbol{b} + \sum_{i=-m}^{m-1} \tilde{c}_i A^i \boldsymbol{b} \neq 0,$$

due to linear independence. Thus $\boldsymbol{v}_{m+1}^{(1)} = p_m(A)\boldsymbol{b} + q_m(A^{-1})\boldsymbol{b}$, with $\deg p_m = m$. Analogously,

$$A^{-1}\boldsymbol{v}_m^{(2)} = A^{-1}r_{m-1}(A^{-1})A^{-1}\boldsymbol{b} + A^{-1}s_{m-1}(A)\boldsymbol{b} = cA^{-m-1}\boldsymbol{b} + \sum_{i=-m}^{m-2} c_i A^i \boldsymbol{b}, c \neq 0$$

Orthogonalization with respect to $\boldsymbol{v}_j^{(1)}, \boldsymbol{v}_j^{(2)}, j = 1, \ldots, m$ and to $\boldsymbol{v}_{m+1}^{(1)}$ gives

$$cA^{-m-1}\boldsymbol{b} + \sum_{i=-m}^{m-1} c_i A^i \boldsymbol{b} + \sum_{i=1}^{m} \left(c_i^{(1)} p_m(A)\boldsymbol{b} + c_i^{(2)} q_m(A^{-1})\boldsymbol{b} \right) = cA^{-m-1}\boldsymbol{b} + \sum_{i=-m}^{m} \tilde{c}_i A^i \boldsymbol{b} \neq 0,$$

again, due to linear independence. Therefore, $\boldsymbol{v}_{m+1}^{(2)} = r_m(A^{-1})A^{-1}\boldsymbol{b} + s_m(A)\boldsymbol{b}$, with $\deg r_m = m$.

□

This proposition shows, that if the algorithm stops with zero basis vector $\boldsymbol{v}_m^{(1)}$ or $\boldsymbol{v}_m^{(2)}$, then an invariant subspace of A associated with $\boldsymbol{b}$ is found, and it contains the exact solution of the given problem. The following lemma says that polynomials in A and A^{-1} are exactly represented in the extended Krylov subspaces, see [13].

Lemma 4.5.2 *Matrix polynomials in A, resp.* A^{-1} *of degree* $k \le m-1$*, respectively of degree* $k \le m$ *are exactly represented in the extended Krylov subspace* $\widetilde{\mathscr{K}_{2m}}$*. In particular,*

$$p_k(A)\boldsymbol{b} = \beta \widetilde{V}_{2m} p_k(\widetilde{H}_{2m})\boldsymbol{e}_1 \in \widetilde{\mathscr{K}_{2m}}(A, \boldsymbol{b}),$$

$k \le m-1$*, and*

$$p_k(A^{-1})\boldsymbol{b} = \beta \widetilde{V}_{2m} p_k(\widetilde{H}_{2m}^{-1})\boldsymbol{e}_1 \in \widetilde{\mathscr{K}_{2m}}(A, \boldsymbol{b}),$$

$k \le m$*.*

Proof of this result for symmetric case can be found in [13], generalization to the non-symmetric case is straightforward.

Convergence theory

We conclude this section by giving an error estimation for the approximation (4.50) computed using the extended Krylov subspace method, see [31], pp. 8-9.

We define $W_1 := W(A)$ and $W_2 = (W(A))^{-1} := \{z^{-1} | z \in W_1\}$, and we asssume, that both W_j, $j = 1, 2$, are symmetric with respect to the real axis $\mathbb{R}$ and strictly lie in the right half-plane. Let D denote the closed unit circle, and let $\psi_j : \bar{\mathbb{C}} \setminus D \to \bar{\mathbb{C}} \setminus W_j$, $\phi_j := \psi_j^{-1}$ be the direct and inverse Riemann mappings for W_j, $j = 1, 2$. Moreover, let $F_{j,k}$, $k \in \mathbb{N}$ denote the corresponding Faber polynomials of degree k, whose definition can be found, e.g. in [40], pp. 3. The kth (ordinary) **Faber polynomial** is defined as the polynomial part of the Laurent expansion at ∞ of $[\phi_j(z)]^k$,

$$[\phi_j(z)]^k = z^k + \sum_{i=-\infty}^{k-1} \beta_{k,i,j} z^i, \quad k \geq 0,$$

i.e.

$$F_{j,k} := z^k + \sum_{i=0}^{k-1} \beta_{k,i,j} z^i, \quad k \geq 0,$$

$j = 1, 2$. Consider the class of functions that can be written as

$$f(z) = \int_{-\infty}^{0} \frac{d\mu(\varsigma)}{z - \varsigma}, \quad z \in \mathbb{C} \setminus (-\infty, 0], \tag{4.51}$$

where μ is a measure such that the integral converges absolutely. For $a > 0$, we split this integral into

$$f(z) = f_1(z) + f_2(z), \quad f_1(z) = \int_{-\infty}^{-a} \frac{d\mu(\varsigma)}{z - \varsigma}, \quad f_2(z) = \int_{-a}^{0} \frac{d\mu(\varsigma)}{z - \varsigma}. \tag{4.52}$$

First, we introduce an auxiliary lemma.

Lemma 4.5.3 *Let f be defined by (4.51) and satisfy (4.52) for some $a > 0$. With the notation above, for any $m \in \mathbb{N}$, $m > 1$,*

$$\left| f_1(z) - \sum_{k=0}^{m-1} \gamma_{1,k} F_{1,k}(z) \right| \leq c_1 \left| \phi_1(-a) \right|^{-m}, \quad z \in W_1,$$

$$\left| f_2(z) - \sum_{k=0}^{m} \gamma_{2,k} F_{2,k}(z) \right| \leq c_2 \left| \phi_2(-a^{-1}) \right|^{-m}, \quad z \in W_1,$$

where c_1, c_2 are positive real constants independent of m, n, and $\gamma_{1,k}, \gamma_{2,k}$ are some real numbers.

For a proof, see [31], pp. 7-8. The following theorem gives the required error estimation, see [13], pp. 9.

Theorem 4.5.4 *Let $A \in \mathbb{R}^{n\times n}$ be nonsingular with $W_1 \subset \mathbb{C}^+$ and let $f(z)$ be (4.51). There exists $a > 0$ such that*

$$\left\| f(A)\boldsymbol{b} - \beta\widetilde{V}_{2m} f(\widetilde{H}_{2m})\boldsymbol{e}_1 \right\| \leq \frac{c}{|\phi_1(-a)|^m}, \tag{4.53}$$

where c is a positive constant depending on W_1 and on the measure μ but independent of n and m.

Proof. Define the functions

$$g(z) = f_1(z) - \sum_{k=0}^{m-1} \gamma_{1,k} F_{1,k}(z), \qquad h(z) = f_2(z) - \sum_{k=0}^{m} \gamma_{2,k} F_{2,k}(z^{-1}). \tag{4.54}$$

Using the splitting (4.52) and Lemma 4.5.3, we have

$$\begin{aligned}
\left\| f(A)\boldsymbol{b} - \beta\widetilde{V}_{2m} f(\widetilde{H}_{2m})\boldsymbol{e}_1 \right\| &= \Big\| f_1(A)\boldsymbol{b} - \sum_{k=0}^{m-1} \gamma_{1,k} F_{1,k}(A)\boldsymbol{b} - \beta\widetilde{V}_{2m} f_1(\widetilde{H}_{2m})\boldsymbol{e}_1 \\
&\quad + \beta\widetilde{V}_{2m} \sum_{k=0}^{m-1} \gamma_{1,k} F_{1,k}(\widetilde{H}_{2m})\boldsymbol{e}_1 + f_2(A)\boldsymbol{b} - \sum_{k=0}^{m} \gamma_{2,k} F_{2,k}(A^{-1})\boldsymbol{b} \\
&\quad - \beta\widetilde{V}_{2m} f_2(\widetilde{H}_{2m})\boldsymbol{e}_1 + \beta\widetilde{V}_{2m} \sum_{k=0}^{m} \gamma_{2,k} F_{2,k}(\widetilde{H}_{2m}^{-1})\boldsymbol{e}_1 \Big\| \\
&= \left\| g(A)\boldsymbol{b} - \beta\widetilde{V}_{2m} g(\widetilde{H}_{2m})\boldsymbol{e}_1 + h(A)\boldsymbol{b} - \beta\widetilde{V}_{2m} h(\widetilde{H}_{2m})\boldsymbol{e}_1 \right\| \\
&\leq \max\{1, \beta\} \left[\|g(A)\| + ||g(\widetilde{H}_{2m})|| + \|h(A)\| - ||h(\widetilde{H}_{2m})|| \right].
\end{aligned} \tag{4.55}$$

