Charles University in Prague

Faculty of Mathematics and Physics

BACHELOR THESIS

Tomáš Gergelits

Krylov subspace methods: Theory, applications and interconnections

Department of Numerical Mathematics

Supervisor of the bachelor thesis:	prof. Ing. Zdeněk Strakoš, DrSc.
Study programme:	Mathematics
Specialization:	General Mathematics

Prague 2011

I am grateful to prof. Ing. Zdeněk Strakoš, DrSc., my supervisor, for introducing me to the world of numerical analysis, for his many suggestions and lasting support during writing this thesis. I would like to thank RNDr. Petr Tichý, Ph.D. for his prompt and efficient help with technical realization.

This bachelor thesis was supported by the Grant Agency of the Academy of Sciences of The Czech Republic within the project Theory of Krylov subspace methods and its relationship to other mathematical disciplines (identification code IAA100300802).

I declare that I carried out this bachelor 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 date August 5, 2011

Název práce: Teorie a aplikace Krylovovských metod v souvislostech
Autor: Tomáš Gergelits
Katedra: Katedra numerické matematiky
Vedoucí bakalářské práce: prof. Ing. Zdeněk Strakoš, DrSc.

Abstrakt: Po seznámení se s vlastnostmi Čebyševových polynomů a základním přehledem stacionárních iteračních metod je práce zaměřena na studium metody konjugovaných gradientů (CG), Krylovovské metody vhodné pro symetrické a pozitivně definitní matice. Je zdůrazněn principiální rozdíl mezi stacionárními a Krylovovskými metodami. Metoda konjugovaných gradientů je odvozena využitím minimalizace kvadratického funkcionálu a detailně je ukázána souvislost s dalšími oblastmi matematiky (Lanczosova metoda, ortogonální polynomy, kvadraturní vzorce, problém momentů). Je vyzdvihnut vliv konečné aritmetiky. Na teoretickou část navazují experimenty, které studují odhad odvozený v přesné aritmetice a který je často uváděný v literatuře. Je ukázáno, že tento odhad nutně selhává v praktických výpočtech. Na závěr práce jsou popsány dva otevřené problémy, jež mohou být předmětem dalšího studia.

Klíčová slova: Metody Krylovovských podprostorů, konergenční vlastnosti, numerická stabilita, spektrální informace, odhady rychlosti konvergence

Title: Krylov subspace methods: Theory, applications and interconnections
Author: Tomáš Gergelits
Department: Department of Numerical Mathematics
Supervisor: prof. Ing. Zdeněk Strakoš, DrSc.

Abstract: After recalling of properties of Chebyshev polynomials and of stationary iterative methods, this thesis is focused on the description of Conjugate Gradient Method (CG), the Krylov method of the choice for symmetric positive definite matrices. Fundamental difference between stationary iterative methods and Krylov subspace methods is emphasized. CG is derived using the minimization of the quadratic functional and the relationship with several other fields of mathematics (Lanczos method, orthogonal polynomials, quadratic rules, moment problem) is pointed out. Effects of finite precision arithmetic are emphasized. In compliance with the theoretical part, the numerical experiments examine a bound derived assuming exact arithmetic which is often presented in literature. It is shown that this bound inevitably fails in practical computations. The thesis is concluded with description of two open problems which can motivate further research.

Keywords: Krylov subspace methods, convergence behaviour, numerical stability, spectral information, convergence rate bounds

Contents

Introduction

This bachelor thesis has three main goals. The first goal is to prepare a solid theoretical background for a study of Krylov subspace methods in our further research. Willing to achieve this goal, a detail description of the Method of conjugate gradients (CG) and of its relationship to many different mathematical areas is given. The second goal of our thesis is to describe a fundamental difference between stationary iterative methods, the Chebyshev semiiterative method and Krylov subspace methods. The third goal is to emphasize the effects of rounding errors in practical computations and the necessity of rigorous analysis of finite precision behaviour. In compliance with this goal we present numerical experiments which examine an upper bound for the energy norm of the error in CG computations.

In the first chapter we present definitions and basic properties of several mathematical objects from various mathematical areas whose knowledge is useful for the further parts of our thesis. Short review of properties of Riemann–Stieltjes integral, orthogonal polynomials, Jacobi matrices and quadrature rules is given. Very interesting and indeed important is a tight connection between Jacobi matrices and orthogonal polynomials. Since the Chebyshev polynomials take an important role in numerical mathematics in general and in iterative methods in particular, we review their extremal properties.

The history of linear iterative methods can be divided into two main periods. The first period is dominated by stationary iterative methods and can be roughly dated between 1950 and 1970. The second period lasts from the beginning of 1970s up to now and is dominated by Krylov subspace methods. In the second part of our thesis we, based on [26, Chapter 4] and [34, Chapter 3 and 4], present a brief summary of basic stationary iterative methods and, following [20], we introduce the principle of Krylov subspace methods. The Chebyshev semiiterative method is also discussed, based on [34, Chapter 5].

The method of conjugate gradients was described by Hestenes and Stiefel in their famous paper [14] and it can be related to various areas of mathematics. Already in the original paper (see [14, Sections 14–18]) the relationship of the CG method to the Riemann-Stieltjes integral, Gauss quadrature, orthogonal polynomials and continued fractions was described. In the third chapter of our thesis we derive CG using the minimalization of the quadratic functional and we reveal the important relationship of the CG method to the Lanczos method. Using a tight connection between the Lanczos algorithm and Jacobi matrices we describe the relationship of CG and Lanczos algorithms to orthogonal polynomials. These polynomials are orthogonal with respect to an inner product defined by the Riemann-Stieltjes integral for specific distribution function. We will see that CG iterations are tightly related to approximations of the Riemann-Stieltjes integral of specific function by Gauss-Christoffel quadrature rule. Using the relationship between the Gauss-Christoffel quadrature and problem of moments we describe how the CG method can be viewed as some kind of model reduction. We also present the formulation of the problem of moments given by Vorobyev because this point of view is convenient also for matrices which are not symmetric and positive definite. In this and the following parts of the thesis we greatly benefit especially from the thorough description in [20, 21, 31].

In exact arithmetic it often happens that the convergence rate in the CG computations accelerates with the number of performed iterations. An explanation of this superlinear convergence is based on the relationship between CG and the Lanczos method. We briefly comment that the analysis of the convergence behaviour can be based also on potential theory.

Practical computations are influenced by rounding errors and it is known that the effect of finite precision arithmetic is often very substantial. The difficulty of analysis of rounding errors and their substantial consequences were reasons why CG and Lanczos method were quite overlooked by numerical analysts for a period of time. We review important works of Paige and Greenbaum which rigorously analyze the effects of rounding errors. Their results allowed to explain the behaviour of CG and the Lanczos method in practical computations.

Since convergence curves in exact and finite precision arithmetic can be substantially different, we emphasize that the analysis of the real convergence behaviour and the derivation of estimates of the error must involve proper mathematical analysis of rounding errors. Numerical experiments in the fourth chapter of our thesis demonstrate that an upper bound which does not take such affects into account can completely fail in practical computations.

Krylov subspace methods still represent an area of very active research. It is believed that detailed understanding of the properties of Krylov subspaces could help in analysis of modern iterative methods especially in case of non-normal matrices. Two open problems about Krylov subspaces are formulated in the last chapter of our thesis. They could be motivate our further research.

Brief summary and short discussion is given at the end. We found convenient to note here that we will mostly consider in this thesis real symmetric or symmetric positive definite (SPD) matrices in this thesis.

1 Required tools

1.1 Some basic concepts

Definition 1.1 (Spectrum of a matrix, spectral radius). *Consider square matrix $A \in \mathbb{R}^{N\times N}$ and denote the eigenvalues of A as $\lambda_i,\ i = 1, \dots, N$.*
Spectrum *$\sigma(A)$ of matrix A is a set of all eigenvalues, i.e.,*

$$\sigma(A) = \{\lambda_1, \dots, \lambda_N\}.$$

Spectral radius *$\rho(A)$ of matrix A is a number*

$$\rho(A) \equiv \max_{i=1,\dots,N} |\lambda_i|.$$

Definition 1.2 (Matrix norm). *Let $\|\cdot\|_V$ be any norm on the space $\mathbb{R}^N$.*
Matrix norm *$\|\cdot\|_M$ induced by the vector norm $\|\cdot\|_V$ can be defined as*

$$\|A\|_M \equiv \frac{\|Ax\|_V}{\|x\|_V}.$$

Theorem 1.1. *For every $\epsilon > 0$ there exists matrix norm $\|\cdot\|_M$ induced by some vector norm $\|\cdot\|_V$ such that*

$$\|A\|_M < \rho(A) + \epsilon \quad \forall\ A \in \mathbb{R}^{N\times N}.$$

Definition 1.3 (Symmetric and positive definite matrix). *We say that $A = (a_{i,j})_{i,j=1}^n$ is* **symmetric** *if*

$$a_{i,j} = a_{j,i}, \quad \textit{for} \quad i, j = 1, \dots, n.$$

We say that the matrix A is positive definite if

$$\begin{aligned} x^T Ax &> 0 \quad \forall\ x \neq 0 \\ x^T Ax &= 0 \quad \Leftrightarrow x = 0. \end{aligned}$$

Symmetric and positive definite matrix A will be often denoted as SPD matrix A in this thesis.

Definition 1.4 (Irreducible matrix). *We say that $A = (a_{i,j})_{i,j=1}^n$ is irreducible, if the associated graph is strongly connected. The associated graph has n vertexes and there is an edge between i^{th} and j^{th} vertex, if $a_{i,j} \neq 0$. Strong connection means, that for every pair of vertexes v, u, there exists directed path form u to v and conversely.*

Definition 1.5 (Irreducibly and strictly diagonally dominant matrices). *We say that $A = (a_{i,j})_{i,j=1}^n$ is diagonally dominant, if it satisfies*

$$|a_{i,i}| \geq \sum_{\substack{j=1\\ j\neq i}}^{n} |a_{i,j}|.$$

A is called strictly diagonally dominant, if the inequality is sharp. A is called irreducibly diagonally dominant, if it is irreducible and diagonally dominant, with sharp inequality for at least one i.

1.2 Riemann-Stieltjes integral

The Riemann-Stieltjes integral is a generalization of the Riemann integral and satisfies classical properties such as linearity, additivity, etc. The generalization is quite straightforward and it does not represent hard task. However, since it may not be so well known, we present it here in detail. This section is adopted from the lecture notes [18]. Exposition of the properties of the Riemann-Stieltjes integral can be found also in [16].

Definition 1.6 (Riemann-Stieltjes integral)**.** *Let $[a, b]$ be a finite interval in the real line and let f, ω be real valued functions of a real variable.*
A **partition** *of the closed interval $[a, b]$ is a subset $P = \{x_0, x_1, \ldots, x_n\}$ of $[a, b]$ with $a = x_0 < x_1 < \cdots < x_n = b$, $(n \geq 1)$.*
The **norm** *of a partition $P = \{x_0, x_1, \ldots, x_n\}$ is the number*

$$\upsilon(P) = \max_{1 \leq k \leq n} (x_k - x_{k-1})$$

If P, Q are two partitions of $[a, b]$ then P is **finer** *than Q if $P \supset Q$. Note that, in this case, $\upsilon(P) \leq \upsilon(Q)$.*
A **tagged partition** *of $[a, b]$ is a pair (P, t) where $P = \{x_0, x_1, \ldots, x_n\}$ is a partition of $[a, b]$ and $t = (t_1, t_2, \ldots, t_n)$ with $x_{k-1} \leq t_k \leq x_k$.*
If (P, t), (Q, s) are tagged partitions of $[a, b]$ then (P, t) is **finer** *than (Q, s) if P is finer than Q. We denote this by $(P, t) > (Q, s)$.*
If (P, t) is a tagged partition of $[a, b]$ with $P = \{x_0, x_1, \ldots, x_n\}$ then

$$S(P, t, f, \omega) = \sum_{k=1}^{n} f(t_k)(\omega(x_k) - \omega(x_{k-1}))$$

is the **Riemann-Stieltjes sum** *of f with respect to ω for the tagged partition (P, t). Function f is Riemann-Stieltjes* **integrable** *with respect to ω if*

$$\exists L \ \forall \epsilon > 0 \ \exists (Q, s) \quad \forall (P, t) : \ (P, t) > (Q, s) \implies |L - S(P, t, f, \omega)| < \epsilon \quad .$$

In this case, the number L is unique and is called the Riemann-Stieltjes integral of f with respect to ω; it is denoted by

$$\int_a^b f \, d\omega \quad or \quad \int_a^b f(\lambda) \, d\omega(\lambda).$$

The set of functions f which are Riemann-Stieltjes integrable with respect to ω is denoted by $\mathcal{R}(\omega, a, b)$. If $\omega(x) = x$ then $\mathcal{R}(\omega, a, b)$ is the set of Riemann integrable functions on $[a, b]$ and is denoted by $\mathcal{R}(a, b)$.

The Riemann-Stieltjes integral can be defined in exactly the same way as the Riemann integral also for infinite intervals of integration.

Theorem 1.2. *Let $f \in \mathcal{R}(\omega, a, b)$ be bounded on $[a, b]$ and suppose that ω is a function on $[a, b]$ with a continuous derivative ω'. Then $f\omega' \in \mathcal{R}(a, b)$ and*

$$\int_a^b f(\lambda) \, d\omega(\lambda) = \int_a^b f(\lambda)\omega'(\lambda) \, d\lambda.$$

In agreement with [20, Section 3.1] we will consider only nondecreasing functions ω defined on the finite interval $[a,b]$ in this thesis. In such a case, the Riemann-Stieltjes integrability is justified for all continuous functions, see the following theorem and its corollary.

Definition 1.7 (Function with bounded variation)**.** *We say that function $\omega : [a,b] \to \mathcal{R}$ has a* **bounded variation** *if*

$$V_a^b(\omega) = \sup \left\{ \sum_{i=1}^{n} |\omega(x_i) - \omega(x_{i-1})|, \ \{x_i\}_{i=0}^{n} \textit{ is a partition of } [a,b] \right\} < \infty$$

Theorem 1.3 (Sufficient condition for existence)**.** *Let f be continuous and ω with bounded variation on the interval $[a,b]$. Then $f \in \mathcal{R}(\omega, a, b)$.*

Corollary 1.4 (Existence of the Riemann-Stieltjes integral for nondecreasing function)**.** *Let f be continuous and let ω be nondecreasing on the interval $[a,b]$. Then $f \in \mathcal{R}(\omega, a, b)$.*

Proof. Since every function with bounded variation on $[a,b]$ can be expressed as a difference of two nondecreasing and bounded functions, the last theorem secures that $f \in \mathcal{R}(\omega, a, b)$ for every f continuous. □

In agreement with [20, Section 3.1], the nondecreasing function ω from the definition of the Riemann-Stieltjes integral will be called **distribution function**, the **point of increase** of ω is a point in the neighborhood of which function ω is not constant, the number of points of increase will be denoted as $\mathfrak{n}(\omega)$. Note that $\mathfrak{n}(\omega)$ can be finite as well as infinite, even uncountable.

We will often consider distribution function ω as a piecewise constant function with N distinct points of increase $\lambda_1, \ldots, \lambda_N$ and positive weights $\omega_1, \ldots, \omega_N$, i.e.,

$$\omega(\lambda) = \begin{cases} 0 & \text{if} \quad a \le \lambda < \lambda_1 \\ \sum_{j=1}^{i} \omega_j & \text{if} \quad \lambda_i \le \lambda < \lambda_{i+1}, \quad i = 1, \ldots, N-1 \\ \sum_{j=1}^{N} \omega_j & \text{if} \quad \lambda_N \le \lambda < b \end{cases} \tag{1.2.1}$$

In this case the Riemann-Stieltjes integral satisfies

$$\int_a^b f(\lambda)\, d\omega(\lambda) = \sum_{i=1}^{N} \omega_i f(\lambda_i). \tag{1.2.2}$$

1.3 Orthogonal polynomials

This section is based on [6, Chapter 1] and [20, Sections 3.2 and 3.3]; see also [8, Chapter 1]. We find convenient to point out different meaning of inner product in [6]. Inner product, as understood in this thesis, corresponds to properties of positive inner product defined in [6].

Definition 1.8 (Mapping $\langle \cdot, \cdot \rangle_\omega$)**.** *Let $\mathcal{P}$ be a space of polynomials and ω be a nondecreasing function on $[a,b]$. We define mapping $\langle \cdot, \cdot \rangle_\omega : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ as*

$$\langle f, g \rangle_\omega = \int_a^b f(\lambda) g(\lambda)\, d\omega(\lambda) \quad \forall f, g \in \mathcal{P}$$

Theorem 1.5 ($\langle \cdot, \cdot \rangle_\omega$ as an inner product)**.** *Let $\mathcal{P}$ be a space of polynomials and ω be a nondecreasing function. Let ω be such a function that*

$$\langle f, f \rangle_\omega \geq 0 \quad \forall f \in \mathcal{P} \quad \textit{with equality only for } f = 0.$$

Then mapping $\langle f, g \rangle_\omega$ defines an **inner product** *on the space $\mathcal{P}$. The associated norm is given by*

$$\|f\|_\omega^2 \equiv \langle f, f \rangle_\omega = \int_a^b f^2(\lambda)\, d\omega(\lambda).$$

The mapping $\langle \cdot, \cdot \rangle_\omega$ is not in general an inner product because it may exist nontrivial polynomial p such that $\|p\|_\omega = 0$. It is not hard to see that the mapping $\langle \cdot, \cdot \rangle_\omega$ defines an inner product for any nondecreasing distribution function ω with **infinite** number of points of increase.

If the distribution function ω has only N distinct points of increase, the mapping $\langle \cdot, \cdot \rangle_\omega$ is an inner product on subspace $\mathcal{P}_{N-1}$ of polynomials of degree less then N (see e.g. [6, Section 1.1]).

General definition of orthogonal polynomials takes into account distribution functions $\widehat{\omega}(\lambda)$ defined on infinite intervals and thus it requires existence of finite moments of all orders, i.e.,

$$\int_{\mathbb{R}} \lambda^r \, d\widehat{\omega}(\lambda) < \infty, \quad r = 0, 1, 2, \ldots \tag{1.3.1}$$

However, Corollary 1.4 implies that in our case is this assumption satisfied.

Definition 1.9 (Orthogonal polynomials)**.** *Let $\langle \cdot, \cdot \rangle$ be an inner product on the space of polynomials $\mathcal{P}$. Sequence of polynomials $\phi_0(\lambda), \phi_1(\lambda), \ldots, \phi_n(\lambda), \ldots$ (finite or infinite) is called* **orthogonal** *if*

$$\langle \phi_i(\lambda), \phi_j(\lambda) \rangle = 0 \quad \forall i \neq j$$

and

$$\langle \phi_i(\lambda), \phi_i(\lambda) \rangle \neq 0 \quad \forall i.$$

The orthogonal polynomials are called **orthonormal** *if*

$$\langle \phi_i(\lambda), \phi_i(\lambda) \rangle = 1 \quad \forall i.$$

The **monic orthogonal** *polynomials are orthogonal polynomials with leading coefficients equal to one.*

Theorem 1.6 (Existence of monic orthogonal polynomials)**.** *[6, Theorems 1.6 and 1.7] For given distribution function ω with infinite points of increase in the interval $[a, b]$ and appropriate inner product $\langle \cdot, \cdot \rangle_\omega$ on the space of polynomials $\mathcal{P}$ there exists infinite unique sequence of monic orthogonal polynomials $\psi_0(\lambda), \psi_1(\lambda), \ldots$.*

For given distribution function ω with N points of increase in the interval $[a, b]$ and appropriate inner product $\langle \cdot, \cdot \rangle_\omega$ on the space of polynomials $\mathcal{P}_{N-1}$ there exists unique sequence of monic orthogonal polynomials $\psi_0(\lambda), \psi_1(\lambda), \ldots, \psi_{N-1}(\lambda)$ which forms a basis of the space $\mathcal{P}_{N-1}$.

The uniquely defined monic orthogonal polynomials can be generated by orthogonalizing the sequence of monomials $1, \lambda, \lambda^2, \dots$ using the Gram-Schmidt method. The sequence of orthonormal polynomials can be obtained by normalizing the sequence of monic orthogonal polynomials. Thus the monic orthogonal polynomials and orthonormal polynomials have the same roots.

Theorem 1.7 (Three-term recurrence)**.** *[6, Theorems 1.27 and 1.29] Let $n < \mathfrak{n}(\omega)$. The sequence of monic orthogonal polynomials $\psi_0, \dots, \psi_n$ is given by a three-term recurrence*

$$\psi_{k+1}(\lambda) = (\lambda - \alpha_k)\psi_k(\lambda) - \delta_k \psi_{k-1}(\lambda), \quad k = 0, \dots, n-1 \tag{1.3.2}$$

where $\psi_{-1}(\lambda) = 0, \psi_0(\lambda) = 1$ and

$$\begin{aligned} \alpha_k &= \frac{\langle \lambda\psi_k, \psi_k \rangle_\omega}{\langle \psi_k, \psi_k \rangle_\omega}, & k &= 0, 1, \dots, n-1 \\ \delta_k &= \frac{\langle \psi_k, \psi_k \rangle_\omega}{\langle \psi_{k-1}, \psi_{k-1} \rangle_\omega}, & k &= 1, 2, \dots, n-1 \\ \delta_0 &= \int_a^b d\omega(\lambda). \end{aligned}$$

The sequence of orthonormal polynomials $\varphi_0, \dots, \varphi_n$ is given by a similar three-term recurrence

$$\sqrt{\delta_{k+1}}\varphi_{k+1}(\lambda) = (\lambda - \alpha_k)\varphi_k(\lambda) - \sqrt{\delta_k}\varphi_{k-1}(\lambda), \quad k < n, \tag{1.3.3}$$

where $\varphi_{-1}(\lambda) = 0, \varphi_0(\lambda) = 1/\sqrt{\delta_0}$.

Consider the special case when $\mathfrak{n}(\omega) = N < \infty$ and suppose that the points of increase $\lambda_i,\ i = 1, \dots, N$ of the distribution function $\omega(\lambda)$ satisfy

$$a < \lambda_1 < \lambda_2 < \cdots < \lambda_N \leq b.$$

Then the previous theorem gives us a sequence of monic orthogonal polynomials $\psi_0, \psi_1, \dots, \psi_{N-1}$ and we can define monic polynomial ψ_N as a result of orthogonalization of the monomial λ^N with respect to the polynomials $\psi_0, \psi_1, \dots, \psi_{N-1}$, i.e., it must hold that

$$\langle \psi_N, \psi_i \rangle_\omega = 0, \quad i = 0, \dots, N-1. \tag{1.3.4}$$

Although $\langle \cdot, \cdot \rangle_\omega$ is an inner product only on the subspace of polynomials of degree less then N, it can be shown that conditions (1.3.4) defines polynomial ψ_N uniquely. Because of the definition of the inner product (1.2.2) it is obvious that with the choice

$$\psi_N = (\lambda - \lambda_1)(\lambda - \lambda_2)\dots(\lambda - \lambda_N)$$

the conditions (1.3.4) are satisfied and thus the monic polynomial defined by the conditions (1.3.4) has roots at the points of increase of the piecewise constant distribution function $\omega(\lambda)$. To sum up, although $\langle \cdot, \cdot \rangle_\omega$ is not an inner product on the space $\mathcal{P}_N$, the polynomials $\psi_0, \dots, \psi_{N-1}, \psi_N$ are monic and they are orthogonal to each other with respect to mapping $\langle \cdot, \cdot \rangle_\omega$.

We have seen that for given distribution function $\omega(\lambda)$ there exists (finite or infinite) sequence of monic polynomials orthogonal to the inner product $\langle\cdot,\cdot\rangle_\omega$. In addition, these monic orthogonal polynomials can be expressed by the three-term recurrence. There is a converse of this theorem. It is usually attributed to Favard but it was known to many mathematicians (e.g. Stieltjes) before.

Theorem 1.8 (Favard's Theorem). *If a sequence (finite or infinite) of monic polynomials $\psi_0, \psi_1, \ldots$ satisfies a three-term recurrence relation such as* (1.3.2) *with real coefficients α_k and with real and positive coefficients δ_k then there exists distribution function $\omega(\lambda)$ such that the polynomials are orthogonal to the inner product $\langle\cdot,\cdot\rangle_\omega$ defined by the Riemann-Stieltjes integral for $\omega(\lambda)$.*

Theorem 1.9 (Properties of the roots). *[20, Lemma 3.2.4] Consider distribution function $\omega(\lambda)$ with finite or infinite number of points of increase. Then the n zeros $\lambda_1^{(n)}, \ldots, \lambda_n^{(n)}$ of the monic orthogonal polynomial ψ_n are distinct and located in the interval $(a, b]$ for $n \leq \mathfrak{n}(\omega)$. If $n < \mathfrak{n}(\omega)$, then the zeros of ψ_n are located in the open interval (a, b).*

Theorem 1.10. *[20, Corollary 3.3.3] Let $\mathfrak{n}(\omega)$ may be finite or infinite, then the roots of two consecutive monic orthogonal polynomials strictly interlace, i.e., for $n < \mathfrak{n}(\omega)$ it holds that*

$$a < \lambda_1^{(n+1)} < \lambda_1^{(n)} < \lambda_2^{(n+1)} < \lambda_2^{(n)} < \cdots < \lambda_n^{(n+1)} < \lambda_n^{(n)} < \lambda_{n+1}^{(n+1)} < b,$$

where $\lambda_i^{(n)}$ are the zeros of ψ_n and $\lambda_i^{(n+1)}$ are the zeros of ψ_{n+1}.

The previous theorem is only a corollary of stronger result [20, Theroem 3.3.1]. We present it here because of its relationship to eigenvalues of Jacobi matrices; see Theorem 1.12.

Theorem 1.11 (Minimalization property of monic orthogonal polynomials). *[6, Theorem 1.24] Let $\psi_0(\lambda), \psi_1(\lambda), \ldots, \psi_n(\lambda)$ be a sequence of the monic orthogonal polynomials, where $n < \mathfrak{n}(\omega)$. The polynomial $\psi_k(\lambda)$ has the smallest norm among the monic polynomials of degree k, where $k \leq n$, i.e.,*

$$\begin{aligned}\psi_k(\lambda) &= \arg\min_{\psi \in \mathcal{M}_k} \|\psi\|_\omega \\ &= \arg\min_{\psi \in \mathcal{M}_k} \|\psi\|_\omega^2 \\ &= \arg\min_{\psi \in \mathcal{M}_k} \int_a^b \psi^2(\lambda)\, d\omega(\lambda),\end{aligned}$$

where $\mathcal{M}_k$ is a set of monic polynomials of degree k.

1.4 Jacobi matrices

The Jacobi matrices represent an important class of matrices. We will see that they can realize the connection orthogonal polynomials and the Lanczos algorithm. Their basic properties can be found in [8, Chapter 3]. In this section we will focus on their connection to orthogonal polynomials.

Definition 1.10 (Jacobi matrix). *Real and symmetric matrix $T_n \in \mathbb{R}^{n\times n}$ is called* **Jacobi matrix** *if it is tridiagonal with positive off-diagonal elements.*

Consider sequence of orthonormal polynomials $\varphi_0, \ldots, \varphi_n$ and coefficients $\{\alpha_i\}_{i=0}^{n-1}$ and $\{\delta_i\}_{i=0}^{n-1}$ from the Theorem 1.7. The coefficients δ_i are positive and thus matrix

$$T_n = \begin{bmatrix} \alpha_0 & \sqrt{\delta_1} & & & \\ \sqrt{\delta_1} & \alpha_1 & \sqrt{\delta_2} & & \\ & \ddots & \ddots & \ddots & \\ & & \sqrt{\delta_{n-2}} & \alpha_{n-2} & \sqrt{\delta_{n-1}} \\ & & & \sqrt{\delta_{n-1}} & \alpha_{n-1} \end{bmatrix}$$

is the Jacobi matrix. Denote by $\Phi_n(\lambda)$ a column vector with orthonormal polynomials $\varphi(\lambda)$ as its entries, i.e.,

$$\Phi_n(\lambda) = [\varphi_0(\lambda), \varphi_1(\lambda), \ldots, \varphi_{n-1}(\lambda)] \,.$$

The three term recurrence (1.3.2) can be written in a matrix form

$$\lambda\Phi_n(\lambda) = T_n\Phi_n(\lambda) + \sqrt{\delta_n}\varphi_n(\lambda)u_n, \tag{1.4.1}$$

where $u_n = [0, \ldots, 0, 1]^T$.

The polynomial $\varphi_n(\lambda)$ has n roots denoted as $\{\lambda_i^{(n)}\}_{i=1}^n$. Putting $\lambda = \lambda_i^{(n)}$ in (1.4.1) gives the following theorem.

Theorem 1.12 (Eigenvalues of the Jacobi matrix). *[6, Theorem 1.31] The zeros $\{\lambda_i^{(n)}\}_{i=1}^n$ of the polynomial $\varphi_n(\lambda)$ are the eigenvalues of the Jacobi matrix T_n and the vectors $\Phi_n(\lambda_i^{(n)})$ are the corresponding eigenvectors.*

Using Theorem 1.8 we can conclude that any Jacobi matrix determines a sequence of polynomials which are orthogonal with respect to certain distribution function. The Cauchy interlacing property formulated as in [8, Theroem 3.3] says that eigenvalues of two consecutive Jacobi matrices strictly interlace each other and it can be viewed as a consequence of Theorem 1.12 and Theorem 1.10. For detailed discussion see [20, Remark 3.3.2].

1.5 Chebyshev polynomials

The Chebyshev polynomials are one of the most important sequences of polynomials. Following [3, Section 3.2.3], we will review and shortly discuss their properties in this section.

