Eigenvalues, orthogonal functions and structured matrices

Andrey CHESNOKOV

Jury:
Prof. dr. ir. Dirk Vandermeulen, chair
Prof. dr. ir. Marc Van Barel, promotor
Prof. dr. ir. Karl Meerbergen
Prof. dr. ir. Sabine Van Huffel
Prof. dr. Raf Vandebril
Prof. dr. Luca Gemignani
(Università di Pisa)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering

November 2011

D/2011/7515/153
ISBN 978-94-6018-452-9

Abstract

English abstract

In this thesis eigenvalues, structured matrices and orthogonal functions are studied from a practical point of view. In general, we try to exploit relations between any of these three concepts to design algorithms in the context of five problems. Each problem is closely related to one of the basic linear algebra problems: solving a system of linear equations or an eigenvalue problem.

Firstly, we study a problem arising from graph theory. There are different ways to map graphs to structured matrices, and depending on the matrix different graph-theoretic properties can be derived from their eigenvalues. An open question whether the regularity of a graph could or could not be derived from the spectrum of a certain class of structured matrices is solved.

The second aspect of the present thesis relates to a common method for eigenvalue computation, namely, a rational Lanczos method. Close relation between this algorithm and a certain minimization problem for orthogonal rational functions gives a possibility for numerical exploration of convergence properties of the algorithm without running it. Such exploration is reduced to solving a constrained weighted energy problem from logarithmic potential theory, which, in its turn, is converted to a linear system. The focus lies on this convergence exploration method.

Then a method to compute recurrence relation coefficients for bivariate polynomials, orthonormal with respect to a discrete inner product, is studied. To compute these polynomials, the inverse eigenvalue problem is posed and solved efficiently and in a stable way. This is achieved by applying Givens rotations to certain structured matrices and yields the generalized Hessenberg matrices, containing the recurrence relation coefficients.

Finally, several linear-algebraic problems with structured matrices are studied

with a continuation method. Such a method defines an easy problem with a known solution and a path between this problem and the one we are wishing to solve. Instead of solving the target problem directly, the solution to the easy one is gradually transformed to the desired one. With this approach we first solve a linear system of equations with Toeplitz coefficient matrix, and later we find all the eigenvalues and eigenvectors of a symmetric diagonal-plus-semiseparable matrix. Both types of structured matrices exhibit close relation with certain classes of orthogonal functions.

Numerical experiments are included for all the proposed methods and illustrate the stability and accuracy of the methods.

Nederlandse samenvatting

In dit werk worden eigenwaarden, gestructureerde matrices en orthogonale functies onderzocht vanuit een praktisch oogpunt. Wij gebruiken de relaties tussen deze drie concepten om nieuwe algoritmen te ontwerpen voor vijf wetenschappelijke problemen. Elk van die problemen is verbonden met klassieke problemen van lineaire algebra: het oplossen van stelsels lineaire vergelijkingen of het berekenen van eigenwaarden en eigenvectoren van een matrix.

Ten eerste wordt het probleem van grafentheorie bestudeerd. Er zijn diverse manieren om grafen naar matrices af te beelden, en afhankelijk van de matrix kunnen verschillende structurele eigenschappen van de onderliggende graf al dan niet gereconstrueerd worden op basis van de eigenwaarden van de matrix. Wij geven een antwoord op de vraag of men de regelmatigheid van een graf op basis van zijn spectrum ten opzichte van een bepaalde gestructureerde matrix kan afleiden.

Het volgende deel van dit werk gaat over een bekende methode voor het berekenen van eigenwaarden, namelijk, de rationale Lanczos methode. Het verband tussen dit algoritme en een bepaald optimalisatie probleem voor rationale orthogonale functies laat toe de convergentie-eigenschappen van zo'n algoritme te bestuderen zonder het algoritme expliciet uit te voeren. Dergelijke studie wordt gereduceerd tot het oplossen van een gewogen energie-probleem met beperkingen binnen logaritmische potentiaaltheorie, dat, op zijn beurt, wordt omgezet in een lineair stelsel. De focus ligt op de methode voor het bestuderen van de convergentie.

Verder wordt een methode ontworpen om de recursie-coëfficiënten te berekenen voor veeltermen in meerdere veranderlijken, die orthonormaal zijn ten opzichte

van een discreet inproduct. Om deze veeltermen te berekenen, wordt een invers eigenwaardeprobleem opgelost op een stabiele en efficiënte manier. Dit wordt bereikt door het toepassen van Givens rotaties op bepaalde gestructureerde matrices en levert de veralgemeende Hessenberg-matrices op. Deze matrices bevatten de recursie-coëfficiënten.

Ten slotte worden verschillende lineair-algebraïsche problemen met gestructureerde matrices bestudeerd met behulp van een voortzettingsmethode. Een dergelijke methode definieert een eenvoudig probleem met een bekende oplossing en een traject tussen dit probleem en datgene wat we willen oplossen. In plaats van het rechtstreeks oplossen van het oorspronkelijke probleem, wordt de oplossing voor het gemakkelijke probleem continu omgevormd naar die voor het moeilijke probleem. Met deze aanpak verwerken we eerst het oplossen van een lineair stelsel met Toeplitz coëfficiëntenmatrix, en later vinden we alle eigenwaarden en eigenvectoren van een symmetrische diagonaal-plus-semiseparabele matrix. Beide soorten van gestructureerde matrices staan in nauw verband met bepaalde categorieën van orthogonale functies.

Numerieke experimenten worden voor alle voorgestelde methoden gegeven en illustreren de stabiliteit en de efficiëntie daarvan.

Acknowledgements

Having the text of this thesis finished it is time to thank some people, without whom I would have never gotten this far.

Foremost, I would like to express my gratitude to Khakim Ikramov for initiating me in the field of linear algebra and guiding me through the academic world on the early stages of my career as a researcher. Being a postgraduate student I would have hard time finding a supervisor who is more motivated to do research and who sets higher standards for the scientific output, being always available and openly expressing his opinion on my work, whenever positive or negative.

Secondly, I would like to thank my current promotor Marc Van Barel who gave me an opportunity to start a PhD at the Department of Computer Science. With him I was lucky to experience a most pleasant balance between freedom to pursue my own ideas and help and council when I got completely stuck. I am very thankful that he always took the necessary time whenever there were any kind of problems.

I would like to thank all the jury members for thoroughly reading my thesis and their valuable comments on the text: Luca Gemignani, Sabine Van Huffel, Karl Meerbergen and Raf Vandebril. Thanks to Dirk Vandermeulen for chairing the jury during my preliminary and public defence, and to Margot Peeters for the administrative support.

Finally I would like to thank my friends. Marc Heijmans and Pedro Bruggemans navigated me through the different aspects of life in a foreign country, providing all different kinds of help and showing in practice how good some Belgian people can be. Gülseli Baysu, with her deep knowledge of human psychology, prevented me several times from stopping this research. We shared our office with Karl Deckers for five years and it was always inspiring to see his strong focus on research and music, and it has even led to our collaboration on one of the papers. And last but certainly not the least I am thankful to Snezhana Dubrovskaya and Sergei Strelkov for sharing with me many cultural

and sport activities, as well as for keeping doors of their houses always open for me.

Thank you all for believing in me!

Contents

Chapter 1

Introduction

1.1 Historical context and motivation

1.1.1 Eigenvalues

Eigenvalues are often introduced in the context of linear algebra or operator theory. Historically, however, they arose during the study of applications, mostly coming from physics.

Euler studied the rotational motion of a rigid body and discovered the importance of the principal axes. Lagrange realized that the principal axes are the eigenvectors of the inertia matrix [77, Sec. 2]. In the early 19th century, Cauchy saw how their work could be used to classify the quadric surfaces, and generalized it to arbitrary dimensions [77, Sec. 3]. Cauchy also coined the term *racine caractéristique* (characteristic root) for what is now called eigenvalue; his term survives in the "characteristic equation" [96, pp. 807-808].

Fourier used the work of Laplace and Lagrange to solve the heat equation by separation of variables in his famous 1822 book *Théorie analytique de la chaleur* [96, p. 673]. Sturm developed Fourier's ideas further and brought them to the attention of Cauchy, who combined them with his own ideas and arrived at the fact that real symmetric matrices have real eigenvalues [77, Sec. 3]. This was extended by Hermite in 1855 to what are now called Hermitian matrices [96]. These were the first results where the structure of the matrix

played an important role. Around the same time, Brioschi proved that the eigenvalues of orthogonal matrices lie on the unit circle [77, Sec. 3], and Clebsch found the corresponding result for skew-symmetric matrices[96, pp. 807-808].

1.1.2 Algebraic graph theory

Since the early years it has been discovered that study of the eigenvalues is useful in seemingly unrelated areas of mathematics, such as graph theory. A possible way to see these applications is to associate matrices, such as an adjacency matrix or a Laplacian matrix, with graphs. This makes it possible to study properties of the underlying graph in relationship to the characteristic polynomial, eigenvalues, and eigenvectors of the matrices. The eigenvalues of these matrices are often called a spectrum of the graph, so the spectrum is dependent on the associated matrix.

After mapping graphs to eigenvalues a natural property to explore is whether any graph-specific information is preserved during the mapping. The question "which graphs are determined by their spectrum?" (futher denoted as "DS") goes back for about half a century, and originates from chemistry. In 1956 Günthard and Primas [74] raised the question in a paper that relates the theory of graph spectra to Hückel's theory from chemistry (see also [44, Chapter 6]). At that time it was believed that every graph is DS until one year later Collatz and Sinogowitz [41] presented a pair of cospectral trees. One more well-known example of cospectral graphs, called the Saltire pair, is presented on Figure 1.1. Both graphs have spectrum $\{[2]^1, [0]^3, [-2]^1\}$, powers denote multiplicities.

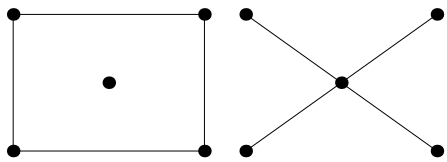


Figure 1.1: Pair of graphs, cospectral wrt adjacency matrix

Another application comes from Fisher [59] in 1966, who considered a question of Kac [91]: "Can one hear the shape of a drum?" He modeled the shape of the drum by a graph. Then the sound of that drum is characterised by the eigenvalues of the graph.

After 1967 many examples of cospectral graphs were found. The most striking result of this kind is that of Schwenk [140] stating that almost all trees are

non-DS. After this result there was no consensus for what would be true for general graphs (see, for example [67, p. 73]). Are almost all graphs DS, are almost no graphs DS, or is neither true? Van Dam and Haemers shed some light on this question in a survey [167]. They leave some open questions that were addressed in [33] and later in [169], namely, they were interested whether a regularity of the graph can be deduced from its spectrum with respect to a certain generalized adjacency matrix, depending on a parameter y.

One part of the solution is given in our work [33], where a cospectral pair of regular and non-regular graphs is firstly found by computer enumeration, and then a general procedure is presented, that allows to construct such a pair for any rational value of y. One of discovered pairs is presented on Figure 1.2. Later in [169] all such pairs on at most eleven vertices are generated and it is shown that such a pair cannot exist for irrational y.

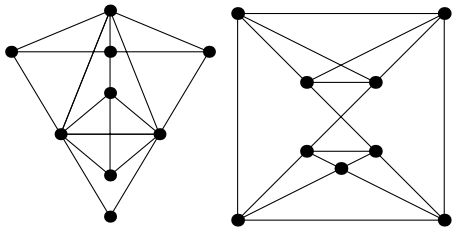


Figure 1.2: Discovered pair of graphs, cospectral wrt generalized adjacency matrix

In 2009 van Dam and Haemers published one more review [168], where they summarized known results about DS and non-DS graphs. They show that the spectrum with respect to different matrices preserves different properties, and present several families of graphs that are DS with respect to the adjacency matrix, the Laplacian matrix, or both. An important development is the new method of Wang and Xu [178] for finding graphs that are DS with respect to the generalized adjacency matrix. Their approach often works for randomly generated graphs, and this strengthens our believe that the statement 'almost all graphs are not DS' (which is true for trees) is false.

1.1.3 Lanczos algorithm

The complexity of the problem for finding eigenvalues of a matrix increases rapidly with increasing size of the matrix. There are exact solutions for dimensions below 5, but for dimensions greater than or equal to 5 there

are generally no exact solutions and one has to resort to numerical methods to find them approximately. Worse, this computational procedure can be very inaccurate in the presence of round-off error. Efficient, accurate methods to compute eigenvalues and eigenvectors of arbitrary matrices were not known until the development of the QR algorithm by Francis [61, 62] and Kublanovskaya [99]. For large Hermitian sparse matrices, the Lanczos algorithm is one example of an efficient iterative method to compute part of the eigenvalues and eigenvectors, among several other possibilities [153].

The Lanczos algorithm is an iterative Krylov subspace algorithm invented by Lanczos [103] that is an adaptation of the power method to find eigenvalues and eigenvectors of a symmetric square matrix or the singular value decomposition of a rectangular matrix. It is particularly useful for very large sparse matrices, especially if a fast procedure for computing a matrix-vector product can be constructed. For the Lanczos algorithm, it can be proved that with exact arithmetic, it constructs an orthogonal basis of a corresponding Krylov subspace. This basis transforms the original matrix to a tridiagonal one of a smaller size, and some of its eigenvalues/vectors (they are called Ritz values/vectors) are then good approximations to the corresponding ones of the original matrix [71, Chapter 9]. Formula (1.1) schematically shows an example of such transformation for a 5×5-matrix, the columns of a narrow matrix are the basis vectors. However, in practice (as the calculations are performed in floating point arithmetic where inaccuracy is inevitable), the orthogonality is quickly lost and in some cases the new Ritz vector could even be linearly dependent on the set that is already constructed [122, Chapter 13]. This troublesome feature complicates the relationship between the eigenvalues of the original matrix and those of the tridiagonal one. Because of this reason the Lanczos algorithm was disregarded by numerical analysts during almost twenty years since its discovery.

$$\begin{pmatrix} \times & \times & 0 \\ \times & \times & \times \\ 0 & \times & \times \end{pmatrix} = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix} \tag{1.1}$$

The search for a practical, easy-to-use Lanczos procedure is rooted in the fundamental error analysis of the method by Paige [117, 118]. An examination of his results motivates several modified Lanczos methods, as described in the well-known book of Golub and Van Loan [71].

1.1.4 Convergence of the Lanczos algorithm: potential theory and zeros of orthogonal functions

It is of basic importance for an appreciation of the Lanczos method to understand which eigenvalues of the original matrix are approximated by the Ritz values. Originally, Trefethen and Bau [153] observed a relationship with electric charge distributions, and they state the following rule of thumb: *The Lanczos iteration tends to converge to eigenvalues in regions of "too little charge" for an equilibrium distribution.* This may be understood as follows. Assume that the eigenvalues of the original matrix are located on the interval $[-1, 1]$, except perhaps for a few outliers. Then one has to compare the distribution of eigenvalues with the equilibrium distribution of $[-1, 1]$. The density of the equilibrium distribution is infinite at the endpoints ± 1. Thus if the eigenvalues of the original matrix are spread out more evenly over the interval $[-1, 1]$, then the Lanczos method tends to find the extreme eigenvalues. On the other hand, if these eigenvalues are distributed like the equilibrium distribution, then the Lanczos iteration is very much useless if the amount of iterations is smaller than the size of the original matrix, and does not find any eigenvalue until these two numbers become equal.

More exact estimates for the classical Lanczos method are presented by Kuijlaars [100, 101], where he utilized the connection between the Lanczos method, a polynomial minimization problem and logarithmic potential theory. The first relationship is known since the very discovery of the method and is due to Stiefel [147]. It could be briefly summarized as follows: the characteristic polynomial of the resultant tridiagonal matrix is a monic polynomial of corresponding degree that minimizes a certain norm among all monic polynomials of the same degree. The zeros of this polynomial are equal to the Ritz values. Rakhmanov [125] characterized the zero distribution of polynomials satisfying a discrete orthogonality by relating it to an extremal problem in potential theory. Using this result, Kuijlaars described, in an asymptotic sense, the region containing those eigenvalues that are well approximated by the Ritz values. The region depends on the distribution of eigenvalues and on the ratio between the size of the matrix and the number of iterations. We refer to [51] for more details on the connection between potential theory and matrix iteration methods.

For the classical Lanczos method and equally distributed eigenvalues this is in accordance with the well-known fact that *eigenvalues on the outskirts of the spectrum converge first.* However, in some applications this convergence behavior is not appreciated and one may be interested, say, only in several internal eigenvalues. As a remedy, a more general Krylov subspace method is presented by Ruhe [129] and further analyzed by him [131, 130] and other

authors [46, 47, 55, 112]. It is suggested to consider rational Krylov spaces instead of classical Krylov spaces, thus replacing monic polynomials by rational functions. It was known since the discovery of the method that in this way it is possible to get good approximations to all the eigenvalues in a union of regions around the poles of the rational functions; see [132]. A typical application that makes use of these algorithms is model reduction of a linear dynamical system, where one wants to get the response over a prescribed range of frequencies; see, e.g., [133] or [64].

Recently, Beckermann, Güttel and Vandebril [10] extended the above-stated results of Kuijlaars [101] and characterized the region of good convergence for the rational Lanczos process. Their description relates the convergence regions to a solution of a more complex constrained weighted energy problem from potential theory. They also presented estimates on the rate of approximation of a given eigenvalue by a rational Ritz value. An explicit solution to the constrained weighted energy problem is known only for some cases, and for other cases some properties can be derived without being able to obtain an explicit solution. Hence it is interesting to obtain an approximate numerical solution. A fast and stable method of solving this problem is suggested in our work [32], where the extremal energy problem is discretized and solved numerically. This method gives the possibility to predict the regions of convergence without actually running the rational Lanczos algorithm.

Such a tool may be useful in large-scale computations to estimate the amount of iterations required to reach a given precision for certain eigenvalues, thus determining computation time and memory requirements before the actual execution of a Lanczos method. On Figures 1.3 and 1.4 we show a convergence behavior of a rational Lanczos algorithm for a certain matrix with eigenvalues equally distributed between -1 and 1, for two different choices of poles. There by a red '+' we denote almost converged Ritz values and the black curve is computed by our new algorithm and represents a border of a region with converged Ritz values.

So, the use of complex poles allows to find first some internal eigenvalues, while real poles just speed up the convergence to outer eigenvalues on one or another side of a segment $[-1, 1]$.

We would like to mention that the use of potential theory is by no means the only tool for studying the convergence behavior of Krylov subspace methods. Important contributions have been made for example in [135, 145, 170, 171], where a priori error bounds are obtained from a refined analysis of the extreme eigenvalues. In contrast to these methods, methods based on potential theory emphasize the global eigenvalue distribution but ignore the local fine structure of eigenvalues.

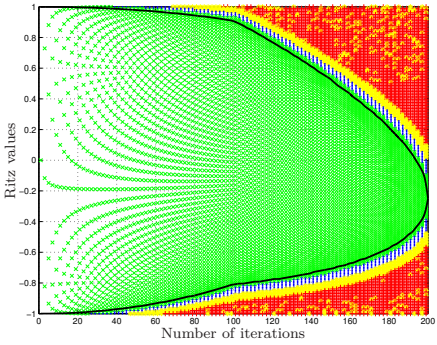


Figure 1.3: Predicted and actual convergence of rational Ritz values with -5 and 1.2 as poles

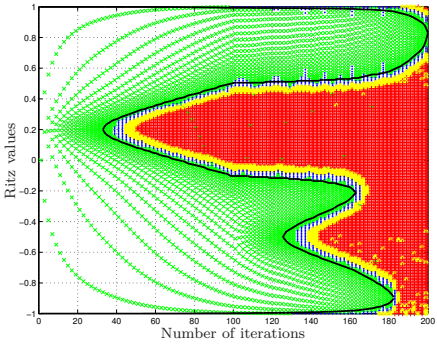


Figure 1.4: Predicted and actual convergence of rational Ritz values with $0.2 + 0.1\mathbf{i}$ and $0.5 + 0.1\mathbf{i}$ as poles

1.1.5 Orthogonal polynomials

Orthogonal polynomials have deep relations not only to the Lanczos iteration method. Originally, the field of orthogonal polynomials developed in the late 19th century from a study of continued fractions by Chebyshev [31] and was pursued by Markov [108] and Stieltjes [148]. In the 20th century the theory of orthogonal polynomials was used in the study of both theoretical and practical problems. Two standard references on this topic are the classic textbook of Szegő [151], and Chihara's work [36] which puts more emphasis on the discrete case. Plenty of applications where different orthonormal polynomials are a valuable and fruitful tool are discussed in the book of Golub and Meurant [70].

As one of the examples, different sequences of orthogonal polynomials have important applications in algebraic graph theory, as described by Godsil [67]. Let us define a matching in a graph as a set of edges without common vertices. Then a matching polynomial (sometimes called an acyclic polynomial) is a generating function of the numbers of matchings of various sizes in a graph. As shown in [67], for the complete bipartite graph one type of matching polynomial is closely related to the generalized Laguerre polynomials, and for a complete graph it is just the Hermite polynomial. We find the Chebyshev polynomials (of the second kind) when studying the characteristic polynomials of the paths, as shown by Schwenk [141]. One more recent application in this area has been given by Chung, Faber and Manteuffel [39], and van Dam and Haemers [166], who gave upper bounds on the diameter of a graph in terms of its spectrum by using Chebyshev polynomials shifted to a proper interval. Some other applications of orthogonal polynomials in graph theory are presented in a recent

work [28].

One more application that brings together orthogonal polynomials, structured matrices and eigenvalue problems, is a discrete least squares approximation problem (see cf. [128, 127, 157]), which is often related to data fitting, as it was shown by Forsythe in his pioneering work [60]. One type of algorithms for the least squares problem will compute implicitly or explicitly an orthonormal basis and the Fourier coefficients of the solution in this basis. The polynomials appearing there are orthogonal on a discrete set, and are usually described by recurrence relations. Special properties of the discrete set (like when all the points are on the real line or all of the points lie on the complex unit circle) lead to a "short recurrence" for the orthogonal polynomials and thus reduce the complexity of such an algorithm. Detailed study of recurrence relations between univariate polynomials orthogonal on the real line are presented in the book of Gautschi [65]. In particular, the already mentioned Hermitian Lanczos method could be adapted to generate polynomials orthogonal on a discrete subset of a real line. This result represents a part of classical numerical analysis; see, e.g. [60, 65, 72].

Bivariate polynomials on a planar region have been studied quite extensively; see [149, 97, 180] and references therein. As shown by Dunkl and Xu [52], bivariate orthogonal polynomials, provided that one uses a graded monomial order, also satisfy a matrix form of recurrence relations, which is a generalization of the three-term relation in one variable. For discrete orthogonal polynomials, depending on the data points, recurrence relation matrices are block-tridiagonal, see [181]. However, the corresponding problem of numerical computation of a discrete bivariate orthogonal basis has not been considered for a long time before the work of Huhtanen and Larsen [88].

In data fitting applications it is especially useful to be able to add points to a discrete set online. As shown by Elhay, Golub and Kautsky [54] and Van Barel and Bultheel [160], effective procedures for updating and downdating the recurrence information for polynomial sequences could be developed. However, the algorithm of Huhtanen and Larsen lacks the useful online updating feature.

Unlike in the univariate case, in the multivariate case the number of nonzero terms in the recurrence relation is growing as the degree of the polynomial grows. This growth is of order $\sqrt{8d}$, where d is the number of polynomials generated so far, see [88] for details. This growth makes the multivariate analogues of univariate algorithms much slower.

In our research [161] we address a multivariate discrete least squares approximation problem. The core of the algorithm constitutes an updating procedure for a polynomial basis, which is a generalization of a univariate procedure by

Van Barel and Bultheel [24, 158].

Similarly to the univariate case, to solve the least squares problem we compute recurrence relation coefficients of discrete orthogonal basis polynomials and in parallel the Fourier coefficients of a target approximant in this basis. The basis polynomials are represented by their recurrence relation coefficients. For bivariate problems, these coefficients are coming from two coupled inverse eigenvalue problems for generalized Hessenberg matrices. The latter are solved by means of a sequence of Givens rotations.

The points where the discrete inner products are prescribed may constitute any pairs of complex points. However, when we take pairs of real points, the generalized Hessenberg matrices representing the recurrence relations become symmetric. The lower bandwidth of these matrices is equal to the depth of recursion, so, as explained above, it is slowly growing together with the size of a matrix. Thus symmetricity does not have that significant effect on the complexity like it had in the univariate case. Compared to the algorithm of Huhtanen and Larsen [88], our algorithm has an updating feature, which is useful in applications. In terms of speed and accuracy the two algorithms are similar.