Since both functions in (4.54) are analytic in W_1 and since $W(\widetilde{H}_{2m}) \subseteq W_1$, we deduce from the result in [8], pp. 668-690, that

$$\max\{\|g(A)\|, ||g(\widetilde{H}_{2m})||\} \leq 11.08 \max_{z\in W_1} |g(z)|,$$

$$\max\{\|h(A)\|, ||h(\widetilde{H}_{2m})||\} \leq 11.08 \max_{z\in W_1} |h(z)|.$$

These inequalities combined with (4.55) and Lemma 4.5.3, give (4.53). Here a is chosen such that

$$|\phi_1(-a)| = \left|\phi_2(-a^{-1})\right|.$$

□

Chapter 5

Numerical experiments

'No amount of experimentation can ever prove me right;
a single experiment can prove me wrong.'
Albert Einstein

The goal of the presented experiments is to compare methods for approximation of $f(A)\boldsymbol{b}$, namely the standart Krylov subspace method, the restarted Krylov subspace method without and with deflation, the extended Krylov subspace method and the polynomial least squares approximation. (Note that some comparision of methods for approximation of $f(A)$ and the standart Krylov subspace method was given in my Bachelor thesis [52]). We focused on the relative error of the approximation $\boldsymbol{f}$ to $f(A)\boldsymbol{b}$, i.e. $\|f(A)\boldsymbol{b} - \boldsymbol{f}\| / \|f(A)\boldsymbol{b}\|$, and the computational time. We use the following Matlab software modified for our purpose:

- the standart and the restarted Krylov subspace method, without and with deflation by Dipl.-Math. Stefan Güttel, Dr. rer. nat.,
  - available online at `http://www.matrixfunctions.com`;
- the extended Krylov subspace method by Prof. Valeria Simoncini;
- the polynomial least squares approximation by M.S. Jie Chen, Ph.D.,
  - available online at `http://www.mcs.anl.gov/~jiechen/software.html`.

The test matrices contain:

- some simple examples constructed by ourselves in order to create a matrix with specific eigenvalues;
- `Lap2D`, a test matrix contained in the software from M.S. Jie Chen Ph.D.;
- `Trefethen2000`, a test matrix from the University of Florida Matrix Sparse Collection (`http://www.cise.ufl.edu/research/sparse/matrices/`);
- matrices obtained by discretization of Heat and Maxwell's equations, donored by my University schoolmate Bc. Lukáš Korous.

For the legend at the pictures, following shortcuts and notation are used:

- KSM - Krylov subspace method;
- LS - least squares;
- ell - deflate number ℓ;
- m - restart length m for the restarted Krylov subspace methods, m=∞ denotes the standart Krylov subspace method.

Further note, that when we speak about the restarted Krylov subspace method, we mean the variant without deflation, unless it is not stated otherwise.

Experiment 1

In the first experiment, we compare the accuracy of the approximation $\boldsymbol{f}$ to $f(A)\boldsymbol{b}$ and the time of computation using the restarted Krylov subspace method for different restart length m, $m = 1, 2, 5, 10, 20, 50$, the standart Krylov subspace method, the extended Krylov subspace method and the polynomial least squares approximation. We compute an approximation of $\exp(A)\boldsymbol{b}$, where A is a diagonal matrix,

$$A = \mathrm{diag}(1, 2, \ldots, 100) \in \mathbb{R}^{100\times 100} \tag{5.1}$$

and $\boldsymbol{b} = (1, 1, \ldots, 1)/\sqrt{100} \in \mathbb{R}^{100}$, with relative error tolerance set for all methods to 10^{-14}.

From the Figure 5.1 (top) we can see, that the smaller m we use in the restarted Krylov subspace method, the more iterations at all we need to obtain the approximate solution with the same relative error. Figure 5.1 (bottom left and bottom right) shows, that the extended Krylov subspace method and the polynomial least squares approximation do not reach the required tolerance.

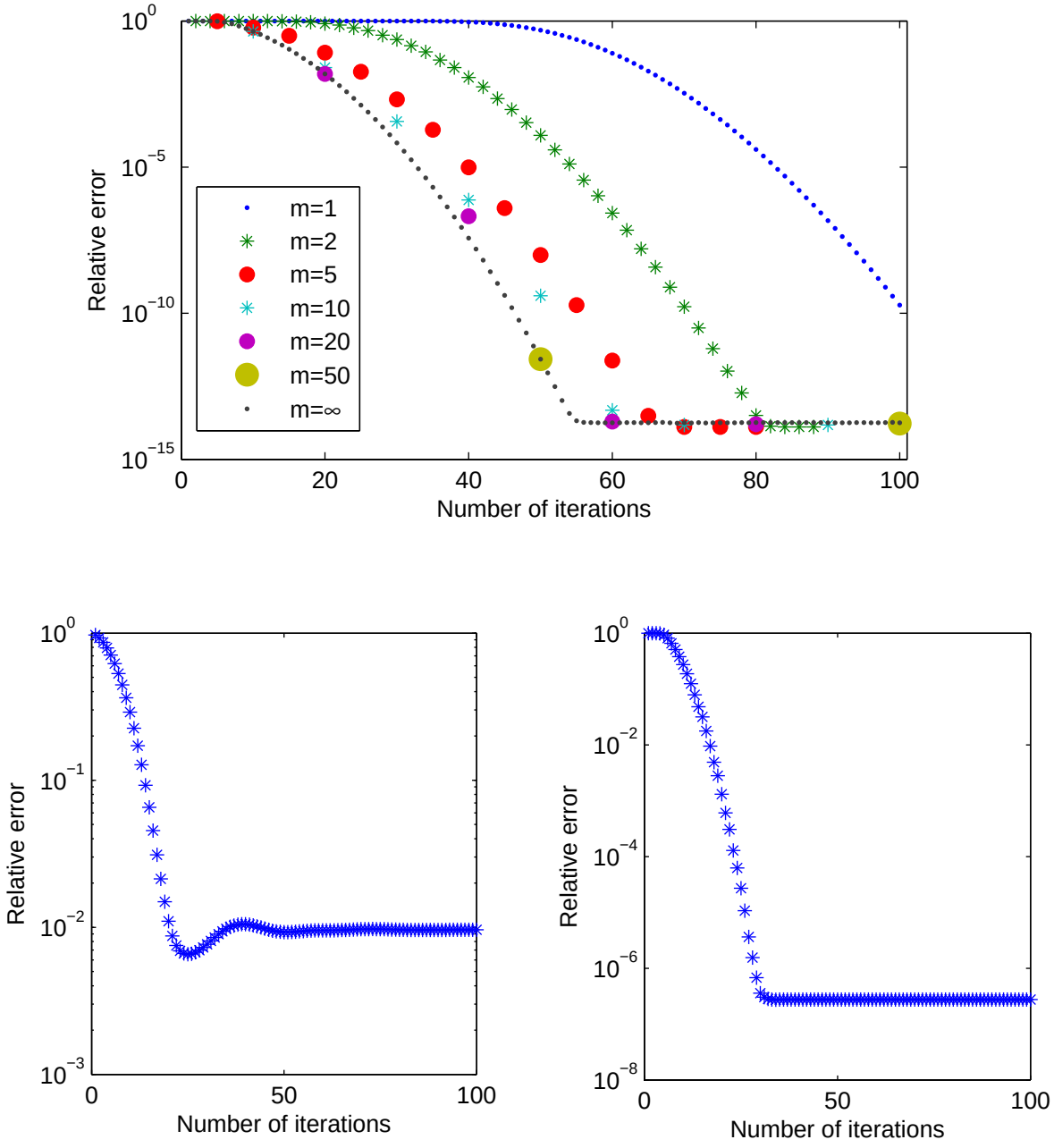


Figure 5.1: Convergence behaviour. Top - the restarted and the standart Krylov subspace method. Bottom left - the extended Krylov subspace method. Bottom right - the polynomial least squares approximation.