Definition 1.11 (Chebyshev polynomials). *The Chebyshev polynomials T_n can be defined as follows.*

$$\mathrm{T}_n(x) = \left\{ \begin{array}{cc} \cos(n \arccos(x)), & x \in [-1, 1] \\ \cosh(n \operatorname{arccosh}(x), & x \notin [-1, 1] \end{array} \right.$$

It is not hard to show that the roots of the Chebyshev polynomials are points

$$x_k = \cos\left(\frac{2k-1}{n}\frac{\pi}{2}\right), \quad k = 1 \ldots n$$

and extremes on $[-1,1]$ *are taken at points:*

$$\widehat{x}_k = \cos\left(\frac{k\pi}{n}\right), \quad k = 0 \dots n$$

and it holds that

$$\mathrm{T}_n(\widehat{x}_k) = (-1)^k. \tag{1.5.1}$$

Theorem 1.13 (Recurrence for Chebyshev polynomials)**.** *[3, Section 3.2.3] The Chebyshev polynomials can be expressed by the following three-term recurrence*

$$\begin{aligned} \mathrm{T}_0(x) &= 1 \\ \mathrm{T}_1(x) &= x \\ \mathrm{T}_{n+1}(x) &= 2x\mathrm{T}_n(x) - \mathrm{T}_{n-1}(x), \quad n \geq 1. \end{aligned} \tag{1.5.2}$$

Proof. This theorem can be considered as a consequence of trigonometric formulas

$$\cos((n+1)\theta) + \cos((n-1)\theta = 2\cos(\theta)\cos(n\theta)$$
$$\cosh((n+1)\theta) + \cosh((n-1)\theta = 2\cosh(\theta)\cosh(n\theta)$$

□

Using the three-term recurrence we obtain

$$\begin{aligned} \mathrm{T}_2(x) &= 2x^2 - 1 \\ \mathrm{T}_3(x) &= 4x^3 - 3x \\ \mathrm{T}_4(x) &= 8x^4 - 8x^2 + 1 \\ \mathrm{T}_5(x) &= 16x^5 - 20x^3 + 5x \\ &\vdots \end{aligned}$$

Theorem 1.14 (The orthogonality of the Chebyshev polynomials I.)**.** *[3, Theorem 4.5.20]*

Define a discrete inner product on the space of polynomials $\mathcal{P}_m$

$$\langle f, g \rangle = \sum_{k=0}^{m} f(x_k) g(x_k),$$

where

$$x_k = \cos\left(\frac{2k+1}{m+1}\frac{\pi}{2}\right), \quad k = 0, \dots, m$$

are the zeros of T_{m+1}.

The sequence of the Chebyshev polynomials $\{\mathrm{T}_k\}_{k=0}^m$ *is orthogonal with respect to this inner product and*

$$\|\mathrm{T}_k(x)\| = \begin{cases} \sqrt{\frac{1}{2}(m+1)} & \text{for } k \neq 0 \\ \sqrt{m+1} & \text{for } k = 0 \end{cases}$$

Theorem 1.15 (The orthogonality of the Chebyshev polynomials II.)**.** *[3, Theorem 4.5.20]*

Define an inner product on the space of polynomials $\mathcal{P}$

$$\langle f, g\rangle_\omega = \int_{-1}^{1} f(x)g(x)\,d\omega(x) = \int_{-1}^{1} f(x)g(x)\omega'(x)\,dx \tag{1.5.3}$$

and define the associated norm

$$\|f\|_\omega = \sqrt{\langle f, f\rangle_\omega}, \tag{1.5.4}$$

where

$$\begin{aligned} \omega(x) &= \frac{2}{\pi}\arcsin(x) \\ \omega'(x) &= \frac{2}{\pi}\frac{1}{\sqrt{1-x^2}}. \end{aligned} \tag{1.5.5}$$

The distribution function ω *is strictly increasing which ensures (see Theorem 1.5) that* (1.5.3) *is really an inner product.*

The sequence of the Chebyshev polynomials $\{\mathrm{T}_k\}_{k=0}^{\infty}$ *is orthogonal with respect to this inner product and*

$$\|\mathrm{T}_k(x)\|_\omega = \begin{cases} 1 & \textit{for } k \neq 0 \\ \sqrt{2} & \textit{for } k = 0 \end{cases}$$

Thus the sequence of polynomials

$$\frac{1}{\sqrt{2}}\mathrm{T}_0, \mathrm{T}_1, \ldots, \mathrm{T}_n, \ldots$$

is orthonormal with respect to the inner product defined by the Riemann-Stieltjes integral with distribution function (1.5.5). If we rewrite the three-term recurrence (1.5.2) as

$$\frac{1}{2}\mathrm{T}_{n+1}(x) = x\mathrm{T}_n(x) - \frac{1}{2}\mathrm{T}_{n-1}(x), \quad n \geq 1$$

then we see that the associated Jacobi matrix has the form

$$\begin{bmatrix} 0 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1/2 & 0 & 1/2 \\ & & & 1/2 & 0 \end{bmatrix}.$$

In compliance with the theory of orthogonal polynomials there is also a sequence of monic orthogonal polynomials

$$\mathrm{T}_0, \mathrm{T}_1, 2^{-1}\mathrm{T}_2, \ldots, 2^{-(n-1)}\mathrm{T}_n, \ldots \tag{1.5.6}$$

We will say that these polynomials are **monic Chebyshev polynomials**.

Theorem 1.16 (Minimalization property I.). *The monic Chebyshev polynomial* $2^{-(n-1)}\mathrm{T}_n$ *minimizes the norm* $\|\cdot\|_\omega$ *among the monic polynomials of degree* n*, i.e.,*

$$2^{-(n-1)}\mathrm{T}_n = \arg\min_{\psi\in\mathcal{M}_n} \|\psi\|_\omega \tag{1.5.7}$$

The monic Chebyshev polynomial $2^{1-n}\mathrm{T}_n(x)$ *is the only polynomial with this minimalization property.*

Theorem 1.16 is only a special case of Theorem 1.11.

Theorem 1.17 (Minimalization property II.). *[3, Theorem 3.2.4] The monic Chebyshev polynomial* $2^{-(n-1)}\mathrm{T}_n$ *minimizes the maximum norm* $\|\cdot\|_\infty$ *among the monic polynomials of degree* n*, i.e.,*

$$2^{-(n-1)}\mathrm{T}_n = \arg\min_{\psi\in\mathcal{M}_n} \|\psi\|_\infty\,, \tag{1.5.8}$$

where the maximum norm is defined as

$$\|f\|_\infty = \max_{x\in[-1,1]} |f(x)|. \tag{1.5.9}$$

The monic Chebyshev polynomial is the only polynomial with this minimalization property.

Proof. Assume, for contradiction, that there exists $p(x) \in \mathcal{M}_n$ such that

$$|p(x)| < |2^{1-n}\mathrm{T}_n(x)| \quad \forall x \in [-1,1].$$

Specially, using (1.5.1) we can write

$$\begin{aligned} p(\widehat{x}_0) &< 2^{1-n}\mathrm{T}_n(\widehat{x}_0) \\ p(\widehat{x}_1) &> 2^{1-n}\mathrm{T}_n(\widehat{x}_1) \\ p(\widehat{x}_2) &< 2^{1-n}\mathrm{T}_n(\widehat{x}_2) \\ &\vdots \end{aligned}$$

We see that the polynomial

$$R(x) = p(x) - 2^{1-n}\mathrm{T}_n(x)$$

must change the sign in every interval (x_k, x_{k+1}), $k = 0, \dots, n-1$. From the continuity of $R(x)$ there is at least one root on every (x_k, x_{k+1}) and thus the polynomial $R(x)$ has at least n roots. However, being a difference of two monic polynomials of degree n, the polynomial $R(x)$ must have degree less then n. Consequently, $R(x) \equiv 0$ which is a contradiction.

The uniqueness of the polynomial $2^{1-n}\mathrm{T}_n$ can be proved in similar way. The only difference is that one must handle with the situation that $R(x)$ may have some zeros in some of the extreme points $\widehat{x}_k$.

□

The previous theorems showed the minimalization property of the Chebyshev polynomials among the class of monic polynomials. However, as we will see several times in this thesis, many problems lead to the minimalization among the class of polynomials which satisfy some constraint condition. Typically, the polynomials have prescribed certain value at some given point ξ.

From now on consider given point

$$\xi \notin [-1, 1]$$

and a constrain condition

$$p(\xi) = 1 \tag{1.5.10}$$

and define the set of polynomials

$$\Pi_n^\xi \equiv \{p(x); \deg p = n,\ p(\xi) = 1\}.$$

The set Π_n^ξ is a set of polynomials of degree n with prescribed value 1 at given point ξ.

Theorem 1.18 (Minimalization property III.)**.** *See e.g. [34, Theorem 5.1]. The scaled Chebyshev polynomial*

$$\frac{\mathrm{T}_n(x)}{\mathrm{T}_n(\xi)}$$

minimizes the maximum norm $\|\cdot\|_\infty$ *among the polynomials of degree n which satisfy the condition* (1.5.10)*, i.e.,*

$$\frac{\mathrm{T}_n(x)}{\mathrm{T}_n(\xi)} = \arg \min_{p \in \Pi_n^\xi} \|p(x)\|_\infty . \tag{1.5.11}$$

The scaled Chebyshev polynomial is the only polynomial with this minimalization property.

Proof. The proof is analogous to the proof of Theorem 1.17. For contradiction we suppose that there exists $p \in \Pi_n^\xi$ with smaller norm. The polynomial

$$R(x) = p(x) - \frac{\mathrm{T}_n(x)}{\mathrm{T}_n(\xi)}$$

has degree n. On the other hand, both polynomials satisfy the condition (1.5.10) and thus $R(\xi) = 0$. The polynomial $R(x)$ of degree n has $n + 1$ roots and thus $R(x) \equiv 0$ which gives the result. □

To sum up, if we specify some constraint condition in order to be able to compare the polynomials, then the scaled Chebyshev polynomials have the smallest maximum norm on the interval $[-1, 1]$. Since

$$\max_{x \in [-1,1]} |\mathrm{T}_n(x)| = 1,$$

we can write

$$\left\| \frac{\mathrm{T}_n(x)}{\mathrm{T}_n(\xi)} \right\| = \frac{1}{|\mathrm{T}_n(\xi)|}.$$

As a consequence of the minimalization property from the last theorem we have the following maximalization property of the Chebyshev polynomials.

Corollary 1.19 (Maximalization property)**.** *The Chebyshev polynomial* T_n *has in every point* $\xi \notin [-1,1]$ *the largest magnitude among the polynomials of degree* n *and of comparable maximum norm, i.e.,*

$$|\mathrm{T}_n(\xi)| > |p(\xi)|, \quad \forall \xi \notin [-1,1], \quad \forall p \in \Omega_n, \quad \textit{where} \tag{1.5.12}$$

$$\Omega_n \equiv \{p(x);\ \deg p = n,\ \|p\|_\infty = 1\} \tag{1.5.13}$$

Proof. Theorem 1.18 implies that for arbitrary polynomial $p(x) \in \Omega_n$ we have inequality

$$\left\| \frac{p(x)}{p(\xi)} \right\| > \left\| \frac{\mathrm{T}_n(x)}{\mathrm{T}_n(\xi)} \right\|.$$

The polynomials $\mathrm{T}_n(x)$, $p(x)$ lie in Ω_n and thus their maximum norm are equal to 1. Consequently;

$$\frac{1}{|p(\xi)|} > \frac{1}{|\mathrm{T}_n(\xi)|},$$

which gives the result. □

We enclose the review of the properties of the Chebyshev polynomials with the statement about approximation of monomial by polynomials of lower degree; see [3, p. 201].

Corollary 1.20 (Best approximation of x^n by the polynomial of lower degree)**.** *Let* $\mathcal{P}_{n-1}$ *be a set of all polynomials of degree less or equal to* $n-1$*. Consider following minimax problem:*

$$\min_{p \in P_{n-1}} \|x^n - p(x)\|_{[-1,1]}.$$

The unique solution is given by polynomial

$$\widehat{p}(x) \equiv x^n - \mathrm{T}_n(x).$$

Proof. Since $\deg(x^n - p(x)) = n$, we want $x^n - p(x) = \mathrm{T}_n(x)$, because $\mathrm{T}_n(x)$ is a solution of minimax problem.

$$\|\widehat{p}(x) - x^n\| = \|\mathrm{T}_n(x)\| = \min_{p \in \Pi_n} \|p(x)\|$$

□

Remark 1.12. *So far, all results were formulated for the interval* $[-1,1]$*. However, arbitrary interval* $[a,b]$ *can be linearly transformed on* $[-1,1]$ *by function*

$$x \longmapsto \frac{2x-b-a}{b-a},$$

so all previous result can be easily reformulated for general $[a,b]$*. For example, let* $\xi \notin [a,b]$*. Then Theorem 1.18 gives*

$$\frac{\mathrm{T}_n\left(\frac{2x-b-a}{b-a}\right)}{\mathrm{T}_n\left(\frac{2\xi-b-a}{b-a}\right)} = \arg \min_{p \in \Pi_n^\xi} \max_{x \in [a,b]} |p(x)|. \tag{1.5.14}$$

For arbitrary interval $[a,b]$ we can define the maximum norm on the space of polynomials $\mathcal{P}$

$$\|p\|_\infty = \max_{x \in [a,b]} |p(x)|, \quad p \in \mathcal{P}$$

The problem of minimalization of the maximum norm $\|\cdot\|_\infty$ over some subspace (e.g. Π_n^ξ, $\mathcal{M}_n$) is in the literature often called minimax problem. The Chebyshev polynomials are said to have minimax property (e.g. see [3]).

1.6 Quadrature rules

Definitions and theorems in this section are from [20, Chapter 3], complex exposition can be found in [4]. Quadrature rules represents one of the most important methods of numerical integration. They are based on the simple idea that the integral is approximated by a linear combination of the values of the integrand, i.e.,

$$\int_a^b f(\lambda)\, d\omega(\lambda) \approx w_1 f(\lambda_1) + w_2 f(\lambda_2) + \ldots + w_n f(\lambda_n)$$

Definition 1.13 (Quadrature rule)**.** *Consider distribution function $\omega(\lambda)$ on the interval $[a, b]$. If f is Riemann-Stieltjes integrable function (e.g. f is continuous on $[a, b]$), then we can write its integral as*

$$I_\omega(f) \equiv \int_a^b f(\lambda)\, d\omega(\lambda) = \sum_{i=j}^{n} w_i f(\lambda_i) + R_n(f), \tag{1.6.1}$$

where $\lambda_i, \ldots, \lambda_n \in (a, b)$ are distinct points called **nodes** *and $w_1, \ldots, w_n$ are called* **weights** *of the n*-**point (mechanical) quadrature rule**

$$I_\omega^n(f) \equiv \sum_{i=1}^{n} w_i f(\lambda_i).$$

The term $R_n(f)$ is called a **remainder** *or error of the quadrature rule.*

When the remainder is equal to zero, the quadrature rule is called **exact**. *We say that the quadrature rule has* **algebraic degree of exactness** *m if $R_n(\lambda^k) = 0$ for $k = 0, \ldots, m$ and $R_n(\lambda^{m+1}) \neq 0$.*

There are several approaches, how to choose the nodes and the weights of the quadrature rule (see [4]) but we will not discuss them here. In this thesis we will deal only with the Gauss-Christoffel quadrature rules because of its relationship with the Jacobi matrices (see Theorem 1.22) and consequently with the Lanczos method and the algorithm of conjugate gradients as will be exploited in detail in Section 3.4.

Definition 1.14 (Lagrange form of the interpolatory polynomial)**.** *For given function f and n distinct points $\lambda_1, \ldots, \lambda_n$, the polynomial $\mathcal{L}_n$ of degree n is called* **interpolatory** *if*

$$\mathcal{L}_n(\lambda_i) = f(\lambda_i), \quad i = 1, \ldots, n.$$

It can be shown that this polynomial is determined uniquely and it can be written in **Lagrange form** *as*

$$\mathcal{L}_n(\lambda) = \sum_{j=1}^{n} f(\lambda_j) l_j(\lambda), \quad \textit{where}$$

$$l_j(\lambda) = \prod_{\substack{i=1 \\ i\neq j}}^{n} \frac{\lambda - \lambda_i}{\lambda_j - \lambda_i}.$$

The polynomial $l_j(\lambda)$ can be also written as

$$l_j(\lambda) = \frac{q_n(\lambda)}{q_n'(\lambda_j)(\lambda - \lambda_j)}, \quad \textit{where} \tag{1.6.2}$$

$$q_n = (\lambda - \lambda_1)(\lambda - \lambda_2)\dots(\lambda - \lambda_n). \tag{1.6.3}$$

Definition 1.15 (n-point interpolatory quadrature)**.** *Given the n distinct nodes $\lambda_i, \dots, \lambda_n \in (a, b)$, the* n**-point interpolatory quadrature** *for approximating the Riemann-Stieltjes integral is defined by*

$$I_\omega^n(f) = \sum_{j=1}^{n} w_j f(\lambda_j), \quad \textit{where}$$

$$w_j = \int_a^b l_j(\lambda).$$

Note that the weights are uniquely determined by the nodes. It can be shown that the algebraic degree of exactness of the n-point interpolatory quadrature rules is at least $n-1$ for arbitrary choice of the nodes $\lambda_1, \dots, \lambda_n$. However, for the special choice of the nodes $\lambda_1, \dots, \lambda_n$, the algebraic degree of exactness can be even higher.

Definition 1.16 (Gauss-Christoffel quadrature rule)**.** *Consider positive integer $n < \mathfrak{n}(\omega)$, the n-point Gauss-Christoffel quadrature rule is interpolatory quadrature with algebraic degree of exactness equal to $2n - 1$.*

As it is stated in the next theorem, this quadrature exists. On the other hand, it can be shown that it is impossible to create a quadrature rule of higher algebraic degree of exactness (consider a square of polynomial q_n with roots at the nodes of the quadrature).

Theorem 1.21 (Nodes and weights of the Gauss-Christoffel quadrature)**.** *[20, Lemma 3.2.5] Consider positive integer $n < \mathfrak{n}(\omega)$, an n-point quadrature rule is the Gauss-Christoffel quadrature rule if and only if it is an interpolatory quadrature with the n nodes given by the roots of the monic orthogonal polynomial $\psi_n(\lambda)$.*

The weights of the Gauss-Christoffel quadrature can be computed using the monic orthogonal polynomial $\psi_n(\lambda)$ (see (1.6.2)*)*

$$w_i = \int_a^b \frac{\psi_n(\lambda)}{\psi_n'(\lambda_i)(\lambda - \lambda_i)} \, d\omega(\lambda)$$

The weights $w_1, \dots, w_n$ are called **Christoffel numbers** and it can be shown that they are strictly positive.

Since the roots of the orthonormal polynomial $\varphi_n(\lambda)$ determine the nodes of the n-point Gauss-Christoffel quadrature rule as well as the eigenvalues of the Jacobi matrix T_n, we can observe a deep connection between the Jacobi matrices and the Gauss-Christoffel quadrature rules, see the following theorem.

Theorem 1.22 (Gauss quadrature and Jacobi matrix)**.** *[20, Theorem 3.4.1] Consider distribution function $\omega(\lambda)$ defined on $[a, b]$ and positive integer $n < \mathfrak{n}(\omega)$. Let $\varphi_0(\lambda), \varphi_1(\lambda), \dots, \varphi_n(\lambda)$ be the orthonormal polynomials associated with the inner product $\langle \cdot, \cdot \rangle_\omega$ given by $\omega(\lambda)$.*

Denote by $\theta_1, \dots, \theta_n$ the eigenvalues and by $z_1, \dots, z_n$ the corresponding eigenvectors of the Jacobi matrix T_n given by the three-term recurrence (1.3.3)*.*

Denote by $\lambda_1, \dots, \lambda_n \in (a, b)$ the nodes and by $w_1, \dots, w_n$ the weights of the n-point Gauss-Christoffel quadrature associated with $\omega(\lambda)$.

Then

$$\theta_i = \lambda_i, \quad (z_i, e_1)^2 = w_i, \quad \textit{for } i = 1 \dots, n.$$

As a consequence, the eigenvalues of the Jacobi matrix T_n are distinct and the first components of its eigenvectors are nonzero.

The proof of Theorem 1.22 is not trivial and it can be done for example with use of the Lanczos algorithm. We mention it here just to sum up the relationship between Jacobi matrix and Gauss-Christoffel quadrature. Logically, the theorem should be written in Section 3.4.

It is worth to note that every quadrature rule with the nodes $\lambda_1, \dots, \lambda_n \in (a, b)$ and positive weights $w_1, \dots, w_n$ can be written as the Riemann-Stieltjes integral for the piecewise constant distribution function $\omega^{(n)}(\lambda)$ (see (1.2.1)), i.e.,

$$\int_a^b f(\lambda)\, d\omega^{(n)}(\lambda) = \sum_{i=1}^{n} w_i f(\lambda_i), \quad \text{where} \tag{1.6.4}$$

$$\omega^{(n)}(\lambda) = \begin{cases} 0 & \text{if } \ a \le \lambda < \lambda_1 \\ \sum_{j=1}^{i} w_j & \text{if } \ \lambda_i \le \lambda < \lambda_{i+1}, \quad i = 1, \dots, n-1 \\ \sum_{j=1}^{n} w_j & \text{if } \ \lambda_n \le \lambda < b \end{cases} \tag{1.6.5}$$

2 Iterative methods for solution of linear systems

The solution of large linear systems of the form

$$Ax = b, \tag{2.0.1}$$

where $A \in \mathbb{R}^{N \times N}$ is a regular matrix and $b \in \mathbb{R}^N$ is a given right-hand side vector, represents one of the most important problems of numerical mathematics. With the importance of the problem corresponds enormous effort of numerical mathematics to design methods which would be able to solve this problem satisfactorily. Through the history, several main approaches were developed.

Direct methods, like LU factorization, are very robust, i.e., they are applicable on many different types of matrices. They are convenient for small and dense matrices. However, they are impractical for large and sparse matrices as they do not preserve the sparsity and have enormous memory requirements consequently.

Iterative methods generate a sequence of approximate solutions which converge to the exact solution. The computation of approximations is essentially based on matrix-vector multiplication and thus it is not essential to store the matrix A. Iterative methods require fewer storage and often also fewer operations then direct methods. We will introduce two main approaches of iterative methods, stationary iterative methods and methods based on Krylov subspaces. Modern multi-grid methods are not included, they are beyond the scope of this thesis.

2.1 Stationary iterative methods

The first ideas of stationary iterative methods for solving linear systems appeared in works of Gauss, Jacobi or Seidel in the 19th century. Term stationary means that the process of generating new approximation does not depend on the iteration step. An important contribution in the field of stationary iterative methods was made by Young in his PhD thesis [35]. He introduced successive overrelaxation method (SOR) which become superior among other stationary iterative methods.

Concept of *matrix splitting* is a nice approach which includes all methods mentioned above. It allows to study necessary and sufficient conditions for convergence and to compare different methods. A lot of mathematicians were interested in analysis of the stationary iterative methods. Nice and well written analysis can be found in [34, 36] or in [13]. The following summary of known results is foremost based on [34], several interesting ideas and comments can be found in [26].

2.1.1 Derivation of several stationary iterative methods

In this section we follow [26, Chapter 4], the convergence analysis is based on the results from [34] and [26]. Consider the linear system (2.0.1) for the regular matrix A with nonzero elements on the main diagonal. The stationary iterative methods generate approximations $x_n = (\xi_1^{(n)}, \ldots, \xi_N^{(n)})$ from the iterative scheme

$$Mx_{n+1} = Lx_n + b, \tag{2.1.1}$$

where

$$A = M - L \tag{2.1.2}$$

is the splitting of the matrix A. It is necessary to construct splittings such that the matrix M is regular and easy to invert. Suppose that the sequence $\{x_n\}_{n=0}^{\infty}$ converges. Then it is easy to see that its limit $\widehat{x}$ is a solution of the problem $Ax = b$. We will define several possible techniques of splitting and review the properties of the associated iterative schemes.

Consider decomposition $A = D - E - F$, where D is a diagonal matrix, and E and F are, respectively, strictly lower and strictly upper triangular matrices.

Since D is a regular matrix, we can define following stationary iterative methods:

Jacobi method

Matrix splitting:

$$M \equiv D, \quad L \equiv E + F$$

Matrix notation:

$$x_{n+1} = D^{-1}(E + F)x_n + D^{-1}b, \tag{2.1.3}$$

Notation for elements:

$$\xi_i^{(n+1)} = -\sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{a_{i,j}}{a_{i,i}} \xi_j^{(n)} + \frac{b_i}{a_{i,i}}. \tag{2.1.4}$$

The Jacobi method is very simple and easy to implement on a computer. It is necessary to save two vectors. The elements of x_{n+1} are computed independently of each other which makes the Jacobi method convenient for parallel computing. Since it holds that

$$(b - Ax_n)_i = b_i - \sum_{\substack{j=1 \\ j \neq i}}^{N} a_{i,j}\xi_j^k - a_{i,i}\xi_i^k,$$

we see that (2.1.4) corrects the i-th component of the residual.

Gauss-Seidel method

Matrix splitting:

$$M \equiv D - E, \quad L \equiv F$$

Matrix notation:

$$x_{n+1} = (D - E)^{-1}Fx_n + (D - E)^{-1}b, \tag{2.1.5}$$

Notation for elements:

$$\xi_i^{k+1} = -\sum_{j=1}^{i-1} \frac{a_{i,j}}{a_{i,i}} \xi_j^{k+1} - \sum_{j=i+1}^{N} \frac{a_{i,j}}{a_{i,i}} \xi_j^k + \frac{b_i}{a_{i,i}}. \tag{2.1.6}$$

The only difference between the Jacobi and the Gauss-Seidel method is that the Gauss-Seidel method uses the first $i-1$ previously computed components of the new approximation to compute the i-th component. Only one vector needs to be stored. For completeness' sake we add that this type of Gauss-Seidel method is called the forward Gauss-Seidel method. The backward variant is defined by $M \equiv D - F, L \equiv E$ and has analogous properties.

Successive Overrelaxation (SOR) method

The SOR approximation x_{n+1} is a weighted mean of the approximation $\hat{x}^{k+1}$ from the Gauss-Seidel method and the SOR approximation x_n from the previous step, i.e.,

$$x_{n+1} = \omega \hat{x}^{k+1} + (1-\omega)x_n,$$

where $\omega \in \mathbb{R}$ is a relaxation parameter. After some manipulation we can write the SOR method using the concept of matrix splitting.
Matrix splitting:

$$M \equiv \frac{1}{\omega}(D - \omega E), \quad L \equiv \frac{1}{\omega}((1-\omega)D + \omega F)$$

Matrix notation:

$$x_{n+1} = (D - \omega E)^{-1}(\omega F + (1-\omega)D)x_n + \omega(D - \omega E)^{-1}b, \tag{2.1.7}$$

Notation for elements:

$$\xi_i^{k+1} = \xi_i^k + \omega\left(-\sum_{j=1}^{i-1}\frac{a_{i,j}}{a_{i,i}}\xi_j^{k+1} - \sum_{j=i+1}^{N}\frac{a_{i,j}}{a_{i,i}}\xi_j^k + \frac{b_i}{a_{i,i}} - \xi_i^k\right). \tag{2.1.8}$$

As in the Gauss-Seidel method, only one vector needs to be stored and both forward and backward variants can be defined. Symmetric SOR (SSOR) is a method which combines these two approaches. One step of SSOR involves one step of each forward and backward SOR. Selecting $\omega = 1$ shows that the Gauss-Seidel method can be considered as a special case of the SOR method.

2.1.2 Analysis of convergence

Using (2.1.1) we can write the iterative scheme

$$x_{n+1} = Gx_n + g, \quad \text{where} \tag{2.1.9}$$

$$G \equiv M^{-1}L, \quad g \equiv M^{-1}b \tag{2.1.10}$$

Definition 2.1 (Convergent iterative scheme). *Iterative scheme* (2.1.9) *is called convergent if and only if the sequence of approximations* $\{x_n\}_{k=0}^{\infty}$ *converges to the solution of linear system* (2.0.1) *for every initial approximation* x^0.

Since every two norms on finite dimensional spaces are equivalent, it is unnecessary to define in which norm we measure the convergence.

Theorem 2.1 (General convergence result). *Iterative scheme* (2.1.1) *is covergent if and only if* $\rho(G) < 1$.

Proof. Let $\rho(G) < 1$. Then from Theorem 1.1 there exists matrix norm $\|\cdot\|_M$ generated by some vector norm $\|\cdot\|_V$, that $\|G\|_M < 1$. Then from (2.1.9) and the identity $\widehat{x} = G\widehat{x} + g$ for the solution $\widehat{x}$ we get

$$\|x_{n+1} - \widehat{x}\|_V = \|G(x_n - \widehat{x})\|_V = \cdots \tag{2.1.11}$$

$$= \left\|G^{k+1}(x^0 - \widehat{x})\right\|_V \leq \left\|(x^0 - \widehat{x})\right\|_V \|G\|_M^{k+1} \overset{k\to\infty}{\longrightarrow} 0 \tag{2.1.12}$$

and thus $\{x_n\}_{k=0}^{\infty}$ converges to the solution $\widehat{x}$ independently on the initial approximation x^0.

Conversely, let $\rho(G) \geq 1$. Then there exists eigenvalue $\lambda \in \mathbb{C}$ and eigenvector $y \in \mathbb{R}^n$ such that $|\lambda| \geq 1$. Then for the special choice $x^0 = \widehat{x} + y$ we get

$$\|x_{n+1} - \widehat{x}\| = \left\|G^k(x^0 - \widehat{x})\right\| = \left\|G^k y\right\| = \left\|\lambda^k y\right\| = \left|\lambda^k\right| \cdot \|y\| \geq \|y\| \;\; \forall k.$$

So we have found x^0 for which the scheme does not converge. That means, that (2.1.1) is not convergent. □

As it will be stated in Section 2.1.4 the rate of the convergence of different iterative schemes can be compared only assymptoticaly, via the spectral radius. Smaller spectral radius means more rapid convergence.

Denote matrices G corresponding to the matrix splitting in the Jacobi and the SOR method as

$$G_J \equiv D^{-1}(E+F),\; G_\omega \equiv (D - \omega E)^{-1}\left(\omega F + (1-\omega)\, D\right). \tag{2.1.13}$$

Perron-Frobenius theory of nonnegative matrices and theory of *regular splitting* allows to formulate some results about the Jacobi and Gauss-Seidel methods. We present brief summary of the most important results.

Theorem 2.2. *Let $G_J \geq 0$. Then the Jacobi method converges if and only if the Gauss-Seidel method converges. In that case, the Gauss-Seidel method converges faster.*

This result is a consequence of the Stein-Rosenberg theorem (see [34, Theorem 3.8]) from 1945 and it can not be generalized. There are matrices, for which converges only one of these methods. In addition, the Jacobi method can be faster than the Gauss-Seidel method, even if both of them are convergent.