An inverse unitary Hessenberg eigenvalue problem was studied by Ammar and He [4], and for a survey of methods on different inverse eigenvalue problems, we refer to Chu and Golub [38]. The relation between inverse eigenvalue problems and univariate discrete least squares approximation is well known, see e.g. Reichel [127] and Elhay, Golub and Kautsky [54]. Different orthogonal polynomials are appearing here in a natural way, and one example follows. Based on the inverse unitary QR algorithm for computing unitary Hessenberg matrices [3], Reichel, Ammar and Gragg [128] solve the approximation problem when the given function values are taken in points on the unit circle. Their algorithm is based on computational aspects associated with the family of polynomials orthogonal with respect to an inner product on the unit circle. Such polynomials are known as Szegő polynomials. Fassbender [57] presents an approximation algorithm based on an inverse unitary Hessenberg eigenvalue problem and shows that it is equivalent to computing Szegő polynomials.

For our numerical experiments we use Padua points (Figure 1.5) as the set of nodes for the discrete inner product. These points were introduced for the first time by Caliari, De Marchi and Vianello in [27]. Such points are an example of optimal points with real coordinates for total degree polynomial interpolation in two variables, with a Lebesgue constant increasing like log squared of the degree, see [18, 19].

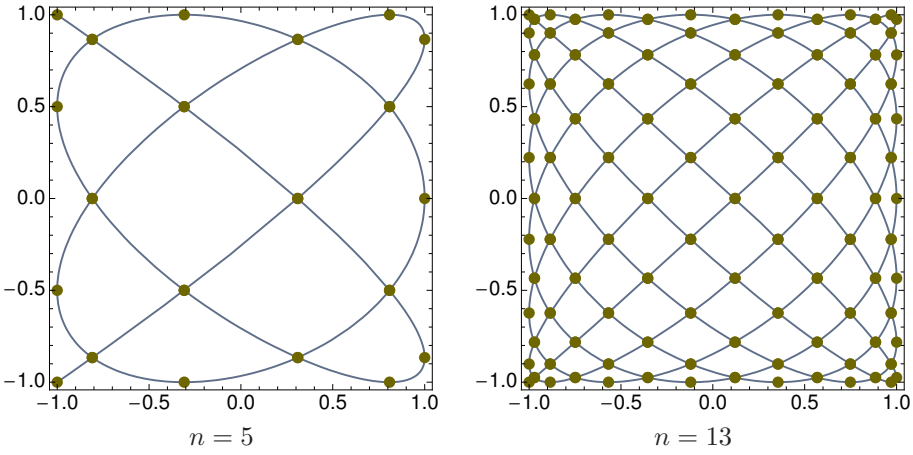


Figure 1.5: Padua points for different n

1.1.6 Structured matrices

Different classes of structured matrices, like, for example, tridiagonal, band-, Toeplitz, Toeplitz-like, Cauchy, Cauchy-like, semiseparable and other matrices, received a close attention of many numerical analysts. As we mentioned in the beginning, the first works on structured matrices go back to the time of Cauchy and Hermite. This attention is due to the fact that while a general, unstructured dense matrix requires the storage space proportional to its size squared, many structured matrices could be stored in space linear in their size.

Even more important, the computational complexity of some algorithms can be reduced enormously when they are applied on structured matrices. This is illustrated by comparing the computational complexity of some frequently used algorithms. So, while a direct solver for a general matrix, such as LU-decomposition, requires the amount of operations cubic in matrix order, tridiagonal and semiseparable matrices allow linear solvers [174], and for Cauchy matrices there exist fast (complexity proportional to the size squared) and superfast (linear up to some logarithmic term) algorithms [119]. The complexity also reduces dramatically for other algorithms, like matrix-vector multiplication (which could be applied within Lanczos methods) or the QR-method for solving the eigenproblem.

Within this research we became interested in two specific classes of matrix structure, namely, in Toeplitz matrices and diagonal-plus-semiseparable matrices.

A matrix is called Toeplitz if the elements along each of the diagonals coincide, so such matrix is fully determined by its first row and column. An example of a 4×4 Toeplitz matrix is presented as (1.2). Toeplitz matrices arise in different problems of applied mathematics, such as approximation of differential equations [107, 124], Padé approximations [156, 20, 23] and polynomial root localisation [176, 79, 94].

$$\begin{pmatrix} 4 & 1 & 2 & 10 \\ 77 & 4 & 1 & 2 \\ 1 & 77 & 4 & 1 \\ 0 & 1 & 77 & 4 \end{pmatrix} \tag{1.2}$$

The fact that a Toeplitz matrix is determined only by an amount of parameters linear in its order has led to several fast and superfast algorithms for the solution of linear systems with Toeplitz coefficient matrices, utilizing the matrix structure. The two types of direct fast solvers that require $\mathcal{O}(n^2)$ operations are Levinson-type and Schur-type solvers. For more references and information about these algorithms, one may refer to [93].

Algorithms with complexity less than $\mathcal{O}(n^2)$ are called superfast. The idea of a superfast Toeplitz solver was first announced in the PhD thesis of Morf [113]. Superfast algorithms were designed by Sugiyama et al. [150], Brent, Gustavson, and Yun [20], Ammar and Gragg [2], Van Barel, Heinig and Kravanja [163]. Many recent algorithms can be found in [45, 66, 78, 116].

However, the above-mentioned methods usually require numerical nonsingularity of the leading principal submatrices in a given Toeplitz matrix. Moreover, if these submatrices are nonsingular, but ill-conditioned, this can cause numerical instability in finite-precision implementations. So, a construction of a general superfast method, that would not be so critical to the properties of a given matrix, is desirable.

To achieve the goal, a suitable tool is needed to define, identify and exploit the matrix structure. Such tool was introduced by Kailath, Kung and Morf in the paper [92] and received the name of displacement rank approach. A matrix M is said to be of low displacement rank, if for some matrices A and B the rank of the matrix $X = M - AMB$ is low. With an appropriate choice of A and B a Toeplitz matrix T can be converted to a rank two matrix X, and a skeleton decomposition of X gives another representation of T, that also requires storage space linear in the order of T.

Under suitable conditions on the matrices A and B this representation makes it possible by using the Fast Fourier Transform (FFT) to multiply a given Toeplitz matrix, as well as its inverse, by a vector using $\mathcal{O}(n \log n)$ arithmetic operations.

The effective representation of the inverses of Toeplitz matrices, allowing fast matrix-by-vector multiplication, was first discovered by Gohberg and Semencul [69], while using a slightly different approach. A tremendous impact of their formula on the field of structured matrices and numerical algorithms is systematically presented in the book of Heinig and Rost [79].

In mathematical modelling, Toeplitz matrices, together with their block versions, arise whenever properties of shift invariance are satisfied by some function in the model. They are encountered, in particular, in fields like image processing and in the numerical solution of differential equations where the shift invariance takes different forms. Very often these matrices are block banded block Toeplitz matrices (further referred as BBBT matrices) accompanied with a Toeplitz structure of the blocks. Such a matrix in a general form is represented as shown in (1.3). A very extensive study of the problems where block banded block Toeplitz matrices arise, is given in [15].

$$\begin{pmatrix} A_0 & A_1 & \ddots & A_k & 0 & 0 & \cdots & 0 \\ A_{-1} & A_0 & A_1 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \ddots & A_{-1} & A_0 & A_1 & \ddots & \ddots & \ddots & 0 \\ A_{-k} & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & A_k \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & A_{-1} & A_0 & A_1 \\ 0 & \dots & 0 & 0 & A_{-k} & \ddots & A_{-1} & A_0 \end{pmatrix} \tag{1.3}$$

Due to the large block size of the matrices (for example, for image processing the product of the two sizes is the number of pixels in the image) it is mandatory to exploit both the outer banded Toeplitz structure and the inner Toeplitz structure to devise efficient algorithms for the solution of these systems. Several iterative techniques for the solution of BBBT systems have been introduced in the literature, in particular we recall PCG methods [29, 143, 144], the multigrid techniques [58] and the algorithms based on the cyclic reduction [13, 15]. There are also some direct methods, such as generalizations of the methods based on the Schur algorithm [14], the generalization of displacement ranks [126] and the deconvolution approach [182].

However the best of these algorithms give $O(kn^4)$ complexity for nonsymmetric $n^2 \times n^2$ BBBT systems with nonbanded Toeplitz blocks, k denotes the (block) bandwidth, and are not easy to program. So, we presented in [34] a new

algorithm that reduces the complexity for banded case to $O(n^4) + O(k^3 n^3)$, and is quite easy to program. Such a direct method may be of particular interest when constructing preconditioners for existing iterative methods like GMRes and BiCGStab.

The proposed algorithm constructs the low-rank circulant transformation of a given BBBT system and then by means of the Sherman-Morrison-Woodbury formula transforms the inverse of such a transformation to the inverse of the original matrix. The block circulant matrix with Toeplitz blocks is converted to a block diagonal matrix with Toeplitz blocks, and the resulting Toeplitz systems are solved by means of a fast Toeplitz solver.

A matrix is called a symmetric semiseparable matrix if all submatrices taken out of the lower and upper triangular part of the matrix are of rank 1 and the matrix is symmetric. A matrix is called a symmetric diagonal-plus-semiseparable matrix if it can be written as the sum of a diagonal and a symmetric semiseparable matrix, and if such a matrix has a form

$$A = \begin{pmatrix} d_1 & u_1v_2 & u_1v_3 & \cdots & u_1v_{N-1} & u_1v_N \\ v_2u_1 & d_2 & u_2v_3 & u_2v_4 & \cdots & u_2v_N \\ v_3u_1 & v_3u_2 & \ddots & \cdots & \cdots & \cdots \\ \vdots & v_4u_2 & \vdots & \ddots & \cdots & \cdots \\ v_{N-1}u_1 & \vdots & \vdots & \vdots & d_{N-1} & u_{N-1}v_N \\ v_Nu_1 & v_Nu_2 & \vdots & \vdots & v_Nu_{N-1} & d_N \end{pmatrix},$$

where u and v are vectors, then it is called a symmetric generator-representable diagonal-plus-semiseparable matrix. These types of matrix structure appear in two different contexts. First, during a specific discretization of Green's function for the two point boundary value problem, symmetric diagonal-plus-semiseparable matrices arise (see [123]). If the kernel of an integral operator can be written as the sum of a semiseparable part and a degenerate one, discretization of the eigenvalue problem of those operators also involves diagonal-plus-semiseparable matrices (see [146]). The inverse of an irreducible tridiagonal matrix has this form too. Second, there exist stable procedures to reduce a dense symmetric matrix into a similar semiseparable (plus diagonal) one by means of orthogonal transformations, as shown in [175, Ch. 2]. Hence, by combining the latter algorithm with an eigenvalue solver for diagonal-plus-semiseparable matrices, a spectral decomposition of any symmetric matrix can be computed.

Several effective algorithms have been proposed to find all the eigenvalues and eigenvectors of a symmetric diagonal-plus-semiseparable matrix, like the QR-algorithm [175, Ch. 5 and Ch. 7] and divide-and-conquer algorithms [110].

Those divide-and-conquer algorithms involve computations with the secular equation, and lots of care and precautions have to be taken to perform these computations in a stable way, as shown in [152]. The method presented in this research has similar complexity, but allows to avoid computations with the secular equation and makes use of a continuation method instead.

Both Toeplitz and semiseparable matrices have close connection with orthogonal functions. In [162] Van Barel, Fasino, Gemignani and Mastronardi describe how to construct an orthonormal basis in the space of proper orthogonal rational functions with prescribed poles by solving an inverse eigenvalue problem, involving generator-representable diagonal-plus-semiseparable matrices. The direct and the inverse eigenvalue problem of generator-representable symmetric diagonal-plus-semiseparable matrices are studied in detail by Fasino and Gemignani in [56]. For a thorough analysis of relations between semiseparable matrices and orthogonal functions we refer to [175, Ch. 12 and Ch. 13].

For basic relations between Toeplitz matrices and polynomials, we refer to the book [84]. Kailath, Vieira and Morf show in [94], that certain formulas for inverting of Toeplitz operators in both discrete and continuous time can be interpreted as versions of the Christoffel-Darboux formula for the biorthogonal Szegő and Krein polynomials on the circle and on the line, respectively. In case of discrete time a Toeplitz operator is represented just by a Toeplitz matrix. The Levinson method, mentioned above, represents the recursions for the so-called discrete Szegő orthogonal polynomials [151]. Van Barel and Bultheel [159] explore the connection between look-ahead schemes for block Toeplitz systems and formal orthogonal matrix polynomials.

1.1.7 Continuation methods

Large scale scientific computing is an active research field. Traditional methods which work well for small problems are not always suitable for large ones, or not suitable for modern computer architectures. For example, the very efficient method for eigenproblems with small matrices – the QR iteration method is highly serial in nature and it is difficult to benefit from many of the advanced architectures, as shown, for example, in [82]. However, recently the QR-algorithm has been implemented on distributed memory architectures [83]. The above-discussed Lanczos method can take advantage of the sparseness structure of a given matrix or exploit a fast matrix-vector multiplication, but is hard to parallelize. A nice achievement in this direction was presented in [89], see also references therein.

The continuation or homotopy method is well-known and has beed used in the past for solving systems of algebraic equations [73, 1] and multi-objective

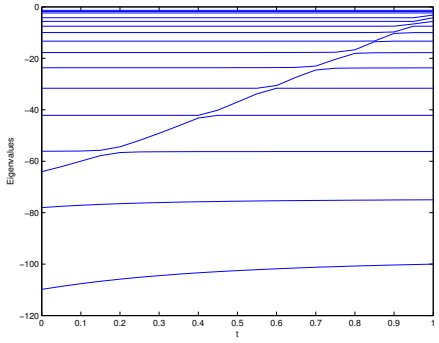


Figure 1.6: Eigenvalue curves before deflation

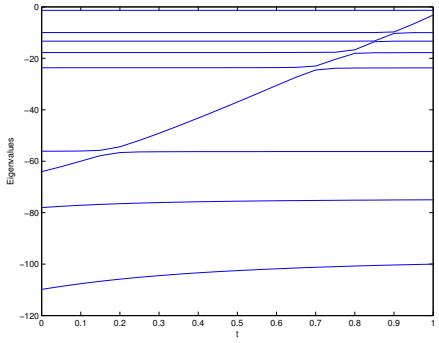


Figure 1.7: Eigenvalue curves after deflation

optimization problems [139]. To develop such a method, one starts with a simple problem whose solution is available and a trajectory that joins this simple problem and the complex problem that has actually to be solved. In the process of moving along the chosen trajectory, the solution to the simple problem continuously transforms into the solution to the complex problem. The trajectory is often defined by indtroducing an additional scalar parameter into the problem and the path tracing could be done by following the solution curve of a certain differential equation, as shown in [1].

In recent years different homotopy methods have also been applied to solving the eigenvalue problem with tridiagonal matrices [37, 105, 87, 115]. These methods came into consideration for the matrix eigenvalue problem because of their natural parallelism. Parallelization is done over eigenpairs; each processor has a copy of the problem and once set up the processor may run independently of the others. Therefore homotopy algorithms are good candidates for modern parallel computers. A new homotopy algorithm for diagonal-plus-semiseparable matrices that is presented in our report [35] has good accuracy and is capable of treating the matrices with clustered eigenvalues effectively. This is achieved by aggressive deflation techniques, similar to the ones described by Oettli in [115] and the use of a divide-an-conquer principle, inspired by Mastronardi, Van Camp and Van Barel [110]. In addition, partial reorthogonalization is used to guarantee orthogonality of the eigenvectors. For a certain model matrix the traced eigenvalues are shown on Figures 1.6 and 1.7, without use of deflation techniques and involving deflation, correspondingly. On the y-axis the actual eigenvalues are plotted. For $t = 0$ they come from some matrix with known eigenvalue decomposition, and for $t = 1$ they are the eigenvalues of the model matrix.

The homotopy approach can also be applied to the inversion of symmetric Toeplitz matrices, when combined with a displacement rank representation. The resulting method starts with a displacement representation of the identity matrix and transforms it to the displacement representation of a target inverse by dividing the trajectory in small chunks and applying an iterative improvement process on each of them. The complexity analysis reveals that the resulting method is superfast. This algorithm is presented and analyzed in our journal paper [164].

1.2 Overview

In the previous section the historical pathway, leading to this research, was described. This pathway also shows many interconnections between the more specific fields that gave rise to the addressed problems. These interconnections are schematically shown on Figure 1.8.

Each of the columns on this Figure corresponds to one of the specific fields that form the title of the research – eigenvalues, orthogonal functions and structured matrices. Each of these fields is considered from applicational and computational points of view. So, in Chapter 2 the eigenvalues serve as a tool to explore several graph-theoretic properties. In Chapter 3 we study the convergence behavior of one algorithm for computing the eigenvalues, namely, the rational Lanczos algorithm. To investigate this behavior, a certain minimization problem for orthogonal functions plays an important role. In Chapter 4 we present an algorithm to compute an orthogonal basis of multivariate polynomials, where, in its turn, structured matrices machinery appears as a key instrument. And finally, in Chapters 5 and 6 we present algorithms that solve several fundamental linear-algebraic problems with structured matrices.

So, every oval on Figure 1.8 contains a concept, a viewpoint, and a number of a chapter where this concept is studied from this viewpoint. Arrows represent the links described in a previous paragraph.

Chapters 2, 3 and 4 may be read independently of each other. Chapter 5 contains preparatory results for Chapter 6, and these two chapters together may be also read independently of the previous ones.

This research is focused on designing, implementing and analyzing efficient and accurate methods, that would involve the three concepts declared in the title – eigenvalues, orthogonal polynomials and structured matrices – and would deliver new results in one of the fields by utilizing concepts from the other one.

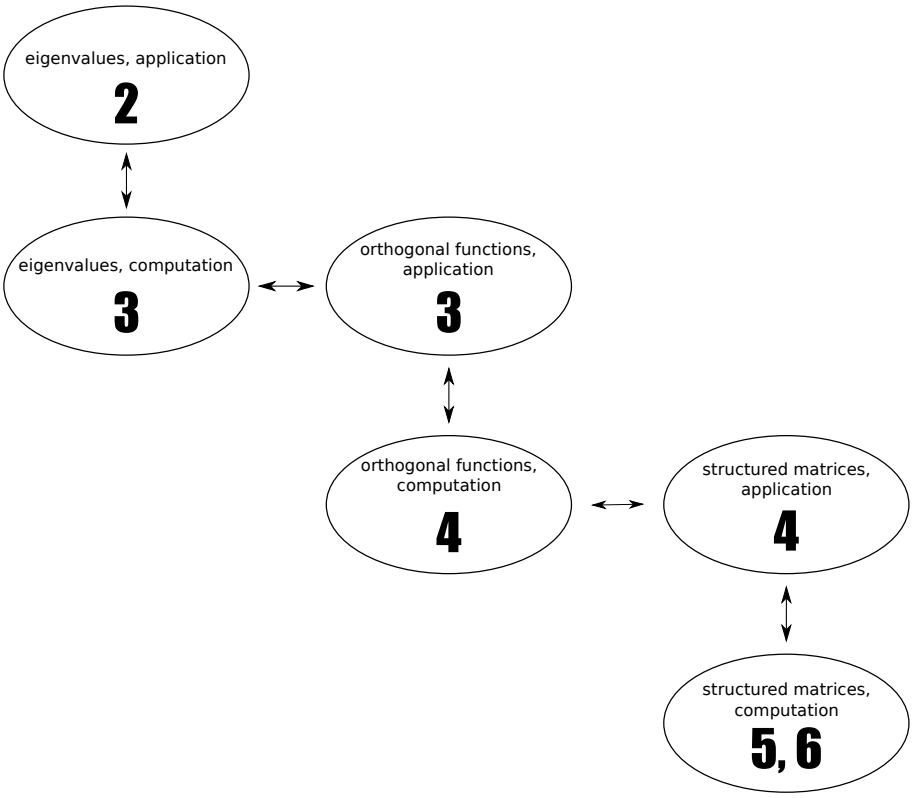


Figure 1.8: Relation between chapters

The problems we address are application-driven.

The first application comes from graph theory, and allows to find a pair of cospectral graphs such that one is regular and another one is not. Graphs are mapped to structured matrices, namely, their generalized adjacency matrices, and by exploring some properties of these matrices and their characteristic polynomials, it becomes possible to see their cospectrality, thus solving the problem.

The second application is model reduction for dynamical systems, where a rational Lanczos method is widely used. Here the fundamental results from the theory of discrete orthogonal rational functions allow to numerically describe the distribution of the Ritz values, thus specifying regions of convergence for the rational Lanczos method.

The third application is the discrete least squares approximation problem. It is solved by reducing it to the problem of computing bivariate orthogonal polynomials. These polynomials, in turn, are found by solving a certain structured inverse eigenvalue problem.

Structured matrices under consideration within this work are also coming from practice. So, for example, Toeplitz and diagonal-plus-semiseparable matrices appear while discretizing differential equations, and linear systems with two-level Toeplitz matrices arise in image deblurring.

The following is a chapter-by-chapter overview.

- Chapter 2 gives an introduction to spectral graph theory. First, several basic concepts from graph theory are defined, including different types of adjacency matrices, spectrum of the graph, and regularity. After this, by means of several known theorems it is shown which graph-theoretic properties could be deduced from the spectrum of the graph. Later, the new result considering whether a regularity of the graph can be deduced from its spectrum with respect to a certain generalized adjacency matrix is derived. The answer is 'no', and several small counterexamples are found by computer enumeration. Finally, a general procedure, allowing to construct more counterexamples, is described. The results of this chapter have been reported in the journal paper [33].

- Chapter 3 is devoted to the convergence behavior of the rational Lanczos method. We begin with necessary concepts from logarithmic potential theory and prove some properties of a weighted logarithmic potential. Then we formulate the classical Lanczos method and briefly describe how to predict its regions of convergence, applying potential theory tools. Later we generalize these ideas to the rational Lanczos case. Finally, we present a novel method to numerically solve the constrained weighted energy problem, describing a distribution of converged rational Ritz values. Several numerical experiments show the validity of the approach. The results of this chapter have been reported in the journal paper [32].

- Chapter 4 develops an algorithm to compute recurrence relation coefficients for bivariate polynomials, orthonormal with respect to a discrete inner product. We start with an application, namely, with the discrete least squares approximation problem. Firstly, we briefly review the existing theory for the univariate case. This includes a relation to an inverse eigenvalue problem. Secondly, we generalize these ideas to the multivariate case and pose a pair of coupled inverse eigenvalue problems, describing the recurrence relation coefficients of the target polynomials. These coupled problems are later solved by means of a novel updating

algorithm. Finally, the algorithm is applied in several numerical examples. The results of this chapter have been reported in the journal paper [161].

- Chapter 5 is technical. First, we define several classes of matrix structure, like Toeplitz and semiseparable. Then we formulate a general concept of a homotopy approach and briefly specify it for matrix inversion and eigenvalue problems. The facts presented in this section are used later in Chapter 6.

- Chapter 6 presents three algorithms for three classes of structured matrices. First, a homotopy approach is applied to solve a linear system with a Toeplitz coefficient matrix. The compact representation of the inverse of the coefficient matrix comes for free while executing the method. Then all the eigenvalues and eigenvectors of a symmetric diagonal-plus-semiseparable matrix are computed by another version of a homotopy algorithm. Finally, we derive a direct method to solve a two-level Toeplitz linear system with banded outer structure. The results of this chapter have been reported in the journal papers [164, 34] and in the report [35].

- Chapter 7 summarizes and discusses the presented work and gives suggestions for future work.

Chapter 2

Spectra of graphs and regularity

In this chapter we study different properties of graph spectra and show which properties of a graph may be deduced from its spectrum. First, we give some basic definitions in Section 2.1. Later in Section 2.2 we present several general results, coupling certain characteristics of matrices, associated with a graph, with graph-theoretic properties of the underlying graph. Section 2.3 discusses several known methods to construct cospectral graphs, such as Seidel switching, Godsil-McKay switching and computer enumeration. Finally, in Section 2.4 we give an answer to an open question whether a regular and non-regular graph could be cospectral with respect to a certain class of generalized adjacency matrices. This section is based on our paper [33].

2.1 Definitions

In this section we will give the necessary definitions from graph theory and present some simple known results on graph spectra.

2.1.1 Graph-theoretic notions

By a *graph* $\Gamma = (V_\Gamma, E_\Gamma)$ we mean a finite set V_Γ together with a set E_Γ of two-element subsets of V_Γ. The elements of V_Γ are called *vertices* and the elements of E_Γ are called *edges*. If those two-element subsets are considered as ordered pairs, Γ is called a *directed* graph, otherwise an *undirected* graph.