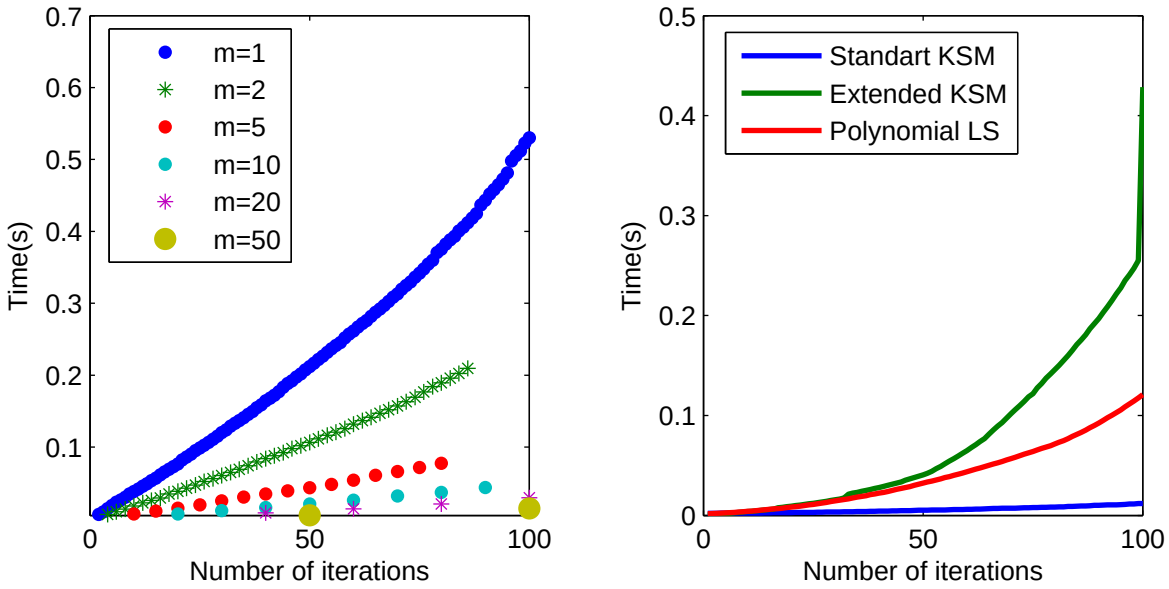


Figure 5.2: Computational time. Left - the restarted Krylov subspace method for different restart lengths m. Right - the standart Krylov subspace method, the extended Krylov subspace method and the polynomial least squares approximation.

Figure 5.2 (left) illustrates, how the restart length impacts the computational time. The time of computation is decreasing with increasing restart length, even thought the method uses long reccurences. The reason is in computation $\boldsymbol{f}$. In case of the standart Krylov subspace method, we compute an approximation of the matrix function only once. In case of the restarted Krylov subspace method, we compute an update of the approximation of the matrix function after each restart. If A is small, the time savings due to the restart do not compensate for computational time required to evaluate the update of $\boldsymbol{f}$. We will discuss this behaviour later, in Experiment 6.

Further we can observe, that the computational time of the extended Krylov subspace method and the polynomial least squares approximation is between the computational time of the standart Krylov subspace method and the restarted Krylov subspace method with restart length $m = 1$ (steepest descent method), see Figure 5.2 (right).

Experiment 2

In the second example, we illustrate, how using deflation in the restarted Krylov subspace method with restart length $m = 10$ impacts the convergence. We discus the results for two matrices. The first one is a diagonal matrix with diagonal elements $A(i, i) = 15 * i$, $i = 1, \ldots, 100$, and the vector $\boldsymbol{b}$ is the same as in Experiment 1. We computed approximation of $\sqrt{A}\boldsymbol{b}$. The matrix A has distinct and well separated eigenvaluesand thus we can see the effect of deflation. Then we show results for the data and the function from Experiment 1, where the eigenvalues of A are closer to each other.

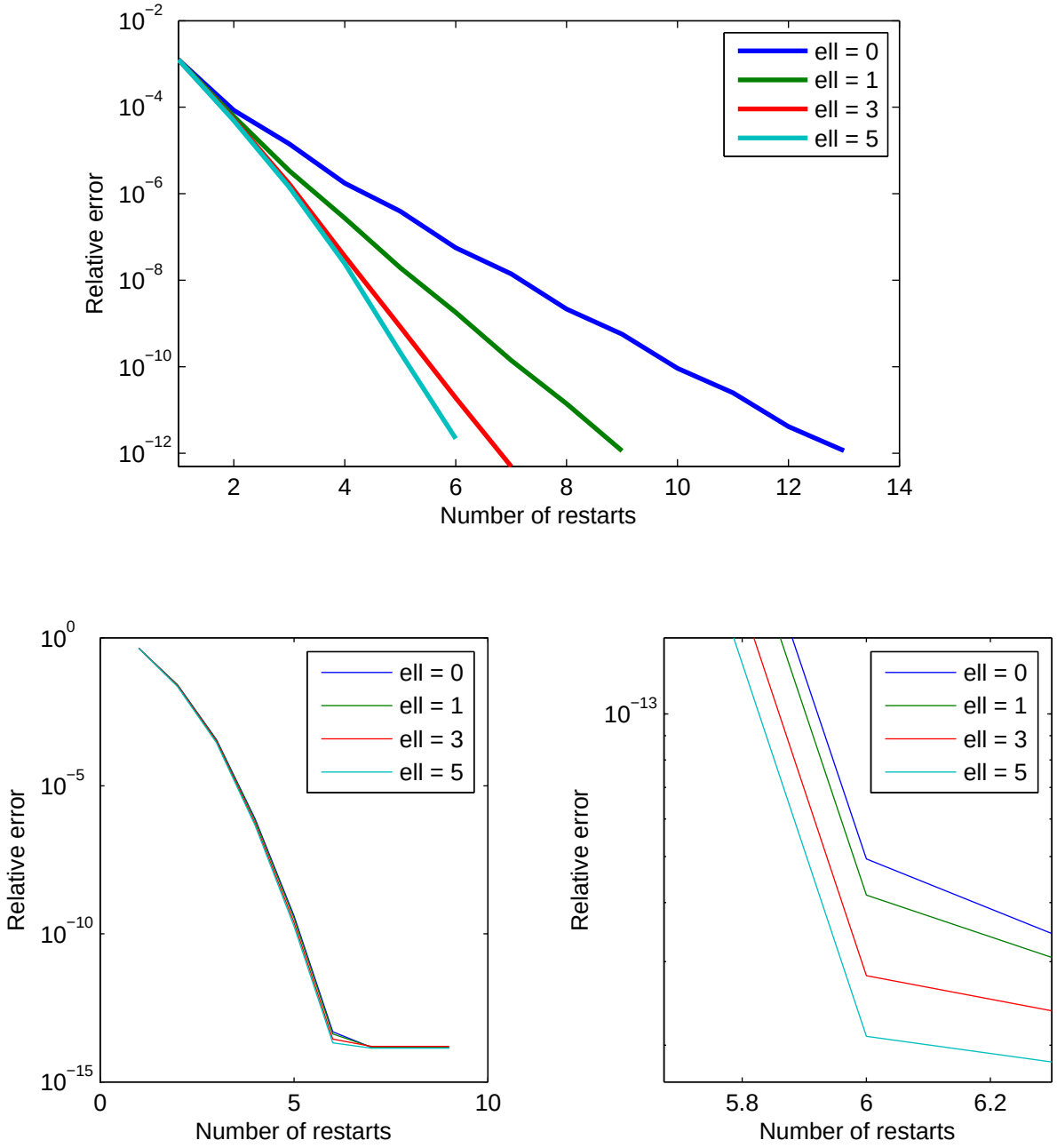


Figure 5.3: Top - effect of deflation for the matrix with diagonal elements $A(i,i) = 15 * i, i = 1, \ldots, 100$*, for deflate number* $\ell = 0$ *(no deflation),* $\ell = 1, \ell = 3$ *and* $\ell = 5$*. Bottom left - effect of deflation for the matrix with diagonal elements* $A(i,i) = i, i = 1, \ldots, 100$*. Bottom right - detail of the corner in the picture at the bottom left.*

In the first problem the speed of convergence can be accelerated using deflation in the restarted Krylov subspace method, and the acceleration depends on the choice of the parameter ℓ. The accelaration of convergence is more obvious for larger values of the parameter ℓ, see Figure 5.3 (top). The acceleration is not significiant for the matrix (5.1) from Experiment 1, see Figure 5.3 (bottom left), for detail, see Figure 5.3 (bottom right).