Theorem 2.3 (Convergence of diagonally dominant matrices)**.** *Let A be irreducibly or strictly diagonally dominant matrix. Then both Jacobi and Gauss-Seidel method defined above are convergent.*

Definition 2.2 (Regular splitting, weak regular splitting)**.** *Consider splitting $A = M - L$.*
It is called **regular** *if M is regular, $M^{-1} \geq 0$ and $N \geq 0$.*
It is called **weak regular** *if M is regular, $M^{-1} \geq 0$ and $M^{-1}N \geq 0$.*

Theorem 2.4 (Convergence of (weak) regular splittings)**.** *Let $A = M - L$ be the regular or weak regular splitting.*
Then $\rho(M^{-1}L) < 1$ if and only if A is regular and $A^{-1} \geq 0$.

Theorem 2.5 (Comparison of rate of asymptotic convergence)**.** *Let $A = M_1 - L_1 = M_2 - L_2$ be two regular splittings, where $A^{-1} \geq 0$. If at least one of the following: $L_2 \geq L_1 \geq 0$, $M_1^{-1} \geq M_2^{-1}$ is true, then holds*

$$1 > \rho(M_2^{-1}L_2) \geq \rho(M_1^{-1}L_1) \geq 0.$$

For strict inequalities in assumptions we get strict inequalities in the result.

2.1.3 Analysis of SOR

The SOR method defined in (2.1.7) is a special case of more general concept of the *relaxation methods*. Specially, there is a lot of variants of the SOR method which are based on the block partitioning of the matrix A. Varga's analysis based on *p-cyclic* and *"consistently ordered"* matrices. Young's analysis is based on *consistently ordered* matrices and on matrices with *Property $\mathcal{A}$*. There is a difference between Varga's and Young's definition of consistently ordered matrix; see [36, Section 5.8] or [26, Section 4.2.5] for details. However, in our summary we will focus only on the basic SOR method. In this case, the matrix A has Property $\mathcal{A}$ if and only if it is 2-cyclic and the main result Theorem 2.9 and its consequences are the same for both approaches. Full-scale exposition can be found in [34] or [36].

Theorem 2.6 (Kahan (1958)). *Consider G_ω defined in* (2.1.13). *Then*

$$\rho(G_\omega) \geq |\omega - 1| \quad \forall\, \omega \in \mathbb{C}.$$

Consequently; for the convergent SOR scheme the relaxation parameter ω lies in the open interval $(0, 2)$*.*

Theorem 2.7 (Ostrowski (1954)). *Consider symmetric matrix A with decomposition $A = D - E - E^T$ where D is symmetric and positive definite and $(D - \omega E)$ is nonsingular for any $\omega \in [0, 2]$. Then we can define G_ω analogously as in* (2.1.13) *and it holds that $\rho(G_\omega) < 1$ if and only if A is positive definite and $\omega \in (0, 2)$.*

It is worth to notice that E need not to be triangular and D need not to be diagonal as in the standard SOR method.

Corollary 2.8 (Convergence of SOR for SPD matrices). *Consider symmetric matrix A with positive elements on diagonal and let $\omega \in (0, 2)$. Then the SOR method is convergent if and only if A is positive definite.*

Definition 2.3 (Consistent ordering (Young)). *The matrix $A = (a_{i,j})$ of order N is* **consistently ordered** *if for some t there exist disjoint subsets $S_1, \ldots, S_t$ of $W = 1, \ldots, N$ such that $\sum_{k=1}^{t} S_k = W$ and such that if $a_{i,j} \neq 0$ or $a_{j,i} \neq 0$, then $j \in S_{k+1}$ for $j > i$ and $j \in S_{k-1}$ for $j < i$ where S_k contains i.*

Theorem 2.9 (Connection between eigenvalues of SOR and Jacobi matrices). *[26, Theorem 4.7] Let A be a consistently ordered matrix with positive elements on diagonal and let $\omega \neq 0$. Consider equation*

$$(\lambda + \omega - 1)^2 = \lambda \omega^2 \mu^2. \tag{2.1.14}$$

Then holds: $\mu \in \sigma(G_J)$ and λ satisfies (2.1.14) $\Rightarrow \lambda \in \sigma(G_\omega)$
Conversely: $\lambda \in \sigma(G_\omega), \lambda \neq 0$ and μ satisfies (2.1.14) $\Rightarrow \mu \in \sigma(G_J)$*.*

Substituting $\omega = 1$ in (2.1.14) gives

$$\lambda = \mu^2$$

which implies that $\rho(G_1) = \rho^2(G_J)$ and thus the Gauss-Seidel method has better asymptotic rate of convergence then Jacobi method and we know even the ratio of the convergence rates.

This theorem allows to compute an optimal value of the relaxation parameter ω.

Theorem 2.10 (Optimal parameter in SOR)**.** *Consider G_J defined in* (2.1.13) *of a consistently ordered matrix A. Then ω_{opt} defined as*

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\rho^2(G_J)}} = 1 + \left(\frac{\rho(G_J)}{1+\sqrt{1\rho^2(G_J)}}\right) \tag{2.1.15}$$

satisfies that

$$\rho(G_\omega) > \rho(G_{\omega_{opt}}) = \omega_{opt} - 1.$$

The last results show that for consistently ordered matrix with nonzeros on the main diagonal the SOR method should be preferred among other stationary iterative methods since it has better asymptotic rate of convergence.

2.1.4 Properties of the asymptotic convergence

In this part we follow [32, Section 9.1.1]. The analysis of the stationary iterative methods is based on the spectral radius of the matrix G defined by (2.1.9). For the error of the k-th step of the iterative process it is possible to write

$$\frac{\|e_n\|}{\|e^0\|} \leq \left\|G^k\right\| \leq \|G\|^k \tag{2.1.16}$$

and we see that the convergent iterative scheme ($\rho(G) < 1$) ensures the asymptotic convergence. The rate of this asymptotic convergence is given by the spectral radius. Although it holds (see e.g. [32, Section 1.7]) that

$$\lim_{k\to\infty} \left\|G^k\right\|^{1/k} = \rho(G), \tag{2.1.17}$$

the main trouble of all stationary iterative methods is that it can happen that

$$\|G\| > 1 > \rho(G).$$

The limit (2.1.17) justifies that

$$\frac{\|e_n\|}{\|e^0\|} \approx \rho(G)^k$$

for large k, but for smaller k it is more convenient to use

$$\frac{\|e_n\|}{\|e^0\|} \leq \ \|G\|^k,$$

where $\|G\|^k$ can be greater then 1.

The information about the spectral radius gives no control of the behaviour of the iteration process at first iterations. Thus the convergence can be very poor. It can even happen that the convergence curve consists of two parts. The norm of the error can be increasing until it reaches the second, asymptotic, part of the convergence curve, where the norm decreases linearly with the rate given by the spectral radius $\rho(G)$.

2.2 Chebyshev semi-iterative method (CSI)

The idea of the CSI method is to accelerate the convergence of the iterative scheme (2.2.1) using more information in generating new approximation. We will show that this acceleration results in strictly decreasing norm of the error. In this section we will follow [34, Section 5.1].

As we have seen in previous chapter, stationary iterative methods can be written in the form

$$x_{n+1} = Gx_n + g, \tag{2.2.1}$$

where G is $n \times n$ matrix with $\rho(G) < 1$. We will assume that convergent matrix G is symmetric.

2.2.1 Derivation of the CSI method

Let $x_0, \ldots, x_n$ be approximations based on some stationary iterative method of the form (2.2.1). We can define new approximation as a linear combination of approximations $x_0, \ldots, x_n$, i.e.,

$$\tilde{x}_n = \sum_{j=0}^{n} \alpha_j(n) x_j. \tag{2.2.2}$$

Approximation $\tilde{x}_n$ is determined by the coefficients $\alpha_j(n)$ and our goal is to set these coefficients to speed up the convergence.

We want to build a consistent method, i.e., a method in which the substitution of the exact solution to the iterative scheme give no error. Thus we impose restriction

$$\sum_{j=0}^{n} \alpha_j(n) = 1 \tag{2.2.3}$$

on coefficients.

If we denote errors of approximations as $\tilde{e}_n = \tilde{x}_n - \widehat{x}$ and $e_n = x_n - \widehat{x}$ and if we use the restriction (2.2.3) we can write

$$\tilde{e}_n = \sum_{j=0}^{n} \alpha_j(n) x_j - \sum_{j=0}^{k} \alpha_j(n) \widehat{x} = \sum_{j=0}^{k} \alpha_j(k) e_j$$

and using the identity $e_n = G^n e^0$ we finally have

$$\tilde{e}_n = p_n(G) e_0, \text{ where } p_n(\lambda) = \sum_{j=0}^{n} \alpha_j(n) \lambda^j. \tag{2.2.4}$$

The restriction (2.2.3) implies that $p_n(1) = 1$. The ratio of errors is bounded by the norm of $p_n(G)$, i.e.,

$$\frac{\|\tilde{e}_n\|}{\|e_0\|} \leq \|p_n(G)\|. \tag{2.2.5}$$

In order to minimize the norm of the error in the n-th step, we must solve the problem

$$\min_{p \in \Pi_n^1} \|p_n(G)\|, \tag{2.2.6}$$

where Π_n^1 is a set of polynomials of degree n satisfying the restriction (2.2.3).

Cayley-Hamilton theorem tells that $q_N(G) = 0$, where q_N is a characteristic polynomial of matrix G and thus we have convergence in at most N steps.

We assume that G is symmetric and thus we can obtain the identity

$$\|p_n(G)\| = \rho(p_n(G)) = \max_{1 \le i \le N} |p_n(\lambda_i)|, \tag{2.2.7}$$

where $-1 < \lambda_1 \le \cdots \le \lambda_N < 1$ are the eigenvalues of the convergent matrix G.

In general we have no additional information about the spectrum of G and thus we maximize over interval instead of over a discrete set of points. To sum up, in order to make the relative error as small as we can, we deal with the problem

$$\min_{p \in \Pi_n^1} \max_{x \in [a,b]} |p(x)|, \tag{2.2.8}$$

where a, b are bounds for the smallest and the largest eigenvalue.

This is a well-known minimax problem, which has on interval $[-1, 1]$ solution given by Chebyshev polynomials. All we have to do is to transform interval $[a, b]$ onto $[-1, 1]$ and to normalize the Chebyshev polynomial in order to satisfy condition $p_n(1) = 1$, remember (1.5.14). Hence we get

$$\tilde{p}_n(x) = \frac{\mathrm{T}_n(\frac{2x-b-a}{b-a})}{\mathrm{T}_n(\frac{2-b-a}{b-a})} \tag{2.2.9}$$

as a solution of our problem (2.2.8). Theorem 1.18 and Remark 1.12 implies that $\tilde{p}_n(x)$ is a unique solution among all polynomials p_n of degree n satisfying $p_n(1) = 1$.

Assume the knowledge of the spectral radius $\rho(G)$ and set $\rho \equiv \rho(G) = b = -a$. Then (2.2.9) becomes

$$\tilde{p}_n(x) = \frac{\mathrm{T}_n(\frac{x}{\rho})}{\mathrm{T}_n(\frac{1}{\rho})}. \tag{2.2.10}$$

Finally, using (2.2.10), the three term recurrence in Theorem 1.13, (2.2.4), definition of n-th error and the identity $(I - G)x = g$ we can derive the Chebyshev semiiterative method (CSI) with respect to iterative scheme (2.2.1):

$$\tilde{x}_{n+1} = \omega_{n+1}(G\tilde{x}_n - \tilde{x}_{n-1} + g) + \tilde{x}_{n-1}, \tag{2.2.11}$$

where

$$\omega_{m+1} := \frac{2\mathrm{T}_m(1/\rho)}{\rho \mathrm{T}_{\mathrm{m+1}}(1/\rho)} = 1 + \frac{\mathrm{T}_{m-1}(1/\rho)}{\mathrm{T}_{m+1}(1/\rho)}, \tag{2.2.12}$$

where the second equality can be proved.

Although $\tilde{x}_{n+1}$ is defined as a sum of $n + 1$ approximations, we have shown it is enough to store two previous approximations $\tilde{x}_n$ and $\tilde{x}_{n-1}$, we even do not need to compute $x_0 \dots x_{n+1}$.

2.2.2 Characteristics and properties of CSI

Theorem 2.11 (Decay of $\|\tilde{p}_n(G)\|$)**.** *Polynomials defined in* (2.2.10) *have strictly decreasing matrix norm with increasing n.*

Proof. Eigenvalues of symmetric matrix are real and G is convergent, so there exists i such that $|\lambda_i| = \rho$. In addition, from definition of the Chebyshev polynomials we know that $|\mathrm{T}_m(\pm 1)| = 1$ and thus using (2.2.7) and (2.2.10) gives

$$\begin{aligned}\|\tilde{p}_m(G)\| &= \max_{1\le i\le N}\left|\frac{\mathrm{T}_m(\lambda_i/\rho)}{\mathrm{T}_m(1/\rho)}\right| \\ &= \frac{1}{\mathrm{T}_m(\frac{1}{\rho})}.\end{aligned}$$

Definition of the Chebyshev polynomials outside the interval $[-1, 1]$ now implies the result, $\|\tilde{p}_n(G)\|$ is decreasing for increasing n. □

The parameters of the CSI method can be also computed using following relations:

$$\omega_{n+1} = \frac{1}{1-\left(\frac{\rho^2\omega_n}{4}\right)},\ n \ge 2,\ \omega_1 = 1,\ \omega_2 = \frac{2}{2-\rho^2}.$$

We have presented method whose convergence is not based only on the spectral radius. In contrast with stationary iterative methods we have control of the behaviour in every step and the norm of the error is decreasing in every step.

In the case when $G = G_J$, i.e., when the iterative scheme (2.2.1) corresponds to the Jacobi method and it is consistently ordered, it is possible to relate the CSI method with the optimal parameter ω_{opt} from the SOR method. It can be shown that

$$\begin{aligned}\|\tilde{p}_n(G)\| &= \frac{2(\omega_{opt}-1)^{n/2}}{1+(\omega_{opt}-1)^n} \\ \rho(\tilde{p}_n(G)) &\overset{n\to\infty}{\longrightarrow} \sqrt{\rho(G_{\omega_{opt}})} = \sqrt{\omega_{opt}-1} \\ \lim_{n\to\infty}\omega_n &= \omega_{opt}.\end{aligned}$$

2.2.3 Comparison of SOR and CSI

The SOR and the CSI method can be considered as very similar methods. The CSI method requires storage of two vectors, the SOR method needs just one vector to be stored. Although the methods are based on different assumptions, it is possible to modify the problem in order to compare them. The comparison of those methods was done by Golub and Varga in 1961. The paper can be found in [7] and we follow it here. For more details see also [34, Section 5.2].

Consider simple iterative method

$$x_{n+1} = Gx_n + g$$

with convergent and symmetric matrix G. We know that under these assumptions the CSI method can be defined. We know that the rate of convergence of the CSI method is determined by polynomial $\tilde{p}_m(x)$, shifted and scaled Chebyshev polynomial.

In order to be able to apply the results of SOR analysis we consider expanded linear system

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 & G \\ G & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} g \\ g \end{pmatrix}$$

Solution of this linear system is the same for x and y and it is also the solution of the original problem. Matrix

$$J = \begin{pmatrix} 0 & G \\ G & 0 \end{pmatrix}$$

has the same spectral radius as the original matrix and is cyclic and consistently ordered. Thus we can apply the results of the SOR theory and determine the optimal relaxation parameter

$$\omega_{opt} = \frac{1}{\sqrt{1 - \rho^2(G)}}.$$

Now we see that the class of matrices convenient for the SOR method is larger than it seemed before. However, we must solve problem of double dimension which might be a serious limitation.

Defining $z_{2n} = x_n,\ z_{2n+1} = y_n$ we obtain iterative scheme of form:

$$z_{n+1} = \omega(Gz_n - z_{n-1} + g) + z_{n-1}, \tag{2.2.13}$$

where ω is the relaxation parameter.

We see that (2.2.13) has the same form as (2.2.11). The only difference is that relaxation parameter is constant for this method. It is easy to see that error of this method is again determined by some polynomial $r_n(G)$ and initial error e_0. It holds that $\|r_n(G)\| > \|\tilde{p}_n(G)\|$ for all $n \geq 2$. However $\lim_{n\to\infty} \|r_n(G)\| = \lim_{n\to\infty} \|\tilde{p}_n(G)\|$. That means, that CSI method should be preferred, since it reduce error more effectively. Requirement of storage of additional vector is not substantial. The asymptotic rate of convergence is not improved.

As it is showed in [34, Section 5.3], similar results can be obtained in the case, where symmetric and convergent matrix G can be written as

$$G = \begin{pmatrix} 0 & F \\ F^T & 0 \end{pmatrix}$$

In this case there is no problem in defining SOR. Detailed analysis of the CSI method reveals some cyclic character and thus we obtain so-called *cyclic Chebyshev semiiterative method.* The comparison of these methods gives the same results. The asymptotic rate of convergence is the same for both methods, but cyclic CSI reduces the norm of error more effectively.

2.3 Iterative methods based on Krylov subspaces

Krylov subspace methods represents one of the most important techniques for solving large linear systems with matrix of coefficients $A \in \mathbb{R}^{N\times N}$. In many applications we do not have A explicitly and the only operation which is easy to execute is a matrix-vector multiplication. Krylov subspace methods are convenient for such problems. In addition, they can be characterized by projection process and thus, assuming exact arithmetic, they find a solution in at most N steps.

In this section we follow especially [20, Chapter 2], for comparison see e.g. [26, Chapter 5–7]. We will describe general projection framework and see, how Krylov subspaces come naturally to play.

2.3.1 Approximation by projections

Let $A \in \mathbb{R}^{N\times N}, b \in \mathbb{R}^N$. General projection method constructs a sequence of approximations given by

$$x_n \in x_0 + \mathcal{S}_n, \quad r_n \perp \mathcal{C}_n \tag{2.3.1}$$

where x_0 is the initial approximation and $\mathcal{S}_n$ is a subspace of dimension n called *search space*. Since we have n degrees of freedom to determine x_n, we need n constraints. These constraints are imposed as orthogonality conditions on residual $r_n = b - Ax_n$. Residual must be orthogonal to the n-dimensional subspace $\mathcal{C}_n$ which is called *constraints space.*

There exists vector $z_n \in \mathcal{S}_n$ such that we can write

$$\begin{aligned} x_n &= x_0 + z_n \\ e_0 &= e_n + z_n \\ r_0 &= r_n + Az_n, \end{aligned}$$

where $e_n = x - x_n$ is the error of the approximation.

Projection process is called to be *well defined* if search and constraints spaces satisfy

$$\mathbb{R}^N = A\mathcal{S}_n \oplus \mathcal{C}_n^\perp. \tag{2.3.2}$$

Because of (2.3.2) there is uniquely defined decomposition of initial residual $r_0 = r_0|_{A\mathcal{S}_n} + r_0|_{\mathcal{C}_n^\perp}$. Then $Az_n = r_0|_{A\mathcal{S}_n}$ is a projection of r_0 on $A\mathcal{S}_n$ orthogonal to $\mathcal{C}_n^\perp$.

Let columns of matrices S_n and C_n creates basis of spaces $\mathcal{S}_n$ and $\mathcal{C}_n$. Then for n-th approximation holds

$$x_n = x_0 + S_n y_n. \tag{2.3.3}$$

After some manipulation and using orthogonality condition $C_n^T r_0 = 0$ we see that y_n is uniquely defined as solution of n dimensional problem

$$y_n = (C_n^T A S_n)^{-1} C_n^T r_0. \tag{2.3.4}$$

Invertibility of $(C_n^T A S_n)$ is equivalent to condition (2.3.2). Therefore, generating approximations x_n can be understood as solving smaller - projected - linear system for y_n.

Residual can be expressed as

$$\begin{aligned} r_n &= (I - P_n) r_0 \\ P_n &= A S_n (C_n^T A S_n)^{-1} C_n^T. \end{aligned}$$

Since $P_n^2 = P_n$, P_n represents a projector. For $A\mathcal{S}_n = \mathcal{C}_n$ we call it *orthogonal projector*, for $A\mathcal{S}_n \neq \mathcal{C}_n$ we consider it as *oblique projector*. This terminology is in agreement with [20], but differs from that in [26]. Reasons for this terminology are commented in [20]. Following observation will give us reason to consider Krylov subspaces to be the search spaces.

Lemma 2.12. *Suppose* (2.3.2), $r_0 \in \mathcal{S}_n$ *and* $A\mathcal{S}_n = \mathcal{S}_n$*. Then* $r_n = 0$.

Proof. $r_0 = r_n + P_n r_0$ represents a decomposition of r_0 into $P_n r_0 \in A\mathcal{S}_n$ and $r_n \in \mathcal{C}_n^\perp$. From the assumptions we get $r_0 = P_n r_0$ which implies $r_n = 0$. □

Definition 2.4 (Krylov subspace)**.** *Let* $A \in \mathbb{R}^{N\times N}$, $v \in \mathbb{R}^N$*. The subspace*

$$\mathcal{K}_k(A, v) = \operatorname{span}(v, Av, A^2v, \ldots, A^{k-1}) \tag{2.3.5}$$

is called Krylov subspace.

There exists uniquely defined $d \leq N$ such that $v, Av, \ldots, A^{d-1}v$ are linearly independent and $A^d v$ can be expressed as a linear combination of that d independent vectors. It can be shown, that $\mathcal{K}_d(A, v) = \mathcal{K}_k(A, v)$ for every $k \geq d$ and that $\mathcal{K}_d(A, v)$ is A*-invariant* for a regular matrix A. It follows quite easily that this integer d equals to the grade of v with respect to A which is defined as a degree of minimal polynomial of v with respect to A, i.e., it is a positive integer d such that there exists a polynomial p of degree d satisfying

$$p(A)v = 0$$

and such that there is no polynomial of lower degree with this property.

Corollary 2.13. *Let A be a regular matrix, set $\mathcal{S}_n = \mathcal{K}_n(A, r_0)$ and suppose that* (2.3.2) *holds. Then $r_d = 0$ and we have found an exact solution in d iterations.*

Proof. Because of linear independence of $r_0, Ar_0, \ldots, A^{d-1}r_0$ holds

$$r_0 \in \mathcal{S}_0 \subset \mathcal{S}_1 \subset \ldots \subset \mathcal{S}_d = A\mathcal{S}_d,$$

where the last equality results from the A-invariancy of Krylov subspace $\mathcal{K}_d(A, r_0)$. Now we see that assumptions of Lemma 2.12 are satisfied, so $r_d = 0$, i.e., $Ax_d = b$. □

Theorem 2.14 (Sufficient condition for well defined projection process)**.** *Suppose that A is symmetric and positive definite and that $\mathcal{C}_n = \mathcal{S}_n$. Then*

$$\mathbb{R}^N = A\mathcal{S}_n \oplus \mathcal{C}_n^\perp \quad ,i.e.,$$

the projection process is well defined.

Proof. Since the subspaces $A\mathcal{S}_n$ and $\mathcal{S}_n$ have dimensions n and $N-n$ it is sufficient to show that the subspaces have trivial intersection. Consider $z \in A\mathcal{S}_n \cap \mathcal{S}_n^\perp$. Then there exists $y \in \mathcal{S}_n$ such that $z = Ay$. However, because z lies also in the orthogonal complement of $\mathcal{S}_n$, it holds that

$$0 = y^T z = y^T Ay.$$

The matrix A is SPD and thus $y = 0$ which implies that also $z = 0$ and the proof is complete. □

3 Method of Conjugate Gradients (CG)

Method of Conjugate Gradients is one of the best known iterative methods for solving systems of linear equations

$$Ax = b \tag{3.0.1}$$

with symmetric and positive definite (SPD) matrix $A \in \mathbb{R}^{N \times N}$ and the right-hand side $b \in \mathbb{R}^N$. The CG method is optimal in a sense that it minimizes the energy norm of the error over given Krylov subspace. As other Krylov subspace methods, the algorithm does not require storage of A.

The method was introduced by Hestenes and Stiefel [14] in 1952 and the authors were from the very beginning aware of very interesting interconnections between the method of conjugate gradients and other areas of mathematics. The original paper reveals the relationship to continued fractions, orthogonal polynomials, Riemann-Stieltjes integral and the Gauss-Christoffel quadrature. They also understood that rounding errors influence the theoretical properties of the CG method and thus they formulated it as an iterative method, although it ideally gives the solution in finite number of steps.

There is a vast literature about the method of conjugate gradients. The exposition of this chapter is based mainly on [20, 21] and many interesting comments and results can be found in [31, 8, 26].

3.1 Derivation of CG through the minimalization of functional

There exist several approaches, how to derive the method of conjugate gradients. Here we will use that one which is based on the minimalization of the quadratic functional

$$F(x) = \frac{1}{2} x^* Ax - x^* b \tag{3.1.1}$$

and we follow [20, Section 2.5.3]. For the SPD matrix A is the minimalization of (3.1.1) equivalent to the solution of the equation

$$\nabla F(x) = Ax - b = 0.$$

Because A is SPD, we can define an inner product

$$(x, y)_A := x^T Ay \tag{3.1.2}$$

and associated norm (called A-norm or **energy norm**)

$$\|x\|_A := \sqrt{(x, x)_A}.$$

Vectors which are orthogonal with respect to the inner product $(\cdot, \cdot)_A$ will be called **A-orthogonal** in the following text.

Let x be the minimum of the functional (3.1.1). For arbitrary approximation x_n we can write

$$\begin{aligned} F(x_n) &= \tfrac{1}{2}(x_n, x_n)_A - x_n^T b \\ &= \tfrac{1}{2}\left\|x - x_n\right\|_A^2 - \tfrac{1}{2}\left\|x\right\|_A^2 + (x_n, x)_A - x_n^T b \\ &= \tfrac{1}{2}\left\|x - x_n\right\|_A^2 - \tfrac{1}{2}\left\|x\right\|_A^2 + x_n^T(Ax - b) \\ &= \tfrac{1}{2}\left\|x - x_n\right\|_A^2 - \tfrac{1}{2}\left\|x\right\|_A^2 . \end{aligned}$$

That demonstrates the fact that the minimalization of the functional $F(y)$ over some subspace S is the same as the minimalization of the energy norm $\|.\|_A$ of vector $x - y$ over the same subspace S. This it is more convenient to measure the distance of the approximation x_n to the solution x in the energy norm rather then in the Euclidean norm.

As we have an approximation x_n, it is natural to create next approximation in a form

$$x_{n+1} = x_n + \alpha_n p_n, \tag{3.1.3}$$

where p_n is carefully chosen direction vector and the coefficient α_n is settled to minimize $\|x - x_{n+1}\|_A$ along the line $x - x_n - \alpha p_n$. Using properties of the inner product we can write

$$\left\|x - x_{n+1}\right\|_A^2 = \left\|x - x_n\right\|_A^2 - 2\alpha(A(x - x_n), p_n) + \alpha^2(Ap_n, p_n)$$

and derivation with respect to α determines the point of extreme

$$\alpha_n = \frac{(r_n, p_n)}{(p_n, p_n)_A}, \tag{3.1.4}$$

where $r_n = A(x - x_n)$ is the residual. From the equation (3.1.3) follow the formulas for the residuals and for the errors $e_n = x - x_n$

$$r_{n+1} = r_n - \alpha_n A p_n \tag{3.1.5}$$

$$e_{n-1} = e_n + \alpha_{n-1} p_{n-1} \tag{3.1.6}$$

Immediate consequence of the choice of the parameter α_n is the orthogonality between the residual and the direction vector, i.e.,

$$(r_{n+1}, p_n) = (r_n - \alpha_n A p_n, p_n) = (r_n, p_n) - \frac{(r_n, p_n)}{(p_n, p_n)_A}(p_n, p_n)_A = 0. \tag{3.1.7}$$

Geometrically, it says that the gradient of $F(x)$ at x_{n+1} is orthogonal to a surface of such y that $F(y) = F(x_{n+1})$. Direction vector p_n is a tangent of this surface.

For better insight into the CG method it is useful to use (3.1.5), definition of α_n (3.1.4), identity $r_n = Ae_n$ and the definition of the inner product (3.1.2) to write

$$e_n = e_{n-1} - \frac{(e_{n-1}, p_{n-1})_A}{(p_{n-1}, p_{n-1})_A} p_{n-1}. \tag{3.1.8}$$

Now we see that the error e_n can be viewed as the A-orthogonalization of the last error vector e_{n-1} against the direction vector p_{n-1}. The A-orthogonality of the components e_n and $\alpha_{n-1} p_{n-1}$ can be shown easily by a computation or we can refer to general properties of projections on Hilbert spaces. The A-orthogonality allows us to use the Pythagorean Theorem

$$\|e_{n-1}\|_A^2 = \|e_n\|_A^2 + \alpha_{n-1}^2 \|p_{n-1}\|_A^2 \tag{3.1.9}$$

The repetitive use of the equation (3.1.6) gives the expansion

$$e_0 = e_n + \sum_{j=1}^{n} \alpha_{j-1} p_{j-1} \tag{3.1.10}$$

which says that the error of the initial approximation is approximated by a linear combination of the direction vectors $\{p_j\}_{j=0}^{k-1}$. Error of this approximation is expressed by the error e_n. The repetitive use of the equation (3.1.9) gives

$$\|e_0\|_A^2 = \|e_n\|_A^2 + \sum_{j=0}^{n-1} \alpha_j^2 \|p_j\|_A^2 . \tag{3.1.11}$$

It is worth to notice that this identity does not follow from the expansion (3.1.10), we do not know anything about the inner products $(p_i, p_j)_A$ of $\{p_j\}_{j=0}^{k-1}$, yet. Also it should be stressed that the expansion (3.1.10) and the identity (3.1.11) holds for arbitrary choice of $\{p_j\}_{j=0}^{n-1}$.

Now we will derive, how to choose the direction vectors p_n. In a method of the steepest descent, the choice is $p_n \equiv r_n$, so in every step we search for the next approximation in a direction of the steepest descent $(-\nabla F(x_n))$. The convergence of this method is guaranteed but can be very poor. The main reason for the poor convergence is that in every step we use the information only from the last iteration and thus we minimalize just over one-dimensional subspace. In order to minimalize over subspaces of larger dimension, the direction vector p_n must combine information from several iteration steps. Very simple choice is to add an information about the previous direction vector p_{n-1} and to compute

$$p_n = r_n + \beta_n p_{n-1}. \tag{3.1.12}$$

We have added as little as we can, but we will see that it is good enough.

The iteration process can stop only if $p_n = 0$ or $\alpha_n = 0$. Independently on the choice of the parameter β_n, the orthogonality between p_{n-1} and r_n (see (3.1.7)) gives

$$(p_n, r_n) = (r_n, r_n) = \|r_n\|^2 \tag{3.1.13}$$

and in both cases of possible breakdown we finally get $(r_n, r_n) = 0$ which is equivalent to $r_n = 0$. That ensures, that the iteration process will stop if and only if we have found an exact solution.