In what follows we will mostly consider only *simple graphs*, namely, finite undirected graphs without loops or multiple edges (a *loop* is an edge with both of its vertices identical).

Two vertices u and v are called *adjacent* if they are connected by an edge $e = (u, v)$. In this case the edge e is said to be *incident* with the vertices u and v.

The number of edges incident with a vertex in a graph is called the *degree* of the vertex. If all the vertices have the same degree r, the graph is called *regular of degree* r.

The *complement* $\overline{\Gamma}$ of a graph Γ is the graph with the same vertex set as Γ, where any two distinct vertices are adjacent if and only if they are non-adjacent in Γ.

Any sequence of consecutive edges in a graph is called a *walk*. A walk can pass through the same edge more than once.

A graph is called connected, if any two of its vertices are joined by a walk. A graph is *disconnected* if it is not connected, and it then consists of two or more parts called *connected components*, two vertices being in different components if they cannot be joined by a walk.

An isomorphism of graphs Γ_1 and Γ_2 is a bijection between the vertex sets V_{Γ_1} of Γ_1 and V_{Γ_2} of Γ_2: $f : V_{\Gamma_1} \to V_{\Gamma_2}$ such that any two vertices u and v of Γ_1 are adjacent in Γ_1 if and only if $f(u)$ and $f(v)$ are adjacent in Γ_2. If an isomorphism exists between two graphs, then the graphs are called *isomorphic*, and otherwise *nonisomorphic*. The graph isomorphism is an equivalence relation on graphs.

2.1.2 Matrices associated to a graph

Let Γ be a simple graph whose vertex set is $\{x_1, \ldots, x_n\}$. The *adjacency matrix* $\mathbf{A}$ of Γ is a square matrix of order n, whose entry $(\mathbf{A})_{ij} = 1$ when vertices x_i and x_j are adjacent in Γ and $(\mathbf{A})_{ij} = 0$ otherwise. We may explicitly specify the graph by appending the index, as $\mathbf{A}_\Gamma$.

The *Laplace matrix* of Γ is a square matrix of order n with zero row sums, where $(\mathbf{L})_{ij} = -(\mathbf{A})_{ij}$ for $i \neq j$. If $\mathbf{D}$ is the diagonal matrix such that $\mathbf{D}_{ii}$ is the degree of x_i, then $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The matrix $\mathbf{Q} = \mathbf{D} + \mathbf{A}$ is called the *signless Laplace matrix* of Γ. An important property of the Laplace matrix $\mathbf{L}$ and the signless Laplace matrix $\mathbf{Q}$ is that they are symmetric positive semidefinite.

Let J denote the square all-ones matrix and I the identity matrix. Consider a matrix $\mathbf{M} = \alpha\mathbf{A} + \beta J + \gamma I$, where α, β, γ are parameters. Any such matrix, with $\alpha \neq 0$, is called a *generalized adjacency matrix* of Γ.

The *Seidel adjacency matrix* of a graph Γ with adjacency matrix $\mathbf{A}$ is the matrix $\mathbf{S}$ defined by $\mathbf{S} = J - I - 2\mathbf{A}$.

2.1.3 The spectrum of a graph

The *spectrum* of a simple graph Γ with respect to some associated matrix $\mathbf{M}$ is by definition the spectrum of this matrix $\mathbf{M}$, that is, its set of eigenvalues together with their multiplicities. One may use more specific terms like *Laplace spectrum* when it's clear which matrix is kept in mind. If the matrix is not specified, the spectrum of the graph means the *ordinary spectrum*, e.g. that of its adjacency matrix $\mathbf{A}$.

Similarly, the *characteristic polynomial* of Γ with respect to some associated matrix $\mathbf{M}$ is that of $\mathbf{M}$, that is, the polynomial $p_{\mathbf{M}}$ defined by $p_{\mathbf{M}}(x) = \det(xI - \mathbf{M})$.

Two nonisomorphic graphs Γ and Δ are called *cospectral* with respect to some associated matrix $\mathbf{M}$, when their spectra with respect to this matrix coincide. Again, if the type of associated matrix is not specified, the adjacency matrix $\mathbf{A}$ is meant.

It is easy to show that the spectrum of any generalized adjacency matrix is obtained by scaling and shifting from that of a matrix of the form $\mathbf{G} = \mathbf{A} - yJ$, so for matters of graph cospectrality we will further restrict ourselves to this special case. For example, the spectrum of the Seidel matrix maps to the spectrum of $\overline{\mathbf{S}} = \mathbf{A} - \frac{1}{2}J$.

We will call two graphs Γ and Δ *y-cospectral* (for some real y) when $\mathbf{A}_\Gamma - yJ$ and $\mathbf{A}_\Delta - yJ$ have the same spectrum. Then 0-cospectral is what we called cospectral, and $\frac{1}{2}$-cospectral is Seidel-cospectral, and 1-cospectrality is cospectrality for the complementary graphs.

2.2 Structural properties and graph spectra

One may become interested whether some structural properties of a graph correspond to certain properties of the spectrum and vice versa. We will briefly summarize some basic results for classical associated matrices, following the lecture notes book [21].

Suppose Γ is simple with n vertices. Since $\mathbf{A}_\Gamma$ is real and symmetric, all its eigenvalues are real. Also, for each eigenvalue λ, its algebraic multiplicity coincides with its geometric multiplicity, so that we may just speak about "multiplicity". Since $\mathbf{A}$ has zero diagonal, its trace $\operatorname{tr}\mathbf{A}$, and hence the sum of the eigenvalues is zero.

Similarly, $\mathbf{L}$ is real and symmetric, thus the Laplace spectrum is real. Moreover, $\mathbf{L}$ is positive semidefinite and singular, so we may denote the eigenvalues by $\mu_1, \dots, \mu_n$, where $0 = \mu_1 \leqslant \mu_2 \leqslant \cdots \leqslant \mu_n$. The sum of these eigenvalues is $\operatorname{tr}\mathbf{L}$, which is twice the number of edges of Γ.

Finally, also $\mathbf{Q}$ has real spectrum and nonnegative eigenvalues (but is not necessarily singular). We have $\operatorname{tr}\mathbf{Q} = \operatorname{tr}\mathbf{L}$.

2.2.1 Regular graphs

We can translate the definition of regularity into the matrix language. By definition, every vertex in a regular graph has precisely k neighbors. So, Γ is regular of degree k precisely when its adjacency matrix $\mathbf{A}$ has row sums k, i.e., when $A\mathbf{1} = k\mathbf{1}$ (or $\mathbf{A}J = kJ$), here $\mathbf{1}$ denotes the all-ones vector. From this definition follows that k is an ordinary eigenvalue of a regular graph.

If Γ is regular of degree k, then for every eigenvalue λ of $\mathbf{A}$ we have $|\lambda| \leqslant k$. (One way to see this, is by observing that if $|t| > k$ then the matrix $tI - \mathbf{A}$ is strictly diagonally dominant, and hence nonsingular, so that t is not an eigenvalue of $\mathbf{A}$.)

If Γ is regular of degree k, then $\mathbf{L} = kI - \mathbf{A}$. It follows that if Γ has ordinary eigenvalues $k = \lambda_1 \geqslant \cdots \geqslant \lambda_n$ and Laplace eigenvalues $0 = \mu_1 \leqslant \mu_2 \cdots \leqslant \mu_n$, then $\lambda_i = k - \mu_i$ for $i = 1, \dots, n$. The eigenvalues of $\mathbf{Q} = kI + \mathbf{A}$ are $2k$, $k + \lambda_2$, $\dots$, $k + \lambda_n$.

2.2.2 Complements

Suppose that a simple graph Γ has adjacency matrix $\mathbf{A}$, then $\overline{\Gamma}$ has adjacency matrix $\overline{\mathbf{A}} = J - I - \mathbf{A}$ and Laplace matrix $\overline{\mathbf{L}} = nI - J - \mathbf{L}$. Because eigenvectors of $\mathbf{L}$ are also eigenvectors of J, the eigenvalues of $\overline{\mathbf{L}}$ are $0, n-\mu_n, \ldots, n-\mu_2$. In particular, $\mu_n \leqslant n$. We keep here the notation μ_i from the previous subsection for Laplace eigenvalues.

If Γ is regular a similar result holds for the ordinary eigenvalues: if Γ is k-regular with eigenvalues $\lambda_1 \geqslant \cdots \geqslant \lambda_n$, then the eigenvalues of the complement are $n-k-1, -1-\lambda_n, \ldots, -1-\lambda_2$.

2.2.3 Walks

From the spectrum one can read off the number of closed walks of a given length.

Proposition 1. *Let h be a nonnegative integer. Then $(\mathbf{A}^h)_{ij}$ is the number of walks of length h from i to j. In particular, $(\mathbf{A}^2)_{ii}$ is the degree of the vertex i, $\operatorname{tr}\mathbf{A}^2$ equals twice the number of edges of Γ; similarly, $\operatorname{tr}\mathbf{A}^3$ is six times the number of triangles in Γ.*

2.2.4 Connectedness

The spectrum of a disconnected graph is easily found from the spectra of its connected components:

Proposition 2. *Let Γ be a graph with connected components Γ_i $(1 \leqslant i \leqslant s)$. Then the spectrum of Γ is the union of the spectra of Γ_i (and multiplicities are added). The same holds for the Laplace and the signless Laplace spectrum.*

The number of connected components itself may be deduced from the Laplace spectrum:

Proposition 3. *The multiplicity of 0 as a Laplace eigenvalue of an undirected graph Γ equals the number of connected components of Γ.*

There is some connection between the ordinary spectrum and regularity:

Proposition 4. *Let the undirected graph Γ be k-regular. Then k is the largest eigenvalue of Γ, and its multiplicity equals the number of connected components of Γ.*

2.3 Finding cospectral graphs

In Section 2.1 we gave a definition of cospectral graphs. One may wonder, whether cospectral graphs do exist at all. The first example of cospectral graphs was found by Collatz and Sinogowitz [41]. They presented a pair of cospectral trees. Another famous example is found by Cvetković [43] and is often called the Saltire pair (since the two pictures superposed give the Scottish flag: *Saltire*), see Figure 2.1. It is easy to verify that both graphs have a spectrum $\{[2]^1, [0]^3, [-2]^1\}$, where powers denote multiplicities. For graphs on less than five vertices, no pair with cospectral adjacency matrices exists, so each of these graphs is *determined by its spectrum*. We will further shorten this notation to DS.

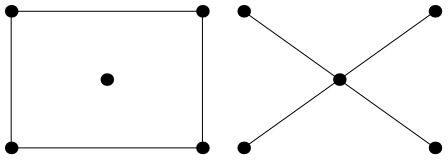


Figure 2.1: Pair of graphs, cospectral wrt $\mathbf{A}$

Since we know that cospectral graphs exist, one may pose the question "how to find those cospectral pairs?" In the next subsections we will present some early answers, following the review [167]. First, we describe several constructive techniques, and later we show the results coming from a computer enumeration.

2.3.1 Constructive answers

There are several ways to construct cospectral pairs. One way is to use simple transformations of graphs themselves. Schwenk [140] went this way while constructing cospectral trees. Another way to discover cospectral mates is to start with some (associated) matrix, and transform it by some similarity transformation to another matrix. If this second matrix happens to be an associated matrix of the same type for another graph, a desired cospectral pair is constructed. Because of the nature of the underlying transformation, this procedure is often called *switching*. Below we present two switching ideas that are due to van Lint and Seidel [173] and Godsil and McKay [68].

Constructing cospectral trees

Consider the adjacency spectrum. Suppose we have two cospectral pairs of graphs. Then the disjoint unions one gets by uniting graphs from different pairs, are clearly also cospectral. Schwenk [140] examined the case of uniting disjoint graphs by identifying a fixed vertex from one graph with a fixed vertex from the other graph. Such a union is called a *coalescence* of the graphs with respect to the fixed vertices. He proved the following lemma.

Lemma 1. *Let* Γ *and* Γ' *be cospectral graphs and let* u *and* u' *be vertices of* Γ *and* Γ' *respectively. Suppose that* $\Gamma - u$ *(that is the subgraph of* Γ *obtained by deleting* u*) and* $\Gamma' - u'$ *are cospectral too. Let* Δ *be an arbitrary graph with a fixed vertex* v*. Then the coalescence of* Γ *and* Δ *with respect to* u *and* v *is cospectral with the coalescence of* Γ' *and* Δ *with respect to* u' *and* v*.*

For example, let $\Gamma = \Gamma'$ be as given below, then $\Gamma - u$ and $\Gamma' - u'$ are cospectral because they are isomorphic. Suppose that Δ and v are as shown on Fig. 2.2. Then it follows from Lemma 1 that the graphs on Fig. 2.3 and Fig. 2.4 are cospectral.

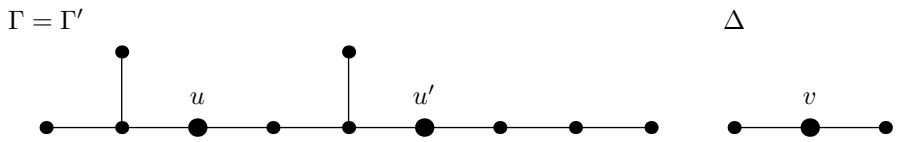


Figure 2.2: Graphs $\Gamma = \Gamma'$ and Δ

Schwenk's method is very suitable for constructing cospectral trees. In fact, the lemma above enabled him to prove his famous theorem:

Theorem 1. *With respect to the adjacency matrix, almost all trees are non-DS.*

Seidel switching

Different associated matrices have different sensitivity when it comes to the DS question. The worst example here is the Seidel matrix $\mathbf{S}$. Van Lint and Seidel [173] introduced a special transformation of the Seidel matrix of a given graph Γ, that creates a cospectral mate for Γ. This transformation was further explored by Seidel in [142].

For a given partition of the vertex set of Γ, consider the following operation on the Seidel matrix $\mathbf{S}$ of Γ:

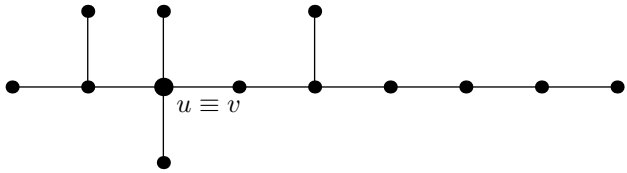


Figure 2.3: Coalescence of Γ and Δ wrt u and v

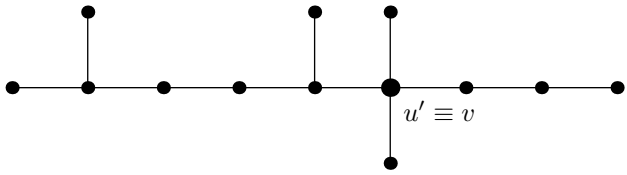


Figure 2.4: Coalescence of Γ' and Δ wrt u' and v

$$\mathbf{S} = \begin{pmatrix} S_1 & S_{12} \\ S_{12}^T & S_2 \end{pmatrix} \sim \begin{pmatrix} S_1 & -S_{12} \\ -S_{12}^T & S_2 \end{pmatrix} = \overline{\mathbf{S}}.$$

Observe that $\overline{\mathbf{S}} = \overline{I}\mathbf{S}\overline{I}^{-1}$, where $\overline{I} = \overline{I}^{-1} = \mathrm{diag}(1, \ldots, 1, -1, \ldots, -1)$, which means that $\mathbf{S}$ and $\overline{\mathbf{S}}$ are similar, and therefore $\mathbf{S}$ and $\overline{\mathbf{S}}$ are cospectral. Let Γ be the graph with Seidel matrix $\mathbf{S}$. The operation that changes Γ into $\overline{\Gamma}$ is called Seidel switching. Note that only in the case that S_{12} has equally many times a -1 as a $+1$, Γ has the same number of edges as $\overline{\Gamma}$. So Γ is hardly ever isomorphic to $\overline{\Gamma}$. And it is easy to check that S_{12} cannot have the mentioned property for all possible partitions. Thus we have:

Proposition 5. *With respect to the Seidel matrix, no graph with more than one vertex is DS.*

It is also clear that if Γ is regular, $\overline{\Gamma}$ is in general not regular.

Godsil-McKay switching

In some cases Seidel switching also leads to cospectral graphs for the adjacency spectrum (for example if the graphs Γ and $\overline{\Gamma}$ are regular of the same degree). Godsil and McKay [68] consider a more general version of Seidel switching

and give conditions under which the adjacency spectrum is unchanged by this operation. We will refer to their method as GM switching.

Theorem 2. *Let N be a $(0,1)$-matrix of size $b \times c$ (say) whose column sums are 0, b or $b/2$. Define $\overline{N}$ to be the matrix obtained from N by replacing each column $\mathbf{v}$ with $b/2$ ones by its complement $\mathbf{1} - \mathbf{v}$. Let B be a symmetric $b \times b$ matrix with constant row (and column) sums, and let C be a symmetric $c \times c$ matrix. Put*

$$M = \begin{pmatrix} B & N \\ N^T & C \end{pmatrix} \quad \text{and} \quad \overline{M} = \begin{pmatrix} B & \overline{N} \\ \overline{N}^T & C \end{pmatrix}. \tag{2.1}$$

Then M and $\overline{M}$ are cospectral.

More recent constructive procedures for cospectral pairs are presented in the review [168].

2.3.2 Computer enumeration

The growing power of computer systems leads to one more procedure for finding cospectral pairs, besides the theoretical exploration, namely – computer enumeration.

The above-mentioned paper [68] by Godsil and McKay already gave interesting computer results for cospectral graphs. In [68] all graphs up to 9 vertices are generated and checked on cospectrality. This enumeration has been extended to 11 vertices by Haemers and Spence [76], and cospectrality was tested with respect to the adjacency matrix $\mathbf{A}$, the Laplacian matrix $\mathbf{L}$, and the signless Laplacian matrix $\mathbf{Q}$.

Table 2.1 represents a part of a similar table by Haemers and Spence [76] with updates from [22]. The columns $\mathbf{A}$, $\mathbf{L}$ and $\mathbf{Q}$ contain the fractions of graphs, having a cospectral mate with respect to a corresponding matrix, The last column gives a fraction of graphs with a cospectral mate which can be constructed by GM switching.

From these results follows, that for the enumerated cases a large part of all cospectral graphs comes from GM switching, and that the fraction of graphs on n vertices with a cospectral mate starts to decrease at some value of $n < 11$ (depending on the matrix). Since the fraction of cospectral graphs on n vertices constructible by GM switching tends to 0 if $n \to \infty$, Haemers and Spence see this as an indication that possibly almost no graph has a cospectral mate. However, at the present time only lower asymptotic bounds for the number of graphs with a cospectral mate are present, see [21].

Table 2.1: Fractions of graphs with cospectral mates, [76]

n	# graphs	**A**	**L**	**Q**	GM
2	2	0	0	0	0
3	4	0	0	0	0
4	11	0	0	0.182	0
5	34	0.059	0	0.118	0
6	156	0.064	0.026	0.103	0
7	1044	0.105	0.125	0.098	0.038
8	12346	0.139	0.143	0.097	0.085
9	274668	0.186	0.155	0.069	0.139
10	12005168	0.213	0.118	0.053	0.171
11	1018997864	0.211	0.090	0.038	0.174
12	165091172592	0.188	NA	NA	NA

Enumeration of cospectral graphs on 12 vertices is studied in the recent paper [22].

2.4 Regularity and generalized adjacency matrix

In this section we will investigate for which matrices one can see from the spectrum whether the graph is regular.

The basic result here comes from the reference book [44], see also [21].

Theorem 3. *For the adjacency matrix, the Laplacian matrix and the signless Laplacian matrix of a graph* Γ*, the following can be deduced from the spectrum.*

1. *The number of vertices.*
2. *The number of edges.*
3. *Whether* Γ *is regular.*

For the adjacency matrix the following follows from the spectrum.

1. *The number of closed walks of any fixed length.*
2. *Whether* Γ *is bipartite.*

For the Laplacian matrix the following follows from the spectrum.

1. *The number of connected components.*
2. *The number of spanning trees.*

For generalized adjacency matrices the following result was proved in [167].

Proposition 6. *With respect to the generalized adjacency matrix* $\mathbf{M}(\alpha, \beta, \gamma) = \alpha\mathbf{A}+\beta J+\gamma I$*, a regular graph cannot be cospectral with a non-regular one, except possibly when* $-1 < \beta/\alpha < 0$.

The statement 'a regular graph cannot be cospectral to a non-regular one' (we further call it an R-statement) is clearly not true if $\beta/\alpha = -1/2$, as follows from the Seidel switching procedure and Proposition 5. For example the triangle (which is regular) and the graph on three vertices with one edge (which is non-regular) have the spectrum $\{-1, -1, \frac{1}{2}\}$ with respect to $A - \frac{1}{2}J$. So, they are cospectral with respect to this matrix and thus with respect to $\mathbf{S}$, see the remark in Subsection 2.1.2. When [167] was written, it was unknown if the above statement is true or false for $-1 < \beta/\alpha < 0$, $\beta/\alpha \neq -1/2$.

To explore this question, the following result is useful:

Proposition 7. *1. (Johnson and Newman [90]) If two graphs are* y*-cospectral for two distinct values of* y*, then they are cospectral for all* y*.*

2. (Van Dam, Haemers and Koolen [169]) If two graphs are y*-cospectral for an irrational value of* y*, then they are cospectral for all* y*.*

For irrational values of $\beta/\alpha \in (-1, 0)$ the R-statement is correct, as follows from Proposition 7, see [169] for details. We will show now that, provided β/α is rational and $-1 < \beta/\alpha < 0$, $\beta/\alpha \neq -1/2$, the R-statement is false.

The part by Johnson and Newman of Proposition 7 says if two graphs are cospectral with respect to two generalized adjacency matrices with different values of β/α, then they are cospectral with respect to all generalized adjacency matrices, and therefore they are both regular or both non-regular. In other words, if one graph is regular and the other one not, the graphs can only be cospectral for one value of β/α. These remarks show some difficulties that should be dealt with in finding counterexamples to the above statement. Both graphs must have the same number of edges (see, for example, [169]), and may not be cospectral for any other value of β/α. This excludes most of the known tricks for constructions of cospectral generalized adjacency matrices.

The first counterexamples to the statement were found by computer enumeration. We will present them in subsection 2.4.1. Then in Subsection 2.4.2 we present a theoretic construction for counterexamples, which may be seen as a generalization of switching techniques.

2.4.1 Computer results

If two graphs are cospectral with respect to $\mathbf{M}(1, \beta/\alpha, 0)$, then their complements are cospectral with respect to $\mathbf{M}(1, -1 - \beta/\alpha, 0)$. So one may use to following algorithm to construct counterexamples.

1. Choose any integer value of β and α such that $-1/2 < \beta/\alpha < 0$ and $\gcd(\alpha, \beta) = 1$.
2. Choose the number of vertices $n \geqslant 3$.
3. Generate all graphs on n vertixes.
4. Compute the characteristic polynomial for each graph with respect to $\mathbf{M}(\alpha, \beta, 0)$.
5. Look through the generated polynomials and find identical ones. See if one of the graphs corresponding to one of such polynomials is regular and the other graph is not.

Graphs were generated using the package `nauty` by McKay [111]. Then graphs were stored on a disc in a compressed form.

The graphs generated were then fed to GAP [134] in such a way that strings were produced comprising the coefficients of the characteristic polynomials, and these were stored in a file, one to each line. To the end of each line a suffix `true` or `false` was added, `true` stands for regular graphs and `false` for nonregular. This file was then sorted using the Linux utility `sort`, after which it was easy to look through this file and take identical polynomials such that one has suffix `true` and another has suffix `false`. Some of the enumeration ideas are taken from [76].