This is caused by the choice of the eigenvalues - there they are not so well separated, so the effect of deflation is negligible.

Using the deflated restart, we approximately identify a subspace of dimension ℓ, which delays the convergence. In case of the matrix square root, it is an A-invariant subspace associated with ℓ eigenvalues close to the singularity, i.e., ℓ smallest eigenvalues. If these eigenvalues are relatively close, the effect of deflation is not so obvious, see [16], pp. 9-12.

Experiment 3

The goal of the third experiment is to compare several methods on a matrix $A = \texttt{Lap2D} \in \mathbb{R}^{10000\times 10000}$, see [6], pp. 16-17, which is the standart Laplacian on a uniform 100×100 grid. The matrix has the block form

$$A = \begin{bmatrix} B & -I & & & & \\ -I & B & -I & & & \\ & -I & B & -I & & \\ & & \ddots & \ddots & \ddots & \\ & & & -I & B & -I \\ & & & & -I & B \end{bmatrix}, \text{ with } B = \begin{bmatrix} 4 & -1 & & & & \\ -1 & 4 & -1 & & & \\ & -1 & 4 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 4 & -1 \\ & & & & -1 & 4 \end{bmatrix} \in \mathbb{R}^{100\times 100},$$

its condition number is $\kappa(A) \approx 4133.6$ and its eigenvalues are given by

$$4\left[\sin^2\left(\frac{i\pi}{202}\right) + \sin^2\left(\frac{j\pi}{202}\right)\right], \quad i,j = 1,2,\ldots,100.$$

The vector $\boldsymbol{b}$ is random with euclidean norm equal to one. We computed an approximation of $\sqrt{A}\boldsymbol{b}$. Methods included in this experiments were the standart and the restarted Krylov subspace method for different restart lengths m, the restarted Krylov subspace method without and with deflation using rational approximation of the square root function, the extended Krylov subspace method and the polynomial least squares approximation.

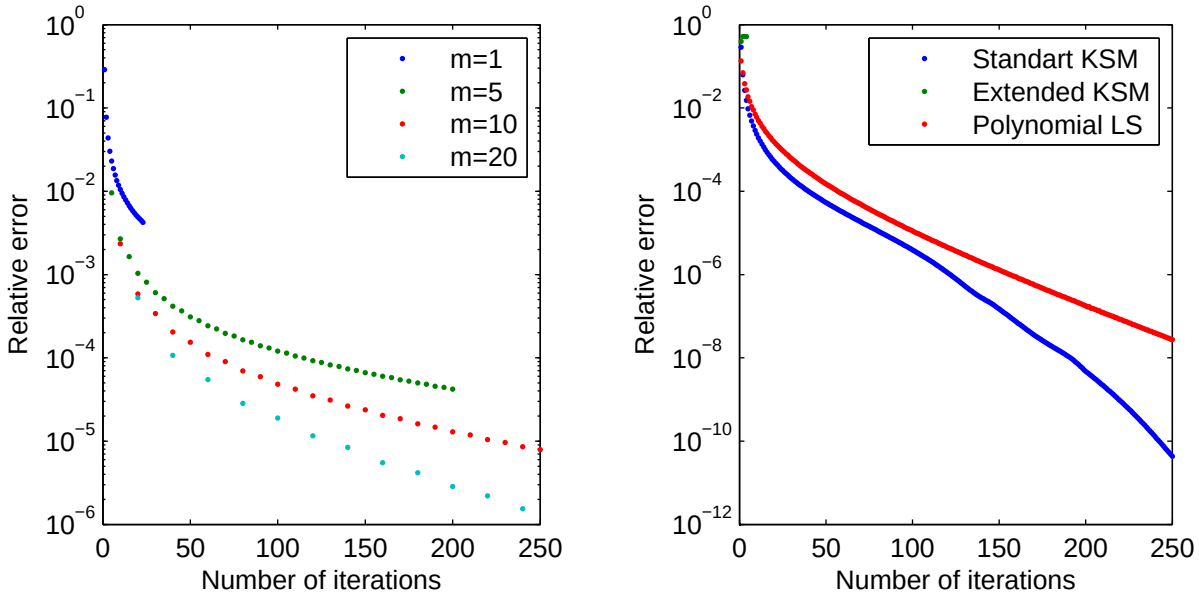


Figure 5.4: Convergence behaviour. Left - the restarted Krylov subspace method. Right - the standart Krylov subspace method, the extended Krylov subspace method and the polynomial least squares approximation.

Figure 5.4 (left) shows the relative error for the restarted Krylov subspace method for restart lengths $m = 1, 5, 10, 20$. We can observe similar behaviour as in Experiment 1 - if we want to obtain the same order of magnitude, the smaller restart length we use, the more iterations we have to compute. Computation using the extended Krylov subspace method was stopped after three iterations, so there is no reliable comparision, see the note bellow. The polynomial least squares approximation achieved similar approximation result as the restarted Krylov subspace method for the restart length $m = 20$, see Figure 5.4 (right).