For a moment let assume, that $\{p_j\}_{j=0}^{n-1}$ are A-orthogonal. Then

$$e_n = e_0 - \sum_{j=1}^{n} \alpha_{j-1} p_{j-1} \tag{3.1.14}$$

represents the A-orthogonal decomposition of the initial error e_0. Consequently, $\|e_n\|_A$ is minimal over all possible approximations x_n in the n-th dimensional subspace generated by the direction vectors $p_0, \ldots, p_{n-1}$, i.e.,

$$\|x - x_n\|_A = \min_{y \in x_0 + \operatorname{span}\{p_0, \ldots, p_{n-1}\}} \|x - y\|_A . \tag{3.1.15}$$

Moreover, the A-orthogonality of the direction vectors implies that $p_N = 0$ and thus the iteration process finds the exact solution in at most N steps. This result is important, however, the same importance should be given to the fact that this result assumed exact arithmetic. In practical computations it is sometimes necessary to run more iterations to attain sufficient accuracy of the approximation. For the details of the numerical behaviour of the CG method see Section 3.7 and references given there.

We have seen that for the A-orthogonal vectors p_i the iteration process in the n-th step minimizes the energy norm over n-dimensional subspace. However, there is only one degree of freedom, the coefficient β_n, so we can ensure the A-orthogonality just between two vectors. We will determine β_n in order to get $(p_n, p_{n-1})_A = 0$. From the definition of the direction vector p_n it is easy to see that

$$\beta_n = -\frac{(r_n, p_{n-1})_A}{(p_{n-1}, p_{n-1})_A}. \tag{3.1.16}$$

The elegance of the CG method lies in the fact that this *local* A-orthogonality implies the *global* A-orthogonality; see the following theorem.

Theorem 3.1 (Properties of r_k, p_k defined in CG). *For an iterative process defined above holds that*

$$\begin{aligned}(p_k, p_j)_A &= 0 \quad \textit{for} \quad j \neq k \\ (r_k, r_j) &= 0 \quad \textit{for} \quad j \neq k\end{aligned}$$

Proof. Proof is based on induction and can be found in many publications. We can refer to the original paper of Hestenes and Stiefel [14, Theorem 5:1]. □

Thus we see, that with the coefficients β_k given by (3.1.16) the assumption of A-orthogonality is satisfied and thus the minimalization property (3.1.15) is guaranteed. Finally, using (3.1.5) and (3.1.13) we can express the coefficients α_k and β_k in a form which is more convenient for the practical computation.

$$\alpha_k = \frac{(r_k, p_k)}{(p_k, p_k)_A} = \frac{\|r_k\|^2}{\|p_k\|_A^2} \tag{3.1.17}$$

$$\begin{aligned}\beta_k &= -\frac{(r_k, p_{k-1})_A}{(p_{k-1}, p_{k-1})_A} = -\frac{r_k^* A p_{k-1}}{p_{k-1}^* A p_{k-1}} \\ &= -\frac{r_k^*(r_{k-1} - r_k)}{\alpha_{k-1}} \frac{\alpha_{k-1}}{p_{k-1}^*(r_{k-1} - r_k)} = -\frac{(r_k, r_{k-1} - r_k)}{(p_{k-1}, r_{k-1} - r_k)} = \frac{\|r_k\|^2}{\|r_{k-1}\|^2}\end{aligned} \tag{3.1.18}$$

Combining the equations (3.1.3), (3.1.12), (3.1.17) and (3.1.18) gives the implementation of the CG method stated in Algorithm (1).

Algorithm (1): The CG method

```
Input: SPD matrix $A \in \mathbb{R}^{N \times N}$, vector $b \in \mathbb{R}^N$, initial approximation
$x_0$, stopping criterion.
Output: Approximation $x_n$ of the exact solution of $Ax = b$.
    Initialization:  $r_0 = b - Ax_0,\ p_0 = r_0$.
```

```
For $k = 1, 2, \dots$
    $\alpha_{k-1} = \frac{(r_{k-1}, r_{k-1})}{(r_{k-1}, r_{k-1})_A}$
    $x_k = x_{k-1} + \alpha_{k-1} p_{k-1}$
    $r_k = r_{k-1} - \alpha_{k-1} A p_{k-1}$
  Stop when the stopping criterion is satisfied.            (3.1.19)
    $\beta_k = \frac{(r_k, r_k)}{(r_{k-1}, r_{k-1})}$
    $p_k = r_k + \beta_k p_{k-1}$
End
```

3.2 CG viewed through the projection process, connection to the Krylov subspaces

If not specified differently, we assume here and in the following sections that CG computation does not stop in iterations $1, \dots, n$. The relation between CG and Krylov subspaces is mentioned in many publications (see e.g. [20]) and the following lemma is often left as an easy exercise.

Lemma 3.2. *The residuals $r_0, \dots, r_n$ form an orthogonal basis of the Krylov subspace $\mathcal{K}_{n+1}(A, r_0)$ and the direction vectors $p_0, \dots, p_n$ form an A-orthogonal basis of the Krylov subspace $\mathcal{K}_{n+1}(A, r_0)$.*

Proof. Because of the orthogonality (resp. A-orthogonality) of the residuals (resp. directions vectors), the vectors $r_0, \dots, r_n$ (resp. $p_0, \dots, p_n$) are linearly independent and thus the linear span of these polynomials is a subspace of dimension $n+1$. Consequently, in order to show that the vectors form a basis of the Krylov subspace $\mathcal{K}_{n+1}(A, r_0)$ it is sufficient to show that

$$r_k \text{ (resp. } p_k) \in \mathcal{K}_{k+1}(A, r_0), \quad k = 0, \dots, n, \tag{3.2.1}$$

since it is obvious that $\mathcal{K}_{k+1}(A, r_0) \subset \mathcal{K}_{n+1}(A, r_0)$ for $k \leq n$. Relations (3.2.1) can be easily proved using induction and the definitions of the residual r_k and the direction vector p_k. For $k = 0$ the relation (3.2.1) is satisfied trivially. Assume that $r_{k-1},\ p_{k-1} \in \mathcal{K}_k(A, r_0)$. Then $Ap_{k-1} \in A\mathcal{K}_k(A, r_0) \subset \mathcal{K}_{k+1}(A, r_0)$ and thus

$$r_k = \overbrace{r_{k-1}}^{\in \mathcal{K}_k(A,r_0)} - \overbrace{\alpha_{k-1} A p_{k-1}}^{\in \mathcal{K}_{k+1}(A,r_0)} \in \mathcal{K}_{k+1}(A, r_0).$$

Consequently,

$$p_k = \overbrace{r_k}^{\in \mathcal{K}_{k+1}(A,r_0)} - \overbrace{\beta_k p_{k-1}}^{\in \mathcal{K}_k(A,r_0)} \in \mathcal{K}_{k+1}(A, r_0).$$

Because the Krylov subspace $\mathcal{K}_{n+1}(A, r_0)$ has dimension $n+1$, we have proved that

$$\operatorname{span}(r_0, \dots, r_n) = \mathcal{K}_{n+1}(A, r_0) = \operatorname{span}(p_0, \dots, p_n). \tag{3.2.2}$$

□

This observation allows us to rewrite some of the equations from the derivation of CG in a language of Krylov subspaces. From the definition of the approximation x_{k+1} (see (3.1.3)) and the last lemma it is obvious that

$$x_{k+1} \in x_0 + \mathcal{K}_{k+1}(A, r_0). \tag{3.2.3}$$

The minimalization property (3.1.15) gives

$$\|x - x_{k+1}\|_A = \min_{y \in x_0 + \mathcal{K}_{k+1}(A, r_0)} \|x - y\|_A \,, \quad \text{i.e.,} \tag{3.2.4}$$

the approximation x_{k+1} minimizes the energy norm of the error over the Krylov subspace $\mathcal{K}_{k+1}(A, r_0)$. The last lemma also allows to express the orthogonality of the residuals as

$$r_{k+1} \perp \mathcal{K}_{k+1}(A, r_0), \quad \forall k \leq n. \tag{3.2.5}$$

These results goes hand in hand with the general projection method defined in Subsection 2.3.1 used for the symmetric and positive definite matrix A with the Krylov subspace $\mathcal{K}_k(A, r_0)$ considered as the search and constraint space (compare (3.2.3) and (3.2.5) with (2.3.1)). From Theorem 2.14 we know, that the projection process is well defined.

3.3 Relationship between CG and the Lanczos algorithm

The Lanczos and Arnoldi algorithms are often used in methods which compute approximations of some eigenvalues of the matrix A and they are tools for computing orthonormal basis of Krylov subspaces $\mathcal{K}_k(A, v)$. The naive basis $v, Av, A^2v, \ldots, A^{k-1}v$ is typically poorly conditioned, and in practical computation, the vectors eventually become linearly dependent. Recall the classical power method – method for approximation of the eigenvector corresponding to the eigenvalue with the largest absolute value.

The Lanczos and CG algorithms are closely linked. We will show that the Lanczos algorithm can be viewed through the projection process and thus that it also determines the CG approximations. We will also derive formulas between the coefficients of both algorithms. We will also mention that the CG method can be even derived from the Lanczos algorithm; for details see [20, Section 2.5], [26, Section 6.7] or [10, Section 9.3].

The exposition about the Lanczos and Arnoldi algorithms is based on [20, Section 2.4], introduction to the Lanczos method and the explanation of the relationship between CG and Lanczos algorithm is based on [21].

The Arnoldi algorithm

The Lanczos algorithm was introduced by Cornelius Lanczos in 1950, the Arnoldi algorithm was introduced later, in 1951 by Walter E. Arnoldi. However, for the purposes of the exposition it is more convenient to describe the Arnoldi algorithm first.

The Arnoldi algorithm can be viewed as the Gram-Schmidt orthogonalization process applied to natural basis of Krylov subspace $(v, Av, A^2v, \ldots, A^{n-1}v)$. It starts with vector $v_1 = v/\|v\|$ and in the k-th step, the vector v_k is generated as the orthogonalization of the vector Av_{k-1} against previous vectors $v_1, \ldots, v_{k-1}$.

Orthogonality of the vectors $v_1, \ldots, v_k$ can be easily proved by induction as well as the fact that they generate $\mathcal{K}_{k-1}(A, v)$. Since each vector is normalized, vectors $v_1, \ldots, v_k$ create an orthonormal basis of the Krylov subspace $\mathcal{K}_{k-1}(A, v)$. The orthogonalization process can terminate only if

$$Av_d \in \operatorname{span}(v_1, v_2, \ldots, v_d),$$

which means that $\operatorname{span}(v_1, v_2, \ldots, v_d)$ become A-invariant. Since

$$\operatorname{span}(v_1, v_2, \ldots, v_d) = \mathcal{K}_{d-1}(A, v),$$

we see that d is the grade of v with respect to A.

Let $V_k \equiv [v_1, \ldots, v_k]$ be a matrix with vectors $v_1, \ldots, v_k$ as a columns and let

$$H_k \equiv \begin{bmatrix} h_{1,1} & h_{1,2} & \ldots & h_{1,k} \\ h_{2,1} & h_{2,2} & \ldots & h_{2,k} \\ & \ddots & \ddots & \vdots \\ & & h_{k,k-1} & h_{k,k} \end{bmatrix}$$

be an upper Hessenberg matrix of coefficients. We can rewrite the Arnoldi algorithm in terms of matrices. After k steps of the Arnoldi algorithm we have

$$AV_k = V_k H_k + h_{k+1,k} v_{k+1} u_k^T, \tag{3.3.1}$$

where u_k represents the last column of the k-dimensional identity matrix. Multiplication of (3.3.1) with V_k^T from the left gives

$$V_k^T A V_k = H_k \tag{3.3.2}$$

The Lanczos algorithm

Now suppose that A is symmetric. Since

$$(H_k)^T = (V_k^T A V_k)^T = V_k^T A^T (V_k^T)^T = V_k^T A V_k = H_k,$$

upper Hessenberg matrix H_k must be also symmetric and thus tridiagonal. It means, that to ensure orthonormality of the vectors v_i it is sufficient to orthogonalize just against two previous vectors. The matrix formulation of the Lanczos algorithm stays as following:

$$AV_k = V_k T_k + \delta_{k+1} v_{k+1} u_k^T, \quad k = 1, 2, \ldots, d-1 \tag{3.3.3}$$
$$AV_d = V_d T_d \tag{3.3.4}$$

where

$$T_k \equiv \begin{bmatrix} \gamma_1 & \delta_2 & & & \\ \delta_2 & \gamma_2 & \delta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{k-1} & \gamma_{k-1} & \delta_k \\ & & & \delta_k & \gamma_k \end{bmatrix}$$

is a Jacobi matrix.

Orthonormality of the vectors v_i in terms of matrices gives

$$V_k^T V_k = I_k, \quad k = 1, \dots, d \tag{3.3.5}$$

$$V_k^T v_{k+1} = 0 \tag{3.3.6}$$

and (3.3.2) transforms into

$$V_k^T A V_k = T_k. \tag{3.3.7}$$

Vectors $v_1, \dots, v_d$ are called *Lanczos vectors* and we see that they satisfy *a three-term recurrence*

$$\delta_{k+1} v_{k+1} = A v_k - \gamma_k v_k - \delta_k v_{k-1}. \tag{3.3.8}$$

The Lanczos method

The Lanczos method computes the approximation of a few dominant eigenvalues of the matrix A. It uses the Lanczos algorithm to construct the matrix T_n and then it computes the eigenvalues and eigenvectors of T_n. The eigenvalues of T_n are called Ritz values and they approximate the eigenvalues of A. We know that the eigenvalues of Jacobi matrix are distinct (see Theorem 1.9 and Theorem 1.12) and thus we can denote the eigenvalues of T_n as $\theta_1^{(n)} < \theta_2^{(n)} < \dots < \theta_n^{(n)}$. The approximation of the eigenvectors can be easily computed as $V_n z_i^{(n)}$ where $z_i^{(n)} = (\zeta_{i,1}^{(n)}, \zeta_{i,2}^{(n)}, \dots, \zeta_{i,n}^{(n)})$ is the corresponding normalized eigenvector of the matrix T_n. The vectors $x_i^{(n)} \equiv V_n z_i^{(n)}$ are called Ritz vectors.

The multiplication of the equation (3.3.3) from the right by the eigenvector $z_i^{(n)}$ gives

$$A x_i^{(n)} - \theta_i^{(n)} x_i^{(n)} = \delta_{n+1} u_n^T z_i^{(n)} v_{n+1} \tag{3.3.9}$$

$$\left\| A x_i^{(n)} - \theta_i^{(n)} x_i^{(n)} \right\| = \delta_{n+1} \left| \zeta_{i,n}^{(n)} \right|, \tag{3.3.10}$$

where $\zeta_{i,n}^{(n)}$ is the last component of the eigenvector $z_i^{(n)}$. We know that the symmetric matrix A can be decomposed as $A = Q \Lambda Q^T$, where Q is a unitary matrix with eigenvectors q_j in a columns and $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_N)$ is a diagonal matrix of eigenvalues. Using this spectral decomposition in (3.3.10) allows to obtain the inequality; see e.g. [21, Section 2.1].

$$\min_{j=1,\dots,N} \left| \lambda_j - \theta_i^{(n)} \right| \le \frac{\delta_{n+1} \left| \zeta_{i,n}^{(n)} \right|}{\left\| x_i^{(n)} \right\|}. \tag{3.3.11}$$

The norm of the Ritz vectors $x_i^{(n)}$ is in exact arithmetic equal to one and thus we see that the quantity $\delta_{n+1} |\zeta_{i,n}^{(n)}|$ determines how well the Ritz value $\theta_i^{(n)}$ approximates some eigenvalue of the matrix A.

Theorem 3.3 (Persistence Theorem). *Let $n < t$. Then,*

$$\min_j \left| \theta_i^{(n)} - \theta_j^{(t)} \right| \le \delta_{n+1} |\zeta_{i,n}^{(n)}|. \tag{3.3.12}$$

The Persistence Theorem was proved in [24] and implies that for every $t > n$ there is an eigenvalue of the associated Jacobi matrix T_t close to $\theta_i^{(n)}$ within $\delta_{n+1}|\zeta_{i,n}^{(n)}|$, we say that eigenvalue $\theta_i^{(n)}$ is stabilized to within $\delta_{n+1}|\zeta_{i,n}^{(n)}|$. Small $\delta_{n+1}|\zeta_{i,n}^{(n)}|$ is thus a criterion for the fact that some eigenvalue of A has been well approximated.

3.3.1 Lanczos algorithm as a solver and equivalence with CG

Using the Lanczos algorithm as a solver of linear systems goes hand in hand with the general principal of projection process; see Subsection 2.3.1. Let the matrix A be symmetric and positive definite and let the Krylov subspace $\mathcal{K}_k(A, r_0)$ be the search space $\mathcal{S}_k$ and also the constraints space $\mathcal{C}_k$.

In compliance with (2.3.1) we can write

$$\begin{aligned} x_k &\in x_0 + \mathcal{K}_k(A, r_0) \\ r_k &\perp \mathcal{K}_k(A, r_0). \end{aligned}$$

Since columns of the matrix V_k represent a basis of $\mathcal{K}_k(A, r_0)$, (2.3.3) and (2.3.4) gives

$$\begin{aligned} x_k &= x_0 + V_k y_k \\ V_k^T A V_k y_k &= V_k^T r_0. \end{aligned} \tag{3.3.13}$$

From the identity (3.3.7) and the orthogonality of $\{v_i\}_{i=1}^k$ (have in mind that $r_0 = v_1 \|r_0\|$) we finally have

$$T_k y_k = \|r_0\| u_1. \tag{3.3.14}$$

To sum up, the k-th approximation x_k can be obtained by solving smaller (projected) problem (3.3.14) for y_k and computation of x_k from (3.3.13).

In Section 3.2 we have shown, that the CG can be viewed as the projection process. The search and constraints spaces are the same as in the projection process determined by the Lanczos algorithm. Well defined projection process is unique and thus the CG approximation x_k is also determined by the solution of the problem (3.3.14) and by the equation (3.3.13).

For the SPD matrix A and the initial residual r_0, the CG algorithm determines the same Jacobi matrix T_n as the Lanczos algorithm applied to the initial vector $r_0 / \|r_0\|$.

Relations between vectors and coefficients

Consider the sequence of the residuals $\{r_i\}_{i=0}^k$ from the CG method and the sequence of the Lanczos vectors $\{v_i\}_{i=1}^{k+1}$ from the Lanczos algorithm with starting vector $v = r_0$ and have in mind their properties:

$$\begin{aligned} \operatorname{span}(r_0, \ldots, r_k) = \mathcal{K}_k(A, r_0) &\quad \operatorname{span}(v_1, \ldots, v_{k+1}) = \mathcal{K}_k(A, r_0) \\ r_i \perp r_j, \quad i \neq j &\quad v_i \perp v_j, \quad i \neq j. \end{aligned}$$

Thus we can write

$$v_{k+1} \in \mathcal{K}_k(A, r_0) \quad r_k \in \mathcal{K}_k(A, r_0) \tag{3.3.15}$$

$$v_{k+1} \perp \mathcal{K}_{k-1}(A, r_0) \quad r_k \perp \mathcal{K}_{k-1}(A, r_0). \tag{3.3.16}$$

Since both vectors v_{k+1}, r_k lie in the same k-dimensional subspace and are orthogonal to the same $(k-1)$-dimensional subspace and since the Lanczos vectors are normalized we see that up to sign, the Lanczos vector is the normalized CG residual. In order to determine the proper sign we compare the formula (3.3.8) with the formula for r_k from Algorithm (1). Since $v_1 = r_0 / \|r_0\|$, we finally get

$$v_{k+1} = (-1)^k \frac{r_k}{\|r_k\|} \tag{3.3.17}$$

The coefficients of the Lanczos algorithm can be expressed using the coefficients of the CG algorithm. In (3.3.8) we have seen that the Lanczos vectors v_{k+1} satisfy a three-term recurrence. Using the formulas from Algorithm (1) we can express r_k in a similar way.

$$\begin{aligned}
r_k &= r_{k-1} - \alpha_{k-1} A p_{k-1} \\
&= r_{k-1} - \alpha_{k-1} A (r_{k-1} + \beta_{k-1} p_{k-2}) \\
-\frac{1}{\alpha_{k-1}} r_k &= A r_{k-1} - \frac{1}{\alpha_{k-1}} r_{k-1} + \frac{\beta_{k-1}}{\alpha_{k-2}} (r_{k-2} - r_{k-1}) \\
-\frac{1}{\alpha_{k-1}} r_k &= A r_{k-1} - \left(\frac{1}{\alpha_{k-1}} + \frac{\beta_{k-1}}{\alpha_{k-2}} \right) r_{k-1} + \frac{\beta_{k-1}}{\alpha_{k-2}} r_{k-2} \qquad (3.3.18)
\end{aligned}$$

Multiplication of (3.3.18) with $\frac{(-1)^{k-1}}{\|r_{k-1}\|}$ gives

$$\begin{aligned}
\frac{1}{\alpha_{k-1}} \frac{\|r_k\|}{\|r_{k-1}\|} \left[(-1)^k \frac{r_k}{\|r_k\|} \right] = A & \left[(-1)^{k-1} \frac{r_{k-1}}{\|r_{k-1}\|} \right] \\
& - \left(\frac{1}{\alpha_{k-1}} + \frac{\beta_{k-1}}{\alpha_{k-2}} \right) \left[(-1)^{k-1} \frac{r_{k-1}}{\|r_{k-1}\|} \right] \\
& - \frac{\beta_{k-1}}{\alpha_{k-2}} \frac{\|r_{k-2}\|}{\|r_{k-1}\|} \left[(-1)^{k-2} \frac{r_{k-2}}{\|r_{k-2}\|} \right]
\end{aligned}$$

Finally, using (3.3.17) and

$$\beta_k = \frac{\|r_k\|^2}{\|r_{k-1}\|^2}$$

from Algorithm (1), we get

$$\frac{\sqrt{\beta_k}}{\alpha_{k-1}} v_{k+1} = A v_{k+1} - \left(\frac{1}{\alpha_{k-1}} + \frac{\beta_{k-1}}{\alpha_{k-2}} \right) v_k - \frac{\sqrt{\beta_{k-1}}}{\alpha_{k-2}}. \tag{3.3.19}$$

Comparison with (3.3.8) gives the formulas between coefficients

$$\begin{aligned}
\delta_{k+1} &= \frac{\sqrt{\beta_k}}{\alpha_{k-1}} \\
\gamma_k &= \frac{1}{\alpha_{k-1}} + \frac{\beta_{k-1}}{\alpha_{k-2}},
\end{aligned} \qquad (3.3.20)$$

where we set

$$\delta_0 = 0, \quad \gamma_{-1} = 1.$$

Conversely, let L_k be a lower bidiagonal matrix with $\frac{\sqrt{\beta_j}}{\alpha_{j-1}}$, $j = 1, \ldots, k-1$ on a subdiagonal and $\frac{1}{\sqrt{\alpha_{j-1}}}$, $j = 1, \ldots, k$ on the main diagonal. Then

$$T_k = L_k L_k^T$$

and thus the coefficients in the CG algorithm can be obtained from the coefficients of the Lanczos algorithm by the LU (Choleski) decomposition of the Jacobi matrix T_k.

3.4 Relationship with orthogonal polynomials and Gauss-Christoffel quadrature

In this section we follow [21, 31] and we will describe the relationship between the CG and Lanczos algorithms and the sequence of orthogonal polynomials. We know that the Lanczos algorithm generates the Jacobi matrix. The Theorem 1.8 implies that this Jacobi matrix defines monic polynomials which are orthogonal to some inner product determined by some distribution function. We will define this distribution function and use it to demonstrate the relationship between the Lanczos algorithm and the Gauss-Christoffel quadrature.

Suppose that the Lanczos algorithm, applied to the symmetric matrix A and the initial vector v_1, does not stop in iterations $1, \ldots, n$, i.e., n is strictly smaller then the grade of v_1 with respect to A. We know that the Lanczos vectors $v_1, \ldots, v_{n+1}$ create an orthonormal basis of the Krylov subspace $\mathcal{K}_{n+1}(A, v_1)$. Thus we can write vector v_{k+1}, $k = 0 \ldots, n$ in terms of a polynomial in the matrix A applied to the initial vector v_1, i.e.,

$$v_{k+1} = \varphi_{k+1}(A)v_1, \quad k = 0, \ldots, n, \tag{3.4.1}$$

where φ_{k+1} is a polynomial of degree k. Substitution of (3.4.1) to the three-term recurrence (3.3.8) gives a three-term recurrence for the polynomials φ_k:

$$\begin{aligned} \delta_{k+1}\varphi_{k+1}(\lambda) &= (\lambda - \gamma_k)\varphi_k(\lambda) - \delta_k\varphi_{k-1}(\lambda) \quad k = 1, \ldots, n \\ \varphi_1(\lambda) &= 1 \\ \varphi_0(\lambda) &= 0. \end{aligned} \tag{3.4.2}$$

Assume now for simplicity that the symmetric matrix A has distinct eigenvalues $\lambda_1 < \cdots < \lambda_N$ and set numbers a, b such that the eigenvalues are enclosed in the open interval (a, b). From the spectral decomposition $A = Q\Lambda Q^T$, where Q is a unitary matrix with eigenvectors q_j in a columns and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is a diagonal matrix of eigenvalues.

Using the orthonormality of the Lanczos vectors we can compute

$$\begin{aligned} \delta_{k,l} &= (v_l, v_k) = v_k^T v_l = (\varphi_k(A)v_1)^T \varphi_l(A)v_1 \\ &= v_1^T Q\varphi_k(\Lambda)Q^T Q\varphi_l(\Lambda)Q^T v_1 \\ &= (Q^T v_1)^T \varphi_k(\Lambda)\varphi_l(\Lambda)Q^T v_1 \\ &= \sum_{j=1}^{N} (v_1, q_j)^2 \varphi_k(\lambda_j)\varphi_l(\lambda_j), \quad k, l = 1, \ldots, n+1 \end{aligned} \tag{3.4.3}$$

where $\delta_{k,l}$ is the Kronecker's delta.

Similarly as in (1.2.1) and (1.2.2) we can define nondecreasing piecewise constant distribution function $\omega(\lambda)$ with at most N points of increase $\lambda_1, \dots, \lambda_N$ and with weights given by the squared component of the vector v_1 in the direction of the invariant subspace determined by the eigenvector q_j, i.e.,

$$\omega_j = (v_1, q_j)^2. \tag{3.4.4}$$

Because the initial vector v_1 is normalized to have unit norm, the sum of the weights is equal to 1, i.e.,

$$\sum_{j-1}^{N} \omega_j = 1. \tag{3.4.5}$$

Define also the associated Riemann-Stieltjes integral

$$\int_a^b f(\lambda)\, d\omega(\lambda) \equiv \sum_{j=1}^{N} \omega_j f(\lambda_j) = v_1^T f(A) v_1. \tag{3.4.6}$$

Similarly as in Definition 1.8 define a mapping

$$\langle p, q \rangle_\omega = \int_a^b p(\lambda) q(\lambda)\, d\omega(\lambda). \tag{3.4.7}$$

Lemma 3.4. *Mapping defined in* (3.4.7) *is an inner product on the subspace* $\mathcal{P}_n$

Proof. The polynomials $\varphi_0, \varphi_1, \dots, \varphi_{n+1}$ span the subspace $\mathcal{P}_n$. The identity (3.4.3) implies that $\varphi_1, \dots, \varphi_{n+1}$ are orthonormal with respect to the mapping $\langle \cdot, \cdot \rangle_\omega$. Thus for arbitrary polynomial $p \in \mathcal{P}_n$ holds that

$$\begin{aligned} \langle p, p \rangle_\omega &= \left\langle \sum_{i=1}^{n} \nu_i \varphi_i, \sum_{i=1}^{n} \nu_i \varphi_i \right\rangle_\omega \\ &= \sum_{i=1}^{n} \nu_i^2 \end{aligned}$$

and thus

$$\langle p, p \rangle_\omega = 0 \iff \nu_i = 0 \ \forall i = 1, \dots, n \iff p = 0.$$

□

We have shown that the polynomials $\varphi_0, \varphi_1, \dots, \varphi_{n+1}$ are orthonormal with respect to the inner product $\langle \cdot, \cdot \rangle_\omega$ on the subspace $\mathcal{P}_n$. Thus we can use the results of Section 1.3 and see that the roots of the polynomial φ_{k+1} are the eigenvalues of the Jacobi matrix T_k. Polynomials $(-1)^k \chi_k$, where χ_k is the characteristic polynomial of the Jacobi matrix T_k, are the monic orthogonal polynomials.

Now we will reveal the relationship between the Lanczos algorithm and the Gauss-Christoffel quadrature for approximation of the Riemann-Stieltjes integral.

Consider the symmetric tridiagonal Jacobi matrix T_n defined by the first n steps of the Lanczos algorithm. Denote as I_n the $n$dimensional identity matrix and let u_i be its columns. Simple identity

$$T_n I_n = I_n T_n, \tag{3.4.8}$$

can be considered as a matrix formulation of the Lanczos algorithm applied to the matrix T_n and the starting vector u_1. It is worth to note here that the comparison with (3.3.4) gives that u_1 is of grade n with respect to T_n. We know that matrix T_n defines a sequence of polynomials $\varphi_1, \ldots, \varphi_n$. Using exactly the same arguments as at the beginning of this section we can write

$$u_{k+1} = \varphi_{k+1}(T_n)u_1, \quad k = 0, \ldots, n-1. \tag{3.4.9}$$

The polynomials are the same as in the first case because they are determined by the same tridiagonal matrix T_n. It is worth to note that we know that the roots of the polynomial φ_{n+1} are the eigenvalues of the matrix T_n and thus the eigenvalues are always distinct (also in the case when A has some multiple eigenvalues); see Theorem 1.9. We denote them as $\theta_1^{(n)} < \theta_2^{(n)} < \cdots < \theta_n^{(n)}$. As a consequence of the interlace theorem they are also enclosed in the open interval (a, b). Consider the spectral decomposition of T_n

$$T_n = Z_n \Theta^{(n)} Z_n^T,$$

where $\Theta^{(n)}$ is a diagonal matrix of eigenvalues $\theta_1^{(n)}, \ldots, \theta_n^{(n)}$ and Z_n is a unitary matrix with normalized eigenvectors $z_j^{(n)}$ in a columns.