We arbitrarily tried some values of α and β. For $\beta/\alpha = -1/4$ we found exactly two regular-nonregular pairs of cospectral graphs on less than 10 vertices. This pair is given in Fig. 2.5. It is clear that one graph is regular, whilst the other one is not. For both graphs the characteristic polynomial with respect to $4A - J$ ($\alpha = 4$, $\beta = -1$, $\beta/\alpha = -1/4$) is

$$x^9 + 9x^8 - 72x^7 - 848x^6 + 19200x^4 + 38912x^3 - 110592x^2 - 393216x - 262144.$$

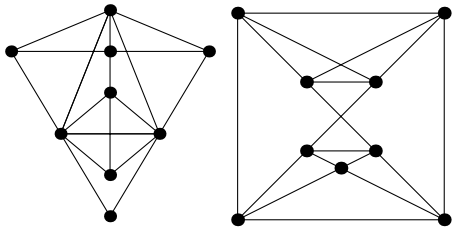


Figure 2.5: First pair of cospectral graphs wrt $4A - J$

Fig. 2.6 presents the second pair of such graphs (Libra and a hexagon with a triangle). Both graphs are disconnected. Their characteristic polynomial with respect to $4A - J$ is

$$x^9 + 9x^8 - 144x^7 - 1312x^6 + 5376x^5 + 54016x^4 - 40960x^3 \\ - 581632x^2 + 262144x + 1376256.$$

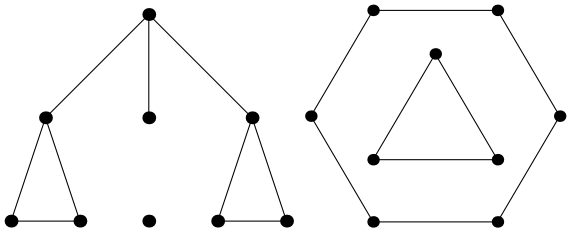


Figure 2.6: Second pair of cospectral graphs wrt $4A - J$

We also tried for $\alpha = 7$, $\beta = -3$ and we found a regular graph cospectral with two nonisomorphic nonregular graphs. Fig. 2.7 presents them. Their characteristic polynomial with respect to $7A - 3J$ is

$$x^9 + 27x^8 - 126x^7 - 6762x^6 - 343x^5 + 545027x^4 - 67228x^3 \\ - 13647284x^2 + 13176688x.$$

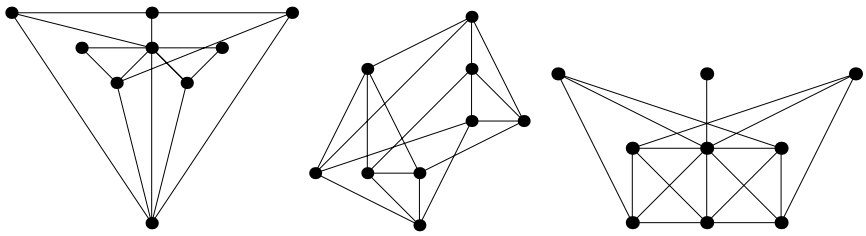


Figure 2.7: Triplet of cospectral graphs wrt $7A - 3J$

There is only one cospectral regular-nonregular pair on nine vertices with respect to $7A - 3J$ (and none on fewer vertices). For more small examples of such cospectral pairs see [169].

2.4.2 Construction of a cospectral pair

Graph partitions

To prove Theorem 4 of this subsection, we will make use of partitions of V_Γ, and therefore several general definitions and basic results are necessary. A detailed study of different graph partitions is presented by Godsil in [67, Ch. 5].

A partition of V_Γ is by definition a set $\pi = (C_1, \ldots, C_k)$, whose elements C_k are themselves disjoint non-empty subsets of V_Γ, and $\cup_i C_i = V_\Gamma$. The elements of partition π are called *cells*. A partition π is called *equitable* if, for all i and j the number of neighbors which a vertex in C_i has in the cell C_j is independent of the choice of the vertex in C_i.

Example 1. Consider a graph on Figure 2.8. Then the partition $\pi = (C_1, C_2)$ with two cells $C_1 = \{1, 2, 4, 5, 7, 8\}$ and $C_2 = \{3, 6\}$ is equitable.

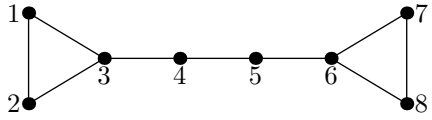


Figure 2.8: McKay's graph

Given an equitable partition $\pi = (C_1, \ldots, C_k)$ of a graph Γ, we now define the square $k \times k$ *quotient matrix* $\mathbf{C} = \mathbf{A}_{\Gamma/\pi}$ of Γ with respect to π by letting $(\mathbf{C})_{ij}$

to denote the number of edges which join a fixed vertex in C_i to vertices in C_j. Thus, in Example 1 we have

$$\mathbf{A}_{\Gamma/\pi} = \begin{pmatrix} 1 & 1 \\ 3 & 0 \end{pmatrix}.$$

The *characteristic vector* $v_i = v_i(\pi)$ of a partition $\pi = (C_1, \ldots, C_k)$ of a set of n elements is a vector of length n such that $(v_i)_j = 1$ if the j-th vertex of Γ is contained in C_i, and 0 otherwise. The *characteristic matrix* $P = P(\pi)$ of partition π is a $n \times k$ matrix with columns formed by all the characteristic vectors of the cells of π.

Recall that $\mathbf{A}_\Gamma$ denotes the adjacency matrix of the graph Γ. The following lemma provides then a neccessary and sufficient condition for an equitable partition.

Lemma 2. *Let π be a partition of the vertex set of the graph Γ with a characteristic matrix P. If π is equitable, then $\mathbf{A}_\Gamma P = P\mathbf{A}_{\Gamma/\pi}$. Conversely, π is equitable only if there is a matrix $\mathbf{B}$ such that $\mathbf{A}_\Gamma P = P\mathbf{B}$.*

The quotient matrix $\mathbf{A}_{\Gamma/\pi}$ provides some information about the eigenvalues and eigenvectors of $\mathbf{A}_\Gamma$, as shown by the following lemma.

Lemma 3 ([67])**.** *Let π be an equitable partition of the graph Γ with c cells. Assume $P = P(\pi)$, $\mathbf{A} = \mathbf{A}_\Gamma$ and $\mathbf{B} = \mathbf{A}_{\Gamma/\pi}$. We have:*

1. *If $\mathbf{B}x = \theta x$ then $\mathbf{A}Px = \theta Px$.*
2. *If $\mathbf{A}y = \theta y$ then $y^T P\mathbf{B} = \theta y^T P$.*
3. *The characteristic polynomial of $\mathbf{B}$ divides the characteristic polynomial of $\mathbf{A}$.*

For the proof of both lemmas we refer to the book [67].

Generalized switching

We will introduce now a new theorem regarding cospectrality with respect to generalized adjacency matrices of the form $\mathbf{A} - yJ$. This result extends the numerical findings of Subsection 2.4.1. While proving the theorem we will describe a generalization of a switching technique, that is, a procedure to explicitly construct cospectral mates with given properties. As remarked before, two graphs are cospectral with respect to a generalized adjacency matrix of the

form $\mathbf{A} - yJ$, if and only if they are also cospectral with respect to $\mathbf{M}(\alpha, \beta, \gamma)$, with $\beta/\alpha = -y$, so the restriction to a simpler class $\mathbf{A} - yJ$ is justified.

Theorem 4. *For every rational value of $y \in (0,1)$, there exists a pair of graphs, one regular and one not, that are cospectral with respect to $\mathbf{A} - yJ$.*

Proof. Write $y = p/q \in (0,1)$, such that p and q are integers and q is even. We will construct two cospectral generalized adjacency matrices $\mathbf{M}$ and $\overline{\mathbf{M}}$ of size $4q + q^2$ with entries $-p$ and $r = q - p$. Define

$$\mathbf{M} = \begin{bmatrix} K & B \\ B^T & C \end{bmatrix} \quad \text{and} \quad \overline{\mathbf{M}} = \begin{bmatrix} K & \overline{B} \\ \overline{B}^T & C \end{bmatrix}.$$

The matrices $K, B, \overline{B}$ and C are built with $q \times q$ blocks with constant row and column sums. The contruction is as follows:

$$K = \begin{bmatrix} -pJ & rJ & rJ & -pJ \\ rJ & -pJ & -pJ & -pJ \\ rJ & -pJ & -pJ & -pJ \\ -pJ & -pJ & -pJ & -pJ \end{bmatrix},$$

$$B = \begin{bmatrix} B_{1,1} & B_{1,2} & B_{1,3} & B_{1,4} & \cdots & B_{1,q-1} & B_{1,q} \\ B_{2,1} & \overline{B}_{2,2} & B_{2,3} & \overline{B}_{2,4} & \cdots & B_{2,q-1} & \overline{B}_{2,q} \\ \overline{B}_{3,1} & B_{3,2} & \overline{B}_{3,3} & B_{3,4} & \cdots & \overline{B}_{3,q-1} & B_{3,q} \\ \overline{B}_{4,1} & \overline{B}_{4,2} & \overline{B}_{4,3} & \overline{B}_{4,4} & \cdots & \overline{B}_{4,q-1} & \overline{B}_{4,q} \end{bmatrix},$$

$$\overline{B} = \begin{bmatrix} \overline{B}_{4,1} & \overline{B}_{4,2} & \overline{B}_{4,3} & \overline{B}_{4,4} & \cdots & \overline{B}_{4,q-1} & \overline{B}_{4,q} \\ \overline{B}_{3,1} & B_{3,2} & \overline{B}_{3,3} & B_{3,4} & \cdots & \overline{B}_{3,q-1} & B_{3,q} \\ B_{2,1} & \overline{B}_{2,2} & B_{2,3} & \overline{B}_{2,4} & \cdots & B_{2,q-1} & \overline{B}_{2,q} \\ B_{1,1} & B_{1,2} & B_{1,3} & B_{1,4} & \cdots & B_{1,q-1} & B_{1,q} \end{bmatrix},$$

where $B_{i,j}$ is any $q \times q$ matrix with $p-1$ times r and $r+1$ times $-p$ in each row and column, and $\overline{B}_{i,j}$ is any $q \times q$ matrix with $p+1$ times r and $r-1$ times $-p$ in each row and column. So $B_{i,j}$ has row sums $-q$ and $\overline{B}_{i,j}$ has row sums q. Notice that the first $4q$ rows of $\mathbf{M}$ all have row sum $q(q-4p)$, whilst the first $4q$ row sums of $\overline{\mathbf{M}}$ take three different values: $q(3q-4p)$, $q(q-4p)$ and $q(-q-4p)$. Also observe that $\overline{B}$ can be obtained from B by reversing the order of the block rows. The matrix

$$C = \begin{bmatrix} C_{1,1} & \cdots & C_{1,q} \\ \vdots & & \vdots \\ C_{q,1} & \cdots & C_{q,q} \end{bmatrix}$$

should be taken such that C is symmetric with diagonal entries $-p$ and all row and column sums equal to $q(q-4p)$ (which makes all row sums of $\mathbf{M}$ equal). All

blocks $C_{i,j}$ must have constant row and column sums. There are many ways to establish this. For instance, take $C_{1,1} = C_{1,2} = C_{1,q} = -pJ$, $C_{1,q/2+1} = rJ$ and for the remaining values of i take for $C_{1,i}$ any $q \times q$ matrix with p times r and r times $-p$ in each row and column. Then put $C = \text{circulant}(C_{1,1}, \ldots, C_{1,q})$.

Since all row (and also column) sums of $\mathbf{M}$ are equal and those of $\overline{\mathbf{M}}$ are not, it is clear that $\mathbf{M}$ represents a regular graph and $\overline{\mathbf{M}}$ represents a non-regular graph. What remains to be proved is that $\mathbf{M}$ and $\overline{\mathbf{M}}$ are cospectral. First observe that the given partition of $\mathbf{M}$ (and $\overline{\mathbf{M}}$) into $(q+4)^2$ blocks of size $q \times q$ is an equitable partition, that is, all blocks have constant row and column sum. The quotient matrix of such a partitioned matrix is the $(q+4) \times (q+4)$ matrix whose entries are the row sums of the blocks. For an equitable partition Lemma 3 tells that the eigenvalues of the quotient matrix are also eigenvalues of the original matrix and that the corresponding eigenvectors are constant over each partition class, that is, the eigenvectors span the column space V of $I \otimes J$. Note that the quotient matrix of $\overline{\mathbf{M}}$ can be obtained from the quotient matrix of $\mathbf{M}$ by multiplying the first four rows and columns by -1. Hence these quotient matrices are cospectral. The remaining eigenvalues of $\mathbf{M}$ and $\overline{\mathbf{M}}$ have eigenvectors in $V^\perp$. This implies that these eigenvalues are not changed if any block $\mathbf{M}_{i,j}$ of $\mathbf{M}$ is replaced by $\mathbf{M}_{i,j} + cJ$ for some constant c. Define

$$\mathbf{M}' = \begin{bmatrix} O & B \\ B^T & C \end{bmatrix} \text{ and } \overline{\mathbf{M}}' = \begin{bmatrix} O & \overline{B} \\ \overline{B}^T & C \end{bmatrix}.$$

Then for the eigenvectors in $V^\perp$, $\mathbf{M}'$ and $\mathbf{M}$ have the same eigenvalues, and so do $\overline{\mathbf{M}}'$ and $\overline{\mathbf{M}}$. But since $\overline{B}$ can be obtained from B by a row permutation, $\mathbf{M}'$ and $\overline{\mathbf{M}}'$ are cospectral. The conclusion is that $\mathbf{M}$ and $\overline{\mathbf{M}}$ have the same eigenvalues for the eigenvectors in V and for the eigenvectors in $V^\perp$. Therefore $\mathbf{M}$ and $\overline{\mathbf{M}}$ have the same spectrum. □

Further results on this topic are presented in [169].

2.5 Conclusion

After defining some basic concepts and stating some of their properties, we have given an answer to an open question in algebraic graph theory, namely, whether a regular and non-regular graph could be cospectral with respect to a certain class of generalized adjacency matrices, depending on some parameter. The answer is positive, and first examples were found by computer enumeration. Later, we proposed a generalization of a switching procedure that allows to construct such cospectral pairs for any rational value of the parameter.

Chapter 3

Potential theory and rational Ritz values

This chapter consists of two basic parts. In Section 3.1 we will briefly recall the necessary concepts from logarithmic potential theory and pose the constrained weighted energy problem. Next, we will present a novel algorithm to solve this problem numerically. In Section 3.2 we will formulate the rational Lanczos algorithm, establish its connection with a certain constrained weighted energy problem, and finally apply the numerical algorithm from Section 3.1 to predict regions of convergence for the rational Lanczos algorithm. We also give several numerical results that show good correspondence between predicted results and actual convergence of the algorithm.

3.1 Constrained weighted energy problem: a numerical approach

In this section we begin with definitions of logarithmic potential and logarithmic energy and pose certain minimization problems for both potential and energy. One of them – constrained weighted energy problem – is later discretized and solved by linear algebra methods. Comparison with the exact solution shows

good quality of the computed solution. Herein we follow our paper [32] (joint work with Deckers and Van Barel).

3.1.1 Preliminaries

The field of complex numbers will be denoted by $\mathbb{C}$ and the Riemann sphere by $\overline{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$. For the real line we use the symbol $\mathbb{R}$. Let $a \in \mathbb{C}$, then $\Re\{a\}$ refers to the real part of a. Further, we denote the imaginary unit by $\mathbf{i}$.

Let $\mathcal{M}(E)$ be the space of all Borel probability measures on $\mathbb{C}$ which are supported on a compact set E; i.e. for any $\mu \in \mathcal{M}(E)$ we have $\mu(\mathbb{C}) = 1$ and $\text{supp}(\mu) \subseteq E$. The logarithmic potential of a compactly supported measure μ is then defined (cf. [154, p. 53]) by

$$U^{\mu}(z) = \int \log \frac{1}{|z - z'|} d\mu(z'), \tag{3.1}$$

and its logarithmic energy is given by

$$I(\mu) = \int \int \log \frac{1}{|z - z'|} d\mu(z') d\mu(z). \tag{3.2}$$

Given a positive Borel measure ν on $\mathbb{C}$, with compact support $\text{supp}(\nu) \subset \mathbb{C} \setminus E$ bounded away from E and $\nu(\mathbb{C}) = s \in [0, 1]$, an important problem in logarithmic potential theory is to minimize the *weighted logarithmic energy* $I(\mu - \nu)$ among all $\mu \in \mathcal{M}(E)$. If there exists a probability measure on E with finite energy, the solution to this problem is unique and is called the *balayage-measure* of the probability measure $\eta = \nu + (1 - s)\delta_{\infty}$ (where δ_z is the unit measure whose support is the point z), from $\overline{\mathbb{C}} \setminus E$ onto E, which will be denoted by μ^{ν}. In this paper we will only consider the case in which E is an interval or a union of disjoint intervals. The minimization problem can also be characterized then in terms of its potential as follows (see e.g. [137]).

Property 5. *Let $\mu^{\nu} \in \mathcal{M}(E)$ be a solution to the problem of minimizing $I(\mu - \nu)$ among all $\mu \in \mathcal{M}(E)$. Then the potential $U^{\mu^{\nu} - \nu}(z)$ is equal to a constant C^{ν} on E and smaller than C^{ν} everywhere else. Moreover, it is the only probability measure with that property.*

We call the potential $U^{\mu - \nu}(z)$ a *weighted potential*. In the special case in which $E = [a, b]$, the density of the balayage-measure μ^{ν} is explicitly known, and given by (see also [48, Thm. 4.3])

$$\frac{d\mu^{\nu}(z)}{dz} = \frac{1}{\pi \sqrt{(b - z)(z - a)}} \int \Re \left\{ \frac{\sqrt{(u - b)(u - a)}}{u - z} \right\} d\eta(u), \quad z \in [a, b], \tag{3.3}$$

where the square root is positive for $u > b$ and the branch cut is $[a, b]$.

Next, suppose $\sigma \in \mathcal{M}(E)$ has finite logarithmic energy $I(\sigma)$, and let $t \in (0, 1)$. A related problem is then to minimize $I(\mu - \nu)$ among all $\mu \in \mathcal{M}(E)$ that satisfy $t\mu \leqslant \sigma$ (in the sense of densities). We call this problem a *constrained weighted energy problem* (CWEP). Again, there is a characterizing property in terms of its potential (see e.g. [137]).

Property 6. *Assume $U^{\sigma}(z)$ is continuous and real-valued and let μ_t^{ν} be a solution to the CWEP. Then $U^{\mu_t^{\nu}-\nu}(z)$ is equal to a constant C_t^{ν} on* $\operatorname{supp}(\sigma - t\mu_t^{\nu})$ *and smaller than or equal to C_t^{ν} everywhere else. Moreover, the only measure $\mu \in \mathcal{M}(E)$, with $t\mu \leqslant \sigma$, for which the weighted potential $U^{\mu-\nu}(z)$ is constant on* $\operatorname{supp}(\sigma - t\mu)$ *and smaller or equal to this constant everywhere else, is μ_t^{ν}.*

The special case of CWEP, in which $\nu = 0$, was historically studied first and is called the (un-weighted) *constrained energy problem* (CEP). Furthermore, the exact solution to the un-constrained problem is called the *equilibrium measure*.

Since $t\mu_t^{\nu} \leqslant \sigma$, the set $\operatorname{supp}(\sigma - t\mu_t^{\nu})$ is just the set where $t\mu_t^{\nu} < \sigma$. We now have the following lemma, where we use the notation ρ^+ to denote the positive part of a signed measure ρ.

Lemma 4. *Suppose for $\mu \in \mathcal{M}(E)$ (not necessarily with $t\mu \leqslant \sigma$) it holds that the weighted potential $U^{\mu-\nu}(z)$ is constant on* $\operatorname{supp}(\sigma - t\mu)$. *Then* $\operatorname{supp}(\sigma - t\mu_t^{\nu})$ *is a subset of* $\operatorname{supp}((\sigma - t\mu)^+)$, *and $\mu_t^{\nu} \geqslant \mu$ on* $\operatorname{supp}(\sigma - t\mu_t^{\nu})$.

PROOF. Define the Borel probability measures $\rho = \frac{\sigma - t\mu}{1-t}$ and $\rho_t = \frac{\sigma - t\mu_t^{\nu}}{1-t}$, and the external field $Q(z) = \frac{1}{1-t}\left(tU^{\nu}(z) - U^{\sigma}(z)\right)$. Then it holds that

$$U^{\rho}(z) + Q(z) = -\frac{t}{1-t}U^{\mu-\nu}(z) = C \quad \text{for} \quad z \in \operatorname{supp}(\rho),$$

and

$$\begin{aligned} U^{\rho_t}(z) + Q(z) &= -\tfrac{t}{1-t}U^{\mu_t^{\nu}-\nu}(z) = -\tfrac{t}{1-t}C_t^{\nu} \quad \text{for} \quad z \in \operatorname{supp}(\rho_t), \\ U^{\rho_t}(z) + Q(z) &= -\tfrac{t}{1-t}U^{\mu_t^{\nu}-\nu}(z) \geqslant -\tfrac{t}{1-t}C_t^{\nu} \quad \text{for} \quad z \in \operatorname{supp}(\rho). \end{aligned}$$

From [102, Lemma 3] it then follows that $\rho_t \leqslant \rho^+$ and $\operatorname{supp}(\rho_t) \subset \operatorname{supp}(\rho^+)$, which ends the proof. □

From the previous lemma it follows that, if there is a $\mu \in \mathcal{M}(E)$ for which the weighted potential $U^{\mu-\nu}(z)$ is constant on $\operatorname{supp}(\sigma - t\mu)$, then on the region where $t\mu \geqslant \sigma$ it holds that $t\mu_t^{\nu} = \sigma$.

3.1.2 Numerical algorithm

An algorithm to solve the constrained energy problem numerically was presented by Helsen and Van Barel [81]. In this section we will update this algorithm to work with weighted potentials. More specifically, we replace the potentials in the algorithm by the weighted potentials. This becomes possible using the lemmas proved in the previous subsection. First, we introduce the main idea of the algorithm, and then we treat the necessary discretization.

Main loop

We devise an algorithmic approach to solve the CWEP on the basis of Lemma 4. Given a positive Borel measure ν with compact support bounded away from E, we first look for a Borel probability measure $\mu^{(1)}$, with $\operatorname{supp}(\mu^{(1)}) = E$, whose weighted potential $U^{\mu^{(1)}-\nu}$ is constant on E. Then, on the region where $t\mu^{(1)} \geqslant \sigma$ we know that $t\mu_t = \sigma$, so that we can put $\mu^{(2)} = \sigma/t$ over there and require $U^{\mu^{(2)}-\nu}$ to be constant on the other region. This process will be repeated until at a certain point $\mu^{(k)} \leqslant \sigma/t$. The solution μ_t^ν will then be equal to $\mu^{(k)}$.

In a high level language this may look like:

Algorithm 1: Continuous version of the CWEP algorithm

begin
  $I = \operatorname{supp}(\sigma)$
  $J = \emptyset$
  $\mu = \infty$
  while $\mu \not\leqslant \sigma/t$ **do**
    $\mu|_J = \frac{1}{t}\sigma|_J$
    solve $\begin{cases} U^{\mu|_I}(z) = C - U^{\mu|_J}(z) + U^\nu(z), & \forall z \in I \\ \mu|_I(\mathbb{C}) = 1 - \mu|_J(\mathbb{C}) \end{cases}$
    $I = \{t\mu < \sigma\}$
    $J = \{t\mu \geqslant \sigma\}$
  end
  return $\mu_t^\nu = \mu$
end

The set I is the region where μ is not known yet, while J is the region where μ is already known to be equal to σ/t. The weighted potential of μ needs to be constant on I, so we solve $U^{\mu-\nu}(z) = U^{\mu|_I}(z) + U^{\mu|_J}(z) - U^\nu(z) = C$ for every

$z \in I$, where C is an unknown constant depending on ν, keeping in mind that μ has to be a probability measure: $\mu(\mathbb{C}) = \mu|_I(\mathbb{C}) + \mu|_J(\mathbb{C}) = 1$.

The output of this algorithm is a probability measure μ_t^ν that satisfies $t\mu_t^\nu \leqslant \sigma$, and whose weighted potential $U^{\mu_t^\nu - \nu}$ is constant on $\operatorname{supp}(\sigma - t\mu_t^\nu)$. If, at the same time, the potential is smaller than or equal to this constant outside of $\operatorname{supp}(\sigma - t\mu_t^\nu)$, then it follows from Property 6 that μ_t^ν is the solution of the CWEP. In the next lemma we will prove that the potential is indeed smaller than or equal to this constant outside of $\operatorname{supp}(\sigma - t\mu_t^\nu)$. In what follows, we represent the intermediate solution after step k in the algorithm by $\mu^{(k)}$, whereas the constant value of its weighted potential is denoted by $C^{(k)}$, and $S_k = \operatorname{supp}(\sigma - t\mu^{(k)})$.

Lemma 5. *For every k, the weighted potential $U^{\mu^{(k)} - \nu}$ is smaller than or equal to the constant $C^{(k)}$ outside S_k.*

PROOF. The proof will use induction on k.

The first intermediate solution $\mu^{(1)}$ is the balayage-measure μ^ν. Its weighted potential is equal to $C^{(1)} := C^\nu$ on $S_1 := E$, and is smaller than $C^{(1)}$ outside S_1.