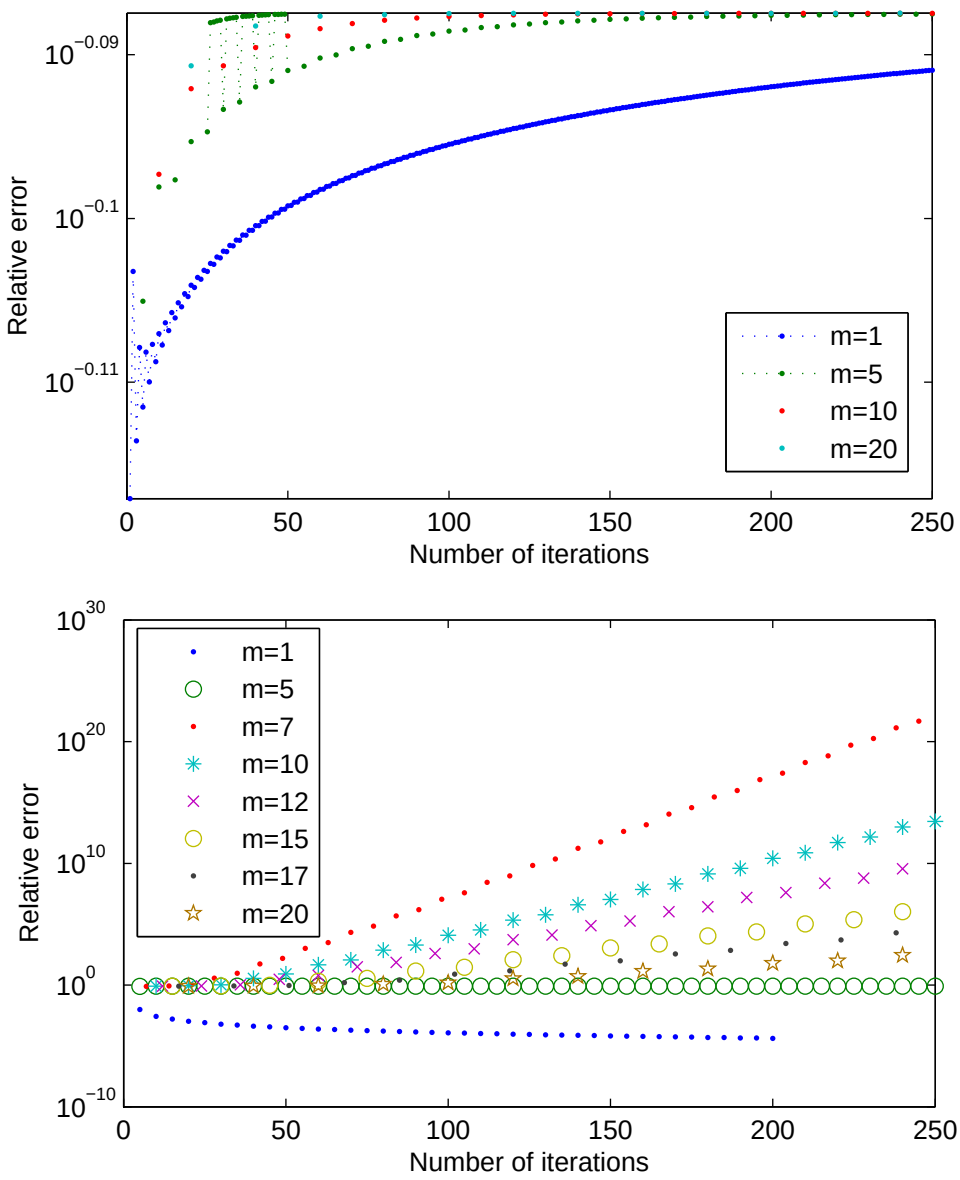


Figure 5.5: Convergence behaviour. Top - the restarted Krylov subspace method using rational approximation of the matrix square root, for different restart length m. Bottom - the restarted Krylov subspace method with deflation, $\ell = 5$, using rational approximation of the matrix square root.

Further, we compared the restarted Krylov subspace method using rational approximation of matrix square root (Zolotarev relative best rational approximation to $z/\sqrt{z}$, see [21], pp. 91-101) for different restart lengths, see Figure 5.5 (top). For all restart lengths, the differences between particular relative errors are negligible, the method does not converge. The reason is in rational approximation of function f, which was not properly computed. The error of the rational approximation was further enhaced by computing additional iterations.

Finally, the approximation of $\sqrt{A}\boldsymbol{b}$ was computed using the restarted Krylov subspace method with deflation, $\ell = 5$, and using the same rational approximation of matrix square root as above. Figure 5.5 (bottom) shows that only the method with restart length $m = 1$ (steepest descent) converge. For larger restart lengths the method does not converge.

Note. Some of the algorithms above automatically terminated before reading sufficient error tolerance. In case of the restarted Krylov subspace method for $m = 1$ and for $m = 5$, and the extended Krylov subspace method, the reason is that the error of the approximation did not decrease or the decrease was negligible. The same problem occured while computing approximation using the restarted Krylov subspace method with deflation, $\ell = 5$.

Experiment 4 - Maxwell's Equations

The modeling of transient electromagnetic fields in inhomogeneous media is a typical task arising, for example, in geophysical prospecting. Such models can be based on the quasi static Maxwell's equations, see [21], pp. 135-139,

$$\mathrm{rot}\boldsymbol{E} + \mu\partial_t\boldsymbol{B} = 0, \tag{5.2}$$

$$\mathrm{rot}\boldsymbol{B} - \sigma\boldsymbol{E} = \boldsymbol{j}, \tag{5.3}$$

$$\mathrm{div}\boldsymbol{B} = 0, \tag{5.4}$$

where

$\boldsymbol{E} = \boldsymbol{E}(\boldsymbol{x}, t)$	is the electric field,
$\boldsymbol{B} = \boldsymbol{B}(\boldsymbol{x}, t)$	is the magnetic field,
$\sigma = \sigma(\boldsymbol{x})$	is the electric conductivity,
$\mu = 4\pi \cdot 10^{-7}$	is the magnetic permeability,
$\boldsymbol{j} = \boldsymbol{j}(\boldsymbol{x}, t)$	is the external source current density.

After eliminating $\boldsymbol{B}$ from (5.3) and putting into (5.2), we obtain the second order partial differential equation

$$\nabla \times \nabla \times \boldsymbol{E} + \mu\sigma\partial_t\boldsymbol{E} = -\mu\partial_t\boldsymbol{j}$$

for the electric field. That arrives at a scalar bidimensional heat equation

$$-\nabla^2\boldsymbol{E} + \mu\sigma\partial_t\boldsymbol{E} = -\mu\sigma_t\boldsymbol{j}.$$

The source term $\boldsymbol{j}$ typically results from a known stationary transmitter with a driving current that is shut off at time $t = 0$, i.e.,

$$\boldsymbol{j}^e(\boldsymbol{x}, t) = \boldsymbol{q}(\boldsymbol{x})H(-t)$$

with the vector field $\boldsymbol{q}$ denoting the spatial current pattern and the Heaviside unit step function H. Discretization of this equation gives a linear ordinary differential equation

$$M\boldsymbol{E}'(t) = K\boldsymbol{E}(t), \quad \boldsymbol{E}(t_0) = \boldsymbol{E}_0, \tag{5.5}$$

with symmetric matrices $K, M \in \mathbb{R}^{728\times 728}$ and vectors $\boldsymbol{E}(t)$, $\boldsymbol{E}_0 \in \mathbb{R}^{728}$. We define $A := M^{-1}K \in \mathbb{R}^{728\times 728}$. Solution of the discretized system of the differential equations (5.5) is

$$f^t(A)\boldsymbol{b} = \exp(tA)\boldsymbol{b}, \quad t > 0.$$

Time discretization was computed on the domain $(-\frac{\pi}{2}, \frac{\pi}{2}) \times (-\frac{\pi}{2}, \frac{\pi}{2})$ using the implicit Euler method of the sixth order.

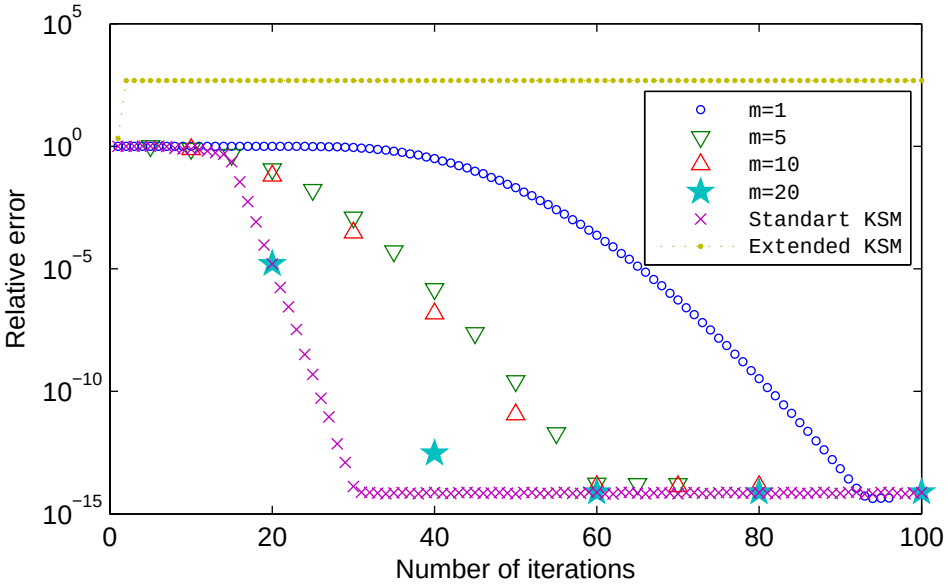


Figure 5.6: Convergence behaviour. The standart, the extended and the restarted Krylov subspace method for different restart lengths.