Similarly as before we can define nondecreasing piecewise constant distribution function $\omega^{(n)}(\lambda)$. We will show that it has exactly n points of increase $\theta_1^{(n)}, \ldots, \theta_n^{(n)}$. The weights are given by the squared component of the vector u_1 in the direction of the invariant subspace determined by the eigenvector $z_j^{(n)}$, i.e.,

$$\omega_j^{(n)} = (u_1, z_j^{(n)})^2. \tag{3.4.10}$$

As before, $\|u_1\| = 1$ gives

$$\sum_{j=1}^{n} \omega_j^{(n)} = 1. \tag{3.4.11}$$

The associated Riemann-Stieltjes integral is defined as

$$\int_a^b f(\lambda)\, d\omega^{(n)}(\lambda) \equiv \sum_{j=1}^{n} \omega_j^{(n)} f(\theta_j^{(n)}) = u_1^T f(T_n) u_1. \tag{3.4.12}$$

The mapping

$$\langle p, q \rangle_{\omega^{(n)}} = \int_a^b p(\lambda) q(\lambda)\, d\omega^{(n)}(\lambda). \tag{3.4.13}$$

defines an inner product on the subspace $\mathcal{P}_{n-1}$. It can be proved similarly as in Lemma 3.4 using the identity (3.4.14)

$$\begin{aligned}
\delta_{k,l} &= (u_l, u_k) = u_k^T u_l = (\varphi_k(T_n)u_1)^T \varphi_l(T_n) u_1 \\
&= u_1^T Z_n \varphi_k(\Theta^{(n)}) Z_n^T Z_n \varphi_l(\Theta^{(n)})) Z_n^T u_1 \\
&= (Z_n u_1)^T \varphi_k(\Theta^{(n)})) \varphi_l(\Theta^{(n)})) Z_n^T u_1 \\
&= \sum_{j=1}^{n} (u_1, z_j^{(n)})^2 \varphi_k(\theta_j^{(n)}) \varphi_l(\theta_j^{(n)}) \\
&= \langle \varphi_k, \varphi_l \rangle_{\omega^{(n)}}, \quad k, l = 1, \ldots, n
\end{aligned} \tag{3.4.14}$$

Suppose that weight $\omega_j^{(n)}$ is zero. Consequently, the distribution function $\omega^{(n)}(\lambda)$ would have less then n points of increase and there would be a polynomial ψ (defined to have roots at the points of increase) of strictly lower degree then n such that $\langle \psi, \psi \rangle_{\omega^{(n)}} = 0$ and thus the mapping $\langle \cdot, \cdot \rangle_{\omega^{(n)}}$ would not be the inner product on $\mathcal{P}_{n-1}$.

Therefore we have shown that the distribution function $\omega^{(n)}(\lambda)$ has exactly n points of increase and that all weights given by (3.4.10) are strictly positive. The polynomials $\varphi_1 = 1, \varphi_2, \ldots, \varphi_{n+1}$ are orthogonal to each other with respect to both inner products $\langle \varphi_k, \varphi_l \rangle_{\omega^{(n)}}$ and $\langle \varphi_k, \varphi_l \rangle_{\omega}$.

Since the polynomial φ_{n+1} has the roots at the points of increase of the distribution function $\omega^{(n)}(\lambda)$, it is obvious that $\langle \varphi_{n+1}, \varphi_{n+1} \rangle_{\omega^{(n)}} = 0$. It is worth to notice that even $\varphi_{n+1}(T_n) = 0$. It can be revealed from the three-term recurrence and the identity (3.4.8) but more elegant way is to use the Cayley-Hamilton theorem (polynomial φ_{n+1} is a multiple of the characteristic polynomial χ_n).

To sum up, the polynomials $\varphi_1, \varphi_2, \ldots, \varphi_n$ represents the sequence of orthonormal polynomials with respect to the inner product $\langle \cdot, \cdot \rangle_{\omega^{(n)}}$ on the space $\mathcal{P}_{n-1}$ and the uniquely defined polynomial φ_{n+1} is orthogonal to them but $\|\varphi_{n+1}\|_{\omega^{(n)}} = 0$; see also a discussion on p. 9.

The following theorem reveals the relationship between the Lanczos algorithm and the Gauss-Christoffel quadrature rule.

Theorem 3.5. *[21, Theorem 2.1] The Riemann-Stieltjes integral* (3.4.12) *defined by the distribution function $\omega^{(n)}(\lambda)$ given by the nodes and weights determined by first n steps of the Lanczos algorithm applied to symmetric matrix A and v_1 represents the n-th Gauss-Christoffel quadrature approximation of the Riemann-Stieltjes integral* (3.4.6) *defined by the distribution function $\omega(\lambda)$ which is given by the nodes and weights determined by the vector v_1 and the spectral decomposition of the matrix A.*

Proof. Consider a polynomial $\phi(\lambda)$ of degree less then $2n$. Then it is possible to write

$$\phi(\lambda) = \varphi_{n+1}(\lambda)\phi_1(\lambda) + \phi_2(\lambda) = \varphi_{n+1}(\lambda)\phi_1(\lambda) + \sum_{j=2}^{n} \nu_j \varphi_j(\lambda) + \nu_1,$$

where ϕ_1, ϕ_2 are the polynomials of degree less then n. The orthogonality of the polynomials $1, \varphi_2, \ldots, \varphi_{n+1}$ gives

$$\int_a^b \phi(\lambda)\, d\omega^{(n)}(\lambda) = \int_a^b \nu_1\, d\omega^{(n)}(\lambda) = \int_a^b \nu_1\, d\omega(\lambda) = \int_a^b \phi(\lambda)\, d\omega(\lambda)$$

□

Using the definitions (3.4.6), (3.4.12) we see that the previous theorem gives the identity

$$v_1^T \phi(A) v_1 = u_1^T \phi(T_n) u_1, \quad \forall \phi \in \mathcal{P}_{2n-1}. \tag{3.4.15}$$

Speaking about the CG algorithm, it is worth noticing the relationship between the initial error and the Gauss-Christoffel quadrature. The identity

$$\|e_0\|_A^2 = e_0^T A e_0 = r_0^T A^{-1} r_0 = \|r_0\|^2 v_1^T A^{-1} v_1$$

and the definition (3.4.6) give

$$\|e_0\|_A^2 = \|r_0\|^2 \, v_1^T A^{-1} v_1 = \|r_0\|^2 \int_a^b \lambda^{-1} \, d\omega(\lambda). \tag{3.4.16}$$

The k-th error e_k satisfies (see e.g. (3.1.10))

$$e_k \in e_0 + \mathcal{K}_k(A, r_0).$$

Since $r_0 = Ae_0$ we can write

$$e_k \in e_0 + \mathrm{span}(Ae_0, \ldots, A^k e_0),$$

and thus

$$e_k = \widehat{\varphi_k}(A) e_0, \tag{3.4.17}$$

where $\widehat{\varphi_k}(\lambda)$ is a polynomial of degree k which is equal to one at the origin. Because of the identity $r_k = Ae_k$ it is also true that

$$r_k = \widehat{\varphi_k}(A) r_0. \tag{3.4.18}$$

Using the identity (3.3.17) between the Lanczos vectors and the CG residuals we can write

$$\widehat{\varphi_k}(A) r_0 = r_k = (-1)^k \, \|r_k\| \, v_{k+1} = (-1)^k \frac{\|r_k\|}{\|r_0\|} \varphi_{k+1}(A) r_0,$$

which implies that polynomials $\widehat{\varphi_k}$ and φ_{k+1} are the same except the multiplication by scalar. Consequently, the condition $\widehat{\varphi_k}(0) = 1$ gives

$$\widehat{\varphi_k}(\lambda) = \frac{\varphi_{k+1}(\lambda)}{\varphi_{k+1}(0)} \tag{3.4.19}$$

Notice that the assumption that A is SPD and the interlace theorem guarantees that the roots of the polynomial φ_{k+1} (resp. the eigenvalues of the Jacobi matrix T_{k+1}) can not be smaller then $\lambda_1 > 0$ and thus $\varphi_{k+1}(0) \neq 0$.

The CG method is optimal in the sense that in the k-th step it minimalizes the error over the k-th dimensional subspace. Now we see that associated quadrature rule is also optimal. The Gauss-Christoffel quadrature rule has the highest possible algebraic degree of exactness.

Conversely, consider SPD A and the initial residual r_0. The Riemann-Stieltjes integral (3.4.6) is defined uniquely as well as its Gauss-Christoffel approximations for $j = 1, \ldots, N$. We know that every quadrature rule can be considered as the Riemann-Stieltjes integral for nondecreasing piecewise constant distribution function (see Section 1.6). Thus we have uniquely defined distribution functions $\omega^{(j)}(\lambda)$ which uniquely determine the Jacobi matrices T_j. Because of the equivalence between the projection process defined by the Lanczos algorithm and the method of CG, the Jacobi matrix together with (3.3.14) and (3.3.13) uniquely determine the CG approximation x_j.

3.5 Matching moments and model reduction

Matching moments model reduction represent an important topic in numerical mathematics. We show in this section that CG and Lanczos method can be considered as that kind of model reduction. We follow [29] and [20, Sections 3.1 and 3.7].

3.5.1 Stieltjes moment problem

In this section we will deal with the problem of moments and its relationship to the Krylov subspace methods. The problem of moments was firstly formulated by Stieltjes, for more details see [20, Section 3.1 and Historical note 3.3.5]. In the version of simplified Stieltjes moment problem we deal with the distribution function $\varpi(\lambda)$ with N points of increase $a < \lambda_1 < \cdots < \lambda_N \leq b$ and with positive weights ϖ_i such that $\sum_{i=1}^{N} \varpi_i = 1$.

For given $n \geq 0$ we want to determine distribution function $\varpi^{(n)}$ on $[a, b]$ with n points of increase $a < \lambda_1^{(n)} < \cdots < \lambda_n^{(n)} \leq b$ and with positive weights $\varpi_1^{(n)}, \ldots, \varpi_n^{(n)}$ such that $\sum_{i=1}^{n} \varpi_i^{(n)} = 1$ in order to match the first $2n$ moments of $\varpi^{(n)}(\lambda)$ with the first $2n$ moments of the original distribution function $\varpi(\lambda)$, i.e.,

$$\int_a^b \lambda^k \, d\varpi^{(n)}(\lambda) = \int_a^b \lambda^k \, d\varpi(\lambda), \quad k = 0, \ldots, 2n-1. \tag{3.5.1}$$

We know that the Riemann-Stieltjes integral given by the distribution function $\varpi^{(n)}(\lambda)$ can be interpreted as a quadrature rule of the integral given by $\varpi(\lambda)$. This means that the solution of the simplified Stieltjes moment problem determines an n-point quadrature rule which is exact for all polynomials up to degree $2n-1$, i.e., it determines the uniquely defined Gauss-Christoffel quadrature rule. Conversely, the distribution function given by the Gauss-Christoffel quadrature rule gives a solution of the simplified Stieltjes moment problem.

Thus the Gauss-Christoffel quadrature can be viewed as a model reduction of the original model represented by the distribution function $\varpi(\lambda)$ (N points of increase) to the reduced model represented by the distribution function $\varpi^{(n)}(\lambda)$ (n points of increase). This model reduction matches the first $2n$ moments.

Using the Lanczos algorithm and its relationship with the Gauss-Christoffel quadrature it is possible to formulate the problem of moments in terms of matrices. With the same setting as in Section 3.4 we can express the moments of the Riemann-Stieltjes integral (3.4.6) in a language of linear algebra,

$$\int_a^b \lambda^k \, d\omega(\lambda) = v_1^T A^k v_1, \quad k = 0, 1, \ldots. \tag{3.5.2}$$

We know that the Lanczos algorithm applied to A and v_1 gives in the n-th step the Jacobi matrix T_n and we can construct the associated distribution function $\omega^{(n)}(\lambda)$. The first $2n$ moments can be expressed in a similar way,

$$\int_a^b \lambda^k \, d\omega^{(n)}(\lambda) = u_1^T T_n^k u_1, \quad k = 0, \ldots, 2n-1. \tag{3.5.3}$$

In the previous section we have shown that the Lanczos algorithm can be formulated in terms of Gauss-Christoffel quadrature approximations given by the distribution function $\omega^{(n)}(\lambda)$ to the original Riemann-Stieltjes integral given by the distribution function $\omega(\lambda)$. Thus the Lanczos algorithm generates the sequence of distribution functions which are solutions of the simplified Stieltjes moment problem and the matching property can be written as

$$u_1^T T_n^k u_1 = v_1^T A^k v_1, \quad k = 0, \ldots, 2n-1. \tag{3.5.4}$$

The Lanczos algorithm can be viewed as a model reduction of the original problem represented by A, v_1 to the reduced model represented by T_n, u_1. This model reduction matches the first $2n$ moments.

With A SPD we can use the results about the equivalence between CG and Lanczos algorithms. From Subsection 3.3.1 we know that the approximation x_n generated by the CG method is also uniquely determined by formulas

$$x_n = x_0 + V_n y_n, \quad T_n y_n = \|r_0\| \, u_1. \tag{3.5.5}$$

Thus the CG approximation x_n can be considered as a result of the model reduction of the original problem $Ax = b$ to the reduced model $T_n y_n = \|r_0\| \, u_1$. This model reduction matches the first $2n$ moments.

3.5.2 Vorobyev moment problem

The Krylov subspace methods are useful also for solving linear systems with nonsymmetric matrices. These methods can be also described as a model reduction with some matching property. However, the interpretation of moment matching which uses the Gauss-Christoffel quadrature involves the extension of the Gauss-Christoffel quadrature to the complex plane which include some nontrivial assumptions.

An operator formulation of the problem of moments suggested by Vorobyev allows to describe the moment matching property without any further assumptions. Unfortunately, the works of Vorobyev are not well known yet. In compliance with the main focus of this thesis, we will illustrate the Vorobyev moment problem for the symmetric matrix A. We will show that in this case the operator formulation of the problem of moments is equivalent to the formulation of the simplified Stieltjes moment problem. For more details about the application of the Vorobyev moment problem to the non-symmetric Krylov subspace methods see [20, Section 3.7] or [29] and references given there.

The symmetric matrix $A \in \mathbb{R}^{N \times N}$ can be viewed as a linear operator on $\mathbb{R}^N$. As before, suppose that columns of V_n represent the orthonormal basis of the n-dimensional Krylov subspace $\mathcal{K}_n(A, v_1)$. The mapping

$$Q_n \equiv V_n V_n^T : \mathbb{R}^N \longrightarrow \mathcal{K}_n(A, v_1)$$

represents the orthogonal projector onto $\mathcal{K}_n(A, v_1)$.

Vorobyev's formulation of the moment problem can be in our case written as follows: Given A and v_1 we wish to construct a linear operator A_n defined on the Krylov subspace $\mathcal{K}_n(A, v_1)$ such that

$$\begin{aligned} A_n v_1 &= A v_1 \\ A_n^2 v_1 &= A^2 v_1 \\ &\vdots \\ A_n^{n-1} v_1 &= A^{n-1} v_1 \\ A_n^n v_1 &= Q_n A^n v_1, \end{aligned} \tag{3.5.6}$$

It can be easily shown that equations (3.5.6) determine the operator A_n uniquely and that the solution is given by the orthogonally projected restriction of the operator A on $\mathcal{K}(A, v_1)$, i.e.,

$$A_n = Q_n A Q_n.$$

Now we will see that the solution of the Vorobyev moment problem gives the same matching property as (3.5.4). Since

$$v_1^T A^k v_1 = v_1^T Q_n A^k Q_n v_1 = u_1^T V_n^T A^k V_n u_1 = u_1^T T_n^k u_1, \tag{3.5.7}$$

we must show that

$$v_1^T A^k v_1 = v_1^T A_n^k v_1, \quad k = 0, \ldots, 2n-1. \tag{3.5.8}$$

The identity is trivial for $k = 0$ and for $k = 1, \ldots, n-1$ it is an immediate consequence of (3.5.6). Since Q_n is a projector, multiplication of the last row of (3.5.6) by Q_n implies that

$$Q_n(A^n v_1 - A_n^n v_1) = 0 \tag{3.5.9}$$

Since Q_n projects orthogonally onto $\mathcal{K}_n(A, v_1)$, the vector $A^n v_1 - A_n^n v_1$ must be orthogonal to all of its basis vectors and thus the use of (3.5.6) and of symmetry of A, A_n gives

$$v_1^T A^j (A^n v_1 - A_n^n v_1) = v_1^T A_n^j (A^n v_1 - A_n^n v_1) = 0, \quad j = 0, \ldots, n-1 \tag{3.5.10}$$

which gives the result.

Summarizing, the matrix formulation of the matching property can equivalently represent Stieltjes and Vorobyev moment problem.

3.6 Convergence of CG, estimates of the energy norm

In this section we will review several results about the convergence of the CG method and express several error bounds. We will see that the relationship of the CG method with the Lanczos algorithm and the Gauss-Christoffel quadrature is not interesting only theoretically but that it allows to compute interesting error bounds.

Since the CG method gives in exact arithmetic the solution in at most N steps, it is not very meaningful to speak about the convergence in the asymptotic sense. In many applications, the approximate of the solution of satisfactory accuracy can be attained in a few steps. It is this what we have in mind when speaking about convergence and convergence behaviour. We care about a decrease of the error from the very beginning of computation.

Now we will review the classical upper bound based on the Chebyshev polynomials, we will follow [32, Section 8.2]. We know (see (3.1.15) and (3.2.4)) that the CG method minimizes in the k-th step the energy norm of the error over the k-dimensional Krylov subspace $\mathcal{K}(A, r_0)$, i.e.,

$$\|x - x_k\|_A = \min_{y \in x_0 + \mathcal{K}_k(A, r_0)} \|x - y\|_A. \tag{3.6.1}$$

Let us also repeat (see (3.4.17) the polynomial expression of the error $e_k = x - x_k$ and the residual r_k,

$$e_k = \widehat{\varphi_k}(A)e_0, \quad r_k = \widehat{\varphi_k}(A)r_0, \quad \widehat{\varphi_k}(\lambda) \in \Pi_k^0, \tag{3.6.2}$$

where Π_k^0 is the set of the polynomials of degree k with value 1 at the origin.

Using the identity $Ae_0 = r_0$ and the spectral decomposition $A = Q\Lambda Q^T$ we can express the energy norm of the error as

$$\begin{aligned} \|e_k\|_A^2 = \|\widehat{\varphi_k}(A)e_0\|_A^2 &= \min_{\varphi \in \Pi_k^0} \|\varphi(A)e_0\|_A^2 = \min_{\varphi \in \Pi_k^0} \|\varphi(A)r_0\|_{A^{-1}}^2 \\ &= \min_{\varphi \in \Pi_k^0} \sum_{i=1}^{N} (r_0, q_i)^2 \frac{\varphi^2(\lambda_i)}{\lambda_i}. \end{aligned} \tag{3.6.3}$$

Thus we see that the rate of the convergence of CG is determined by two factors: by the size of the components of the initial residual in the direction of the invariant subspaces determined by the eigenvectors and by the distribution of the eigenvalues of A.

Let us focus on the estimate of the right side in (3.6.3). Since

$$\|\varphi(A)e_0\|_A = \left\|\varphi(A)A^{1/2}e_0\right\| \le \|\varphi(A)\| \left\|A^{1/2}e_0\right\| = \|\varphi(A)\| \, \|e_0\|_A$$

and

$$\|\varphi(A)\| = \|\varphi(\Lambda)\| = \rho(\varphi(\Lambda)) = \max_{i=1,\dots,N} |\varphi(\lambda_i)|$$

we can write that

$$\frac{\|e_k\|_A}{\|e_0\|_A} \le \min_{\varphi \in \Pi_k^0} \|\varphi(A)\| \quad \le \min_{\varphi \in \Pi_k^0} \max_{i=1,\dots,N} |\varphi(\lambda_i)| \, . \tag{3.6.4}$$

A standard way to analyze the minimax problem (3.6.4) is to substitute the discrete set of the eigenvalues by an interval which includes all eigenvalues, i.e.,

$$\frac{\|e_k\|_A}{\|e_0\|_A} \le \min_{\varphi \in \Pi_k^0} \max_{\lambda \in [\lambda_1, \lambda_N]} |\varphi(\lambda_i)| \, . \tag{3.6.5}$$

The minimax problem in (3.6.5) has the unique solution given by the shifted and scaled Chebyshev polynomials (see Remark 1.12) and thus it is possible to write

$$\frac{\|e_k\|_A}{\|e_0\|_A} \le \left| \mathrm{T}_k \left(\frac{\lambda_N + \lambda_1}{\lambda_N - \lambda_1} \right) \right|^{-1}.$$

After some manipulation (see e.g. [26, Theorem 6.6]) we can write the well known upper bound

$$\frac{\|e_k\|_A}{\|e_0\|_A} \le 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k , \tag{3.6.6}$$

where $\kappa = \lambda_N / \lambda_1$ is the condition number of the matrix A.

Although this upper bound is well known, it was not always understood correctly. The inequality (3.6.6) implies that if the matrix A is well conditioned (the condition number κ is close to 1), the convergence of CG is very rapid. However, large condition number does not imply poor convergence. The upper bound

(3.6.6) is often very pessimistic. That is because the upper bound (3.6.6) is based only on the information about the extreme eigenvalues λ_1, λ_N but we know that the rate of convergence is determined by the distribution of all eigenvalues. It often happens that after a few iterations the rate of convergence accelerates. This phenomenon is called superlinear convergence we will and it will be explained in Subsection 3.6.2.

3.6.1 Estimates of the energy norm

Several interesting estimates of the energy norm are based on the relationship between the CG and the Gauss-Christoffel quadrature and its modifications like Gauss-Radau or Gauss-Lobato quadrature rules; see e.g. [8, Chapter 12]. Specially, it can be proved that the scaled squared energy norm of the error in the n-th step is the remainder of the n-point Gauss-Christoffel quadrature for function λ^{-1} with the distribution function $\omega(\lambda)$ defined in Section 3.4, i.e.,

$$\frac{\|e_n\|_A^2}{\|r_0\|^2} = (T_N^{-1})_{11} - (T_n^{-1})_{11}, \tag{3.6.7}$$

where the identities

$$\int_a^b \lambda^{-1}\, d\omega(\lambda) = (T_N^{-1})_{11}, \quad \int_a^b \lambda^{-1}\, d\omega^{(n)}(\lambda) = (T_n^{-1})_{11} \tag{3.6.8}$$

follow from (3.4.16), the identity $AV_N = V_N T_N$ and the definition (3.4.12). The identity (3.6.7) was probably known to Stieltjes and it was proved by several authors in different ways; for an elegant proof see e.g. [8, Theorem 12.1] or [9].

Since at iteration n we do not know $(T_N^{-1})_{11}$, the formula (3.6.7) cannot be used directly as the estimate of the norm of the error. However, subtraction of (3.6.7) for the iteration n from (3.6.7) for the iteration $n+d$ eliminates the unknown $(T_N^{-1})_{11}$ and we get

$$\|e_n\|_A^2 = \|e_{n+d}\|_A^2 + \|r_0\|^2 \left((T_{n+d}^{-1})_{11} - (T_n^{-1})_{11} \right), \tag{3.6.9}$$

where d is some given positive integer. Recall that the energy norm of the error is strictly decreasing. If d is chosen such that

$$\|e_n\|_A^2 \gg \|e_{n+d}\|_A^2 \tag{3.6.10}$$

then neglecting $\|e_{n+d}\|_A^2$ gives the lower bound of $\|e_n\|_A^2$

$$\eta_{n,d} = \|r_0\|^2 \left((T_{n+d}^{-1})_{11} - (T_n^{-1})_{11} \right). \tag{3.6.11}$$

The difference $(T_{n+d}^{-1})_{11} - (T_n^{-1})_{11}$ can be computed by the CGQL algorithm; see [8, p. 205–207]. Modifications which use the Gauss–Radau or Gauss–Lobato quadrature rules allow to obtain also the upper bound of the energy norm of the error.

Another estimate of the energy norm of the error is based on the identity

$$\|e_k\|_A^2 - \|e_l\|_A^2 = \sum_{i=k}^{l-1} \alpha_i \|r_i\|^2, \quad 0 \le k < l \le N \tag{3.6.12}$$

which was revealed already in the original paper by Hestenes and Stiefel (see [14, relation (6:2)]). The possibility of using of (3.6.12) as a stopping criterion was emphasized in [31]. The derivation of the identity (3.6.12) is easy and it uses only the local orthogonality, for the details see [31, Section 3 and 4]. Similarly as before we can consider positive integer d and write

$$\|e_n\|_A^2 = \|e_{n+d}\|_A^2 + \sum_{i=n}^{n+d-1} \alpha_i \|r_i\|^2 . \tag{3.6.13}$$

Under the assumption (3.6.10) we get the reasonably tight lower bound

$$\nu_{n,d} = \sum_{i=n}^{n+d-1} \alpha_i \|r_i\|^2 . \tag{3.6.14}$$

It is worth noticing that this bound is very simple and it uses the quantities which are at our disposal during the run of the CG algorithm.

The choice of the positive integer d in order to get tight lower bounds $\eta_{n,d}, \nu_{n,d}$ represents difficult open problem. Usually, the larger d, the better is the estimate. On the other hand, it is necessary to run more iterations of the CG algorithm.

In the next section it will be stressed that the theoretical properties of the CG are substantially influenced by rounding errors in the practical computations. Here we would like to stress that rounding errors might play a significant role also in the application of any bounds derived assuming exact arithmetic. The bounds estimate quantities which might be orders of magnitude different from their exact arithmetic counterparts. Without rigorous analysis of the bounds in finite precision (FP) arithmetic there is no justification that the estimates will work in the practical computation.

The numerical stability of the bounds based on the Gauss–Christoffel (the bound $\eta_{n,d}$), Gauss–Radau and Gauss–Lobato quadrature rules was with some limitations justified in [9]. The numerical stability of the bound $\nu_{n,d}$ was explained in [31, Sections 7–10]. In Chapter 4 we will give an example of the upper bound derived assuming exact arithmetic which does not work in finite precision arithmetic and we will explain why it is so.

3.6.2 Superlinear convergence

The superlinear convergence of the CG computations can be explained via the convergence of the Ritz values to the eigenvalues of A. We know from the previous subsection that the energy norm of the CG error is connected with the remainder of the Gauss–Christoffel quadrature and it is easy to see that (3.6.7) can be also written as

$$\frac{\|e_n\|_A^2}{\|r_0\|^2} = \sum_{j=1}^{N} \frac{(z_j^{(N)})}{\lambda_j} - \sum_{i=1}^{n} \frac{(z_i^{(n)})}{\theta_i^{(n)}} . \tag{3.6.15}$$

This identity reflects that there is a relationship (although complicated) between the norm of error and the convergence of Ritz values. Another expression of that relationship is given in the following theorem.

Theorem 3.6. *[21, Theorem 3.3] For all k, there exists $\vartheta_n \in [\lambda_1, \lambda_N]$ such that the energy norm of the error is given by*

$$\|e_n\|_A^2 = \frac{\|r_0\|^2}{\vartheta_n^{2n+1}} \sum_{i=1}^{N} \left(\omega_i \prod_{j=1}^{n} (\lambda_i - \theta_j^{(n)})^2 \right), \tag{3.6.16}$$

where $\omega_i = |(v_1, q_i)|^2$.

This theorem shows that the convergence of Ritz value to some eigenvalue of A implies elimination of the component of the initial residual which lies in the direction of the corresponding eigenvector. In other words,

> “ . . . as soon as one of the extreme eigenvalues is modestly well approximated by a Ritz value, the procedure converges from then on as a process in which this eigenvalue is absent, i.e., a process with a reduced condition number”[33, p. 51].

Convergence analysis based on potential theory

The minimax problems like in (3.6.4) or (3.6.5) can be also analyzed using the methods of potential theory. Indeed, the relationship between potential theory and the Krylov subspace methods were observed by several authors; see e.g. [5, 17].

Mathematically rigorous introduction to potential theory is beyond the scope of this paper and thus we limit only to some comments about the usefulness of analysis based on potential theory.

As it is demonstrated in [17], potential theory can give interesting results about the minimax problem over the discrete set as in (3.6.4). Potential theory can tell us which eigenvalues are very well approximated by the zeros and which are not. It can give an improved asymptotic convergence factor [17, Section 9] which corresponds with the superlinear convergence of CG.

However, there are many assumptions which limit the practical use of this results. The improved asymptotic convergence factor is valid only in asymptotic sense which means that both dimension of the problem and the number of carried iterations must be very large. It also depends on certain asymptotic distribution of eigenvalues.

3.7 Analysis of rounding errors

This section is based on nice and well written review paper [21] and on [31, Section 5]. In the previous sections we have described a method with interesting theoretical properties and we have revealed the relationship of the CG method to the Lanczos algorithm, the orthogonal polynomials, the Riemann-Stieltjes integral and the Gauss–Christoffel quadrature. However, in practical computations it is necessary to consider the effects of rounding errors. Without rigorous analysis and justification it is not possible to be sure that any of the identities, formulas or theoretical properties derived in exact arithmetic holds also in finite precision arithmetic.

It has been known since the introduction of the Lanczos algorithm that the numerical behaviour can be strongly affected by rounding errors and that it does not fulfill its theoretical properties. In particular, the global orthogonality of the Lanczos vectors is typically lost after a few iterations. As a consequence of the loss of orthogonality, the elements of the computed Jacobi matrix may differ by several orders of their exact arithmetic counterparts. In addition, the multiple approximation of the dominant eigenvalues appear and thus the approximation of the other eigenvalues is delayed. Moreover, there is no guarantee that the norm of the Ritz vector $x_i^{(n)} = V_k z_i^{(n)}$ is close to one. It can even numerically vanish and thus it is not clear whether a small value of $\delta_{n+1}|\zeta_{i,n}^{(n)}|$ means convergence of the Ritz value to any eigenvalue of A as (3.3.11) suggests.

The residual vectors in the CG lose their orthogonality similarly as the Lanczos vectors do. Since in exact arithmetic the rate of the convergence depends on how well the eigenvalues of A are approximated by the eigenvalues of T_n (see Subsection 3.6.2), we may expect the same in finite precision. The appearance of the multiple copies of dominant eigenvalues then cause a delay in CG convergence.