Now suppose that the weighted potential of $\mu^{(k-1)}$ is smaller than or equal to $C^{(k-1)}$ outside S_{k-1} for $k > 1$. By construction it holds that $S_k \subset S_{k-1}$. Thus, it is sufficient to prove that

$$U^{\mu^{(k)}} - C^{(k)} \leqslant U^{\mu^{(k-1)}} - C^{(k-1)}. \tag{3.4}$$

On S_k, the relation

$$U^{\mu^{(k)} - \mu^{(k-1)}} = C^{(k)} - C^{(k-1)} \tag{3.5}$$

clearly holds true. Outside S_k it holds that $\mu^{(k)} = \sigma/t$ and $\mu^{(k-1)} \geqslant \sigma/t$, and hence, that $\mu^{(k)} - \mu^{(k-1)}$ is a negative measure. So, from

$$U^{\mu^{(k)} - \mu^{(k-1)}} = U^{(\mu^{(k)} - \mu^{(k-1)})|_{S_k}} + U^{(\mu^{(k)} - \mu^{(k-1)})|_{S_k^c}}$$

we learn that $U^{\mu^{(k)} - \mu^{(k-1)}}$ is subharmonic outside S_k, because the first term is harmonic outside S_k and the second term is subharmonic being the potential of a negative measure. The inequality in (3.4) now follows from (3.5) and the fact that a subharmonic function reaches its maximum on the boundary. □

Discretization

For notational simplicity, we will assume in this subsection that E is connected, but the results that follow are easily extended to the case in which E is a union of disjoint intervals. Furthermore, we will assume that an explicit representation $\kappa(x)$ exists for $U^\nu(x)$ on E in terms of basic operations on elementary functions.

Lemma 5 tells us that, whenever the theoretical algorithm of the previous subsection converges, the output solves the CWEP. Suppose we have a discretization $\{y_1, y_2, \ldots, y_N\}$ of $\operatorname{supp}(\sigma)$, so that the measure μ can be represented by a vector $\mathbf{v}$ containing the values μ_j of the density $d\mu/dy$ in the discretization points y_j. Algorithm 1 is then translated to the discretization by requiring the (in-)equalities of the CWEP to hold only in the discretization points.

In order to be able to compute the mass of a measure μ represented in this way, we will consider it to be piecewise linear with respect to the Lebesgue measure; i.e.,

$$d\mu(y) = (a_j y + b_j)dy \quad \text{for } y \in [y_{j-1}, y_j]. \tag{3.6}$$

The mass of the piecewise linear measure is then given by

$$\frac{1}{2}\sum_{j=2}^{N}(\mu_{j-1} + \mu_j)(y_j - y_{j-1}).$$

This expression is linear in the μ_j's, so that we can create a vector $\mathbf{m}$, only depending on the discretization points y_j, in such a way that the equality $\mathbf{m}^T\mathbf{v} = \mu(\mathbb{C})$ holds for every piecewise linear measure μ with discretization $\mathbf{v} = [\mu_1\ \mu_2\ \ldots\ \mu_N]^T$.

To compute the potential of a piecewise linear measure, we use the following primitive function for $y \mapsto \log 1/|x-y|$:

$$f(y,x) = \begin{cases} (x-y)(\log|x-y| - 1), & \text{if } y \neq x, \\ 0, & \text{if } y = x, \end{cases}$$

and for $y \mapsto y \log 1/|x-y|$:

$$g(y,x) = \begin{cases} \frac{1}{2}\log|x-y|(x^2-y^2) + \frac{1}{4}(x+y)^2, & \text{if } y \neq x, \\ y^2, & \text{if } y = x. \end{cases}$$

This gives us the following expression for the potential of μ:

$$\begin{aligned} U^{\mu}(x) &= \int_E \log \frac{1}{|y-x|} d\mu(y) = \sum_{j=2}^{N} \int_{y_{j-1}}^{y_j} \log \frac{1}{|y-x|} (a_j y + b_j) dy \\ &= \sum_{j=2}^{N} a_j (g(y_j, x) - g(y_{j-1}, x)) + b_j (f(y_j, x) - f(y_{j-1}, x)). \end{aligned} \tag{3.7}$$

Further, the a_j's and b_j's can be expressed in terms of the μ_j's by means of (3.6):

$$\begin{cases} \mu_{j-1} = a_j y_{j-1} + b_j \\ \mu_j = a_j y_j + b_j \end{cases} \Rightarrow \begin{cases} a_j = \dfrac{\mu_j - \mu_{j-1}}{y_j - y_{j-1}} \\ b_j = \mu_j - a_j y_j = \dfrac{y_j \mu_{j-1} - y_{j-1} \mu_j}{y_j - y_{j-1}}. \end{cases} \tag{3.8}$$

Plugging this into (3.7), we obtain an expression for $U^{\mu}(x)$ that is linear in the μ_j's, and hence, there is a matrix $\mathbf{P}$, only depending on the discretization points y_j, so that for every piecewise linear measure μ with discretization $\mathbf{v}$ it holds that

$$U^{\mu}(y_j) = (\mathbf{P}\mathbf{v})_j.$$

Consequently, with $U^{\nu}(y_j) = \kappa(y_j)$ we get that

$$U^{\mu-\nu}(y_j) = (\mathbf{P}\mathbf{v})_j - \kappa(y_j).$$

With this we can write down the discretized version of the CWEP. Suppose we have a set of discretization points $\{y_1, y_2, \ldots, y_N\}$ with corresponding vector $\mathbf{m}$ and matrix $\mathbf{P}$. Let $\mathbf{s}$ and $\mathbf{k}$ be the discretization of the constraint σ and the function κ respectively. Then the problem is: find a vector $\mathbf{v}$ satisfying $\mathbf{m}^T \cdot \mathbf{v} = 1$ and $t\mathbf{v} \leqslant \mathbf{s}$ (elementwise), so that $\mathbf{P}\mathbf{v} - \mathbf{k}$ is constant on the components where $t\mathbf{v} < \mathbf{s}$ and smaller or equal to it everywhere else.

In a high level language this may look like:

Algorithm 2: Discretized version of the CWEP algorithm

begin
  $I = \{1, 2, \ldots, N\}$
  $J = \emptyset$
  $\mathbf{v} = \infty\mathbf{e}(I)$
  while $\mathbf{v} \not\leqslant \mathbf{s}/t$ **do**
    $\mathbf{v}(J) = \frac{1}{t}\mathbf{s}(J)$
    solve $\begin{cases} \mathbf{P}(I,I)\cdot\mathbf{v}(I) = C\mathbf{e}(I) - \mathbf{P}(I,J)\cdot\mathbf{v}(J) + \mathbf{k}(I) \\ \mathbf{m}(I)^T\cdot\mathbf{v}(I) = 1 - \mathbf{m}(J)^T\cdot\mathbf{v}(J) \end{cases}$
    $I = \{i \mid t\mu_i < s_i\}$
    $J = \{j \mid t\mu_j \geqslant s_j\}$
  end
  return $\mathbf{v}_t^\nu = \mathbf{v}$
end

Here, I is the set of indices where $\mathbf{v}$ is not known yet and $J = \{1, 2, \ldots, N\} \setminus I$ is the set of indices where $\mathbf{v}$ is already known to be equal to $\mathbf{s}/t$. Further, $\mathbf{e}$ is the vector defined by $\mathbf{e} = [1\ 1\ \ldots\ 1]^T$. The vector $\mathbf{v}(J)$ is the vector consisting of the components of $\mathbf{v}$ with indices in J and the matrix $\mathbf{P}(I, J)$ is the matrix consisting of the rows and columns of $\mathbf{P}$ with row indices in I and column indices in J. Since in every step at least one discretization point is added to J, it is clear that Algorithm 2 will eventually terminate. (When no discretization point is added, the stopping criterion is fulfilled.)

Practically, we solve the following augmented system:

$$\begin{pmatrix} \mathbf{P}(I,I) & \mathbf{e}(I) \\ \mathbf{m}(I)^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}(I) \\ -C \end{pmatrix} = \begin{pmatrix} -\mathbf{P}(I,J)\cdot\mathbf{v}(J) + \mathbf{k}(I) \\ 1 - \mathbf{m}(J)^T\cdot\mathbf{v}(J) \end{pmatrix}.$$

It is easy to check that this system is equivalent to the inner loop system of Algorithm 2.

Convergence analysis

In the previous subsection we have developed a discrete algorithm by considering the measure being piecewise linear and requiring the (in-)equalities of CWEP to hold only in the discretization points. The following theorem shows that the solution of a discrete problem converges to the solution of a continuos problem, provided that the discretization set is dense in E.

Theorem 1. *Consider a CWEP and related weighted potentials, as defined in Property 6. Consider also a discretization* $E' = \{y_1, y_2, \ldots, y_N\}$ *of* E*, and let us denote by* μ_N *the corresponding output of Algorithm 2 for the selected CWEP and discretization. Suppose that* E' *is dense in* E *as* $N \to \infty$*. Then, as* $N \to \infty$*, measures* μ_N *converge to a measure* μ_∞*, such that it is the solution to CWEP.*

PROOF. The measure μ_N by construction satisfies the following conditions:

$$\int_E d\mu_N(y) = 1, \tag{3.9}$$

$$\forall y \in E: \ \mu_N(y) \leqslant \frac{1}{t}\sigma(y), \ \text{where } t \in (0,1). \tag{3.10}$$

Conditions (3.9)–(3.10) mean that there exists a subsequence of measures $\{\mu_{N_k}\}_{k=1}^{\infty}$ that converges to a certain μ_∞, such that

$$\int_E d\mu_\infty(y) = 1,$$

$$\forall y \in E: \ \mu_\infty(y) \leqslant \frac{1}{t}\sigma(y), \ \text{where } t \in (0,1),$$

and for any function $f(y)$, that is continous and bounded on E holds that

$$\lim_{N_k \to \infty} \int_E f(y) d\mu_{N_k}(y) = \int_E f(y) d\mu_\infty(y).$$

The function $f(y) = \log|1/(x-y)|$ is not bounded when $x \in E$. However, for any fixed x it can be approximated by a sequence of functions $\{f_m(y)\}_{m=1}^{\infty}$ such that

- $\forall m > 0$ the function $f_m(y)$ is continuos and bounded on E;
- $f_m(y) \geqslant 0 \ \forall y \in E$;
- $\forall k$ and $\forall m$ such that $k > m$, $f_k(y) \geqslant f_m(y) \ \forall y \in E$;
- $\lim\limits_{m \to \infty} f_m(y) = \log|1/(x-y)| + C, \ \forall y \in E$.

It follows then that a sequence $\int_E f_m(y) d\mu_\infty(y)$ is nondecreasing when $m \to \infty$ and it is bounded from above by $\dfrac{1}{t}U^\sigma(x)$. So, there exists a limit

$$\lim_{m \to \infty} \int_E f_m(y) d\mu_\infty(y) = U^{\mu_\infty}(x) + C \int_E d\mu_\infty(y) = U^{\mu_\infty}(x) + C.$$

Since for every m holds that

$$\lim_{N_k\to\infty}\int_E f_m(y)d\mu_{N_k}(y)=\int_E f_m(y)d\mu_\infty(y),$$

we conclude that $\forall x\in E$

$$\lim_{N_k\to\infty}U^{\mu_{N_k}}(x)=U^{\mu_\infty}(x). \tag{3.11}$$

The functions $U^{\mu_{N_k}}(x)$, as well as $U^{\mu_\infty}(x)$, are continuos and bounded on E. This means that the convergence in (3.11) is uniform $\forall x\in E$. Since we have supposed that the set $E'=\lim_{N\to\infty} y_{N,k}{}_{k=1}^{N}$ is dense in E, it follows that μ_∞ is a solution of the same CWEP that was fed to Algorithm 2, but in a continuos form. Property 6 states that there exists only one such solution, so it is equal to μ_∞.

Numerical examples

Let $\mathcal{P}_n$ denote the space of polynomials of degree less than or equal to n. For a fixed value of n we then say that a polynomial $p_m\in\mathcal{P}_n$ of degree $m\leqslant n$ has a zero at infinity of multiplicity $n-m$. So, consider now the polynomial

$$p_m(x)=\prod_{j=1}^{m}(x-\alpha_j)\in\mathcal{P}_n,$$

where the α_j's are finite and bounded away from E. Further, let $\eta_{m,n}$ denote the normalized zero counting measure defined by

$$\eta_{m,n}=\nu_{m,n}+\left(1-\frac{m}{n}\right)\delta_\infty,\qquad \nu_{m,n}=\frac{1}{n}\sum_{j=1}^{m}\delta_{\alpha_j}, \tag{3.12}$$

so that $\nu_{m,n}(\mathbb{C})=s=\frac{m}{n}\in[0,1]$. The normalized zero counting measure assigns mass $1/n$ to each zero of p_m (including those zeros that are at infinity) and the zeros are counted according to their multiplicity. From the definition of p_m and $\nu_{m,n}$ it is easy to see that

$$U^{\nu_{m,n}}(x)=-\frac{1}{n}\log|p_m(x)|=:\kappa_{m,n}(x). \tag{3.13}$$

In the special case in which $E=[a,b]$, it follows from (3.3) that the density of the exact solution to the weighted energy problem is given by

$$\frac{d\mu^{\nu_{m,n}}(x)}{dx}=\frac{\left[\sum_{j=1}^{m}\Re\left\{\frac{\sqrt{(\alpha_j-b)(\alpha_j-a)}}{\alpha_j-x}\right\}+(n-m)\right]}{n\pi\sqrt{(b-x)(x-a)}},\quad x\in[a,b]. \tag{3.14}$$

Suppose E is of the form

$$E = \bigcup_{j=1}^{J} [a_j, b_j], \quad J \geqslant 1,$$

where $a_j < b_j$ for every j and $[a_j, b_j] \cap [a_i, b_i] = \emptyset$ for $j \neq i$. To discretize E, we use for every segment $[a_j, b_j]$ rational Chebyshev points with respect to the first Chebyshev weight function $1/\sqrt{1-x^2}$ on $[-1, 1]$ (as described in [172]), based on a given sequence of N_j poles $\overline{B}_{N_j} = \{\beta_1, \ldots, \beta_{N_j}\} \subset \overline{\mathbb{C}} \setminus [-1, 1]$, and map them on $[a_j, b_j]$ by means of the transformation

$$y = \tau^{[a_j,b_j]}(x) = \frac{1}{2}\{(b_j - a_j)x + (b_j + a_j)\}.$$

In what follows, the inverse transformation will be denoted by $x = \tau^{-[a_j,b_j]}(y)$. Note that the classical Chebyshev points are a special case of rational Chebyshev points when $\beta_k = \infty$ for $k = 1, \ldots, N_j$.

Example 2. First, consider the case in which $E = [-1, 1]$ and $\nu := \nu_{m,n} = \frac{m}{n}\delta_\alpha$, with $n = m + 1 = 22$ and $\alpha = 0.5 + 0.1\mathbf{i}$. We then compare the solution given after the first iteration of the inner loop in Algorithm 2 (i.e., when the constraint is not active yet) with the exact solution computed by means of (3.14). For the discretization of E we use the sequence of poles $\overline{B}_{800} = \{\beta_k\}_{k=1}^{800}$, with

$$\beta_k = \begin{cases} \alpha, & k \leqslant 200 \\ \infty, & k > 200. \end{cases}$$

The computed solution to the weighted energy problem (WEP) is plotted on Figure 3.1, whereas the relative error of this solution (compared to (3.14)) is plotted in semi-logarithmic scale on Figure 3.2.

From (3.8) it follows that differences between the discretization points appear in the denominators during the construction of matrix $\mathbf{P}$. This causes an increase of the condition number of $\mathbf{P}$ when a finer mesh is used. The condition number of $\mathbf{P}$ in this example is equal to 1×10^5.

Example 3. Secondly, consider the case in which the constraint is given by

$$d\sigma(x) = \frac{dx}{\pi x\sqrt{(x-\alpha)(\beta-x)}}, \quad \alpha = \frac{1}{2}, \quad \beta = 2 \tag{3.15}$$

on $E = [\alpha, \beta] = [1/2, 2]$. This is the asymptotic eigenvalue distribution of a family of Toeplitz matrices with a specific structure. We take $\nu := \delta_\xi$, where

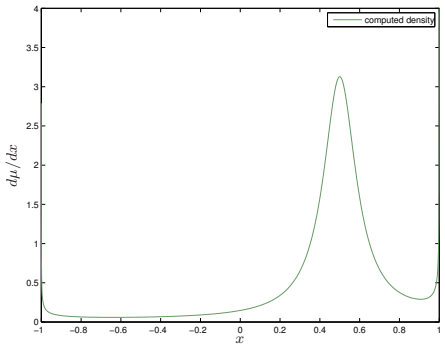


Figure 3.1: Computed solution of the WEP from Example 2.

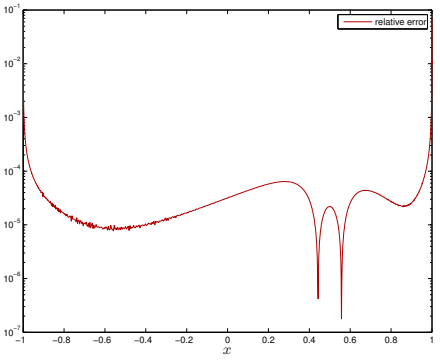


Figure 3.2: Relative error of the computed solution.

$\beta < \xi = 10$. For the discretization of E we use the sequence of poles $\overline{B}_{500} = \{\beta_k\}_{k=1}^{500}$, with

$$\beta_k = \begin{cases} \tau^{-[\alpha,\beta]}(\xi), & k \leqslant 200 \\ \infty, & k > 200. \end{cases}$$

Let t_0 and $b(t) \in [\alpha, \beta]$ be given by:

$$t_0 = \frac{1}{\beta}\sqrt{\frac{\xi-\beta}{\xi-\alpha}}, \quad b(t) = \begin{cases} \beta, & \text{if } t < t_0, \\ \frac{\xi}{t^2\beta(\xi-\alpha)+1}, & \text{if } t \geqslant t_0. \end{cases}$$

Then in [10, Lemma A.3] it has been proved that the exact solution to this CWEP is given by

$$\frac{td\mu_t^{\nu}(x)}{dx} = \begin{cases} \frac{d\sigma(x)}{dx} + \frac{t\sqrt{(\beta-\alpha)(\beta-b(x))}}{\pi(\xi-x)\sqrt{(x-\alpha)(b(t)-x)}} - \frac{\sqrt{\alpha b(t)}}{\pi x\sqrt{(x-\alpha)(b(t)-x)}}, & x \in [\alpha, b(t)] \\ \frac{d\sigma(x)}{dx}, & x \in (b(t), \beta] \end{cases}$$

Figure 3.3 shows the relative error of the computed solution to the CWEP for $t = 0.5 > t_0 = 2/\sqrt{19}$. The error is equal to zero on the segment $F = [b(t), \beta] = [40/23, 2]$, where it's not depicted. Segment F is the set where the solution coincides with the constraint (3.15).

Example 4. Further, consider the case in which the constraint is given by $d\sigma(x) = \frac{2}{\pi}\sqrt{1-x^2}dx$ on $E = [-1, 1]$, and $\nu_t := \nu_{100t-1,100t}$, where $\nu_{m,n}$ is given by (3.12), $t \in (0, 1)$ is such that $100t$ is a natural number, and

$$\alpha_j = \begin{cases} -0.9 + 0.1\mathbf{i}, & j \leqslant 50 \\ 0.5 - 0.1\mathbf{i}, & j > 50. \end{cases}$$

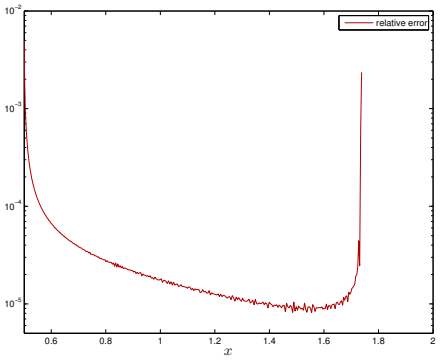


Figure 3.3: Relative error of the computed CWEP solution, Ex. 3.

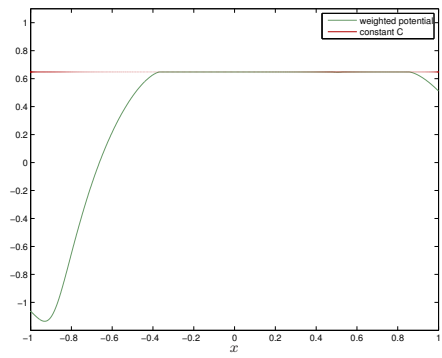


Figure 3.4: Property 6 of weighted potentials, Example 4

For the discretization of E we use the sequence of poles $\overline{B}_{1000} = \{\beta_k\}_{k=1}^{1000}$, with

$$\beta_k = \begin{cases} -0.9 + 0.1\mathbf{i}, & k \leqslant 200 \\ 0.5 - 0.1\mathbf{i}, & 200 < k \leqslant 400 \\ \infty, & k > 400. \end{cases}$$

Figure 3.5 then shows the computed solution to the CWEP for $t = 0.05 + 0.15r$, with $r = 0, \ldots, 5$. The density of σ is plotted by a thick black dashed line.

Figure 3.4 illustrates Property 6 for this example. Namely, the green line is the computed weighted potential $U^{\mu_t - \nu_t}$ for $t = 0.65$ and the red line is the constant C_t^{ν}. On some subset of E these two coincide, on its complement the potential is smaller than the constant. We would like to mention that the potential is also smaller than the constant on the segment (approximately) $[0.475, 0.535]$, but the difference is of order 10^{-3}, and hence, not visible on the figure.

Example 5. Next, consider the case in which the constraint is given by $d\sigma(x) = \frac{1}{2}dx$ on $E = [0,1] \cup [2,3]$, and $\nu := \delta_\alpha$, where $\alpha = 0.7 + 0.1\mathbf{i}$. For the discretization of $[0,1]$ we use the sequence of poles $\overline{B}_{400} = \{\beta_k\}_{k=1}^{400}$, with

$$\beta_k = \begin{cases} \tau^{-[0,1]}(\alpha), & k \leqslant 100 \\ \infty, & 200 < k \leqslant 400, \end{cases}$$

whereas for segment $[2,3]$ we use the sequence of poles $\overline{B}'_{400} = \{\beta_k\}_{k=401}^{800}$, with

$$\beta_k = \begin{cases} \tau^{-[2,3]}(\alpha), & 400 < k \leqslant 500 \\ \infty, & k > 500, \end{cases}$$

The computed solution to the CWEP is then plotted on Figure 3.6 for $t = 0.05 + 0.15r$, with $r = 0, \ldots, 5$. The density of σ is plotted by a thick black dashed line.

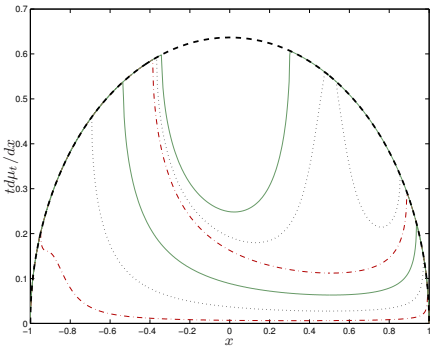


Figure 3.5: Computed solution of the CWEP from Example 4.

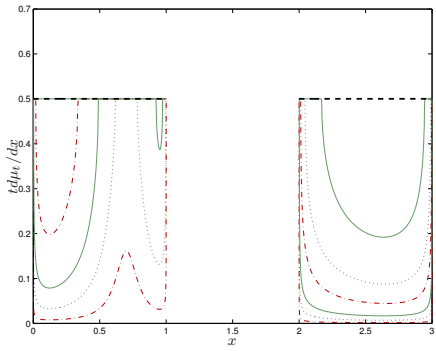


Figure 3.6: Computed solution of the CWEP from Example 5.

Example 6. Finally, consider the case in which $\nu := \delta_\alpha$, and the constraint σ is the balayage-measure of the measure δ_β onto $E = [-1, 1]$, where $\beta = -0.6{+}0.1\mathbf{i}$; i.e.,

$$d\sigma(x) = \frac{1}{\pi\sqrt{1-x^2}}\Re\left\{\frac{\sqrt{1-1/\beta^2}}{1-x/\beta}\right\}dx.$$

For the discretization of E we use the sequence of poles $\overline{B}_{800} = \{\beta_k\}_{k=1}^{800}$, with

$$\beta_k = \begin{cases} \alpha, & k \leqslant 200 \\ \infty, & k > 200. \end{cases}$$

The computed solution to the CWEP is then plotted in semi-logarithmic scale on the Figures 3.7, 3.8 and 3.9 for different values of α, and with $t = 0.05{+}0.15r$, where $r = 0, \ldots, 5$. The density of σ is plotted by a thick black dashed line.