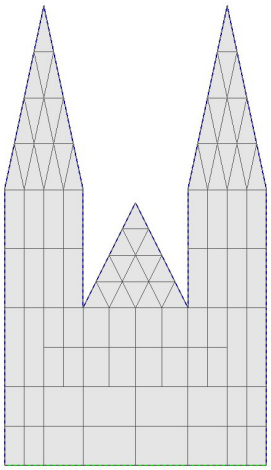
In this experiment, we compared the standart, the restarted and the extended Krylov subspace methods. We take $t = 1$. Because the matrix A has complex eigenvalues, it was not possible to compute approximation via the polynomial least squares approximation (we could not approximate the function by splines). On Figure 5.6 we can see, that the behaviour of the standart and the restarted Krylov subspace method is similar as in Experiment 1 and Experiment 3. Restart length is inversely proportional to the number of iterations needed to obtain the same order of accuracy. The extended method did not converge.

Experiment 5 - Heat Equation

In this experiment we consider the initial-boundary value problem for the heat equation on the unit cube in 2 dimensions,

$$\begin{aligned} \partial_t u &= \Delta u && \text{in } \Omega, t > 0, \\ u(\boldsymbol{x}, t) &= 0 && \text{on } \Gamma = \partial\Omega, t > 0, \\ u(\boldsymbol{x}, 0) &= u_0(x) && \text{in } \Omega. \end{aligned} \tag{5.6}$$

The equation was discretized using implicit SDIRK method of the second order, second order elements, and as a domain Ω a kind of naive simulation of a cross-section of cathedral was taken, see the picture below.



After discretization, the problem (5.6) reduces to the initial value problem

$$\begin{aligned} \boldsymbol{u}'(t) &= A\boldsymbol{u}(t),\ t > 0 \\ \boldsymbol{u}(0) &= \boldsymbol{b}. \end{aligned} \tag{5.7}$$

The obtained matrix $A \in \mathbb{R}^{328\times 328}$ is symmetric. Its eigenvalues are distinct, real and lie in the interval $(0.097, 13.3361)$, condition number of this matrix is 294.8203. The solution of (5.7) is given by

$$\boldsymbol{u}(t) = f^t(A)\boldsymbol{b}, \text{ where } f^t(z) = \exp(tz).$$

In this experiment, we compare efficienty and time of computation of several methods - the standart and the restarted Krylov subspace method for different restart lengths, the extended Krylov subspace method and the polynomial least squares approximation. The approximation is computed in time $t = 1$.

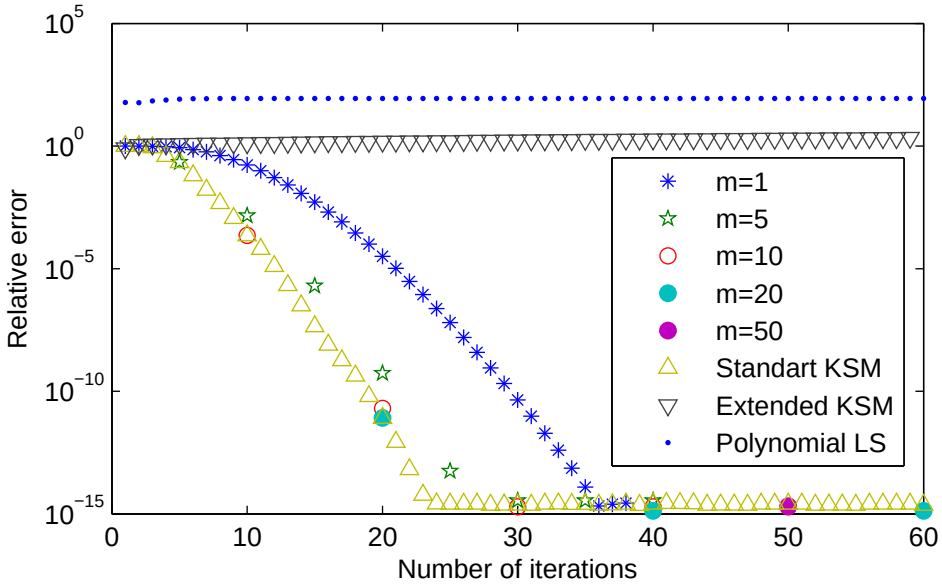


Figure 5.7: Convergence behaviour. The standart, the restarted and the extended Krylov subspace method and the polynomial least squares approximation.

The extended Krylov subspace method and the polynomial least squares do not converge. In the case of polynomial least squares approximation, the knots $t_0, \ldots, t_n$ might not be properly chosen. In Experiment 1 and Experiment 3, the extremal eigenvalues of A are known. Here, we have to compute them numerically using the Matlab command `eig`. If the computed values are inaccurate, then this error is reflected in further computation. The standart and the restarted Krylov subspace method have similar behaviour as in the previous experiments, see Figure 5.7. Because the extended Krylov subspace method and the polynomial least squares do not converge, we show the time comparision only for the standart and the restarted Krylov subspace method, see Figure 5.8.

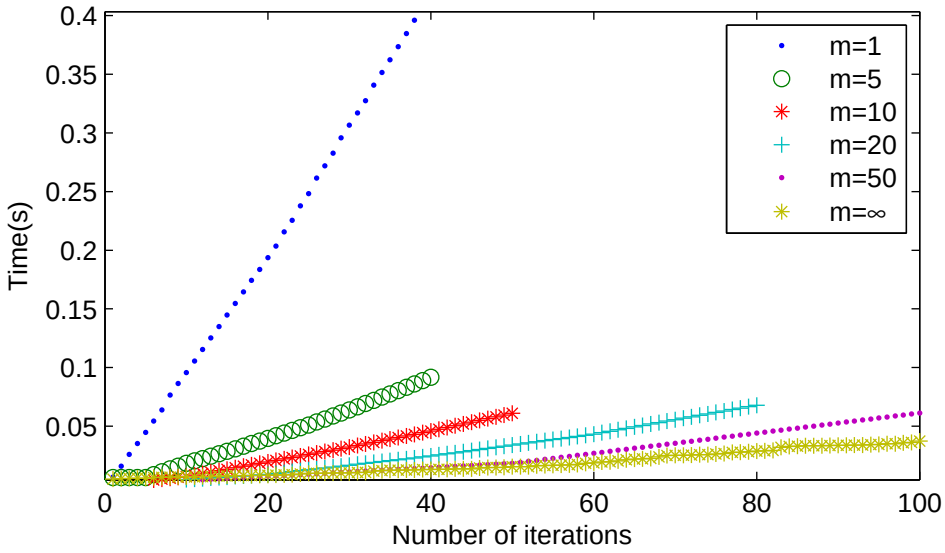


Figure 5.8: Time comparision. The standart and the restarted Krylov subspace method for different restart lengths.

Experiment 6

In the last experiment, we compare time of computation for different methods on a larger matrix. We chose a matrix from the University of Florida matrix sparse collection, namely the matrix `Trefethen2000` $\in \mathbb{R}^{2000\times 2000}$. The differences are more obvious than in the previous experiments.

Figure 5.9 (top left) shows for relatively small restart lengths in the restarted Krylov subspace method, computational time grows faster with the number of iterations if m is smaller. This is caused by the cost of evaluation of approximation $f(A)\boldsymbol{b}$ after each restart, see also Experiment 1. The advantage of restart appears when the restart becomes significantly larger, compare the curve for $m = 100$ and $m = 150$ in Figure 5.9 (top right). Similarly in Figure 5.9 (bottom left) we can see that the standart Krylov subspace method is faster than the restarted Krylov subspace method only for some limited number of the first iterations. This difference is more visible for large matrices A, where restarting can accelerate the time of computation. For completeness, Figure 5.9 (bottom right) compares the polynomial least squares with the standart and the extended Krylov subspace methods.