3.7.1 Paige's analysis and its consequences

Despite the loss of orthogonality, i.e., the invalidity of the fundamental principle of the Lanczos method, the Lanczos method gives surprisingly reasonable results. However the serious loss of orthogonality caused that the Lanczos method was neglected by the numerical analysts for more than 20 years. This was changed in 1970s by works of Chris Paige (see e.g. [25] or [24]). He presented rigorous mathematical analysis of rounding errors in the Lanczos algorithm which allows to understand the numerical behaviour of the Lanczos method. Despite the common wisdom of that time, he clarified that the Lanczos method can be used as a reliable and efficient numerical tool for computing highly accurate approximations of dominant eigenvalues of large sparse matrices.

Since the detail exposition of his work is far beyond the scope of this thesis, we just give the summary of the most important results:

- Small $\delta_{n+1}|\zeta_{i,n}^{(n)}|$ really implies that $\theta_i^{(n)}$ is close to some eigenvalue of A, regardless the size $\left\|x_i^{(n)}\right\|$, i.e., that the last elements of the eigenvectors of the computed T_n indeed reliably tell us how well the eigenvalues of A are approximated by Ritz values.

- Until $\delta_{n+1}|\zeta_{i,n}^{(n)}|$ is very small, the scalar product of v_{n+1} and the Ritz vector $x_i^{(n)}$ is small. In combination with the first point it says that the orthogonality can be lost only in directions of converged Ritz values. In other words, v_{n+1} does not exhibit any substantial loss of orthogonality to Ritz vectors. And if so, then it must be Ritz vector associated with converged Ritz value.

- In contrast to this, there is no proof that the convergence of Ritz value is necessary accompanied by the loss of orthogonality. The relation between loss of orthogonality and the convergence of Ritz values is not equivalence but only implication where the sufficient condition is the loss of orthogonality and the necessary condition is the convergence of Ritz value.

- Until the loss of orthogonality the numerical behaviour of the Lanczos algorithm is nearly the same is the same as of the Lanczos algorithm with full reorthogonalization. Within the small inaccuracy, the computed Ritz values all lie between extreme eigenvalues of A. The last statement justifies the use of upper bound (3.6.6) also in finite precision arithmetic.

3.7.2 Backward-like analysis of Greenbaum

Another fundamental step in the analysis of the numerical behaviour of the Lanczos and CG methods was made in 1989 by Anne Greenbaum; see the original paper [11]. She has proved that for given number of iterations k, the behaviour in the first k steps of the finite precision Lanczos and (with a small inaccuracy) CG computations is identical to the behaviour of exact Lanczos and CG computations applied to a particular matrix $\bar{A}(k)$. Matrix $\bar{A}(k)$ has more eigenvalues than A, but these eigenvalues all lie within tiny intervals about the eigenvalues of A. In addition, in [28, Theorem 4.2] it is proved that for any eigenvalue λ of A for which the corresponding eigenvector has a non-negligible component in the starting vector (in the Lanczos algorithm) or in the initial residual (in the CG algorithm), the matrix $\bar{A}(k)$ has at least one eigenvalue close to λ.

In [12], a joint work of the authors of [11] and [28], it is numerically demonstrated that the behaviour of finite precision Lanczos and CG computations is very similar to the behaviour of exact Lanczos and CG computations applied to matrix $\widehat{A}$ which has many eigenvalues clustered in tiny intervals about each eigenvalue of A. In comparison with matrix $\bar{A}(k)$, the matrix $\widehat{A}$ does not depend on the number of iterations. For detail explanation see [21, Section 4.3, Section 5.2], [31, Section 5] or original papers [11, 28, 12].

3.7.3 Consequences to the CG method

In exact arithmetic there is no difference between computed elements of Jacobi matrix T_n by the Lanczos algorithm or by the CG algorithm and identities between coefficients (3.3.20). In the finite precision arithmetic, these elements are generally different but it is known (see e.g. [21, Theorem 5.1]) that these differences are small and thus the relationship between the CG and Lanczos algorithms is preserved except small inaccuracy. As we stated in the previous section, the relation between energy norm of the error and the remainder of the Gauss-Christoffel quadrature (3.6.7) is also preserved up to small inaccarucy.

As a consequence of the analysis by Greenbaum we can study the numerical behaviour of the CG method through the analysis of exact computations applied to different problem. Thus not only CG in exact arithmetic but also CG in finite precision can be related to Riemann-Stieltjes integral and Gauss-Christoffel quadrature as before, related distribution function has more points of increase and they are clustered.

The loss of orthogonality between residual vectors implies that the computed Krylov subspaces do not have their full dimension and thus there is a delay of convergence in CG computations. Using the results of Greenbaum, it is possible to reveal that these rank deficiencies are determined by the number of multiple copies of the original eigenvalues.

Consequently, it is convenient to measure the difference between exact and FP arithmetic horizontally, i.e., it is convenient to compare the number of needed iterations for given level of accuracy.

3.8 Comparison of CSI and CG

In this section we would like to compare the CSI and CG methods and to show the principle difference between stationary iterative methods and Krylov subspace methods. Results presented in this section are from [8, Section 5.5].

The CSI method was derived from the basic iterative scheme $x_{n+1} = Gx_n + g$, where G is a result of matrix splitting. In order to be able to compare the methods, we will consider simple splitting $G = I - A$. Then both methods for solving the linear system $Ax = b$ with a SPD matrix A can be written as

$$x_{n+1} = x_{n-1} + \omega_{n+1}(\delta_n r_n + x_n - x_{n-1}), \tag{3.8.1}$$

where parameters $\omega_{n+1},\ \delta_n$ depend on the given method.

Both methods give in exact arithmetic solution in at most N steps. The CSI method has the advantage that it does not require computing any inner products during the iterations. This can be important for parallel computing. On the other hand, for the use of the CSI method we need to have estimates of the extreme eigenvalues of the matrix of coefficients. However, the main difference between methods is that CG takes into account the distribution of all eigenvalues (see (3.6.3)) but the CSI method takes into account only information about extreme eigenvalues (see (2.2.8)).

The minimax problem (3.6.4) in the CG method is considered over a discrete set of points, whereas the minimax problem (2.2.8) in the CSI method is over the entire interval. From that point of view we can consider the convergence rate of the CSI method as an estimate (due to the superlinear convergence often very pessimistic) of the convergence rate of the CG method which shows the superiority of CG over the CSI method.

The superiority of the CG method can be also explained by the orthogonality. As a consequence of orthogonality of direction vectors the n-th CG approximation minimizes the energy norm of error over n-dimensional subspace. The CSI method does not satisfy any orthogonal criterion and consequently does not satisfy such strong minimalization property.

Difference between stationary iterative and Krylov subspace methods

Orthogonality is a key idea of all Krylov subspace methods. As a consequence, Krylov subspace methods find solution in at most N steps in exact arithmetic. In addition, orthogonal conditions can be often reformulated into some statements about optimality of computed approximations. Krylov subspace methods search for approximation in Krylov subspaces which naturally extracts the information about the behaviour of system matrix and thus they go insight the structure of problem.

Conversely, stationary iterative methods do not exhibit any optimality properties. The concept of matrix splitting can not in principle extract dominant information contained in the matrix of coefficients. On the other hand, they are typically easy to derive and implement. However, their convergence to solution is only asymptotically which may cause a lot of problems; see Subsection 2.1.4.

The difference between Krylov subspace methods and stationary iterative methods is fundamental. Krylov subspace methods try to extract as many information about the behaviour of the system as possible and then they use these information to generate approximation with some kind of optimal property.

3.9 Preconditioning

In solving real world problems, the convergence of iterative methods may be very poor and preconditioning is necessary in order to attain convergence in a reasonable amount of time. Roughly said, preconditioning transforms the original linear system into another system which has more favorable properties for iterative solution. A preconditioner is a matrix which performs such transformation. Preconditioning, as it is understood nowadays, tries to improve the spectral properties of the matrix of coefficients. We understand that it represents a fundamental and unavoidable part of practical computations. However, this thesis is not focused on the analysis of preconditioners or preconditioned systems and thus we present only the main ideas of preconditioning. These ideas are applicable not only to CG algorithm. This section is based on a well written survey [1] and on [26, Chapters 9 and 10].

Consider linear system $Ax = b$ and assume that M is a regular matrix which in some sense approximates the matrix A. The preconditioner M should be chosen such that it is easy to solve linear systems $Mx = b$ because they must be solved in every iteration of all preconditioned algorithms. It is possible to precondition the system from the left, from the right or we can split the preconditioner, i.e.,

$$M^{-1}Ax = M^{-1}b \tag{3.9.1}$$

$$AM^{-1}y = b, \quad x = M^{-1}y \tag{3.9.2}$$

$$M_1^{-1}AM_2^{-1}y = M_1^{-1}b, \quad x = M_2^{-1}y, \quad M = M_1M_2. \tag{3.9.3}$$

There are two main approaches to constructing preconditioners. The application-specific approach is popular mainly in the applications involving PDEs and it construct preconditioners convenient for a narrow class of problems. The construction of such a preconditioner often requires knowledge of the original PDE problem. These preconditioners may be very effective but typically only on very small class of problems and they may be also very sensitive to the details of the problem. Conversely, purely algebraic methods are universally applicable and they use only information contained in the matrix A. They are not optimal to any particular problem but they are often easier to develop. We restrict ourselves to algebraic preconditioners and we mention basic principles of two main approaches. For more detailed review see [1] and for extensive exposition about preconditioning see [26, Chapters 9–12].

Incomplete LU factorization methods

Gaussian elimination applied to sparse matrices usually cause serious fill-in of factors L and U. However, if we impose some constraint on the level of fill-in we can obtain quite powerful preconditioners in the form $M = \bar{L}\bar{U}$, where $\bar{L}$ and $\bar{U}$ are incomplete LU factors.

Which nonzero elements of the LU factorization will be omitted to avoid fill-in is based on several different criteria, such as position, value, or their combination.

First possibility is to prescribe directly a set of elements which can be filled-in. This family of preconditioners is called ILU. If the set of prescribed nonzero elements is the same as a structure of nonzeros of A then we obtain method of no-fill ILU factorization (ILU(0)). However, no-fill preconditioners are often very poor approximations of A and thus preconditioners with some additional fill-in are often constructed. The idea of preconditioner ILU(p) is to add an information about level of fill to every element processed by Gaussian elimination. Then there is some criterion which computes the level of fill and all elements with larger level of fill than p are discarded. ILU(1) is the most common variant and the structure of nonzeros is given by the nonzeros of product of the factors L and U from the ILU(0). These preconditioners ignore numerical values which may cause difficulties in many applications.

Another possibility is to discard elements which are smaller than prescribed value of drop tolerance τ. However, it is difficult to choose a good value τ and it is not possible to predict storage requirements. Thus it may be convenient to combine both criteria. Preconditioner ILUT(τ, p) discard all elements which are smaller than multiplication of τ with the Euclidean norm of the appropriate row and from the remaining nonzeros keep p largest ones.

Sparse approximate inverses

The inverse of sparse matrix is usually dense but often with many small elements. Thus the idea of preconditioning techniques based on sparse approximate inverses is to explicitly compute sparse approximation of the inverse of A and use it as a preconditioner M.

The main advantage of this approach is that it is easy to parallelize. These techniques are also remarkably robust. There are many different techniques and completely different algorithms for computing a sparse approximate inverse and each of them has its weaknesses and strengths. Here we will mention only the principle of Frobenius norm minimization approach, interested reader is referred to [1] or [26, Section 10.5] for more information about sparse approximate inverse technique.

Frobenius norm minimization technique is based on the solution of the constrained minimization problem

$$\min_{M \in \mathcal{S}} \|I - AM\|_F\,, \tag{3.9.4}$$

where $\mathcal{S}$ is a set of sparse matrices and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. Since

$$\|I - AM\|_F^2 = \sum_{i=1}^{N} \|u_i - Am_i\|_2^2\,, \tag{3.9.5}$$

we can compute the approximate inverse M by solving N independent linear least square problems. That would be too costly but once $\mathcal{S}$ is given, using the sparsity pattern of $\mathcal{S}$ allows to solve much smaller unconstrained least squares problems instead of original constrained least squares problems. Consequently, the computation of M can be implemented efficiently.

4 Numerical experiments

The idea of support preconditioning was inspired by the works of Axelsson and Jennings (see e.g. [15]). The support preconditioning (see [2]) is a method of preconditioning which aims at nearly linear time cost of the CG computations. Analysis is based on the distribution of eigenvalues. The properties of the spectrum are used to propose upper bounds of the error in the CG computations (see e.g. [27]), which are more explanatory than the classical linear bound based on the minimax problem over the interval.

However, these works are based on exact arithmetic and they do not take into account the effects of rounding errors. In the presence of large outliers, the approach gets into significant troubles.

In our experiment we will illustrate that in finite precision arithmetic (FP) the proposed estimates are in general useless. This statement can also be justified from the proper theoretical analysis of FP computation. It should be noted that Jennings in [15] observed during his FP computations some troubles. However, at that time, the analysis of FP computations using CG was not yet developed and it could not explain these troubles satisfactorily.

4.1 Theoretical background

4.1.1 Idea of the proposed upper bound

As before, consider a symmetric positive definite matrix $A \in \mathbb{R}^{N\times N}$, right hand side b and the problem $Ax = b$. Recall that

$$||e_k||_A^2 = \min_{p\in\Pi_k} ||p(A)e_0||_A^2 \tag{4.1.1}$$

$$= \min_{p\in\Pi_k} \left\{ \sum_{i=1}^{N} \frac{(r_0, q_i)^2}{\lambda_i} p^2(\lambda_i) \right\} \tag{4.1.2}$$

$$\frac{||e_k||_A}{||e_0||_A} \leq \min_{p\in\Pi_k} \max_{i=1,\dots,N} |p(\lambda_i)| \leq \tag{4.1.3}$$

$$\leq \min_{p\in\Pi_k} \max_{\lambda\in[\lambda_1,\lambda_N]} |p(\lambda)|, \tag{4.1.4}$$

where $A = Q\Lambda Q^T$ and q_i denotes the i-th column of Q. Set of polynomials p of degree k and scaled to have value 1 at the origin is denoted as Π_k^0.

Set $\xi = 0,\ 0 < a < b$ and recall the solution of the classical minimax problem (1.5.14) on the interval $[a, b]$.

$$\mathrm{C}_k^{[a,b]}(\lambda) = \arg\min_{p\in\Pi_k} \max_{\lambda\in[a,b]} |p(\lambda)|, \tag{4.1.5}$$

where

$$\mathrm{C}_k^{[a,b]}(\lambda) = \frac{\mathrm{T}_k\left(\dfrac{2\lambda - b - a}{b - a}\right)}{\mathrm{T}_k\left(\dfrac{a + b}{a - b}\right)}. \tag{4.1.6}$$

Polynomial T_k is the k-th Chebyshev polynomial of the first kind and $\mathrm{C}_k^{[a,b]}$ is the Chebyshev polynomial shifted on the interval $[a, b]$ and scaled to have value 1 at the origin. Also have in mind that for $a = \lambda_1, b = \lambda_N$ the solution of the previous minimax problem gives the classical upper bound for the rate of the convergence,

$$\frac{||e_k||_A}{||e_0||_A} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k, \tag{4.1.7}$$

where $\kappa = \lambda_N/\lambda_1$ is the condition number of a matrix A.

The idea of the support preconditioning is to transform the spectrum of the matrix to a spectrum, which has not necessarily significantly smaller condition number, but which concentrates $N - m$ eigenvalues in an interval of small condition number and remaining m eigenvalues are out of this interval. These eigenvalues are called outliers.

Proposed upper bounds are based on the polynomial R_k^m

$$R_k^m(\lambda) = \mathrm{C}_{k-m}^{[\lambda_1,\lambda_{N-m}]}(\lambda)\left(1 - \frac{\lambda}{\lambda_{N-m+1}}\right)\dots\left(1 - \frac{\lambda}{\lambda_{N-1}}\right)\left(1 - \frac{\lambda}{\lambda_N}\right). \tag{4.1.8}$$

This polynomial is a product of the shifted and scaled Chebyshev polynomial of degree $k-m$ on the interval $[\lambda_1, \lambda_{N-m}]$ and a factor of degree m that is zero at each of the outliers and less than one in magnitude at each of the other eigenvalues.

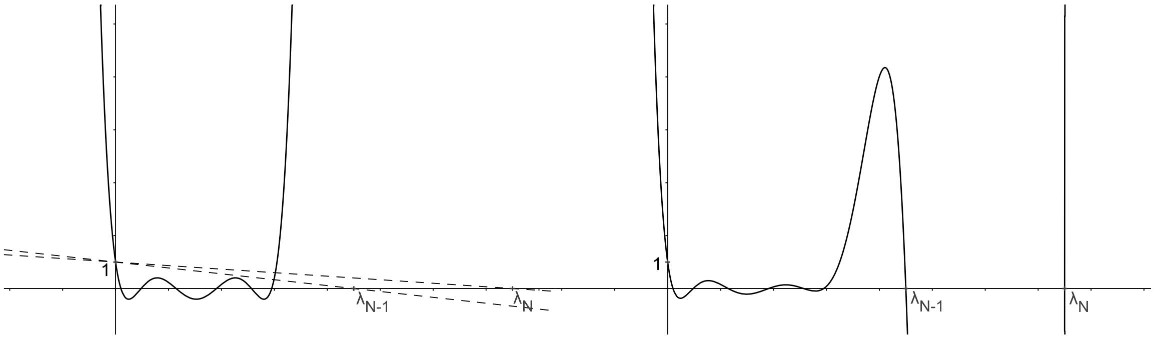

Figure 4.1: Illustration of the polynomial R_k^m from (4.1.8). Left: graphs of $\mathrm{C}_{k-m}^{[\lambda_1,\lambda_{N-m}]}$ (solid line) and of the linear polynomials with the roots at the outliers (dashed line). Right: a graph of R_k^m. Note, how steep is the polynomial near the outlying roots.

From the construction of the polynomial R_k^m it follows that the error satisfies

$$||e_k||_A^2 \leq \sum_{i=1}^{N-m} \frac{(r_0, q_i)^2}{\lambda_i}\left(\mathrm{C}_{k-m}^{[\lambda_1,\lambda_{N-m}]}(\lambda_i)\right)^2 \tag{4.1.9}$$

and the relative error satisfies

$$\frac{||e_k||_A}{||e_0||_A} \leq \max_{\lambda\in[\lambda_1,\lambda_{N-m}]}\left|\mathrm{C}_{k-m}^{[\lambda_1,\lambda_{N-m}]}(\lambda)\right|. \tag{4.1.10}$$

Using (4.1.6) we get

$$\frac{||e_k||_A}{||e_0||_A} \leq \frac{1}{\left|\mathrm{T}_{k-m}\left(\frac{\lambda_1+\lambda_{N-m}}{\lambda_1-\lambda_{N-m}}\right)\right|} \tag{4.1.11}$$

and using the same arguments as in derivation of the bound (4.1.7) we get

$$\frac{||e_k||_A}{||e_0||_A} \leq 2\left(\frac{\sqrt{\kappa_{N-m}}-1}{\sqrt{\kappa_{N-m}}+1}\right)^{k-m}, \tag{4.1.12}$$

where $\kappa_{N-m} = \lambda_{N-m}/\lambda_1$ can be substantially smaller than original κ. The smaller κ_{N-m} indicates asymptotically more rapid convergence . This improvement is not just for free. Note that in the k-th iteration we deal with Chebyshev polynomial of degree $k-m$. There is a delay of m iterations which are necessary to zero out the large outlying eigenvalues.

To sum up, since

$$\begin{aligned}\max_{i=1,\dots,N} |R_k^m(\lambda_i)| = &\max_{i=1,\dots,N-m} |\mathrm{C}_{k-m}^{[\lambda_1,\lambda_{N-m}]}(\lambda_i)| \leq & (4.1.13)\\ \leq &\max_{\lambda\in[\lambda_1,\lambda_{N-m}]} |\mathrm{C}_{k-m}^{[\lambda_1,\lambda_{N-m}]}(\lambda)| \leq & (4.1.14)\\ \leq &2\left(\frac{\sqrt{\kappa_{N-m}}-1}{\sqrt{\kappa_{N-m}}+1}\right)^{k-m},\end{aligned}$$

the construction of the polynomial R_m^k and usage

$$\max_{i=1,\dots,N} |R_k^m(\lambda_i)| \tag{4.1.15}$$

as an upper bound for the relative error computed in the energy norm gives a significant improvement of the asymptotic rate of convergence.

All this is true, however, *only in exact arithmetic.* We will show and explain, why this approach *must* fail in FP arithmetic.

4.1.2 Consequences of the backward-like analysis

Now we will prove that there is absolutely no reason to consider (4.1.15) as an upper bound for the relative error computed in energy norm of finite precision CG computations. Main ingredient for our explanation will be the results of the backward-like analysis of Anne Greenbaum, see Subsection 3.7.2. For arbitrary and fixed number of iterations k there exists matrix $\bar{A}(k)$ such that the behaviour of FP CG computations applied on A is nearly identical with the behaviour of exact CG computations applied on $\bar{A}(k)$. So it is the same to analyse suitability of (4.1.15) for FP CG computations applied on A and for exact CG computations applied on $\bar{A}(k)$. The upper bound (4.1.15) take into consideration the eigenvalues of A but the matrix $\bar{A}(k)$ has many more eigenvalues. Now we see, that (4.1.15) has nothing in common with the exact CG computations applied on $\bar{A}(k)$ because it does not take into consideration all eigenvalues of $\bar{A}(k)$. Thus (4.1.15) is not an upper bound for the exact CG computations applied on $\bar{A}(k)$ in general.

Since the first k iterations of FP CG computations applied on A can be viewed as the first k iterations of exact CG computations applied on $\bar{A}(k)$, we have shown that it can not be guaranteed that in the k-th iteration (4.1.15) gives an upper bound for the relative error computed in energy norm of finite precision CG computations.

An appropriate upper bound must take into consideration all eigenvalues of $\bar{A}(k)$ and thus

$$\max_{\lambda \in \sigma(\bar{A}(k))} |R_k^m(\lambda)|, \tag{4.1.16}$$

where $\sigma(\bar{A}(k))$ is a spectrum of $\bar{A}(k)$, is the appropriate upper bound. However, in the Section 4.3 we will demonstrate that clustered eigenvalues and the properties of the Chebyshev polynomials cause that this upper bound is totally useless.

The unsuitability of (4.1.15) will be numerically demonstrated in the Section 4.3. In our numerical experiments we construct matrix $\widehat{A}$ with many eigenvalues spread throughout tiny intervals about eigenvalues of A. In compliance with [12], the behaviour of FP CG computations applied on A is very similar to the behaviour of exact CG computations applied on $\widehat{A}$ and thus we can numerically demonstrate the unsuitability of (4.1.15) using matrix $\widehat{A}$ instead of matrix $\bar{A}(k)$.

4.2 Description of the experiment

4.2.1 Quantities to be observed

Figures show comparison among the convergence of exact and finite precision (FP) CG computations applied to diagonal matrices A specified in the next subsection and the proposed upper bound of that convergence. The exact CG computations is simulated by saving residual vectors and applying double full reorthogonalization at each iteration. The relative error computed in energy norm, i.e., the A-norm of the error at each iteration divided by the A-norm of the initial error

$$\frac{||x - x_k||_A}{||x - x_0||_A}$$

is plotted. The exact solution x is approximated using functions of Matlab as $x = A^{-1}b$. The approximation x_k is computed using the routine `cglan`, this routine allows to reorthogonalize residual vectors and thus it is used also for simulation of exact CG computations. The convergence of FP CG computations is plotted as a *solid line*, the convergence of exact CG computations is plotted as *dash-dotted line.* The proposed upper bound

$$R(A, k, m) \equiv \max_{i=1,\ldots,N} |R_k^m(\lambda_i)| \tag{4.2.1}$$

from the inequality (4.1.10) is plotted as *dots.*

We will see that the proposed estimate is not an upper bound for the energy norm of the relative error in FP arithmetic. In order to illustrate why it is so, we will use the relationship between FP CG computations applied to linear system $Ax = b$ and exact CG computations applied to larger linear system $\widehat{A}\widehat{x} = \widehat{b}$, where $\widehat{A}$ is a matrix with many eigenvalues spread throughout tiny intervals about the eigenvalues of A. The construction of the matrix $\widehat{A}$ and the choice of the right hand side $\widehat{b}$ is discussed in the next subsection. Exact CG computation of this larger problem is also simulated by applying double reorthogonalization on the residual vectors at each iteration of the CG algorithm (routine `cglan`). The relative error computed in energy norm, i.e., the $\widehat{A}$-norm of the error at each iteration divided by the $\widehat{A}$-norm of the initial error

$$\frac{||\widehat{x} - \widehat{x}_k||_{\widehat{A}}}{||\widehat{x} - \widehat{x}_0||_{\widehat{A}}} \tag{4.2.2}$$

is plotted as a *dotted line*. The exact solution $\widehat{x}$ is approximated using functions of MATLAB as $\widehat{A}^{-1}\widehat{b}$. Similarly as x_0, the initial approximation $\widehat{x}_0$ is set as zero. In analogy with (4.1.16) in Subsection 4.1.2, the polynomial R_k^m is used to generate an upper bound for the relative error (4.2.2). This upper bound

$$R(\widehat{A}, k, m) \equiv \max_{\lambda \in \sigma(\widehat{A})} |R_k^m(\lambda)|, \tag{4.2.3}$$

where $\sigma(\widehat{A})$ is a spectrum of matrix $\widehat{A}$, is plotted as a *dashed line*.

4.2.2 Input data

We perform our experiments on a linear system with right hand side b of ones and with diagonal symmetric positive definite matrix $A \in \mathbb{R}^{N \times N}$. The initial approximation x_0 is set as zero. We deal with two different types of spectrum of matrix A. However, both types are just slight modifications of a spectrum designed in [28]:

Spectrum$(N, \lambda_1, \lambda_N, \rho)$

Given λ_1 and λ_N, we generate the inner eigenvalues by the formula

$$\lambda_i = \lambda_1 + \frac{i-1}{N-1}(\lambda_N - \lambda_1)\rho^{N-i} \quad i = 2, \ldots, N-1.$$

For $\rho = 1$ the spectrum is distributed uniformly. For $\rho < 1$ the eigenvalues tend to cumulate near λ_1. Parameter $\rho \in (0, 1]$ determines the non-uniformity of the spectrum.

This type of spectrum is valuable for the analysis of convergence in finite precision arithmetic, as was shown in [28].

Matrix01$(n, m, \lambda_1, \lambda_N, \rho_1, \rho_2)$

This diagonal matrix has dimension $N = n + m$ and its spectrum is created in two steps. In the first step we distribute N eigenvalues in the interval $[\lambda_1, \lambda_N]$ using `Spectrum(`$N, \lambda_1, \lambda_N, \rho_1$`)`.

In the second step we consider the m largest eigenvalues as outliers and let them unchanged. Remaining n eigenvalues are redistributed using `Spectrum-(`$n, \lambda_1, \lambda_{N-m}, \rho_2$`)`.

Note that for $\rho_2 = \rho_1$ the second step does not change the distribution of the eigenvalues and they are distributed just as in `Spectrum(`$N, \lambda_1, \lambda_N, \rho_1$`)`.

Matrix02$(n, m, \lambda_1, \lambda_n, \rho, out_a, out_b)$

This diagonal matrix has dimension $N = n + m$. The first n eigenvalues are distributed using `Spectrum(`$n, \lambda_1, \lambda_n, \rho$`)`. Remaining m eigenvalues represent the outliers and are distributed uniformly in the interval $[out_a, out_b]$. We consider the case $\lambda_n < out_a$. It allows us to denote these m eigenvalues as $\lambda_{n+1}, \ldots, \lambda_{n+m}$. If $m = 1$ then $\lambda_{n+1} = out_a$.

blurring($exp, count$)

In our numerical experiment we compare the finite precision CG computations for $Ax = b$ to the exact CG computation for $\widehat{A}\widehat{x} = \widehat{b}$ For a given matrix A of dimension N we construct larger diagonal matrix $\widehat{A}$ of dimension $N \times count$ with clustered eigenvalues about the eigenvalues of A (routine `blur`). The matrix $\widehat{A}$ is uniquely defined by the parameters *exp* and *count*. The parameter *count* determines the number of eigenvalues in each cluster. The eigenvalues in each cluster are uniformly distributed in the tiny interval of width 2×10^{-exp}. We will use a notation `blurring(`$exp, count$`)` to specify the matrix $\widehat{A}$ for given matrix A.

The right hand side

$$\widehat{b} = (\widehat{\beta}_{1,1}, \ldots, \widehat{\beta}_{1,count}, \widehat{\beta}_{2,1} \ldots, \widehat{\beta}_{2,count}, \ldots, \widehat{\beta}_{n,1}, \ldots, \widehat{\beta}_{n,count})^T$$

is chosen as in [12]:

$$\widehat{b}_{i,1} = \widehat{\beta}_{i,2} = \ldots = \widehat{\beta}_{i,count} \quad \text{and} \quad \sum_{j=1}^{count} (\widehat{\beta}_{i,j})^2 = \beta_i^2, \quad \text{for} \quad i = 1, \ldots, n,$$

where $b = (\beta_1, \ldots, \beta_n)^T$.

4.2.3 Technical details on realization

All experiments were performed on a personal computer using MATLAB 7.11. Routines which were used are listed below with short characterization. Detail description is a part of the MATLAB code; see also Appendix A.

main This routine sets all parameters which are necessary for the experiment, calls subroutines to get results and plots them.

cglan This routine was taken over from the pack of software for [23]. It represents the merits of the computation. It contains the CG algorithm for computing an approximation x_k. It is also possible to run CG algorithm with the reorthogonalization of residual vectors, this variant is used for simulating the exact CG computations. The original routine was modified in order to get the relative error of the approximation computed in energy norm as an output.

ortho_poly This routine computes the values of the Chebyshev polynomials of the first kind. It was downloaded from the web sites `www.mathworks.com`.

cheb_on_interval This routine computes the proposed upper bounds $R(A, k, m)$ and $R(\widehat{A}, k, m)$.

blur This routine is used to modify the original spectrum to a larger spectrum with clusters of eigenvalues around the original eigenvalues. This modification depends on how tight are the clusters and how many eigenvalues create each cluster. The number of eigenvalues in each cluster is determined by the variable `count`. The eigenvalues in each cluster are uniformly distributed in the tiny interval of width 2×10^{-exp}, where `exp` is a parameter of the experiment. We will denote the modified matrix with clusters as $\widehat{A}$.