3.1.3 Time complexity

Let N denote the number of discretization points. Creating the potential matrix $\mathbf{P}$ then takes $\mathcal{O}(N^2)$ operations, while solving a system with it takes $\mathcal{O}(N^3)$ operations when using a direct method. Further, for the special case in which the measure ν is given by (3.12), it takes $\mathcal{O}(N)$ operations to compute the discretization points based on this measure ν and to compute the potential of ν by means of (3.13), assuming that the number of different zeros of $p_m(x)$ is negligibly small compared to N. Since on each step of Algorithm 2 at least one point is moved from the set I to the set J, the upper bound for the

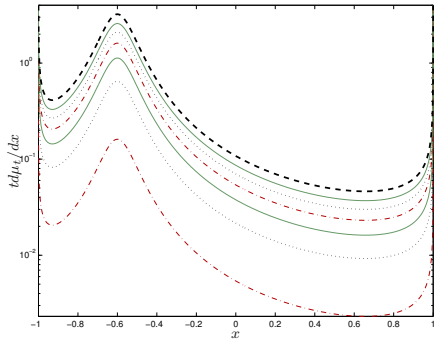


Figure 3.7: Computed solution of the CWEP from Example 6 with $\alpha = 0.1\mathbf{i} - 0.6 = \beta$.

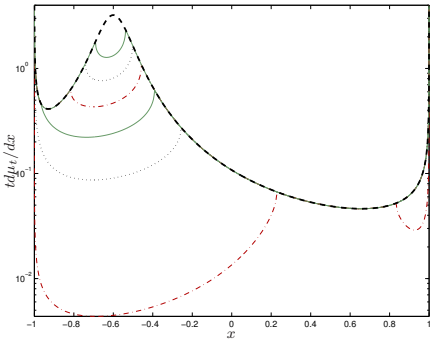


Figure 3.8: Computed solution of the CWEP from Example 6 with $\alpha = 0.5 - 0.1\mathbf{i} \neq \beta$.

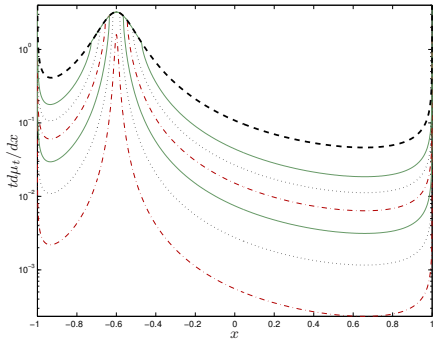


Figure 3.9: Computed solution of the CWEP from Example 6 with $\alpha = -0.6 + 0.01\mathbf{i} \neq \beta$.

number of iterations is N. In practice, however, this upper bound seems to be a serious overestimation; e.g., for each value of t in Example 4, the algorithm converged after about 10 iteration steps. Thus, the total computational cost is of order $\mathcal{O}(N^3)$.

We implemented the algorithm in MATLAB. For $t = 0.05$ in Example 4, it takes about 1.2 s to compute the solution of the CWEP on a PC with 2.93 GHz Intel Core 2 processor and 2 Gb of memory, running Debian Lenny with 2.6.26 kernel. The larger t, the faster the algorithm becomes; e.g., for $t = 0.8$ it only takes about 0.2 s to complete the computations.

3.2 Convergence of rational Ritz values

In this section we will formulate first the classical Lanczos algorithm for eigenvalue problems. Then we will describe in an asymptotic sense the regions of converged eigenvalues. This is a well-known result, and the goal is achieved by exploiting connections between the Lanczos method, a polynomial minimization problem and logarithmic potential theory. We follow the pathway presented in the work [100] by Kuijlaars.

Later in Subsection 3.2.2 we will study a more complex case – a rational Lanczos algorithm. Like in the classical case, we show that it is again possible to predict regions with converged eigenvalues by solving a certain constrained weighted energy problem. Finally, we will solve that energy problem by means of the numerical algorithm from Subsection 3.1.2, and the results exhibit a very good correlation between the prediction and actual convergence. This part is based on our paper [32] (joint work with Deckers and Van Barel). For a complete theoretical investigation we refer to the paper [10] by Beckermann, Güttel and Vandebril. In what follows we assume exact arithmetic and do not take into account the effects of rounding errors on Lanczos iterations.

3.2.1 Classical Lanczos algorithm and potential theory

Formulation of the algorithm: matrix language

The Lanczos iteration is a popular method to approximate part of the eigenvalues of large Hermitian matrices. So, for a given Hermitian matrix $\mathbf{A}$ of size $N \times N$, the Lanczos method starts from a nonzero vector $b \in \mathbb{C}^N$ and generates two sequences of numbers (α_k) and (β_k) as follows. Put $\beta_0 = 0$,

$v_0 = 0$, $v_1 = b/\|b\|_2$, and for $k = 1, 2, \ldots,$

$$\alpha_k = \langle v_k, \mathbf{A}v_k\rangle, \quad \beta_k v_{k+1} = \mathbf{A}v_k - \alpha_k v_k - \beta_{k-1}v_{k-1},$$

where β_k is taken such that $\|v_{k+1}\|_2 = 1$. The vectors $v_1, v_2, \ldots, v_n$ form an orthonormal basis of the so-called *n-th Krylov subspace* $K_n(\mathbf{A}, b) = \operatorname{span}\{b, \mathbf{A}b, \ldots, \mathbf{A}^{n-1}b\}$. The coefficients α_k and β_k are collected in the tridiagonal matrices

$$\mathbf{T}_n = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{pmatrix}$$

for $n \leqslant N$. The eigenvalues of $\mathbf{T}_n$ are called *Ritz values*, and, compared to the eigenvalues of $\mathbf{A}$, they are easier to compute because of the tridiagonal nature of $\mathbf{T}_n$ and because n is smaller than N. The key to the popularity of the method lies in the fact that some of the Ritz values turn out to be accurate approximations of some of the eigenvalues of $\mathbf{A}$ even when n is much smaller than N. The Lanczos method is discussed in many books, e.g., [71, 49, 122, 136]. In what follows we assume that no early breakdown (see [71, Theorem 9.1-1]) happens during the Lanczos iterations.

From matrix language to polynomials

Here we will show how the Lanczos iteration could be reformulated in polynomial language. Basically, it is equivalent to the following polynomial minimization problem. Let $p_n(\lambda) = \det(\lambda I - \mathbf{T}_n)$ be the characteristic polynomial of $\mathbf{T}_n$. Then p_n is a monic polynomial of degree n that minimizes $\|p_n(\mathbf{A})b\|_2$ among all monic polynomials of degree n. The zeros of p_n are equal to the Ritz values. The norm is equal to

$$\|p_n(\mathbf{A})b\|_2 = \left(\sum_{k=1}^{N} \langle b, v_k\rangle^2 p_n(\lambda_k)^2\right)^{1/2}, \tag{3.16}$$

where $\lambda_1, \ldots, \lambda_N$ are the eigenvalues of $\mathbf{A}$ and $v_1, \ldots, v_N$ are their corresponding orthonormal eigenvectors. Thus p_n is orthogonal with respect to a discrete measure

$$\sum_{k=1}^{N} \langle b, v_k\rangle^2 \delta_{\lambda_k},$$

which has the mass $\langle b, v_k\rangle^2$ at the eigenvalue λ_k. Here we denote by δ_{λ_k} the unit measure whose support is the point λ_k.

Further, consider the situation where both N and n tend to infinity. We assume that we have a sequence of matrices $(\mathbf{A}_N)$ with $\mathbf{A}_N$ being a Hermitian matrix of size $N \times N$. The distinct eigenvalues of $\mathbf{A}_N$ are denoted by

$$\lambda_{1,N} < \lambda_{2,N} < \cdots < \lambda_{N,N},$$

and for ease of notation here we assume that they are all distinct. This is not an essential restriction, and the results below will remain valid, as shown in [100].

We also assume that the eigenvalues $\lambda_{k,N}$ are all contained in a fixed bounded interval E and that

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \delta_{\lambda_{k,N}} = \sigma, \tag{3.17}$$

with σ being a Borel probability measure on $\mathbb{R}$ with compact support, $\operatorname{supp}\sigma \subseteq E$. The convergence is in the sense of weak convergence of measure. If the relation (3.17) holds, we say that σ is the *asymptotic distribution of eigenvalues* of a matrix sequence $(\mathbf{A}_N)$.

In many practical situations, matrices $\mathbf{A}_N$ appear as discretizations of a continuous operator. The size N is related to the mesh size of the discretization. A relation like (3.17) may then hold very well, and the measure σ would be determined by the spectral properties of the continuous operator; see, e.g., [11].

In the definition of Lanczos iterations we had also a starting vector b. Here for each N we denote a starting vector by $b_N \in \mathbb{R}^N$. Let us take this vector normalized so that $\|b_N\|_2 = 1$. Thus

$$\sum_{k=1}^{N} \langle b_N, v_{k,N}\rangle^2 = 1,$$

where $(v_{k,N})_{k=1}^N$ is an orthonormal basis of eigenvectors of $\mathbf{A}_N$.

It is known for the Lanczos method (see e.g. [122]) that in exact arithmetics it is essential that the starting vector has a component in the direction of each of the eigenvectors of $\mathbf{A}_N$, and usually a random vector fulfills this condition. We can reformulate this in an asymptotic sense as "vectors b_N are chosen sufficiently random", meaning that none of their Fourier coefficients in the basis $(v_{k,N})$ is exponentially small as $N \to \infty$. That is,

$$\lim_{N\to\infty} \left(\min_{1\leqslant k\leqslant N} |\langle b_N, v_{k,N}\rangle| \right)^{1/N} = 1. \tag{3.18}$$

Further, a technical condition is needed on the spacings of the eigenvalues, which prevents them from being too close. The extensive discussion leading to this condition is presented in [101, Section 4]. Following this discussion, we assume that whenever, for each N, an index $k = k_N \in \{1, \dots, N\}$ is chosen such that

$$\lim_{N\to\infty} \lambda_{k,N} = \lambda \in \mathbb{R},$$

then

$$\lim_{N\to\infty} \frac{1}{N} \sum_{j=1, j\neq k}^{N} \log|\lambda_{k,N} - \lambda_{j,N}| = \int \log|\lambda - \lambda'| d\sigma(\lambda'). \tag{3.19}$$

In particular, as shown by Dragnev and Saff [50], (3.19) forbids $\mathbf{A}_N$ from having two exponentially close eigenvalues, i.e., $|\lambda_{k+1,N} - \lambda_{k,N}| \leqslant e^{-cN}$ for some $c > 0$.

From polynomials to potential theory

Let us recall the definition (3.1) of a logarithmic potential. It is clear that the right-hand side of (3.19) is equal to $-U^\sigma(\lambda)$. It can be also shown that $U^\sigma(\lambda)$ is a continuous function of $\lambda \in \mathbb{C}$.

Expression (3.17) gave a definition of an asymptotic distribution of the eigenvalues. Now we formulate the result that describes the asymptotic distribution of zeros of a Lanczos polynomial in terms of a logarithmic potential.

For $0 \leqslant n \leqslant N$, we denote by $p_{n,N}$ the n-th degree monic Lanczos polynomial associated with $\mathbf{A}_N$. The zeros of $p_{n,N}$ are real and simple and we denote them by

$$\theta_{1,n,N} < \theta_{2,n,N} < \cdots < \theta_{n,n,N}.$$

The following result was first proven by Rakhmanov[125] and later generalized by Dragnev and Saff [50, Theorem 3.3].

Theorem 7. *Assume (3.17), (3.18) and (3.19). Let $n, N \to \infty$ in such a way that $n/N \to t \in (0,1)$. Then there is a Borel probability measure μ_t , depending only on t and σ, such that*

$$\lim_{N\to\infty} \frac{1}{n} \sum_{j=1}^{n} \delta_{\theta_{j,n,N}} = \mu_t, \tag{3.20}$$

and a real constant F_t such that

$$\lim_{N\to\infty} \|p_{n,N}(\mathbf{A}_N) b_N\|_2^{1/n} = \exp(-F_t). \tag{3.21}$$

The measure μ_t satisfies

$$0 \leqslant t\mu_t \leqslant \sigma, \quad \int d\mu_t = 1 \tag{3.22}$$

and minimizes the logarithmic energy (3.2) $I(\mu)$ among all measures μ satisfying $0 \leqslant t\mu \leqslant \sigma$, $\int d\mu = 1$. The logarithmic potential U^{μ_t} of μ_t is a continuous function on $\mathbb{C}$, and the constant F_t is such that

$$U^{\mu_t}(\lambda) = F_t \quad \text{for } \lambda \in \operatorname{supp}(\sigma - t\mu_t), \tag{3.23}$$

$$U^{\mu_t}(\lambda) \leqslant F_t \quad \text{for } \lambda \in \mathbb{C}. \tag{3.24}$$

The relations (3.20)–(3.24) characterize the pair (μ_t, F_t).

This result enabled Kuijlaars [100] to prove his famous theorem:

Theorem 8. *Assume (3.17), (3.18) and (3.19). Let $n, N \to \infty$ in such a way that $n/N \to t \in (0,1)$. Then the Ritz values generated by the Lanczos iteration with starting vector b_N are asymptotically distributed according to the measure μ_t, which is the solution of the extremal energy problem in Theorem 7.*

Kuijlaars [100] and later Beckermann [9] also presented estimates for the convergence rates of Ritz values in terms of the potential U^{μ_t}. For exact formulas we refer here to their works.

Practical characterization of regions of convergence

We will show now how to characterize the regions with converged Ritz values in terms of the solution of the extremal energy problem of Theorem 7.

Kuijlaars [101] has proven the following lemma:

Lemma 6. *Assume (3.17), (3.20) and (3.23). Then for any interval (a,b) one has*

$$\lim_{N\to\infty} \frac{\#\{j : \theta_{j,n,N} \in (a,b)\} - \#\{j : \lambda_{j,N} \in (a,b)\}}{N} = 0$$

if and only if

$$(a,b) \cap \operatorname{supp}(t - \mu_t) = \emptyset.$$

From this lemma follows that one can expect convergence of Ritz values only outside the support of $\sigma - t\mu_t$. However, it is possible to restrict this set even more. Consider $\Lambda(t;\sigma)$ defined in terms of μ_t and F_t as

$$\Lambda(t;\sigma) = \{\lambda \in \mathbb{R} : U^{\mu_t}(\lambda) < F_t\}. \tag{3.25}$$

It is clear from (3.23) that $\Lambda(t;\sigma) \subset \mathbb{R}\setminus \operatorname{supp}(\sigma - t\mu_t)$, but equality need not hold in general. The sets (3.25) form the *saturated region*. It is the region where the n-th Ritz values converged to an eigenvalue of $\mathbf{A}_N$.

3.2.2 Convergence analysis of the rational Lanczos iterations

In [129], Ruhe presented a rational Krylov method as an extension of the shift-and-invert Arnoldi process allowing for varying shifts. We will consider here a variant of this method for Hermitian matrices and we will formulate it using the orthogonal functions approach.

Let us consider a compact set $E \subset \mathbb{C}$. Then consider a given sequence of fixed complex poles $\mathcal{A}_N = \{\alpha_1, \ldots, \alpha_{N-1}, \alpha_N\} \subset \overline{\mathbb{C}}$ bounded away from the convex hull of E, which we will denote in the remainder by $c(E)$, and suppose $\alpha_\emptyset \in c(E)$. We then define the factors

$$Z_k(x) = \frac{x - \alpha_\emptyset}{\left(1 - \frac{x-\alpha_\emptyset}{\alpha_k - \alpha_\emptyset}\right)}, \quad k = 1, \ldots, N,$$

and products

$$b_0(x) \equiv 1, \quad b_k(x) = Z_k(x) b_{k-1}(x), \quad k = 1, \ldots, N.$$

Or, equivalently, with $\pi_k(x)$ given by

$$\pi_0(x) \equiv 1, \qquad \pi_k(x) = \left(1 - \frac{x - \alpha_\emptyset}{\alpha_k - \alpha_\emptyset}\right) \pi_{k-1}(x), \quad k = 1, \ldots, N,$$

we have that

$$b_k(x) = \frac{(x - \alpha_\emptyset)^k}{\pi_k(x)}, \ k = 1, \ldots, N.$$

Next, suppose $\mathbf{A}_N$ is a Hermitian $N \times N$ matrix with eigenvalues $\{\lambda_{1,N}, \ldots, \lambda_{N,N}\} \subset E$ and eigenvectors $\mathbf{u}_{1,N}, \ldots, \mathbf{u}_{N,N}$, and let there be given a nonzero column vector $\mathbf{q}_N \in \mathbb{C}^N$. We then consider the nested sequence of *rational Krylov subspaces*

$$K_{n+1}(\mathbf{A}_N, \mathbf{q}_N, \mathcal{A}_N) = \operatorname{span}\left\{\mathbf{q}_N, b_1(\mathbf{A}_N)\mathbf{q}_N, \ldots, b_n(\mathbf{A}_N)\mathbf{q}_N\right\}, \ n = 0, \ldots, N.$$

For $n < N$ the *rational Lanczos iterations* produce a sequence of orthonormal vectors $\mathbf{v}_k$, $k = 1, \ldots, n+1$, for $K_{n+1}(\mathbf{A}_N, \mathbf{q}_N, \mathcal{A}_N)$ in the following way. (Here we suppose that no early breakdown happens.) Put $\mathbf{v}_1 = \mathbf{q}_N / \|\mathbf{q}_N\|$, then for

$k = 1, \ldots, n$, the $\mathbf{v}_{k+1}$ are defined by orthogonalization of $Z_k(\mathbf{A}_N)\mathbf{v}_k$ against $\mathbf{v}_1, \ldots, \mathbf{v}_k$, followed by normalization[1]:

$$Z_k(\mathbf{A}_N)\mathbf{v}_k = \sum_{j=1}^{k+1} h_{j,k}\mathbf{v}_j, \quad k = 1, \ldots, n. \tag{3.26}$$

Let $\mathbf{V}_n = (\mathbf{v}_k) \in \mathbb{C}^{N\times n}$, $\mathbf{H}_n^{[\alpha_n]} = (h_{j,k}) \in \mathbb{C}^{n\times n}$ upper Hessenberg, $\mathbf{D}_n^{[\alpha_n]} = \operatorname{diag}((\alpha_1 - \alpha_\emptyset)^{-1}, \ldots, (\alpha_n - \alpha_\emptyset)^{-1})$, and define $\mathbf{A}_n$ and $\mathbf{B}_n^{[\alpha_n]}$ by

$$\begin{aligned}\mathbf{A}_n &:= \mathbf{V}_n^H \mathbf{A}_N \mathbf{V}_n \\ \mathbf{B}_n^{[\alpha_n]} &:= \mathbf{H}_n^{[\alpha_n]} \left(\mathbf{I}_n + \mathbf{H}_n^{[\alpha_n]}\mathbf{D}_n^{[\alpha_n]}\right)^{-1} + \alpha_\emptyset \mathbf{I}_n,\end{aligned}$$

where the superscript ^{H} denotes the Hermitian transpose, and the superscript $^{[\alpha_n]}$ means that we consider the last pole α_n variable, whereas the other poles are assumed to be fixed. In matrix notation, (3.26) then becomes:

$$\mathbf{A}_n = \mathbf{B}_n^{[\alpha_n]} - \frac{h_{n+1,n}}{\alpha_n - \alpha_\emptyset}\mathbf{V}_n^H \mathbf{A}_N \mathbf{v}_{n+1}[0 \ \ldots \ 0 \ 1] \left(\mathbf{I}_n + \mathbf{H}_n^{[\alpha_n]}\mathbf{D}_n^{[\alpha_n]}\right)^{-1}. \tag{3.27}$$

Note that the left-hand side of (3.27) is independent of α_n. So, taking $\alpha_n = \infty$ in the right-hand side of (3.27), we find that $\mathbf{A}_n = \mathbf{B}_n^{[\infty]}$.

By definition, $\mathbf{v}_k = \varphi_{k-1}(\mathbf{A}_N)\mathbf{v}_1$, with $\varphi_0(x) = 1$ and $\varphi_k(x) = \frac{p_k(x)}{\pi_k(x)}$. Since

$$\mathbf{v}_j^H \mathbf{v}_k = \delta_{j,k} = \langle \varphi_k \, , \varphi_j \rangle \, ,$$

where the inner product $\langle \cdot \, , \cdot \rangle$ is defined by

$$\langle f \, , g \rangle = (g(\mathbf{A}_N)\mathbf{v}_1)^H \, f(\mathbf{A}_N)\mathbf{v}_1,$$

it follows that the φ_k are orthonormal rational functions (ORFs) with poles in $\mathcal{A}_k$. In [47] it was proved for the special case of all real poles that the eigenvalues of $\mathbf{B}_n^{[\alpha_n]}$ are the zeros of the ORF φ_n, and hence, are all real and in $c(E)$. The restriction to all real poles is in fact not necessary, but then the eigenvalues of $\mathbf{B}_n^{[\alpha_n]}$ are all real and in $c(E)$ iff α_n is real or infinite. So, the Ritz values (i.e., the eigenvalues of $\mathbf{A}_n$) are zeros of an ORF too. More specific, they are zeros of the ORF $\tilde{\varphi}_n$ with poles in $\alpha_1, \ldots, \alpha_{n-1}, \infty$, and they are all real and in $c(E)$. Further, the orthonormality for $\tilde{\varphi}_n(x) = \frac{\tilde{p}_n(x)}{\pi_{n-1}(x)}$ reads:

$$\left\langle \tilde{\varphi}_n \, , \frac{p}{\pi_{n-1}} \right\rangle = \sum_{j=1}^{N} \tilde{p}_n(\lambda_{j,N})\overline{p(\lambda_{j,N})} \left| \frac{\mathbf{q}_N^H \mathbf{u}_{j,N}}{\pi_{n-1}(\lambda_{j,N})} \right|^2 = \begin{cases} 0 & \text{if } \deg(p) < n \\ 1 & \text{if } p = \tilde{p}_n \, . \end{cases}$$

[1] When including the value $n = N$, $\mathbf{v}_{N+1}$ theoretically should be the zero-vector in $\mathbb{C}^N$.

Thus, the rational Lanczos minimization problem is to minimize $\left\| \frac{p(\mathbf{A}_N)}{\pi_{n-1}(\mathbf{A}_N)} \mathbf{q}_N \right\|$ among all monic polynomials $p(x)$ of degree n. Similarly as has been done in [100, 101, 80], we can now characterize the region of the Ritz values that converged to an eigenvalue of $\mathbf{A}_N$ (depending on the number of iterations) by means of a CWEP from potential theory. For this, we consider the situation where both N and n tend to infinity in such a way that $n/N \to t \in (0,1)$. So, let us now make the following assumptions (more details and proofs are given in [10]):

1. We have a sequence of Hermitian matrices $(\mathbf{A}_N) \in \mathbb{C}^{N\times N}$, with N distinct eigenvalues $\{\lambda_{i,N}\}_{i=1}^N \subset E$. The asymptotic distribution of the eigenvalues is given by

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \delta_{\lambda_{k,N}} = \sigma \in \mathcal{M}(E),$$

where convergence is in the weak sense; i.e., for any continuous function f with compact support, $\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} f(\lambda_{k,N}) = \int f d\sigma$. Further, the eigenvalues are sufficiently separated; i.e., whenever an index $k = k_N \in \{1, \ldots, N\}$ is chosen for every N so that

$$\lim_{N\to\infty} \lambda_{k,N} = \lambda \in \mathbb{R},$$

then

$$\lim_{N\to\infty} \frac{1}{N} \sum_{j=1, j\neq k}^{N} \log|\lambda_{k,N} - \lambda_{j,N}| = \int \log|\lambda - \lambda'| d\sigma(\lambda').$$

2. The asymptotic distribution of the poles is given by

$$\lim_{N\to\infty} \frac{1}{tN} \sum_{k=1}^{tN} \delta_{\alpha_k} = \nu_t + (1-s)\delta_\infty = \eta_t,\ \eta_t(\overline{\mathbb{C}}) = 1,$$

where convergence again is in the weak sense, and the support of η_t is bounded away from $c(E)$.

3. For every N we have a starting vector $\mathbf{q}_N \in \mathbb{C}^N$, which is normalized ($\|\mathbf{q}_N\| = 1$) and chosen sufficiently random so that

$$\lim_{N\to\infty} \left(\min_{1\leqslant k\leqslant N} |\mathbf{q}_N^H \mathbf{u}_{k,N}|\right)^{1/N} = 1.$$

Under the previous assumptions we have for the n-th Ritz values

$$\theta_{1,n} < \theta_{2,n} < \ldots < \theta_{n,n}$$

that there is a Borel probability measure μ_t so that

$$\lim_{N\to\infty} \frac{1}{n} \sum_{k=1}^{n} \delta_{\theta_{k,n}} = \mu_t \in \mathcal{M}(E).$$

The measure μ_t satisfies $t\mu_t \leqslant \sigma$ and minimizes the weighted logarithmic energy $I(\mu - \nu_t)$ among all probability measures $\mu \in \mathcal{M}(E)$ satisfying $t\mu \leqslant \sigma$.