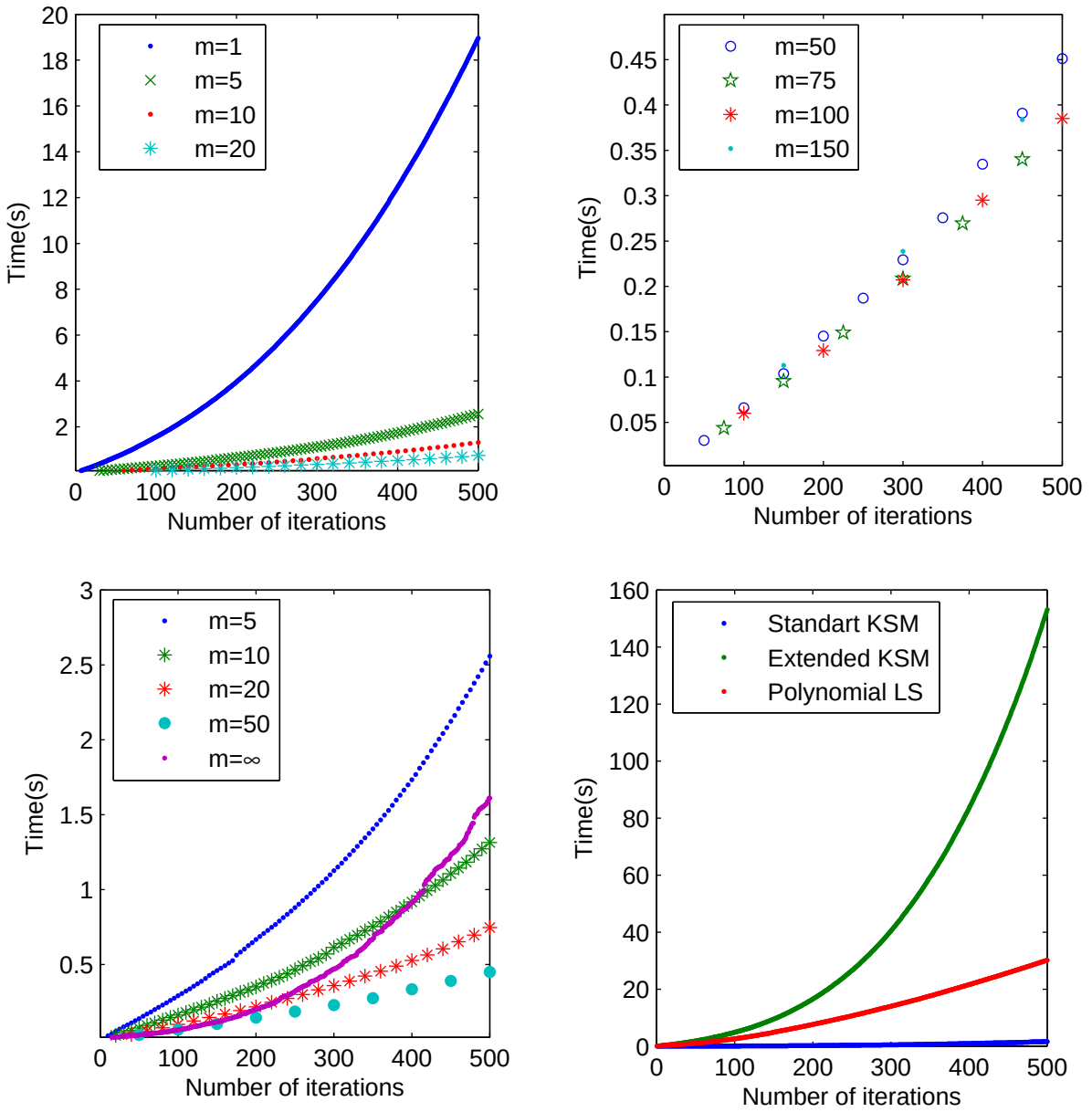


Figure 5.9: Time comparision. Top left - the restarted Krylov subspace method for restart lengths $m = 1, 5, 10, 20$. Top right - the restarted Krylov subspace method for restart lengths $m = 50, 75, 100, 150$. Bottom left - the standart Krylov subspace method and the restarted Krylov subspace method with restart lengths $m = 5, 10, 20, 50$. Bottom right - the standart Krylov subspace method, the extended Krylov subspace method and the polynomial least squares approximation.

Conclusion

'I think and think for months and years. Ninety-nine times, the conclusion is false. The hundredth time I am right.'
Albert Einstein

In the presented work, we described the ways of defining matrix functions and we summarized their properties. We mentioned methods for approximation of $f(A)$, and then we turned to methods for approximation of $f(A)\boldsymbol{b}$, which were the main object of our interest. From the methods of non-Krylov type, we focused on the polynomial least squares approximation. Here, the basis of orthogonal polynomials was generated using the three-term Stieltjes reccurence. Further, we studied methods of the Krylov type - the standart Krylov subspace method, the restarted Krylov subspace method, the restarted Krylov subspace method with deflation and the extended Krylov subspace method.

In the numerical experiments, we compared the polynomial least squares approximation and the Krylov subspace methods. We shown, that for a small matrix A, the computation using the standart Krylov subspace method without restarting is faster than using restart after a predefined number of iterations. The advantage of restarting in acceleration of the computation may appear only for large problems. Here the standart Krylov subspace method is faster then the restarted one at the beginning of computation, but exceeding a certain number of iterations, the evaluation of the approximation of $f(A)\boldsymbol{b}$ using the standart method, can become more computationaly chalenging then updating the approximation after each restart in the restarted method. Another way, how to accelarate the convergence of the restarted Krylov subspace method, is using restarting with deflation. As we have shown, the effect of deflation is negligible if the distance among the eigenvalues of A is smaller.

We performed an experiment using the modification of the Krylov subspace method based on the rational approximation of a function f. We have seen, that if we use the restarted Krylov subspace method with deflation, the choice of restart length m is important for the convergence. For some of the choices, the method diverged, the best results were obtained for $m = 1$, i.e. the method of the steepest descent for matrix functions.

Further, we have illustrated that the extended Krylov subspace method is slower than the standart or the restarted Krylov subspace method. In the extended method the considered subspace contains also information about the matrix inverse, and its dimension is twice as large as the dimension of the subspace in the standart or the restarted Krylov subspace method. In some experiments, the method did not converge.

Finally, we have shown, that the polynomial least squares approximation can be more efficient than the extended Krylov subspace methods. Unfortunately, this method cannot

be used for problems with a matrix A having complex eigenvalues.

Bibliography

[1] M. Afanasjew, M. Eiermann, O. G. Ernst, S. Güttel, *A generalization of the steepest descent method for matrix functions*, Electronic Transactions on Numerical Analysis, volume 28, 2008, pp. 206-222.

[2] M. Afanasjew, M. Eiermann, O. G. Ernst, S. Güttel, *Implementation of a restarted Krylov subspace method for the evaluation of matrix functions*, Linear Algebra and its Applications, volume 429, 2008, pp. 2293-2314.

[3] W. E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quarterly of Applied Mathematics, volume 9, 1951, pp. 17-29.

[4] R. Byers, *Solving the algebraic Riccati equation with the matrix sign function*, Linear Algebra and its Applications, volume 85, 1987, pp. 267-279.

[5] A. J. Carpenter, A. Ruttan, R. S. Varga: *Extended numerical computations on the "1/9" conjecture in the rational approximation theory*, Springer-Verlag, Lecture Notes in Mathematics, volume 1105, 1990, pp. 383-411.

[6] J. Chen, M. Anitescu, Y. Saad, *Computing $f(A)b$ via least squares polynomial approximations*, SIAM Journal on Scientific Computing, volume 33(1), 2011, pp. 195-222.

[7] W. J. Cody, G. Meinardus, R.S. Varga: *Chebyshev rational approximations to e^{-x} in $[0, +\infty)$ and applications to heat-conduction problems*, Journal of Approximation Theory, volume 2(1), 1969, pp. 50-65.

[8] M. Crouzeix, *Numerical range and functional calculus in Hilbert space*, Journal of Functional Analysis, volume 244(2), 2007, pp. 668-690.

[9] P. I. Davies, N. J. Higham: *Computing $f(A)b$ for matrix functions f*, QCD and numerical analysis III, Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, volume 47, 2005, pp. 15-24.

[10] C. Davis: *Explicit functional calculus*, Linear Algebra and its Applications, volume 6, 1973, pp. 193-196.

[11] P. J. Davis, *Interpolation and approximation*, Dover Publications, Inc., New York, NY, 1975.

[12] V. Druskin, A.Freenbaum, L. Knizhnerman: *Using nonorthogonal Lanczos vectors in the computation of matrix functions*, SIAM, volume 19(1), 1998, pp. 38-54.