4.3 Computed results

Our observations are summarized in several points.

a) Delay of convergence in FP computation cause that the proposed estimate is not in general an upper bound and that it can be totally useless. An explanation is based on the results of the backward-like analysis presented before.

b) Since there is a strong relationship between finite precision CG computations and exact CG computations applied to a matrix with clustered eigenvalues, we can try to use an upper bound derived for the latter problem in order to attain an upper bound for the FP CG computations. However, this upper bound is absolutely useless because after some number of iterations it starts to grow very fast. This effect will be called *blow-up*.

c) The proposed estimate might be unsuitable also for small and well conditioned problems, its unsuitability is a matter of the mathematical principle. For larger problems the effect is typically more visible and the consequences are more serious.

d) Violation of the validity of the proposed estimate can be observed also for small numbers of outliers. Several examples with just one outlier are plotted.

e) The proposed estimate does not reflect the distribution of the outliers at all. On the contrary, finite precision computation is strongly affected by the particular distribution of the outliers. This again illustrate, why is the proposed estimate unsuitable.

f) Even a small change of the parameter of the distribution of the eigenvalues inside the interval $[\lambda_1, \lambda_{N-m}]$ can cause a dramatic differences in the rate of the CG convergence and thus influence the suitability of the proposed estimate. This observation coincides with a sensitivity of FP CG to a small change of the distribution of the eigenvalues, see [28].

A detailed discussion and attempt of explanation is given in the following paragraphs.

Point a) Using $R(A, k, m)$ as an upper bound was justified assuming the exact arithmetic. However, FP computation is affected by rounding errors and as a consequence there is a delay of convergence. In order to reach some accuracy level, FP CG computation may need many more iterations then hypothetical exact CG computation. In Figure 4.2 we plot the convergence of FP CG and exact CG computations applied to diagonal matrices given by `Matrix01`-$(60, 12, 0.1, 1000, 0.7, 0.95)$ and `Matrix02`$(24, 5, 1, 2, 0.9, 10, 50)$. There is a significant difference between the FP (solid line) and the exact (dash-dotted line) CG computations. The idea of using $R(A, k, m)$ for m outliers is simple. The more eigenvalues are considered as outliers, the smaller is the interval over which the properly shifted and scaled Chebyshev polynomial is evaluated and consequently the more strict is the decrease of the upper bound. On the other hand, the degree of the Chebyshev polynomial decreases with increasing m. To show how these estimates work, we plot $R(A, k, m)$ for $m = 0, \ldots, 20$ (dashed lines) in both figures. Note that there is a huge improvement in Figure 4.2 (right) of the proposed upper bound for $m = 5$. That is because $\lambda_{24} = 2$ is much more smaller than $\lambda_{25} = 10$ and thus the improvement of the active condition number is substantial.

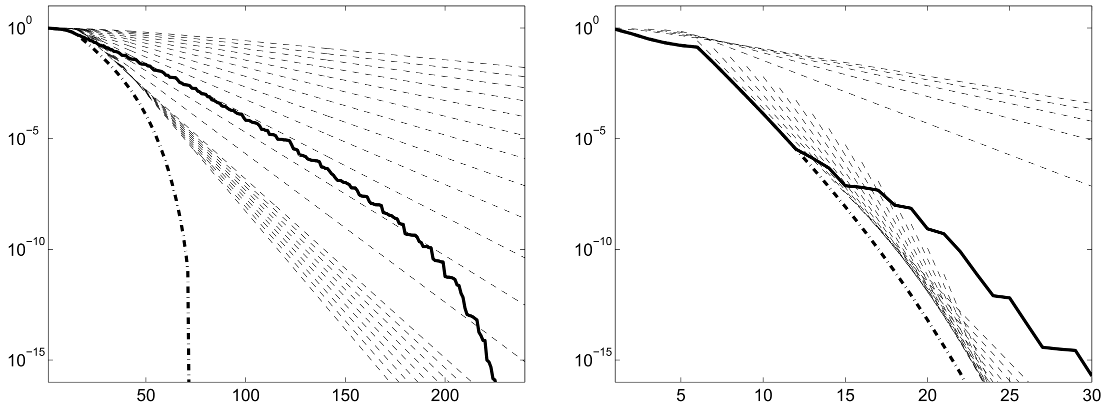

Figure 4.2: Matrices given by: left: `Matrix01`$(60, 12, 0.1, 1000, 0.7, 0.95)$; right: `Matrix02`$(24, 5, 1, 2, 0.9, 10, 50)$. Last $m = 0, \ldots, 20$ eigenvalues are considered as the outliers. We see that $R(A, k, m)$ (represented by dashed lines) is an upper bound for the exact CG algorithm (dash-dotted line) for all m. However, for large m, it is not an upper bound for the FP CG algorithm (solid line).

We see that $R(A, k, m)$ gives an upper bound for the exact CG computation for all m and that for large k close to the size of the problem is the upper bound more tight with increasing m. However, that is not the case with the FP CG computation. We see that for more than 12 outliers for the first matrix and for more than 5 outliers for the second one, the proposed estimate is no more an upper bound for the relative error in finite precision computation.

In Subsection 4.1.2 we have proved that there is *no guarantee that the proposed estimate is an upper bound.* Now we will explain, why in many cases *the proposed estimate is actually not an upper bound.* The explanation is based on the relationship between the Lanczos method for approximation of the eigenvalues and the CG algorithm for solving linear system. It is proved (see for example [21]) that, up to a small inaccuracy, the relationship holds also in FP computation.

Using a language of the Lanczos algorithm, the estimate $R(A, k, m)$ can be for well separated outliers interpreted as follows: If the first m steps are used for approximation of the m largest eigenvalues, the further iterations deal with the rest of the spectrum. Using a language of the CG algorithm viewed as Krylov subspace method, the estimate is based on the fact (which is true in exact arithmetic; see Subsection 3.6.2) that if the largest eigenvalues are well approximated, the CG behaves in the subsequent steps as if they are not present in the problem.

However, as it is written in Section 3.7, the Lanczos algorithm in FP arithmetic tends to generate multiple copies of the eigenvalues which were approximated earlier. As a consequence, there is a delay of convergence in the FP CG computation. In other words we can not guarantee that we need only m iterations to deal with the outliers. We must take into account also the iterations in which the multiple copies are repeatedly formed.

The previous thoughts are relevant and tell us that we must be careful, but we can not use them as quantitative arguments for proper analysis. However, the backward-like analysis gives a theoretical background for our observations and gives a mathematically rigorous arguments to prove the unsuitability of the proposed upper bound $R(A, k, m)$ as it is done in Subsection 4.1.2.

We have shown that there is no reason to relate the relative error of the FP CG computations computed in the energy norm with the estimate $R(A, k, m)$. The problem is clearly fundamental. It is not a question of particular spectrum of a matrix, it is a question of principle of behaviour of CG in finite precision computations. Analysis of finite precision computations can not be based on bounds derived for the behaviour of exact computations.

The unsuitability of $R(A, k, m)$ is demonstrated also in the following figures which are focused on some other specifics of our numerical experiment.

Point b) Consider matrix $\widehat{A}$ so that the exact CG computations for the larger problem with $\widehat{A}$ can be used for the analysis of the FP computations of the original problem with A. We know that $R(\widehat{A}, k, m)$ is an upper bound for the exact CG computation for $\widehat{A}\widehat{x} = \widehat{b}$, but it is not a useful upper bound. Actually, after some number of iterations a blow-up occurs. This blow-up is a consequence of the clustered eigenvalues about eigenvalues of A. As it was shown in Figure 4.1.1, polynomial R_k^m is very steep in the nearby of the outlying eigenvalues. Although there is a small distance (approximately 10^{-exp}) between eigenvalues $\widehat{\lambda}$ in the cluster and the original eigenvalue λ, for increasing k $|R_k^m(\widehat{\lambda})|$ grows very fast for $\widehat{\lambda} \neq \lambda$ in the cluster around the original eigenvalue λ, because of the growth of the Chebyshev polynomials $\mathrm{T}_s(\xi)$ with increasing s for $\xi \notin [-1, 1]$. Consequently, there will be a *blow up* of the upper bound $R(\widehat{A}, k, m)$.

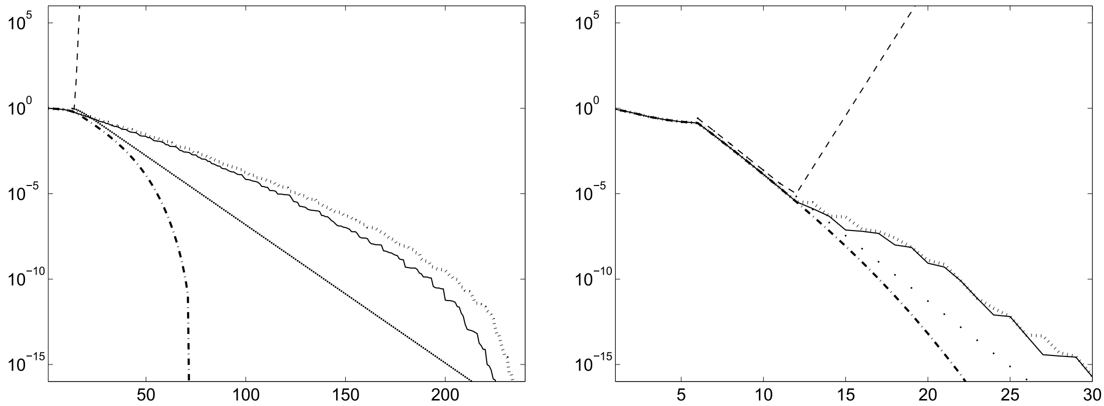

Figure 4.3: Left: `Matrix01`$(60, 12, 0.1, 1000, 0.7, 0.95)$, 12 largest eigenvalues are considered as outliers, the matrix $\widehat{A}$ is given by `blurring`$(12, 15)$. Right: `Matrix02`$(24, 5, 1, 2, 0.9, 10, 50)$, 5 outliers, `blurring`$(14, 11)$. The proposed upper bound $R(A, k, m)$ is plotted by dots, "corrected" upper bound $R(\widehat{A}, k, m)$ as dashed line, the FP CG as a solid line, the exact CG for $\widehat{A}$ as a dotted line and the exact CG for A as a dash-dotted line.

In Figure 4.3 we show convergence of the same problems as in Figure 4.2. In Figure 4.3 (left) , $R(A, k, m)$ is plotted just for $m = 7$ (dots), to the plots of FP and exact CG computations we add plots of exact CG computations applied to $\widehat{A}$ (dotted line) and $R(\widehat{A}, k, 7)$ (dashed line). The matrix $\widehat{A}$ is given by `blurring(12,15)` which means that it has 11 eigenvalues uniformly distributed throughout each of N tiny intervals of width 2×10^{-12} about the eigenvalues of A. In Figure 4.3 (right), $R(A, k, 5)$ is plotted, to the plots of FP and exact CG computations we add plots of exact CG computations applied to $\widehat{A}$ and $R(\widehat{A}, k, 5)$. The matrix $\widehat{A}$ is given by `blurring(14,11)`.

Point c) The aim of this paragraph is to show that the proposed estimate is in general unsuitable for small problems as well as for large ones. Depending on the distribution of the eigenvalues of A, the effect of rounding errors can be very significant also for matrices with small condition numbers. For larger condition numbers the unsuitability of an estimate is more obvious. Trying to attain some level of accuracy, we can observe substantial difference between number of iterations predicted by the upper bound $R(A, k, m)$ and actually needed number of iterations for the FP CG computations. We will plot four figures in Figure 4.4 and Figure 4.5 to see that the estimate $R(A, k, m)$ can fail for 1) dimension n small, condition number κ small, 2) dimension n large, condition number κ large, 3) dimension n small, condition number κ large, 4) dimension n large, condition number κ small.

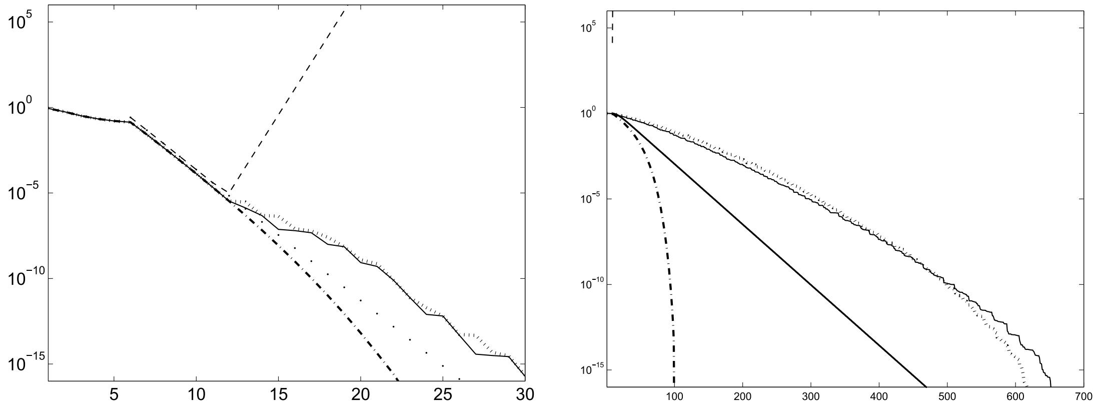

Figure 4.4: Left: `Matrix02(24,5,1,2,0.9,10,50)`, `blurring(14,11)`, $n = 29$, $\kappa = 50$, 5 outliers. Right: `Matrix01(92,8,0.1,10^6,0.3,0.95)`, `blurring(9,61)`, $n = 100$, $\kappa = 10^7$, 8 outliers.

In Figure 4.4 (left) we plot the proposed upper bounds and the convergence of CG computations applied to the matrix given by `Matrix02(24,5,1,2,0.9,10,50)`. That means that the first 24 eigenvalues are given by `Spectrum(24,1,2,0.9)` and 5 eigenvalues are distributed uniformly in the interval $[10, 50]$. In spite of the fact that this problem is very well conditioned ($\kappa = 50$), we see that there is a significant delay of convergence and that for $k > 12$ the upper bound $R(A, k, 5)$ can not be used for the FP CG computation. The upper bound predicts that the computation would converge to the level of accuracy 10^{-16} in less than 26 iterations but the FP CG computation actually needs 30 iterations. The difference seems negligible, but the upper bound undervalued the number of iterations by more than 13%. The FP CG computation is compared to the exact CG computation applied to the matrix $\widehat{A}$ given by `blurring`(14, 11).

In Figure 4.4 (right) we plot the proposed upper bounds and the convergence of CG computations applied to the matrix given by `Matrix01`$(92, 8, 0.1, 10^6, 0.3, 0.95)$. That means that 8 largest eigenvalues are distributed according to `Spectrum`$(100, 0.1, 10^6, 0.3)$ and `Spectrum`$(92, 0.1, \approx 60.5, 0.95)$ determines remaining 92 eigenvalues. This problem is poorly conditioned ($\kappa = 10^7$) and we see, that using $R(A, k, 8)$ as an upper bound for the FP CG computation is pointless. The upper bound predicts that the computation would converge in less than $\approx$ 470 iterations. However, the FP CG computation actually needs $\approx$ 650 iterations. If we would stop the computation after 470 iterations we would have a relative error of magnitude $\approx 10^{-9}$ which is greater by 7 orders than the norm of the relative error predicted by $R(A, k, 8)$. The matrix $\widehat{A}$ is given by `blurring`(9, 61).

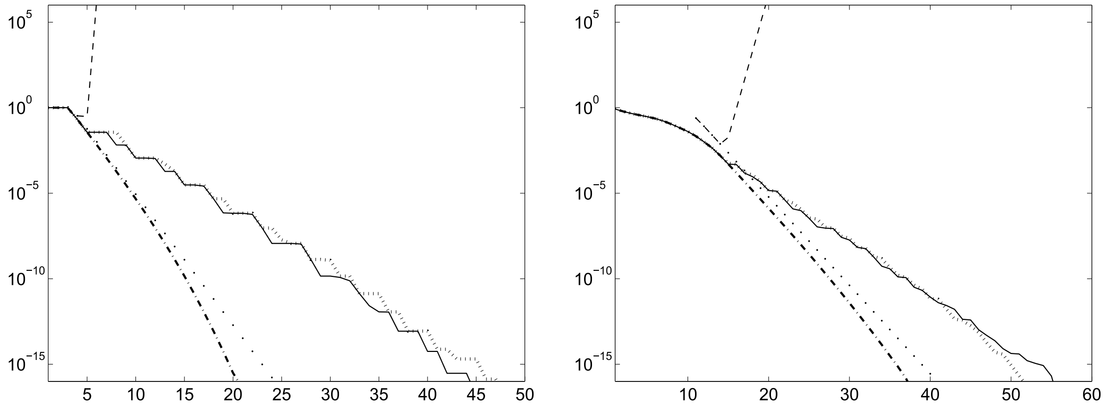

Figure 4.5: Right: `Matrix02`$(24, 3, 1, 2, 0.9, 10^6, 10^7)$, `blurring`(8, 11), $n = 27$, $\kappa = 10^7$, 3 outliers. Left: `Matrix01`$(90, 10, 1, 100, 0.7, 0.95)$, `blurring`(14, 11), $n = 100$, $\kappa = 100$, 10 outliers.

In Figure 4.5 (left) we plot the proposed upper bounds and the convergence of CG computations applied to the matrix given by `Matrix02`$(24, 3, 1, 2, 0.9, 10^6, 10^7)$, so there are 24 small eigenvalues in the interval $[1, 2]$ and 3 large eigenvalues uniformly distributed in the interval $[10^6, 10^7]$. This problem has small dimension ($N = 27$) and is poorly conditioned ($\kappa = 10^7$). Again we see that the convergence of the FP CG computation and $R(A, k, 3)$ have nothing in common. The estimate $R(A, k, 3)$ converges in 24 iterations but FP computation actually needs 45 iterations. If we would stop the computation after 24 iterations we would have a relative error of magnitude $\approx 10^{-8}$ which is greater by 8 orders than it was predicted by $R(A, k, 3)$. The matrix $\widehat{A}$ is given by `blurring`(8, 11).

In Figure 4.5 (right) we plot the proposed upper bounds and the convergence of CG computations applied to the matrix given by `Matrix01`$(90, 10, 1, 100, 0.7, 0.95)$. It can be considered as a large (dimension $N = 100$) and well conditioned ($\kappa = 100$) problem. As before, $R(A, k, 10)$ is not an upper bound for the FP CG computation and we can not use it in that way. The upper bound predicts that the computation would converge in less than 41 iterations but the FP CG computation actually needs 55 iterations. The matrix $\widehat{A}$ is given by `blurring`(14, 11).

Point d) In all paragraphs in this section we are trying to emphasize that the unsuitability of $R(A,k,m)$ is in the principle of that estimate. We will introduce two examples which will demonstrate that the estimate can fail even if we consider only the largest eigenvalue as an outlier. That will again show, that the unsuitability of proposed estimate is a matter of mathematical principle. The FP CG computations do not behave as the exact CG computations and we can not use the results of analysis for the exact CG also for the FP CG.

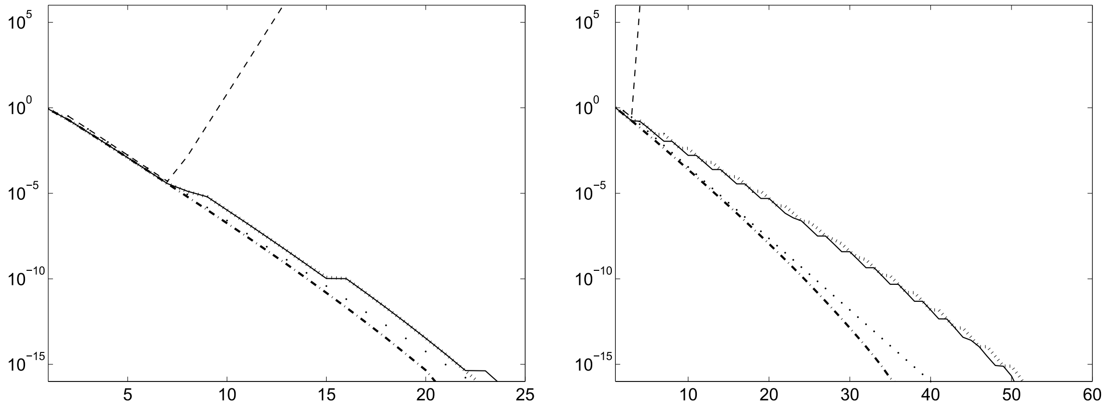

Figure 4.6: Comparison of $R(A,k,1)$ (dots) and the convergence of the FP CG computation (solid line); only the largest eigenvalue considered as outlier. Left: `Matrix02`$(24,1,1,2,1,100,100)$, `blurring`$(14,3)$. Right: `Matrix02`$(48,1,1,5,1,10^7,10^7)$, `blurring`$(7,20)$

In Figure 4.6 (left) we plot the proposed upper bounds and the convergence of CG computations applied to the matrix given by `Matrix02`$(24,1,1,2,1,100,100)$. This spectrum has 24 eigenvalues distributed uniformly in the interval $[1,2]$ and the largest eigenvalue is set to be 100. Although it is a small and well conditioned problem, we can see that delay of convergence between iterations 7 and 9 causes that $R(A,k,1)$ is no more an upper bound to the FP CG computation. The matrix $\widehat{A}$ is given by `blurring`$(14,3)$.

In Figure 4.6 (right) we plot the proposed upper bounds and the convergence of CG computations applied to the matrix given by `Matrix02`$(48,1,1,5,1,10^7,10^7)$. It is a larger system than the previous one, the most important difference is that the outlier is set to be very well separated from the rest of the spectrum and is equal to 10^7. Using a language of the Lanczos algorithm, the multiple copies of such a dominant eigenvalue are formed much more frequently. As a consequence, the behaviour of the FP CG computation is completely different from the behaviour of the exact CG computation and using $R(A,k,1)$ as an upper bound to the FP CG is worthless. The matrix $\widehat{A}$ is given by `blurring`$(7,20)$.

Point e) Because the polynomial R_k^m has the roots at the outliers, the upper bound $R(A,k,m)$ does not depend at all on a distribution of the outlying eigenvalues. It depends only on their number which determines a delay of the estimate based on the Chebyshev polynomials. On the contrary, FP computations do depend on that distribution. In fact, the distribution of the outlying eigenvalues can affect strongly the convergence of FP computations.

In Figure 4.7 we plot the proposed upper bounds and the convergence of CG computations applied to the matrix given by `Matrix02`$(48, 5, 0.1, 10, 0.9, a, b)$. Intervals $[a, b]$ are set as following: In Figure 4.7 (top-left) is $a = 20$ and $b = 30$, in Figure 4.7 (top-right) is $a = 5 \times 10^5$ and $b = 5 \times 10^5 + 2$, in Figure 4.7 (bottom-left) is $a = 10^5$ and $b = 5 \times 10^5$ and in Figure 4.7 (bottom-right) is $a = 10^6$ and $b = 5 \times 10^6$. The corresponding matrix $\widehat{A}$ is given subsequently: top-left: `blurring`$(12, 21)$, top-right: `blurring`$(10, 21)$, bottom-left: `blurring`$(9, 15)$, bottom-right: `blurring`$(9, 25)$.

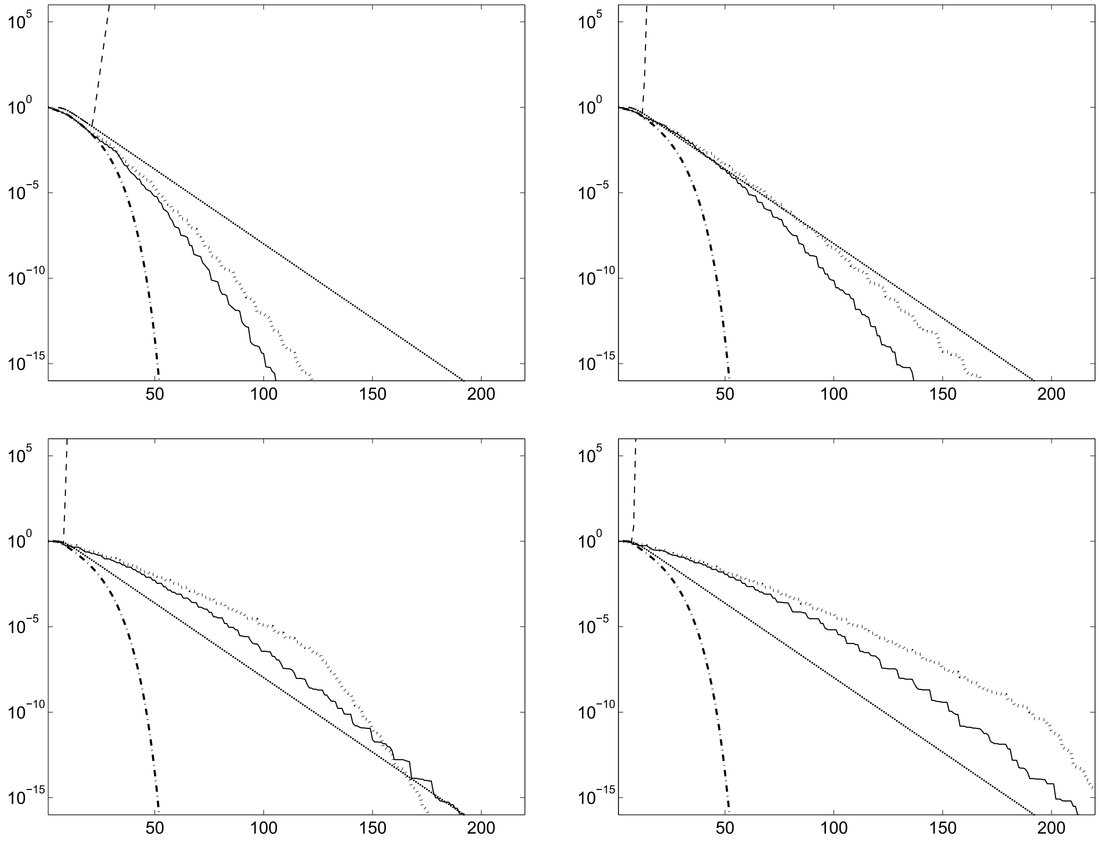

Figure 4.7: Convergence of the CG computations for the matrix given by `Matrix02`$(48, 0.1, 10, 0.9, a, b)$, where a and b are set as: Top-left: $a = 20,\ b = 30$; Top-right: $a = 5 \times 10^5,\ b = 5 \times 10^5 + 2$; Bottom-left: $a = 10^5,\ b = 5 \times 10^5$; Bottom-right: $a = 10^6,\ b = 5 \times 10^6$.

While the upper bound $R(A, k, m)$ and the exact CG computation are not affected by the changes of distribution of the outliers, the FP CG computation is affected strongly. We can observe that the rounding errors have greater effect for larger and more spread outliers. It is in a compliance with behaviour of the Lanczos algorithm in FP arithmetic. The larger eigenvalue, the more multiple copies will be formed and as a consequence there will be more serious delay of convergence of FP CG computations. On the other hand, if the outliers are concentrated around one point, the convergence is faster.

The insensitivity of the proposed upper bound to the distribution of the outliers give us another reason to deny $R(A, k, m)$ as an upper bound for FP CG computations.

Point f) In the last paragraph we have shown the insensitivity of $R(A,k,m)$ to the distribution of the outlying eigenvalues. Now we will focus on the distribution of the eigenvalues which are not outliers, i.e., which lies in the interval $[\lambda_1, \lambda_{N-m}]$. We will see that only a small change of the parameter of their distribution can cause a substantial difference of the convergence of FP computations. In [28] it was shown that matrices with spectrum given by $\texttt{Spectrum}(N, \lambda_1, \lambda_N, \rho)$ are very sensitive to the change of the parameter ρ. We will see (but it is not really surprising) that their modification, matrices $\texttt{Matrix01}(n, m, \lambda_1, \lambda_N, \rho_1, \rho_2)$, are sensitive as well. We will use this sensitivity to demonstrate the unsuitability of the proposed upper bound $R(A,k,m)$.

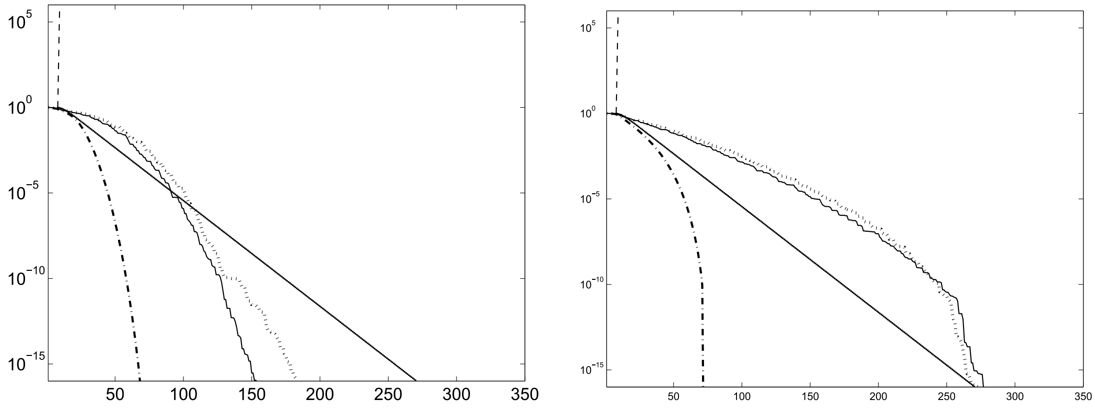

Figure 4.8: Matrix given by $\texttt{Matrix01}(65, 7, 0.1, 10^5, 0.3, \rho_2)$, where ρ_2 is set as: Left: $\rho_2 = 1$; Right: $\rho_2 = 0.95$. The matrix $\widehat{A}$ is given by $\texttt{blurring}(10, 25)$ in both cases.

In Figure 4.8 we plot the proposed upper bounds and the convergence of CG computations applied to very similar matrices which differ just in the parameter ρ_2. The eigenvalues are given by $\texttt{Matrix01}(65, 7, 0.1, 10^5, 0.3, \rho_2)$. In Figure 4.8 (left) we set $\rho_2 = 1$ and in Figure 4.8 (right) we set $\rho_2 = 0.95$. Matrix $\widehat{A}$ was in both cases constructed using parameters *count* $= 25$, *exp* $= 10$. It is obvious that the estimate $R(A,k,m)$ must be the same for both problems. On the contrary, we see that the convergence of the FP CG computations is substantially different. In order to reach the level of accuracy 10^{-16} it was sufficient to compute 150 iterations with $\rho_2 = 1$. But for $\rho_2 = 0.95$ we need over 270 iterations in order to reach the same level of accuracy. Since the upper bound $R(A,k,7)$ is the same for both problems, there is a significant difference in the suitability of the upper bound $R(A,k,7)$. Its usage as an upper bound for the FP CG computations has not serious consequences in the first case but in the second case it does.