Finally, let us denote the *free region* by S_t (i.e., the set where the upper constraint is not active), given by

$$S_t = \operatorname{supp}(\sigma - t\mu_t). \tag{3.28}$$

Then the complement of S_t, which is called the *saturated region* (where the two measures σ and $t\mu_t$ agree), is the region where the n-th Ritz values converged to an eigenvalue of $\mathbf{A}_N$ with a rate of convergence described by the weighted potential $U^{\mu_t - \nu_t}$.

3.2.3 Numerical examples

In the numerical experiments that follow, the rational Lanczos method with full re-orthogonalization and $\alpha_\emptyset = 0 \in E$ is applied to different diagonal matrices $\mathbf{A} \in \mathbb{R}^{200\times 200}$ with starting vector $\mathbf{q} = [1\ 1\ \ldots\ 1]^T$. Note that this starting vector has the same component in each of the eigenvalue directions. For a given sequence of poles we then computed the n-th Ritz values for $n = 1, 2, \ldots, 200$, and indicated in the figures that follow the converged Ritz values. For this, we consider a Ritz value to be converged if in the next iteration there is a Ritz value within some prescribed distance. Although this is not a truly safe convergence check, it works well in our examples. In the figures, the markers from Table 1 are used to display the smallest distance between Ritz values from successive iterations. To make the pictures more readable, we only plot the odd iterations.

Marker	Color	Distance to nearest Ritz value from the next iteration
$+$	Red	less than 0.5×10^{-14}
$*$	Yellow	between 0.5×10^{-14} and 0.5×10^{-8}
$\cdot$	Blue	between 0.5×10^{-8} and 0.5×10^{-4}
$\times$	Green	more than 0.5×10^{-4}

Table 1: Markers and colors for the figures

Like in the polynomial Lanczos case, the convergence plot basically remains the same if we increase the size N of the matrix. Only the horizontal axis has to be re-scaled. This means that the good region of converged eigenvalues only depends on the ratio $t = n/N$, where n is the number of Lanczos iterations.

For fixed values of $t \in (0, 1)$ for which $200t$ is a natural number, we assume the asymptotic distribution of the poles is given by

$$\eta_t = \frac{1}{200t} \sum_{j=1}^{200t-1} \delta_{\alpha_j} + \frac{1}{200t}\delta_\infty = \nu_t + (1 - s_t)\delta_\infty,$$

such that the logarithmic potential for ν_t is given by

$$U^{\nu_t}(x) = -\frac{1}{200t} \sum_{j=1}^{200t-1} \log|\alpha_j - x| =: \kappa_{200t-1,200t}(x). \tag{3.29}$$

Note that this is corresponding to a sequence of $n - 1$ finite poles and one pole at infinity. From the previous subsection it then follows that the boundary between the set where the eigenvalues are found and the set where the eigenvalues are not found yet is the boundary between the free region, given by (3.28), and the saturated region. In the figures that follow, this boundary is computed by means of Algorithm 2 for several values of t, and plotted in function of $n = 200t$ by means of a black line.

In some of the examples that follow we have used complex poles alongside with real ones. While complex poles do not complicate matters significantly (the only price to pay is a conversion from real-valued arithmetics to complex-valued, which is a constant factor), they illustrate our software better. Moreover, the use of complex poles allows to find some internal eigenvalues first, which is not feasible with only real poles, as we will show now. On one hand, one prefers to use zeros of ORF's as discretization points for an interval, and on the other hand, the discretization set has to be asymptotically dense in this interval, as required by the convergence conditions (see Theorem 1). These two requirements start to contradict when we choose one of the poles of ORF's to be inside the interval; adding some imaginary offset is a good solution in this case.

Equally spaced eigenvalues

Suppose the eigenvalues of $\mathbf{A}$ are equally distributed on $E = [-1, 1]$, e.g.,

$$\lambda_{k,200} = -1 + \frac{2(k-1)}{199}, \qquad k = 1, \ldots, 200.$$

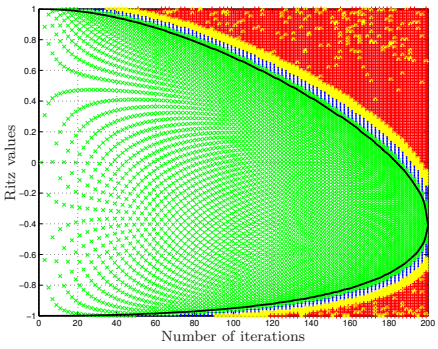


Figure 3.10: Convergence of the Ritz values for Ex. 7 with $\alpha = 2$.

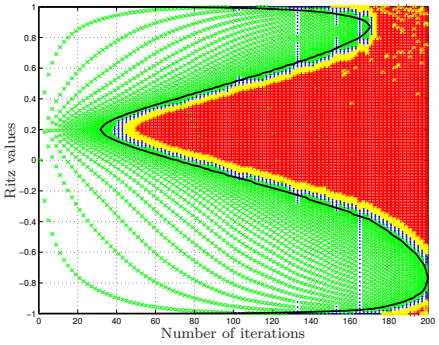


Figure 3.11: Convergence of the Ritz values for Ex. 7 with $\alpha = 0.2 + 0.1\mathbf{i}$.

The constraint for the CWEP is then given by the Lebesgue measure $d\sigma(x) = \frac{1}{2}dx$ on $[-1, 1]$.

Example 7. First, consider the case of one multiple pole at α. The predicted as well as the actual zones of convergence are then plotted on Figures 3.10 and 3.11 for $\alpha = 2$ and $\alpha = 0.2 + 0.1\mathbf{i}$ respectively. These figures clearly show that the pole attracts Ritz values (those closer to the pole, tend to converge first), and that choosing a complex pole close to the interval makes it possible to find inner eigenvalues first.

Example 8. Next, consider the case of two different poles α_1 and α_2, each with multiplicity 100. We then distinguish two cases:

(a) the case in which the poles are ordered as $\{\alpha_1, \ldots, \alpha_1, \alpha_2, \ldots, \alpha_2\}$,

(b) the case in which the poles are ordered as $\{\alpha_1, \alpha_2, \alpha_1, \alpha_2, \ldots\}$.

Figures 3.12 and 3.14 show the results for case (a) with $\alpha_1 = -5$ and $\alpha_2 = 1.2$, and $\alpha_1 = 0.2 + 0.1\mathbf{i}$ and $\alpha_2 = -0.5 + 0.1\mathbf{i}$ respectively. The results for case (b) are plotted on Figures 3.13 and 3.15 respectively.

On the basis of (3.29) it is easy to explain the differences between the figures for case (b) and those for case (a). On the latter there is no effect of the pole α_2 during the first 100 iterations, while for case (b) the figures are more balanced. Further, it clearly follows from Figure 3.13 that the pole closer to the interval has more effect on the convergence behavior of the Ritz values.

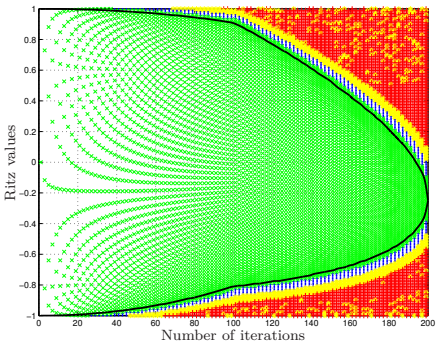


Figure 3.12: Convergence of the Ritz values for Example 8 (a) with $\alpha_1 = -5$ and $\alpha_2 = 1.2$.

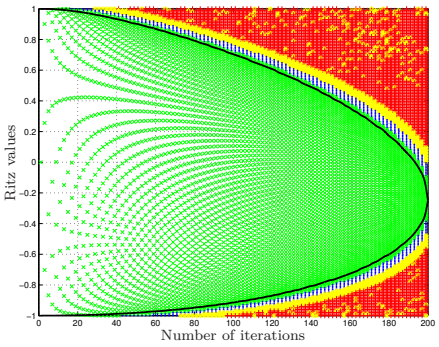


Figure 3.13: Convergence of the Ritz values for Example 8 (b) with $\alpha_1 = -5$ and $\alpha_2 = 1.2$.

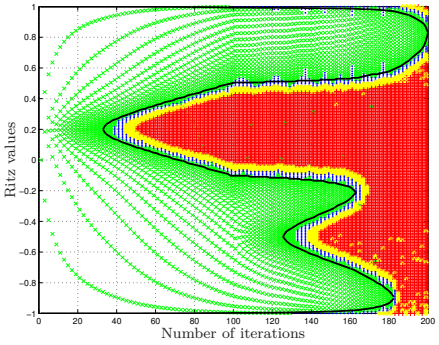


Figure 3.14: Convergence of the Ritz values for Example 8 (a) with $\alpha_1 = 0.2 + 0.1\mathbf{i}$ and $\alpha_2 = -0.5 + 0.1\mathbf{i}$.

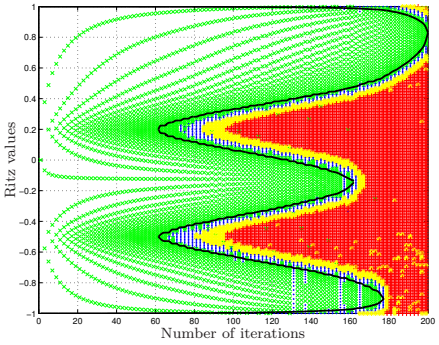


Figure 3.15: Convergence of the Ritz values for Example 8 (b) with $\alpha_1 = 0.2 + 0.1\mathbf{i}$ and $\alpha_2 = -0.5 + 0.1\mathbf{i}$.

Example 9. Finally, consider the case in which the eigenvalues of $\mathbf{A}$ are equally distributed on $E = [0, 1] \cup [2, 3]$. Figure 3.16 then shows the predicted as well as the actual zones of convergence for the case of one multiple pole $\alpha = 0.7 + 0.1\mathbf{i}$.

Eigenvalues distributed according to the balayage-measure

As has been proved in [101], an eigenvalue distribution according to the equilibrium measure is the worst case for the convergence of the Lanczos iteration. No eigenvalues are well-approximated whenever $n < N$. However,

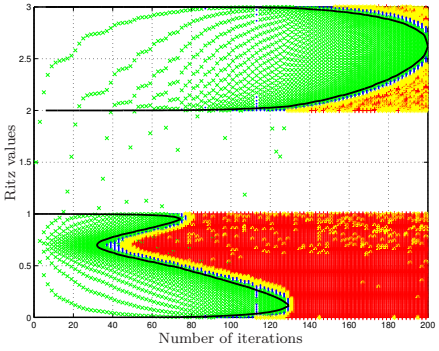


Figure 3.16: Convergence of the Ritz values for Example 9.

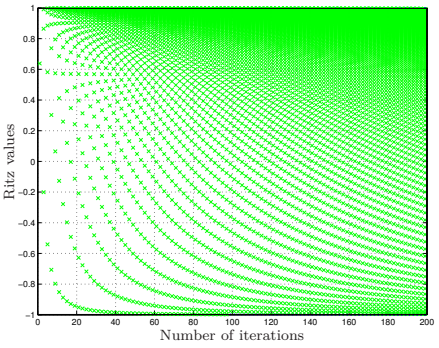


Figure 3.17: Convergence of the Ritz values for Example 10.

keeping the same eigenvalue distribution but using a rational Lanczos method instead, it is possible to find eigenvalues for $n < N$.

In the rational case, the same occurs whenever the eigenvalues are distributed according to the balayage-measure of a probability measure η from $\overline{\mathbb{C}}\backslash E$ onto E, while the asymptotic distribution of the poles is given by a probability measure η_t so that

$$\eta - t\eta_t > 0 \quad \text{on} \quad \operatorname{supp}(\eta - t\eta_t) \neq \emptyset, \quad \text{for every } t \in (0,1).$$

Example 10. Consider the case in which the diagonal matrix $\mathbf{A} \in \mathbb{R}^{200\times 200}$ has eigenvalues equal to the rational Chebyshev points on $E = [-1,1]$ (cf. Subsection 3.1.2), based on the sequence of poles

$$\overline{B}_{200} = \{\beta_1 = \ldots = \beta_{199} = 1.1, \beta_{200} = \infty\}.$$

The asymptotic distribution of the eigenvalues is then given by:

$$d\sigma(x) = \frac{1}{\pi\sqrt{1-x^2}}\frac{1}{200}\left(199\frac{\sqrt{1-1/1.1^2}}{1-x/1.1}+1\right)dx.$$

Applying n steps of the rational Lanczos algorithm with one multiple pole at $\alpha = 1.1$ yields no converged Ritz values for any $n < 200$, as one can clearly see on Figure 3.17. This corresponds to the solution of the CWEP: no saturation is present for any $t = n/200 < 1$. Indeed, for μ_t given by:

$$d\mu_t(x) = \frac{1}{\pi\sqrt{1-x^2}}\frac{1}{200t}\left((200t-1)\frac{\sqrt{1-1/1.1^2}}{1-x/1.1}+1\right)dx,$$

we have that

$$\frac{d(\sigma - t\mu_t)(x)}{dx} = \frac{(1-t)}{\pi\sqrt{1-x^2}} \frac{\sqrt{1-1/1.1^2}}{1-x/1.1} > 0,$$

for every $x \in [-1, 1]$ and every $t \in (0, 1)$. Applying a rational Lanczos method with poles different from 1.1 does make it possible to find eigenvalues for $n < 200$ (see also Figures 3.7–3.9).

3.3 Conclusion

Together with some auxiliary lemmas, we presented an algorithm to numerically solve the constrained weighted energy problem (CWEP), which appears in logarithmic potential theory. Our algorithm is based on an equivalent formulation of the CWEP in terms of a weighted logarithmic potential. First, we formulated the continuous version of the algorithm, and then we discretized it. Compared with the continuous version, the discretized version has the advantage that the algorithm always stops, producing a solution which is accurate in comparison to the exact solution when known. Finally, we used the algorithm to predict the region of convergence of Ritz values obtained by applying the rational Lanczos method for symmetric eigenvalue problems. In all cases our algorithm estimated the region of convergence of Ritz values in an accurate way.

Chapter 4

Multivariate orthogonal polynomials

This chapter is organized as follows. In Section 4.1 we state the problem of polynomial least squares approximation and show how multivariate orthogonal polynomials appear naturally in its context. In Section 4.2 we show how generalized Hessenberg matrices arise in recurrence relations between these orthogonal polynomials and how the original problem of constructing these polynomials is reduced to an inverse eigenvalue problem. In Section 4.3 we present an algorithm to solve this inverse eigenvalue problem and in Section 4.4 we give several numerical examples. Within this chapter we follow our paper [161].

4.1 Polynomial least squares approximation

Consider some discrete inner product $\langle \cdot, \cdot \rangle$, and let us measure distances by its associated norm. In this section we reduce a discrete least squares approximation problem to computing the corresponding orthogonal basis polynomials. The starting problem is: given some function f, find a polynomial $p \in \mathcal{P}$ that minimizes $\|f - p\|_{\langle \cdot, \cdot \rangle}$. Such a polynomial can be found as follows. Construct a basis $\{\phi_1, \ldots, \phi_n\}$ for $\mathcal{P}$ which is orthonormal with respect to the inner product $\langle \cdot, \cdot \rangle$. The solution p is then the Fourier expansion of f with

respect to this basis, truncated after some term. An algorithm that solves this problem will explicitly compute the recurrence coefficients for an orthonormal basis and the Fourier coefficients.

4.1.1 Definitions

Let $\{\zeta_k\}_{k=1}^m$ be a set of nodes – pairs of complex numbers and $\{w_k^2\}_{k=0}^m$ a set of positive weights (let us assume that $w_k > 0$). Expression (4.1) represents then a positive semidefinite inner product in some bivariate polynomial space $\mathcal{P}$:

$$\langle p, q\rangle = \sum_{i=0}^{m} w_i^2 \overline{p(\zeta_i)} q(\zeta_i). \tag{4.1}$$

This is a positive definite inner product for the space of vectors $(p(\zeta_0), \ldots, p(\zeta_m))$ representing the function values at the given nodes.

Let us further specify the polynomial spaces we are working in. Let $R[\mathbf{x}]$ with $\mathbf{x} = (x_1, x_2)$ be the ring of all polynomials in two variables. Fix a monomial order $\prec$, say the graded lexicographical order, and let $\mathrm{lt}(f)$ denote the leading term of the polynomial $f \in R[\mathbf{x}]$ according to the monomial order. This monomial order induces the order $\prec$ on the polynomials, namely, we say that $p(x, y) \prec q(x, y)$ iff $\mathrm{lt}(p) \prec \mathrm{lt}(q)$. Consider any ordered sequence of terms $t_0 \prec \cdots \prec t_n$ and define by $\mathcal{P}_n$ its linear span. We say that the polynomial $p \in \mathcal{P}_n$ has length k iff all the coefficients at the terms t_i, $k < i \leqslant n$, are zero and $t_k \neq 0$. It is assumed that together with each term $x^i y^j$ all the terms $x^p y^q$, $p \leqslant i$, $q \leqslant j$ are also in $\mathcal{P}_n$ and are preceding $x^i y^j$ wrt chosen monomial order.

Although the technique presented below works for any monomial order as defined above, in the applications it is usually more practical to use some total degree monomial order (like graded lexicographical order).

Given an inner product $\langle \cdot, \cdot \rangle$ defined on $\mathcal{P}_m \times \mathcal{P}_m$ and some bivariate function f, the polynomial $p \in \mathcal{P}_n$ of length at most $n \leqslant m$, which minimizes the error

$$\|f - p\|_{\langle \cdot, \cdot \rangle}, \quad p \in \mathcal{P}_n \tag{4.2}$$

is called a *least squares approximant*. This polynomial could be represented as

$$p = \sum_{k=0}^{n} a_k \phi_k, \quad a_k = \langle f, \phi_k \rangle, \tag{4.3}$$

where the $\{\phi_k\}_0^n$ form an orthonormal set of polynomials:

$$\phi_k \in \mathcal{P}_k - \mathcal{P}_{k-1}, \quad \mathcal{P}_{-1} = \emptyset, \quad \langle \phi_k, \phi_l \rangle = \delta_{kl}.$$

The inner product we consider here is of discrete form (4.1) where ζ_i are distinct pairs of complex numbers. When $m = n$, the least squares solution is the interpolating polynomial. Because of the discrete form of the inner product, the only information needed about the function f is the set of its values in the points ζ_i. This also means that (the continuos) f does not necessarily have to belong to $\mathcal{P}_m$.

4.1.2 Reduction to the construction of an orthonormal basis

Let us illustrate now how the orthogonal polynomials do appear in this context. We start with some multivariate polynomial basis $\{\psi_k\}$, $\psi_k \in \mathcal{P}_k - \mathcal{P}_{k-1}$. Let us set

$$p = \sum_{k=0}^{n} c_k \psi_k, \quad c_k \in \mathbb{C}.$$

Then the least squares problem can be reformulated as finding the weighted least squares solution of a system of linear equations. More precisely, such solution is given by

$$\min_{c_k} \sum_{k=0}^{n} w_i^2 (c_k \psi_k(\zeta_i) - f(\zeta_i))^2, \quad i = 0, \ldots, m,$$

which is the least squares solution of the system $W\Psi_n \mathbf{c}_n = W\mathbf{f}$:

$$\min_{\mathbf{c}_n} \|W(\Psi_n \mathbf{c}_n - \mathbf{f})\|_2,$$

where $W = \operatorname{diag}(w_0, \ldots, w_m)$ and

$$\Psi_n = \begin{pmatrix} \psi_0(\zeta_0) & \ldots & \psi_n(\zeta_0) \\ \vdots & & \vdots \\ \psi_0(\zeta_m) & \ldots & \psi_n(\zeta_m) \end{pmatrix}, \quad \mathbf{c}_n = \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f(\zeta_0) \\ \vdots \\ f(\zeta_m) \end{pmatrix}. \quad (4.4)$$

The normal equations for this system are

$$(\Psi_n^H W^H W \Psi_n)\mathbf{c}_n = \Psi_n^H W^H W \mathbf{f}. \quad (4.5)$$

When the ψ_k are chosen to be the orthonormal polynomials ϕ_k, then $\Psi_n^H W^H W \Psi_n = I_{n+1}$ and the previous system has the solution $\mathbf{c}_n = \Psi_n^H W^H W \mathbf{f}$.

For more details on the relation between the least squares problem and polynomial bases one can address [16, Chapter 8].

4.2 Generalized Hessenberg matrices and recurrence relations

From the previous discussion it follows that the central problem is to construct the orthonormal basis $\{\phi_k\}$. This polynomial basis is described by means of recurrence relations between the orthonormal basis polynomials. Further in this section we show how these recurrence coefficients come from a certain inverse eigenvalue problem, at first, for the univariate case. Later this result is generalized to the multivariate case.

4.2.1 Univariate case

Let us recall the one-variable case. In general, the polynomial $z\phi_{k-1}(z)$ can be expressed as a linear combination of the polynomials $\phi_0, \ldots, \phi_k$, leading to a relation of the form

$$z\phi_{k-1}(z) = \eta_{kk}\phi_k(z) + \cdots + \eta_{0k}\phi_0(z), \quad k = 1, \ldots, m+1.$$

We can express the previous relations as

$$z[\phi_0(z), \ldots, \phi_m(z)] = [\phi_0(z), \ldots, \phi_m(z)]H + e_{m+1}^T \phi_{m+1}(z)\eta_{m+1,m+1}, \quad (4.6)$$

where H is an upper Hessenberg matrix and $e_{m+1}^T = [0, 0, \ldots, 0, 1]$.

Since we identify the function with the $(m+1)$-vector of its function values in ζ_k, $k = 0, \ldots, m$ (being just complex numbers in the one-dimensional case), our "functional space" is a space of $(m+1)$-vectors, which is $(m+1)$-dimensional. This means that the $(m+2)$-nd orthogonal polynomial will be orthogonal to the whole space, hence it must be zero. Thus, if ϕ_k are these orthogonal polynomials, then $[\phi_{m+1}(\zeta_0), \ldots, \phi_{m+1}(\zeta_m)]^T$ is the zero vector. Hence, $\phi_{m+1}(z) = \Pi_i(z - \zeta_i)$.

Let us define the matrix Φ_m similarly to Ψ_m (4.4), replacing polynomials ψ with ϕ. Let us set $\Phi = \Phi_m$ and rewrite relation (4.6) as

$$Z\Phi = \Phi H,$$

with $Z = \mathrm{diag}(\zeta_0, \ldots, \zeta_m)$.

Multiplying with the diagonal matrix W and using $ZW = WZ$, we are led to

$$H = (W\Phi)^H Z(W\Phi) = Q^H ZQ, \quad (4.7)$$

which means that the diagonal matrix Z and the Hessenberg matrix H are unitarily similar. The constant polynomial ϕ_0 is normalized when it is equal to η_{00}^{-1} with η_{00} given by

$$Q^H \mathbf{w} = [\eta_{00}, 0, \ldots, 0]^T,$$

where $\mathbf{w} = [w_0, \ldots, w_m]^T$. Since $Q = W\Phi$ and $\|\phi_0\| = 1$, we see that all the entries in $Q^H\mathbf{w}$ are zero by orthogonality, except for the first one, which is $1/\phi_0$.

Thus the problem of constructing a one-variable orthonormal polynomial basis is reduced to the following inverse eigenvalue problem: given the complex points $Z = \mathrm{diag}(\zeta_i)$ and the weights $\mathbf{w} = (w_i)$, find unitary Q and upper Hessenberg H such that

$$Q^H \mathbf{w} = \|\mathbf{w}\|\mathbf{e}_1, \quad Q^H Z Q = H. \tag{4.8}$$

4.2.2 Multivariate case

In what follows we use the abbreviation OP for orthonormal polynomials. To generalize the results of the previous subsection to the bivariate case we must recall that we have a choice between multiplication by x and y to proceed from a current OP $\phi_{k-1}(x, y)$ to one of the following OPs. This choice is predetermined by the ordering of terms chosen in the definition of $\mathcal{P}_n$.

Recurrence relations for the two-variable OPs could be written in a manner similar to (4.6):

$$\begin{aligned} x[\phi_0(x,y), \phi_1(x,y), \phi_2(x,y), \ldots] &= [\phi_0(x,y), \phi_1(x,y), \phi_2(x,y), \ldots]H_x, \\ y[\phi_0(x,y), \phi_1(x,y), \phi_2(x,y), \ldots] &= [\phi_0(x,y), \phi_1(x,y), \phi_2(x,y), \ldots]H_y. \end{aligned} \tag{4.9}$$

However, the matrices H_x and H_y are not anymore Hessenberg. They are what we call generalized Hessenberg and their structure becomes clear from the following example.