[13] V. Druskin and L. Knizhnerman, *Extended Krylov subspaces: approximation of the matrix square root and related functions*, SIAM Journal on Matrix Analysis and Applications, volume 19(3), 1998, pp. 755-771.

[14] R. G. Edwards, U. M. Heller, R. Narayanan: *Chiral fermions on the lattice*, Parallel Computing, volume 25, 1999, pp. 1395-1407.

[15] M. Eiermann, O. G. Ernst, *A restarted Krylov subspace method for the evaluation of matrix functions*, SIAM Journal on Numerical Analysis, volume 44(6), 2006, pp. 2481-2504.

[16] M. Eiermann, O. G. Ernst, S. Güttel, *Deflated restarting for matrix functions*, SIAM Journal on Matrix Analysis and Applications, volume 32, 2011, pp. 621-641.

[17] J. van den Eshof, A. Frommer, T. Lippert, K. Schilling, H. A. van der Vorst: *Numerical methods for the QCD overlap operator. I. Sign-function and error bounds*, Computer Physics Communications, volume 146, 2002, pp. 203-224.

[18] R. W. Freund and M. Hochbruck, *Gauss-quadratures associated with the Arnoldi process and the Lanczos algorithm*, in Linear Algebra for Large-Scale and Real-Time Applications, M. S. Moonen, G. H. Golub, and B. L. R. de Moor, eds., volume 232 of NATO Advanced Science Institutes Series E: Applied Sciences, Kluwer Academic Publishers, Dordrecht, 1993, pp. 377-380.

[19] A. Frommer, V. Simoncini: *Matrix functions*, Model Order Reduction: Theory, Research Aspects and Applications, Mathematics in Industry, Schilders, W. H. Schilder and H. A. van der Vorst, eds, Springer, Heidelberg, 2008, pp. 1-25.

[20] S. H. Golub, C. F. Van Loan: *Matrix computation*, The Johns Hopkins University Press, 1996, pp. 352-368, 555-574.

[21] S. Güttel, *Rational Krylov methods for operator functions*, Dissertation Thesis, Technische Universität Bergakademie Freiberg, 2010.

[22] N. J. Higham, *Accuracy and stability of numerical algorithms*, SIAM, Philadelphia, 1996, pp. 292-294.

[23] N. J. Higham: *Evaluating Padé approximants of the matrix logarithm*, SIAM Journal on Matrix Analysis and Applications, volume 22(4), 2001, pp. 1126-1135.

[24] N. J. Higham: *Functions of matrices*, MIMS EPrint, 2005, pp. 1-23.

[25] N. J. Higham: *Functions of matrices: theory and computation*, SIAM, 2008.

[26] N. J. Higham: *Newton's method for the matrix square root*, Mathematics of Computation, volume 46, 1986, pp. 537-549.

[27] N. J. Higham: *Stable iteration for the matrix square root*, Numerical Algorithms, volume 15(2), 1997, pp. 227-242.

[28] N. J. Higham: *The scaling and squaring method for the matrix exponential revisited*, SIAM, Journal on Matrix Analysis and Applications, volume 26(4), 2005, pp. 1179-1193.

[29] R. A. Horn, C. R. Johnson: *Topics in matrix analysis*, Cambridge University Press, 1991, pp. 425 - 428.

[30] C. S. Kenney, A. J. Laub: *Rational iterative methods for the matrix sign function*, SIAM Journal on Matrix Analysis and Applications, volume 12(2), 1991, pp. 273-291.

[31] L. Knizhnerman, V. Simoncini, *A new investigation of the extended Krylov subspace method for matrix function evaluations*, Numerical Linear Algebra with Applications, volume 17(4), 2010, pp. 615-638.

[32] A. N. Krylov, *On the numerical solution of the equation by which, in technical matters, frequencies of small oscillations of material systems are determined*, Izv. Akad. Nauk SSSR Ser. Fiz.-Mat., 1931, pp. 491-539.

[33] C. Lanczos *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, Journal of Research of the National Bureau of Standards, volume 45, 1950, pp. 255-282.

[34] C. F. Van Loan, *A study of the matrix exponential*, Numerical Analysis Report No. 10, University of Manchester, Manchester, UK, August 1975.

[35] B. Meini: *The matrix square root from a new functional perspective: Theoretical results and computational issues*, SIAM Journal on Matrix Analysis and Applications, volume 26(2), 2004, pp. 362-376.

[36] C. Moler, C. V. Loan: *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review, volume 45(1), 2003, pp. 3-49. A portion of this paper originally appeared in SIAM Review, volume 20(4), 1978, pp. 801-836.

[37] J. D. Roberts: *Linear model reduction and solution of the algebraic Riccati equation by use of the sign function*, International Journal of Control, volume 32(4), 1980, pp. 677-687.

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

[39] I. Moret and P. Novati, *Interpolating functions of matrices on zeros of quasi-kernel polynomials*, Numerical Linear Algebra with Applications, volume 12, 2005, pp. 337-353.

[40] I. Moret and P. Novati, *The computation of functions of matrices by truncated Faber series*, Numerical Functional Analysis and Optimization, volume 22, 2001, pp. 697-719.

[41] P. Novati, *A method based on Fejér points for the computation of functions of non-symmetric matrices*, Applied Numerical Mathematics, volume 44, 2003, pp. 201-224.

[42] G. Opitz, *Steigungsmatrizen*, Zeitschrift für Angewandte Mathematik und Mechanik, volume 44, 1964, pp. T52-T54.

[43] B. Philippe, R. B. Sidje, *Transient solutions of Markov processes by Krylov subspaces*, Research Report, RR-1989, INRIA, 1995.

[44] M. J. D. Powell, *Approximation theory and methods*, Cambridge University Press, Cambridge, UK, 1981, pp. 72-84.

[45] T. J. Rivlin, *An introduction to the approximation of functions*, Dover Publications, 1981, pp. 22.

[46] Y. Saad: *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM Journal on Numerical Analysis, volume 29(1), 1992, pp. 209-228.

[47] Y. Saad, *Iterative solution of indefinite symmetric systems by methods using orthogonal polynomials over two disjoint intervals*, SIAM Journal on Numerical Analysis, volume 20(4), 1983, pp. 784-881.

[48] M. H. Schultz, *Spline analysis*, Prentice Hall, 1973.

[49] V. Simoncini, *A new iterative method for solving large-scale Lyapunov matrix equations*, SIAM Journal on Scientific Computing, volume 29(3), 2007, pp. 1268-1288.

[50] T. J. Stieltjes *Sur l' évaluation approchée des intégrales*, C. R. Acad. Sci. Paris, volume 97, 1883, pp. 740-742, 798-799. Reprinted in Oeuvres I (P. Noordhoff, Groningen), 1914, pp. 314-318.

[51] G. W. Stewart, *An Arnoldi-Schur algorithm for large eigenproblems*, SIAM Journal on Matrix Analysis and Applications, volume 23(3), 2002, pp. 601-614.

[52] D. Suchá, *Numerické aproximace maticových funkcí*, Bachelor Thesis, Charles University in Prague, Faculty of Mathematics and Physics, 2009.

[53] P. K. Suetin, *Series of Faber polynomials, analytical methods and special functions*, Gordon and Breach Science, Amsterdam, 1998. Originally published in Russion as Riady po mnogochlennam Fabera by Nauka, Moscow, 1984.

[54] 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*, Journal of Computational and Applied Mathematics, volume 18(2), 1987, pp. 249-263.

[55] H. A. van der Vorst: *Solution of $f(A)x = b$ with projection methods for the matrix A*, In Numerical Challenges in Lattice Quantum Chromodynamics, Lecture Notes in Computational Science and Engineering, Springer, Berlin, volume 15, 2000, pp. 18-28.

[56] A. Wragg, C. Davies: *Computation of the exponential of a matrix II: Practial considerations*, Journal of the Institute of Mathematics and Its Applications, volume 15(3), 1975, pp. 273-278.

[57] V. Zakian, *Rational approximants to the matrix exponential*, Electronic Letters, volume 6(25), 1970, pp. 814-815.