Does this observation tell us something interesting about properties of FP CG computations? It tells us that even a small change of the parameter of the distribution inside the interval $[\lambda_1, \lambda_{N-m}]$ can have a great effect on how often will be formed multiple copies of the largest eigenvalues. Actually, if there was not such a strong connection and the small change was not affecting a frequency of multiple copies, the upper bound would be as suitable as it was before that change. That is because we know that the unsuitability of the proposed upper bound goes hand in hand with a delay of convergence, i.e., with forming multiple copies of eigenvalues approximated earlier.

4.4 Conclusions of the numerical experiment

In our experiment we have studied an upper bound for CG computations which is based on the composed polynomial R_k^m which is a product of factor with roots at several largest eigenvalues (called outliers) and of the shifted and scaled Chebyshev polynomial on the interval which contains the rest of the spectrum, for definition see (4.1.8). We have proved and numerically demonstrated that this upper bound can not be used in finite precision CG computations. The reason is that this upper bound was developed assuming an exact arithmetic and the behaviour of finite precision CG computations is typically significantly different from the behaviour of exact CG computations.

We have demonstrated that the problem is fundamental. The proposed upper bound might be unsuitable also for small and well conditioned problems. The problems can be observed also for systems, where only the single largest eigenvalue is considered as an outlier. We have also illustrated several other weaknesses of the proposed approach.

Theoretical background for our numerical experiment is formed by the backward-like analysis introduced by Anne Greenbaum in 1989; see [11]. It allows to analyse finite precision CG computations via exact precision CG behaviour applied to a matrix with clustered eigenvalues. As a consequence, constructing of the bounds must be based on polynomials which are small on the union of tiny intervals containing the eigenvalues of original matrix. Although it was stressed in several papers (see for example [21, Section 5.2]) this result is not widely accepted and correctly understood. The correct approach was used by Notay in [22] where the author present bounds of finite precision CG computations in the presence of isolated outlying eigenvalues.

We would like to stress that using bounds and estimates derived assuming exact arithmetic may be without appropriate analysis of rounding error very hazardous. On the other hand, it should not be confusing that estimates convenient also for practical computations presented in Subsection 3.6.1 were originally derived assuming arithmetic. Their validity in finite precision computation is justified by rigorous and nontrivial mathematical proofs and observations which take into account effect of rounding errors. Introducing an estimate based on exact arithmetic without analysis of influence of rounding errors would be of little practical use.

5 Formulation of several open problems

In this chapter we would like to introduce two open problems [30] which may represent the topic of our further research.

5.1 Discontinuity in Krylov subspace methods

For a given positive integer N consider symmetric positive definite (symmetric) diagonal matrix $A \in \mathbb{R}^{N\times N}$ with distinct eigenvalues. Without loss of generality assume that the eigenvalues are given in ascending order and let us denote them as $\lambda_1, \ldots, \lambda_N$. Let us define a diagonal matrix Λ of dimension $N-1$ as matrix A without the last column and row, i.e.,

$$A = \left[\begin{array}{cc} \Lambda & 0 \\ 0 & \lambda_N \end{array} \right].$$

For a given positive integer p and a sufficiently small number δ let us define the matrix $A_p(\delta)$ of dimension $N+2p$ as

$$A_p(\delta) = \operatorname{diag}\left(\Lambda, \lambda_N - p\delta, \lambda_N - (p-1)\delta, \ldots, \lambda_N + (p-1)\delta, \lambda_N + p\delta\right).$$

The matrix $A_p(\delta)$ has $2p+1$ eigenvalues clustered around λ_N in an interval of width 2δ. Sufficiently small number δ means that

$$\lambda_N - p\delta > \lambda_{N-1}$$

which secures that $A_p(\delta)$ has no multiple eigenvalues.

Let $b \equiv (b^-, \beta) \in \mathbb{R}^N$ be a vector such that the CG algorithm applied to A and b converges to the exact solution exactly in N iterations. Let us define the vector $b_p(\delta) \equiv (b^-, \beta_{split}) \in \mathbb{R}^{N+2p}$ where β_{split} is a vector of length $2p+1$ with no zero component satisfying

$$\left\|\beta_{split}\right\|^2 = \left\|\beta\right\|^2.$$

Denote as

$$k_{CG}(\widehat{A}, \widehat{b}) \tag{5.1.1}$$

the number of CG iterations needed in exact arithmetic to reach the exact solution of the problem $\widehat{A}x = \widehat{b}$.

Formulation of the question Since $A_p(\delta)$ has $N+2p$ distinct eigenvalues for every $\delta \neq 0$ we know that $k_{CG}(A_p(\delta), b_p(\delta)) = N+2p$. Since $A_p(0)$ has only N distinct eigenvalues which are the same as the eigenvalues of A (the eigenvalue λ_N has the multiplicity $2p$) we know that $k_{CG}(A_p(0), b_p(\delta)) = N$. Thus we observe some kind of discontinuity of Krylov subspace methods applied to $A_p(\delta), b_p(\delta)$, in particular,

$$N + 2p = \lim_{\delta\to 0} k_{CG}(A_p(\delta), b_p(\delta)) \neq k_{CG}(A_p(0), b_p(\delta)) = N. \tag{5.1.2}$$

5.2 Invariants of Krylov subspaces

Based on the results about the sensitivity of computing Jacobi matrices from the knowledge of corresponding distribution function we formulate a question of some kind of invariants of Krylov subspaces.

Consider symmetric matrix $A \in \mathbb{R}^{N \times N}$ with distinct eigenvalues and vector $b \in \mathbb{R}^N$ and their modifications $\widetilde{A}$ and $\widetilde{b}$. Denote as $\lambda_1, \ldots, \lambda_N$ the eigenvalues of the matrix A and as $\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_N$ the eigenvalues of the matrix $\widetilde{A}$. The matrix $\widetilde{A}$ is a modification of the matrix A in the sense that it is also symmetric and its eigenvalues are slightly perturbed, i.e.,

$$\left|\lambda_i - \widetilde{\lambda}_i\right| < \delta, \quad i = 1, \ldots, N,$$

where δ is sufficiently small prescribed real number. The vectors b and $\widetilde{b}$ are close to each other in a sense that

$$\left|\beta_i - \widetilde{\beta}_i\right| < \delta, \quad i = 1, \ldots, N,$$

where β_i and $\widetilde{\beta}_i$ are, respectively, the i-th components of the vectors b and $\widetilde{b}$.

Consider Krylov subspaces $\mathcal{K}_n(A, b)$ and $\mathcal{K}_n(\widetilde{A}, \widetilde{b})$ where n is a positive integer such that

$$n \leq \min\{d, \widetilde{d}\},$$

where d (resp. $\widetilde{d}$) is the grade of b (resp. $\widetilde{b}$) with respect to A (resp. $\widetilde{A}$). The restriction on n ensures full dimension of our Krylov subspaces.

We would like to study the relationship between $\mathcal{K}_n(A, b)$ and $\mathcal{K}_n(\widetilde{A}, \widetilde{b})$. The difference between Krylov subspaces can in general grow exponentially in dependence of the difference $E = A - \widetilde{A}$. However, the relationship with Riemann-Stieltjes integral and orthogonal polynomials shows that in some sense, the Krylov subspaces $\mathcal{K}_n(A, b)$ and $\mathcal{K}_n(\widetilde{A}, \widetilde{b})$ may have a lot in common.

Analogously as in Section 3.4 we define nondecreasing distribution function $\omega(\lambda)$ (resp. $\omega(\widetilde{\lambda})$) with N points of increase $\lambda_1, \ldots, \lambda_N$ (resp. $\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_N$) and with weights given by decomposition of b (resp. $\widetilde{b}$) in the basis given by normalized eigenvectors of matrix A (resp. $\widetilde{A}$). We know that there exist sequences of polynomials which are orthogonal with respect to the Riemann–Stieltjes integrals $\int d\omega(\lambda)$, $\int d\widetilde{\omega}(\lambda)$. The recursion coefficients of associated orthogonal polynomials compose tridiagonal matrices T_n and $\widetilde{T}_n$.

Using the results of the perturbation analysis about the sensitivity of computing coefficients of orthogonal polynomials from the knowledge of piecewise constant distribution function (see [23, Section 3]), we can conclude that the matrices T_n and $\widetilde{T}_n$ are for small δ close to each other. Specification of δ and of the tightness of matrices T_n and $\widetilde{T}_n$ is quite involved and we do not discuss it here; for details see e.g. [19, 23]. On the other hand, these matrices are also result of the Lanczos algorithm applied on A, b and thus they are strongly connected with the Krylov subspaces $\mathcal{K}_n(A, b)$ and $\mathcal{K}_n(\widetilde{A}, \widetilde{b})$ which can be very distinct.

In spite of the exponentially increasing difference between the Krylov subspaces, it might be doable to reveal and express some invariants through the analysis of the relationship between Krylov subspaces and corresponding tridiagonal matrices.

Conclusion

This thesis was made with the intention to present coherent and solid theoretical background of the CG method. We have revealed close link to the Lanczos method and we have described interesting interconnections to Riemann-Stieltjes integral or Gauss-Christoffel quadrature. We have pointed the main differences between the CG method and the CSI method whose behaviour is determined by the extremal properties of Chebyshev polynomials.

We have accented the influence of rounding errors and we have recalled the main results of rigorous mathematical analysis of CG behaviour in finite precision computations. We have emphasized that the analysis of estimates and bounds must take into account effects of rounding errors and we have shown a failure of one still quite popular approach.

In our thesis, mathematical objects defined in the first chapter were understood as tools useful for the detailed insight into the properties of the CG method. It is worth mentioning that this point of view can be reversed. For example, the CG algorithm can be considered as a tool for computation of quadrature rules (see e.g. [8]).

This bachelor thesis could serve as an introduction to the covered topics. Especially it could be useful for students searching for the text which introduces basic properties of CG algorithm in relation with other areas of mathematics. We hope that reading of this thesis can help to convince the reader that making links is good and that it often represents an important step toward the solution of a problem. Thesis references to extensive literature where an interested reader can find further details or more complex analysis.

Bibliography

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

[2] BOMAN, E. G., AND HENDRICKSON, B. Support theory for preconditioning. *SIAM J. Matrix Anal. Appl. 25*, 3 (2003), 694–717 (electronic).

[3] DAHLQUIST, G., AND BJÖRCK, Å. *Numerical methods in scientific computing. Vol. I.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.

[4] DAVIS, P. J., AND RABINOWITZ, P. *Methods of numerical integration.* Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers] New York-London, 1975. Computer Science and Applied Mathematics.

[5] DRISCOLL, T. A., TOH, K.-C., AND TREFETHEN, L. N. From potential theory to matrix iterations in six steps. *SIAM Rev. 40*, 3 (1998), 547–578.

[6] GAUTSCHI, W. *Orthogonal polynomials: computation and approximation.* Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2004. Oxford Science Publications.

[7] GOLUB, G. H. *Milestones in matrix computation: selected works of Gene H. Golub, with commentaries.* Oxford Science Publications. Oxford University Press, Oxford, 2007. Edited by Raymond H. Chan, Chen Greif and Dianne P. O'Leary.

[8] GOLUB, G. H., AND MEURANT, G. *Matrices, moments and quadrature with applications.* Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2010.

[9] GOLUB, G. H., AND STRAKOŠ, Z. Estimates in quadratic formulas. *Numer. Algorithms 8*, 2-4 (1994), 241–268.

[10] GOLUB, G. H., AND VAN LOAN, C. F. *Matrix computations*, vol. 3 of *Johns Hopkins Series in the Mathematical Sciences.* Johns Hopkins University Press, Baltimore, MD, 1983.

[11] GREENBAUM, A. Behaviour of slightly perturbed Lanczos and conjugate-gradient recurrences. *Linear Algebra Appl. 113* (1989), 7–63.

[12] GREENBAUM, A., AND STRAKOŠ, Z. Predicting the behavior of finite precision Lanczos and conjugate gradient computations. *SIAM J. Matrix Anal. Appl. 13*, 1 (1992), 121–137.

[13] HAGEMAN, L. A., AND YOUNG, D. M. *Applied iterative methods.* Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1981. Computer Science and Applied Mathematics.

[14] HESTENES, M. R., AND STIEFEL, E. Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards 49* (1952), 409–436 (1953).

[15] Jennings, A. Influence of the eigenvalue spectrum on the convergence rate of the conjugate gradient method. *J. Inst. Math. Appl. 20*, 1 (1977), 61–72.

[16] Kolmogorov, A. N., and Fomīn, S. V. *Základy teorie funkcí*, first ed. Translated from the Russian by Doležal, V. and Tichý Z. SNTL - Nakladatelství technické.

[17] Kuijlaars, A. B. J. Convergence analysis of krylov subspace iterations with methods from potential theory. *SIAM Review 2006*, 2006.

[18] Labute, J. P. MATH 255: Lecture notes [online], (accessed June 15, 2011). URL: http://www.math.mcgill.ca/labute/courses/255w03/notes.html.

[19] Laurie, D. P. Accurate recovery of recursion coefficients from Gaussian quadrature formulas. *J. Comput. Appl. Math. 112(1-2)* (1999), 165–180.

[20] Liesen, J., and Strakoš, Z. *Principles and Analysis of Krylov subspace Methods*. Oxford University Press. to appear.

[21] Meurant, G., and Strakoš, Z. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numer. 15* (2006), 471–542.

[22] Notay, Y. On the convergence rate of the conjugate gradients in presence of rounding errors. *Numer. Math. 65*, 3 (1993), 301–317.

[23] O'Leary, D. P., Strakoš, Z., and Tichý, P. On sensitivity of Gauss-Christoffel quadrature. *Numer. Math. 107*, 1 (2007), 147–174.

[24] Paige, C. C. *The computation of eigenvalues and eigenvectors of very large and sparse matrices*. PhD thesis, London University, London, England, 1971.

[25] Paige, C. C. Accuracy and effectiveness of the lanczos algorithm for the symmetric eigenproblem. *Linear Algebra and Its Applications 34* (1980), 235–258.

[26] Saad, Y. *Iterative methods for sparse linear systems*, second ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[27] Spielman, D. A., and Woo, J. A note on preconditioning by low-stretch spanning trees. *Computing Research Repository* (2009).

[28] Strakoš, Z. On the real convergence rate of the conjugate gradient method. *Linear Algebra Appl. 154/156* (1991), 535–549.

[29] Strakoš, Z. Model reduction using the Vorobyev moment problem. *Numer. Algorithms 51*, 3 (2009), 363–379.

[30] Strakoš, Z., 2011. Personal communication.

[31] Strakoš, Z., and Tichý, P. On error estimation in the conjugate gradient method and why it works in finite precision computations. *Electron. Trans. Numer. Anal. 13* (2002), 56–80 (electronic).

[32] Tebbens, J. D., Hnětýnková, I., Plešingr, M., Strakoš, Z., and Tichý, P. *Analýza metod pro maticové výpočty I.* Matfyzpress. to appear.

[33] van der Vorst, H. A. *Iterative Krylov Methods for Large Linear Systems.* Cambridge University Press, 2003.

[34] Varga, R. S. *Matrix iterative analysis.* Prentice-Hall Inc., Englewood Cliffs, N.J., 1962.

[35] Young, D. M. Iterative methods for solving partial difference equations of elliptic type. *Transactions of The American Mathematical Society 76* (1954), 92–92.

[36] Young, D. M. *Iterative solution of large linear systems.* Academic Press, New York, 1971.

List of Abbreviations

CG	method of Conjugate Gradients
CSI	Chebyshev semi-iterative method
CGQL	Conjugate Gradient with Quadrature and Lanczos algorithm
FP	finite precision
PDE	partial differential equation
SOR	successive overrelaxation method
SPD	symmetric positive definite
SSOR	symmetric SOR

A Source code

main

```
function main(n,m,l1,ln,rho,exp,count,t,rho2orouta,outb)
%  NAME:
%     main - main program
%  DESCRIPTION:
%     Typing "main" without any parameteres you run
%     numerical experiment with prescribed data.
%     Result is one figure with different convergence curves.
%     This numerical experiment ilustrates the behavior of CG in
%     FP arithmetic
%  INPUT:
%     2 different types of matrices are used, for more information
%     see Section 4.2 of the thesis.
%    matrix01
%     N = n + m  ... dimension of a matrix
%     m          ... number of eigenvalues which are left unchanged
%                ... these eigenvalues represent outliers
%     n          ... number of eigenvalues whir are re-distributed
%     l1         ... smallest eigenvalue
%     ln         ... largest eigenvalue
%     rho,rho2   ... parameter which determines the distribution
%    matrix02
%     n + m      ... dimension of matrix
%     l1         ... smallest eigenvalue
%     ln         ... largest eigenvalue in the first part of spectrum
%     rho        ... parameter which determines the distribution
%     outa, outb ... over this interval is distributed m eigenvalues
%     m          ... number of eigenvalues in interval [outa,outb]
%
%  m     ... number of eigenvalues which are considered as outliers
%  count ... how many eigenvalues are in each cluster (we vote it odd)
%  exp   ... 2*10^(-exp) is a size of clusters
%  t     ... number of iterations which are executed
%  OUTPUT:
%     figure with 5 graphs, which indicates relative error,
%     computed in energy norm
%  USAGE:
%     Call m-file BAT.m, where are the settings used in the thesis.
if (nargin==9)
    matice='matrix01';
    rho2=rho2orouta;
elseif (nargin==10  )
    matice='matrix02';
    outa=rho2orouta;
elseif (nargin==0)
```

```
%     some default setting
    n=24;l1=0.1;ln=100;rho=0.4;m=4;exp=12;count=5;t=2*n;rho2=0.95;
    matice='matrix01';
else
    error('Bad number of input parameters, see help');
end;

% hhhhhh           Setting an experiment            hhhhhhhh
% hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh

% construction of the matrix
if strcmp(matice,'matrix01')
    l=spectrum(n+m,l1,ln,rho);
    l(1:n)=spectrum(n,l1,l(n),rho2);
elseif strcmp(matice,'matrix02')
    l=spectrum(n,l1,ln,rho);
    l=[l,linspace(outa,outb,m)];
end;
%n=length(l);

% construction of the larger matrix
l2=blur(l,exp,count);

% hhhhhh        Computing of the errors             hhhhhhhh
% hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh

w=ones(length(l),1)';
% FP arithmetic on original problem
xw=[l',w'];
[~,~,AN]=cglan(t,xw,0);

% exact arithmetic on original problem (2x reorthogonalization)
[~,~,exact_AN]=cglan(length(l),xw,2);

% exact arithmetic on larger problem (simulation of FP arithmetic)
w=ones(length(l2),1)'.*(sqrt(1/count));
xw=[l2,w'];
[~,~,ANlrg]=cglan(t,xw,2);

% estimate based on support preconditioning for original problem
est_er   = cheb_on_interval(l,m,t,1);

% estimate based  on support preconditioning for larger problem
est_error = cheb_on_interval(l2',m,t,count);


% hhhhhh          Plotting the results              hhhhhhhh
% hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh
```

```
figure; clf;
semilogy(abs(exact_AN),'-.','LineWidth',2,'MarkerSize',1);
hold on;
semilogy(abs(AN),'-','LineWidth',2,'MarkerSize',1);
hold on;
semilogy(abs(ANlrg),'.','LineWidth',5,'MarkerSize',7);
hold on;
semilogy(abs(est_er),':','LineWidth',3,'MarkerSize',1);
hold on;
semilogy(abs(est_error),'--','LineWidth',3,'MarkerSize',1);
hold on;

axis([1 t 1e-16 1e6]);
set(0,'DefaultAxesFontSize',16);
hold off;
```

BAT

```
% % ... This is a collection of settings, these settings creates
% % ... figures, which are described and analysed in the thesis.
% % ...
% % ... hhhhh          Matrix01             hhhhhh
% % ... hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh
% % ...
% % ... n + m     ... dimension of a matrix
% % ... l1        ... smallest eigenvalue
% % ... ln        ... largest eigenvalue
% % ... rho,rho2 ... parameter which determines the distribution
% % ... m         ... number of eigenvalues which are left unchanged
% % ...           ... these eigenvalues represent outliers
% % ... count     ... how many extra eigenvalues are in each cluster
% % ... exp       ... 2*10^(-exp) is a size of clusters
% % ... t         ... number of iterations which are executed
% % ... main( n,  m, l1,   ln, rho,  exp,count, t, rho2)

main(92,8,0.1,10^6,0.3,9,61,700,0.95);
        ...fig1.4b, paragraph c, file: num1 !!time consuming
main(90,10,1,100,0.7,14,11,60,0.95);
        ...fig1.5b, paragraph c, filne: num2
main(65,7,0.1,10^5,0.3,10,25,350,0.95);
        ...fig1.8a, paragraph f, file: num3
main(65,7,0.1,10^5,0.3,10,25,350,1);
        ...fig1.8b, paragraph f, file: num4

% ... hhhhh          Matrix02             hhhhhh
% ... hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh
% ...
% ... n + m      ... dimension of matrix
% ... l1         ... smallest eigenvalue
% ... ln         ... largest eigenvalue in the first part of spectrum
% ... rho        ... parameter which determines the distribution
% ... outa, outb ... over this interval is uniformly distributed
% ...            ... /outcount/ eigenvalues
% ... m          ... number of eigenvalues in interval [outa,outb]
% ... count      ... how many eigenvalues are in each cluster
% ... exp        ... 2*10^(-exp) is a size of clusters
% ... t          ... number of iterations which are executed
% ... main( n,  m, l1, ln, rho,   exp,count, t, outa, outb, outcount)
main(24,5,1,2,0.9,14,11,30,10,50);
        ...fig1.4a, paragraph c, file: num5
main(24,3,1,2,0.9,8,11,50,10^6,10^7);
        ...fig1.5a, paragraph c, file: num6
main(24,1,1,2,1,14,3,25,100,100);
        ...fig1.6a, paragraph d, file: num7
main(48,1,1,5,1,7,20,60,10^7,10^7);
        ...fig1.6b, paragraph d, file: num8
main(48,5,0.1,10,0.9,12,21,220,20,30);
        ...fig1.7a, paragraph e, file: num9
main(48,5,0.1,10,0.9,10,21,220,5*10^4,5*10^4+4);
        ...fig1.7b, paragraph e, file: num10
main(48,5,0.1,10,0.9,9,15,220,10^4,5*10^4);
        ...fig1.7c, paragraph file: num11
main(48,5,0.1,10,0.9,9,25,220,10^5,5*10^5);
        ...fig1.7d, paragraph e, file: num12
```

cglan

```
function [ab,V,AN] = cglan(t,xw,reo);
%
%  NAME:
%       cglan - the CG-like implementation of the Lanczos algorithm
%
%  DESCRIPTION:
%       The conjugate gradient is used to implement the Lanczos
%       algorithm. Given the discrete inner product whose nodes
%       are contained in the first column, and whose weights are
%       contained in the second column of the array xw, the cglan
%       algorithm generates the first t recurrence coefficients "ab"
%       and the corresponding lanczos vectors V. A reorthogonalization
%       is possible.
%       Algorithm is modified to compute the energy norm of the error
%       (from known exact solution xacc).
%
%  USAGE:
%       [ab,V] = cglan(t,xw,reo);
%       t .......... how many steps of cglan should be performed
%       xw ......... two-column array, 1st column = nodes,
%                        2nd column = weights
%       reo ......... how many times the new vector should be
%                 reorthogonalized against previous vectors, reo=0
%                 means without reorthogonalization
%
%  OUTPUT:
%       ab ..... reccurence coefficients after "t" steps of
%                     the Lanczos algorithm
%       V ...... the corresponding lanczos vectors
%       AN ..... the relative error of the k-th iteration,
%                  compuited in the energy norm
%          ..... this routine can be easily modified to compute
%                  squared energy norm of the error of the k-th
%                  iteration, see lines 62, 63
%
%  (c) Dianne O'Leary, Zdenek Strakos and Petr Tichy, 21.01.2007
%   slightly modificated by Tomas Gergelits, 15.05.2011
%

n       = length(xw(:,1));

ab      = zeros(t,2);
r       = sqrt(xw(:,2));
X       = xw(:,1);

p       = r;
x       = zeros(n,1);
xacc    = diag(X)\r;
inanormsq = xacc'*r;

gamma_old   = 1.0;
delta_old   = 0.0;
rr          = r'*r;
rr0         = rr;
sgn         = -1;
ab(1,2)     = 1.0;

for j = 1 : t,

    sgn     = -sgn;
    V(:,j)  = sgn*r/sqrt(r'*r);
    ap      = X.*p;
    gamma   = r'*r/(p'*ap);
    x       = x + gamma*p;
    r       = r - gamma*ap;
    AN(j)   = sqrt(((xacc - x)'*r)/inanormsq);
%   AN(j)   = (xacc - x)'*r;
%
```

```
    rr_old    = rr;
    rr        = r'*r;
    delta     = rr/rr_old;
%
    for count = 1 : reo,
        for k = 1 : j,
            r = r - (V(:,k)'*r)*V(:,k);
        end;
    end;
%
    p = r + delta*p;
%
    alpha     = 1/gamma + delta_old/gamma_old;
    beta      = delta/(gamma^2);
    ab(j,1)   = alpha;
    ab(j+1,2) = beta;
%
    delta_old = delta;
    gamma_old = gamma;
end;
```

cheb_on_interval

```
function error = cheb_on_interval(lambda,outcount,t,cluster)
%
%  NAME:
%       ChebyshevOnInterval - computes the estimate
%       defined in the thesis
%
%  DESCRIPTION:
%       This subroutine computes the estimate of the relative
%       error which is concretized in the thesis
%       Estimate is based on polynomial which is
%       a multiplication of shifted and scaled
%       Chebyshev polynomial of smaller degree
%       and linear polynomials with the roots
%       at the outlying eigenvalues.
%       outcount .... number of outliers
%
%  USAGE:
%     lambda .. spectrum of the matrix
%     t ....... for how many iterations should be estimates computed
%
%  OUTPUT:
%     error ... length of this vector is t, error(i) is an estimate
%        of the relative error in the i-th iteration
%           ... this subroutine can be easily modified to compute an
%             estimate for squared energy norm of the error,
% see lines 55, 56
```

```
%

lambda  = lambda';
n       = length(lambda);
n_cheb  = n-(outcount*(cluster));
a       = lambda(1);
b       = lambda(n_cheb);


%   ... when computing the estimate for the original spectrum,
%   it is unnecessary to compute terms which we know they are
%   zero. It is good to do that because for large k, there
% would be computed inf*0 = NaN

if cluster ==1
    l   = lambda(1:n_cheb);
    lin = ones(n_cheb,1);
else
    l   = lambda;
    lin = ones(n,1);
end;

argument= (2*l-a-b)/(b-a);

for i=1:outcount
    linear  = (1-l/lambda(n_cheb+1+fix(cluster/2)+(cluster)*(i-1)));
    lin = lin.*linear;
end;

e=0;

for k=outcount+1:t;
    tr_Cheb     = ortho_poly(1,argument,k-outcount)/    ...
                  ortho_poly(1,(-b-a)/(b-a),k-outcount);
    if any(isnan(tr_Cheb))
        tr_Cheb=tr_Cheb+Inf;
    end;
    p      = tr_Cheb.*lin;
%   e(k)   = sum(p.^2./l); ... if computing absolute value of error
    e(k)   = max(abs(p));
end;
error = e';
```

ortho_poly

```
function pl=ortho_poly(kf,x,n)

% This is a code downloaded from the website of MIT.
% http://ceta.mit.edu/comp_spec_func/
%    ==========================================================
%    Purpose: Compute orthogonal polynomials: Tn(x) or Un(x),
%             or Ln(x) or Hn(x), and their derivatives
%    Input :  KF --- Function code
%                 KF=1 for Chebyshev polynomial (First kind) Tn(x)
%                 KF=2 for Chebyshev polynomial (Second kind) Un(x)
%             n ---  Order of orthogonal polynomials
%             x ---  Argument of orthogonal polynomials
%    Output:  PL(n) --- Tn(x) or Un(x) or Ln(x) or Hn(x)
%             DPL(n)--- Tn'(x) or Un'(x) or Ln'(x) or Hn'(x)
%    =========================================================

% The only improvement in this program is it accepts
% vector arguments for x

% make sure that x is a row or column vector and not a matrix.
[r,c]=size(x);
if r==1 | c==1
    rowvec = 0;
    if r==1
        x=x';
        rowvec = 1;
    end
else
    error('x must be a vector, and cannot be a matrix');
end
lenx = length(x);

if n==0
    if rowvec
        pl = ones(1,lenx);
    else
       pl = ones(lenx,1);
    end
else
    pl = zeros(lenx,n);

    a=2;
    b=0;
    c=1;
    y0=1;
    y1=2.*x;

    % the i'th position in pl corresponds to the i'th term
    % don't bother storing pl = 1;

    pl(:,1)=2.*x;
    if (kf == 1)
        y1=x;
        pl(:,1)=y1;
    end

    for  k=2:n
        yn=(a.*x+b).*y1-c*y0;
        pl(:,k)=yn;
        y0=y1;
        y1=yn;
    end
    if rowvec
        pl = pl(:,n)';
    else
       pl = pl(:,n);
    end
end
```

blur

```
function new_l = blur(lambda,exp,count);
%
%  NAME:
%       blur - creates spectrum with clusters
%
%  DESCRIPTION:
%       This subroutine creates a modification of the original
%       spectrum. It creates clusters of eigenvalues in tiny
%       intervals around original eigenvalues. Motivation is
%       in backward-like analysis of computing CG and Lanczos
%       algorithms in finite precision (see Greenbaum(1989),
% Greenbaum and Strakos (1992))

%  USAGE:
%       lambda ....... original spectrum
%       outliers ..... number of eigenvalues around which
% will be clusters  generated
%                ..... clusters are made around largest eigenvalues
%       exp .......... 2*10^(-exp) is a size of clusters
%       count ........ how many eigenvalues are in every cluster
%  OUTPUT:
%       new_l .......... new (larger) spectrum with clusters
%

n       = length(lambda);
l       = zeros(n*count,1);
size    = 10^(-exp);
```

```
ind=1;
for i=1:n
    l(ind:ind+count-1)=linspace(lambda(i)-size,lambda(i)+size,count);
    ind=ind+count;
end;
new_l=l;
```