Consider the graded lexicographic ordering of terms (the numbers in the second table are the order numbers of terms in the same place in the first table):

$$\begin{array}{c|ccccc} & y^4 & & & & \\ & y^3 & xy^3 & & & \\ y & y^2 & xy^2 & x^2y^2 & & \\ & y & xy & x^2y & x^3y & \\ & 1 & x & x^2 & x^3 & x^4 \\ \hline & & & x & & \end{array} \qquad \begin{array}{c|ccccc} & 15 & & & & \\ & 10 & 14 & & & \\ y & 6 & 9 & 13 & & \\ & 3 & 5 & 8 & 12 & \\ & 1 & 2 & 4 & 7 & 11 \\ \hline & & & x & & \end{array}$$

.

Let us write the recurrence relations for the first six polynomials, denoting by boldface the polynomial that is being determined from each equation:

$$\begin{aligned} \boldsymbol{\phi_1} &= const \\ x\phi_1 &= [\phi_1, \boldsymbol{\phi_2}] \cdot H_x(1:2,1) \\ y\phi_1 &= [\phi_1, \phi_2, \boldsymbol{\phi_3}] \cdot H_y(1:3,1) \\ x\phi_2 &= [\phi_1, \phi_2, \phi_3, \boldsymbol{\phi_4}] \cdot H_x(1:4,2) \\ y\phi_2 &= [\phi_1, \phi_2, \phi_3, \phi_4, \boldsymbol{\phi_5}] \cdot H_y(1:5,2) \\ x\phi_3 &= [\phi_1, \phi_2, \phi_3, \phi_4, \boldsymbol{\phi_5}] \cdot H_x(1:5,3) \\ y\phi_3 &= [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \boldsymbol{\phi_6}] \cdot H_y(1:6,3) \end{aligned} \tag{4.10}$$

For the case of real ζ_i the matrices H_x and H_y are symmetric and have the following structure ($\times$, $\bowtie$ and $\boxtimes$ all denote (possibly) nonzero elements, $\boxtimes$ is the pivot essential for the previous system of equations):

$$H_x = \begin{pmatrix} \times & \bowtie & & & & \\ \boxtimes & \times & \bowtie & \bowtie & & \\ & \bowtie & \times & \bowtie & \bowtie & \bowtie \\ & \boxtimes & \bowtie & \times & \bowtie & \bowtie \\ & & \boxtimes & \bowtie & \times & \bowtie \\ & & & \bowtie & \bowtie & \times \end{pmatrix}, \quad H_y = \begin{pmatrix} \times & \bowtie & \bowtie & & & \\ \bowtie & \times & \bowtie & \bowtie & \bowtie & \\ \boxtimes & \bowtie & \times & \bowtie & \bowtie & \bowtie \\ & \bowtie & \bowtie & \times & \bowtie & \bowtie \\ & \boxtimes & \bowtie & \bowtie & \times & \bowtie \\ & & \boxtimes & \bowtie & \bowtie & \times \end{pmatrix}. \tag{4.11}$$

Polynomial $\phi_5(\zeta)$ is the first polynomial that could be reached by multiplication both by x and y. This gives two equations to determine it. The following scheme illustrates one of the possibilities (we call it downside-up scheme). Vertical arrows denote multiplication by y, horizontal arrows denote

multiplication by x.

$$
\begin{array}{c|ccccccc}
 & 10 & & & & & & \\
 & \uparrow & & & & & & \\
 & 6 & & 9 & & & & \\
y & \uparrow & & \uparrow & & & & \\
 & 3 & & 5 & & 8 & & \\
 & \uparrow & & \uparrow & & \uparrow & & \\
 & 1 & \rightarrow & 2 & \rightarrow & 4 & \rightarrow & 7 \\
\hline
 & & & x & & & &
\end{array}
\tag{4.12}
$$

Given the monomial ordering the structure of the matrices H_x and H_y is fixed. If one of the pivot elements $\boxtimes$ becomes zero on some step, this means that on this step an interpolating polynomial is constructed. This polynomial is zero in all points ζ_i used up till this step.

To determine the basis polynomial ϕ_k we introduce the following inductive scheme. Basis: fix ϕ_0 to be constant. Inductive step: suppose that all ϕ_i, $0 \leqslant i < k$, are computed. For ϕ_k look at the scheme (4.12) and decide whether to take coefficients from H_x or H_y matrix. Take the first unused column in this matrix and build on its basis the recurrence equation like in (4.10). Mark the column as used. If there is a recurrence relation for the same polynomial in the second matrix, then mark the first unused column there as used also.

This scheme is easily generalized to polynomials of more than two variables. The only thing that needs to be updated is the monomial order and the scheme (4.12), determining the recurrence relations.

Let us now derive the inverse eigenvalue problem, similar to (4.8) in the univariate case.

We may recall again that our inner product is discrete and thus we work with functions as with the $(m+1)$-vector of its values in $\zeta_k = (x_k, y_k)$, $k = 0, \ldots, m$. It means that the $(m+1)$-st orthogonal polynomial will be orthogonal to the whole space, hence it must be zero. This makes possible to rewrite the relations (4.9) as

$$X\Phi = \Phi H_x,$$

$$Y\Phi = \Phi H_y,$$

with

$$\Phi = \Phi_m = \begin{pmatrix} \phi_0(\zeta_0) & \ldots & \phi_n(\zeta_0) \\ \vdots & & \vdots \\ \phi_0(\zeta_m) & \ldots & \phi_n(\zeta_m) \end{pmatrix}.$$

Using a similar technique as in the beginning of this section, the problem of constructing the bivariate orthonormal polynomial basis is reduced to the following inverse eigenvalue problem: given the complex points $\boldsymbol{\zeta}_i = (x_i, y_i)$, $X = \mathrm{diag}(x_i)$, $Y = \mathrm{diag}(y_i)$, the weights $\mathbf{w} = (w_i)$ and the ordering scheme, find unitary Q and upper generalized Hessenberg matrices H_x and H_y such that

$$Q^H \mathbf{w} = \|\mathbf{w}\| \mathbf{e}_1, \quad Q^H X Q = H_x, \quad Q^H Y Q = H_y. \tag{4.13}$$

4.3 Inverse eigenvalue problem and updating algorithm

In this section we present first a general formulation of an algorithm to solve the inverse eigenvalue problem (4.13), and then study it in detail for a 6×6-example.

4.3.1 General formulation of the algorithm

We will now describe an algorithm which, given the initial data (the points $\boldsymbol{\zeta}_i$, the weights $\mathbf{w}_i$ and the ordering scheme $\pi(i)$), computes the matrices H_x

Algorithm 3: Transformation of $D = [\mathbf{w}|X|Y]$ into $[Q^H\mathbf{w}|Q^HXQ|Q^HYQ]$ having zeros below the pivots, function $\pi(j)$ is a $\mathbf{w}/x/y$ switch

begin
  for $i = 2 : n$ **do**
    for $j = 1 : i-1$ **do**
      - make element $d_{i,\pi(j)}$ zero
        by Givens rotation G^H with the pivot element $(j, \pi(j))$:
        $D = G^H D$
      - $D = D \begin{pmatrix} 1 & & \\ & G & \\ & & G \end{pmatrix}$ (similarity transformation)

    end
  end
end

and H_y – the building blocks of the recurrence relation generating the desired orthonormal polynomials. The pseudocode formulation is presented below as Algorithm 3.

The algorithm starts with the following matrix:

$$\begin{pmatrix} w_1 & x_1 & & & y_1 & & \\ w_2 & & x_2 & & & y_2 & \\ \vdots & & & \ddots & & & \ddots \\ w_m & & & x_m & & & y_m \end{pmatrix} = [\mathbf{w}|X|Y] \in \mathbb{C}^{m\times(2m+1)} \tag{4.14}$$

and transforms it using unitary similarity transformations into

$$[\|\mathbf{w}\|\mathbf{e}_1|H_x|H_y] = [Q^H\mathbf{w}|Q^HXQ|Q^HYQ] = Q^H[\mathbf{w}|X|Y]\begin{pmatrix} I_n & & \\ & Q & \\ & & Q \end{pmatrix}. \tag{4.15}$$

The matrix Q is unitary, such that $[Q^H\mathbf{w}|Q^HXQ|Q^HYQ]$ has zeros below the pivot positions $(i, \pi(i))$, $i = 1, 2, \ldots, m$. Algorithm 3 will add, for each i, the point $\boldsymbol{\zeta}_i$ with corresponding weight w_i. Each iteration changes the underlying inner product, adding a new point ζ_i to it.

4.3.2 6×6 example

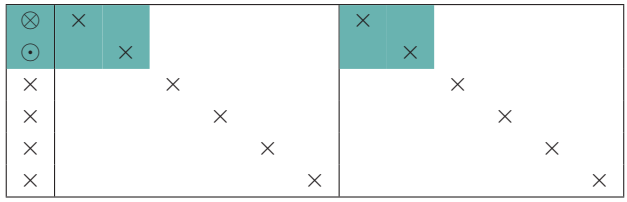
Let us restrict ourselves to the problem of finding the first 6 OPs using the downside-up ordering scheme (4.12). Here $\times$ denotes the original elements of the matrix $D = [\mathbf{w}|X|Y]$, $\bowtie$ stands for some possible nonzero elements, $\boxtimes$ marks the positions of pivots essential for the recurrence relations, and at each step $\odot$ is the element to be annihilated by the Givens rotation built on the element $\otimes$. By [ ] we mark the part of the matrix D that is actually changing at the corresponding *inner* iteration of Algorithm 3. Elements marked by [ ] together with elements marked by [ ] represent the part of the matrix D that is being processed during the current *outer* iteration of Algorithm 3.

We also specify the transformation matrix at each step. $G^w(i, j)$ means the Givens rotation constructed using the i-th and j-th element of the (transformed) weight vector; $G^X(i, j)$ means the Givens rotation constructed using an element

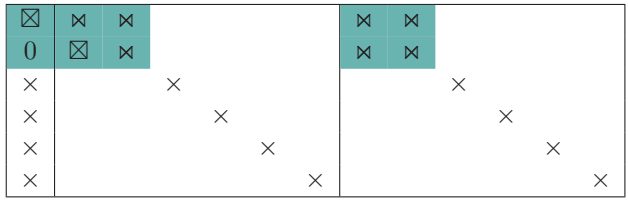
in the X part of the matrix D, placed in i-th row to annihilate the corresponding element placed in j-th row; the column is then $\pi(j)$, and similarly to it we define $G^Y(i,j)$.

We restrict ourselves to pairs of real points $\boldsymbol{\zeta}_i$, resulting in matrices H_x and H_y being symmetric.

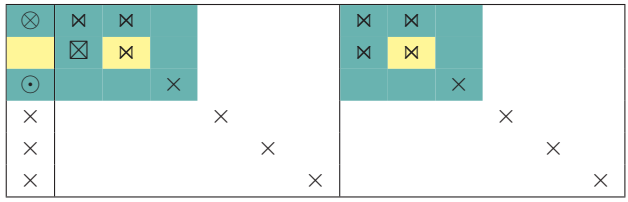
We start with the inner product based on 2 points, so 2×2-submatrices of D are being processed. The ordering scheme tells that the second polynomial comes from the X part, and we mark there its corresponding pivot with $\boxtimes$.



$\downarrow G^w(1,2) \downarrow$

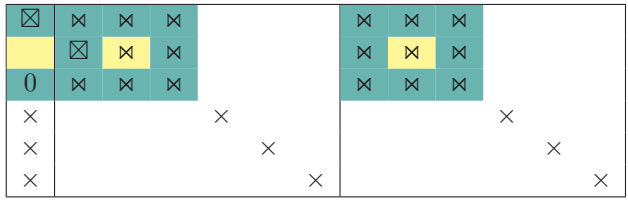


Then we add one more point and start again with the weight vector.

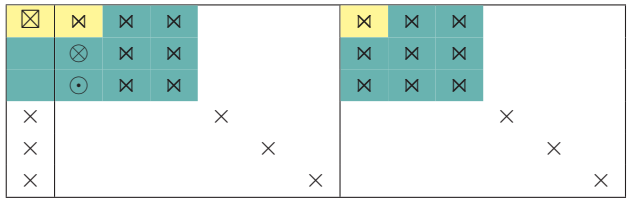


$\downarrow G^w(1,3) \downarrow$

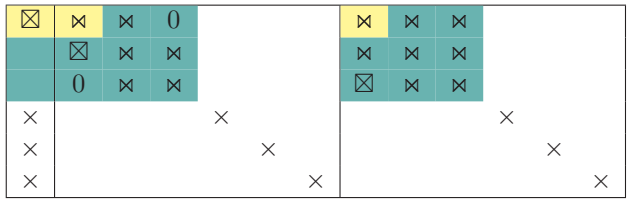
$\downarrow G^w(1,3) \downarrow$



At this moment we have to restore the (generalized) Hessenberg structure in the first column of the X part, using the pivot. Because of symmetry two zeros will appear in the X part. Then we mark the pivot position for the third polynomial, it is in the Y part according to (4.12).

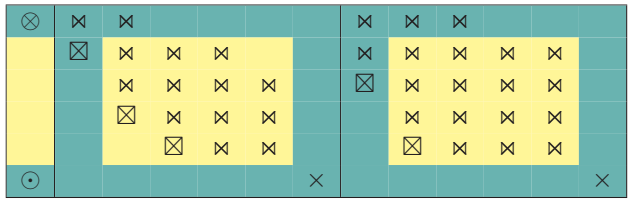


$\downarrow G^X(2,3) \downarrow$



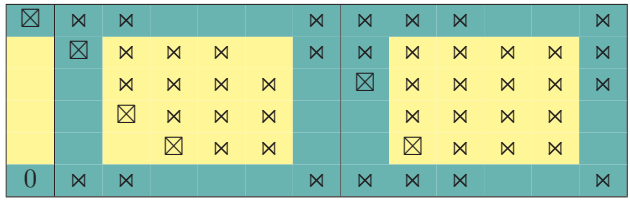
Let us skip the stages of working with 4 and 5 points and proceed directly to the procedure of adding the 6th point. Again we start with the weight vector. Note that there are two pivot positions in the 5th row. This corresponds to the fact that the 5th OP can be derived either by multiplying by x or by y.

At first, we make a zero element in **w**-vector:

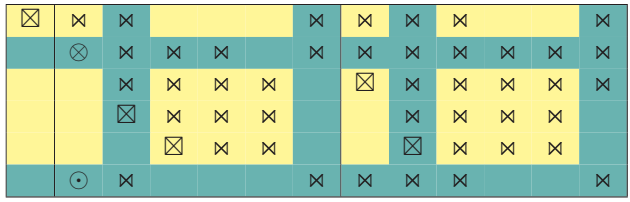


$\downarrow G^w(1,6) \downarrow$

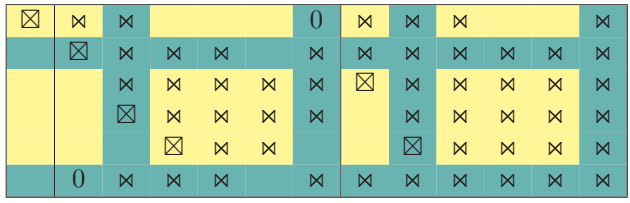
$\downarrow G^w(1,6) \downarrow$



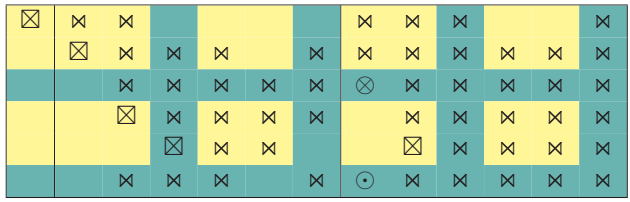
Now the generalized Hessenberg structure is destroyed, so we start chasing the nonzeros in the last row from left to right. The first pivot comes from the X part, second row.



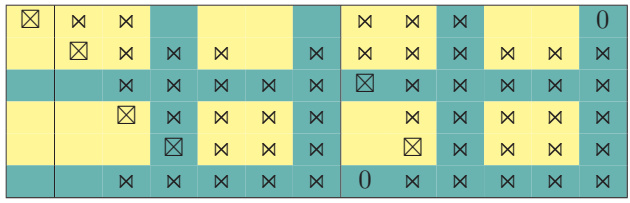
$\downarrow G^X(2,6) \downarrow$



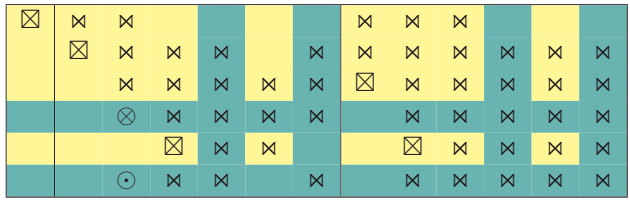
The next pivot comes from the Y part, third row.



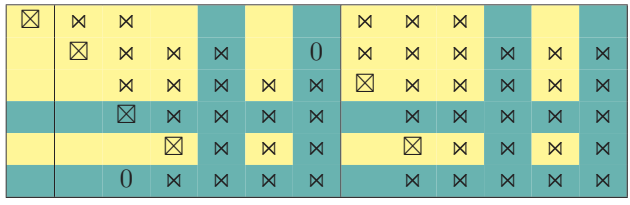
$\downarrow G^Y(3,6) \downarrow$



The next pivot comes from the X part, fourth row.

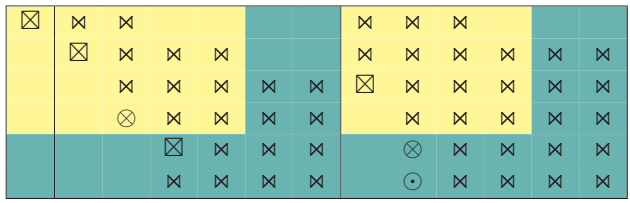


$\downarrow G^X(4,6) \downarrow$

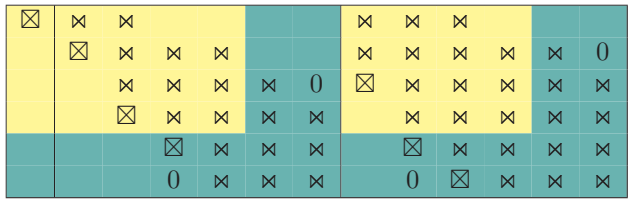


The next pivot comes from the fifth row, but we have a choice between the X part and the Y part. Our ordering scheme (4.12) tells us to take the pivot from the Y part. However, we could also choose the pivot from the X-part, e.g. if two pivots differ in some orders of magnitude, then choosing the larger one may enhance the numerical stability. Note that the element below the pivot position in the 5th row in X part also becomes zero. At this step we mark by

$\boxtimes$ the pivot for the 6th polynomial in the Y-part, its position is in the 6th row.



$\downarrow G^Y(5,6) \downarrow$



The elements of the last matrix are exactly the values appearing in the recurrence system of equations (4.10).

4.4 Numerical experiments

We have implemented Algorithm 3 in Matlab and applied it to several problems.

In the implementation we use the ordering scheme (4.12). As the points for the discrete inner product we use Padua points. The software for working with Padua points is described in [26].

The $m+1 = (\delta+1)(\delta+2)/2$ Padua points corresponding to degree $\delta > 0$ are the set of points

$$\mathrm{Pad}_\delta = \{\boldsymbol{\zeta} = (\zeta_1, \zeta_2)\} = \left\{ \gamma\left(\frac{k\pi}{\delta(\delta+1)}\right), \quad k = 0, \ldots, n(n+1) \right\}$$

where $\gamma(t)$ is their "generating curve"

$$\gamma(t) = (-\cos((\delta+1)t), -\cos(\delta t)) \quad t \in [0, \pi].$$

Notice that two of these points are consecutive vertices of the square, $2\delta - 1$ other points are on the edges of the square and the remaining (interior) points are corresponding to self-intersections of the generating curve, see Figure 4.1.

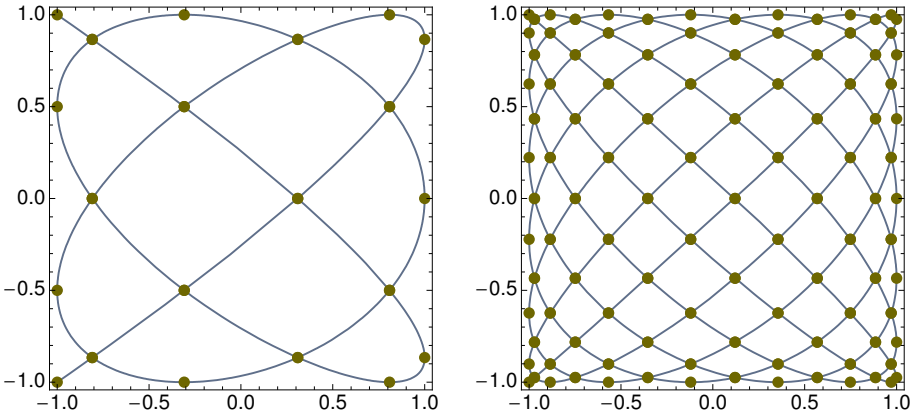


Figure 4.1: Padua points for different n

Example 1: orthogonality test

Let us recall the inverse eigenvalue problem: find a unitary matrix Q (transformation matrix) and generalized upper Hessenberg matrices H_x and H_y such that $Q^H \mathbf{w} = \|\mathbf{w}\|\mathbf{e}_1$, $Q^H X Q = H_x$ and $Q^H Y Q = H_y$. Denote by $\phi_i(\zeta)$ the computed orthonormal polynomials and let $\Phi = [\phi_0(\boldsymbol{\zeta}_i)\, \phi_1(\boldsymbol{\zeta}_i)\, \ldots\, \phi_m(\boldsymbol{\zeta}_i)]_{i=0}^{m+1}$, $W = \mathrm{diag}(w_i)$. Then, as it is proven before, $W\Phi = Q$. Computing the values of the OPs at the nodes $\boldsymbol{\zeta}_i$ by means of the recurrence relations based on H_x, H_y gives us a possibility to check numerically the orthogonality of the matrix $W\Phi$.

As the nodes $\boldsymbol{\zeta}_i$ we take $N = 5151$ Padua points of degree $n = 100$, and the identity matrix as the weight matrix. The values of the OPs are stored in the matrix $V = W\Phi$ and we denote by $R = |V^H V - I|$ (modulus is taken elementwise). In Figure 4.2 we plot $\max R(1:k, 1:k)$ for $k = 10:100:N$.

Example 2: least squares solution

Recall that the solution $p(z)$ to the least squares (LS) problem (4.2) is $p(\zeta) = \sum_{j=0}^{n} c_j \phi_j(\zeta)$. Then $\mathbf{c} = (c_j)$ is given by $\mathbf{c} = \Phi^H W^H W \mathbf{f} = [\langle \phi_i, \mathbf{f} \rangle]$, where the matrices Φ and W are defined in the previous example, so we perform the same row operations on $W\mathbf{f}$ as on $\mathbf{w}$ in Algorithm 3.

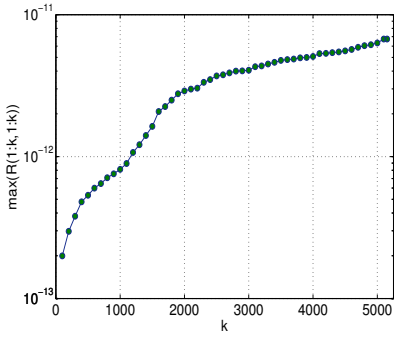


Figure 4.2: Max orthogonality error for the first k OPs, $N = 5151$ points

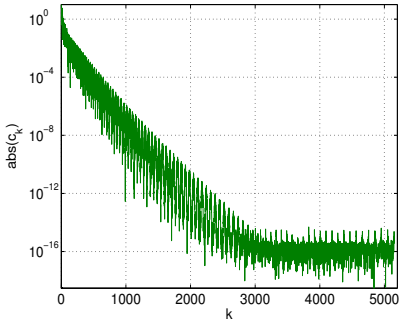


Figure 4.3: LS solution coefficients for the Franke function, $N = 5151$ points

As the first test function we consider the Franke function

$$F(x,y) = \frac{3}{4}e^{-\frac{(9x-2)^2}{4}-\frac{(9y-2)^2}{4}} + \frac{3}{4}e^{-\frac{(9x+1)^2}{49}-\frac{9y+1}{10}}$$
$$+ \frac{1}{2}e^{-\frac{(9x-7)^2}{4}-\frac{(9y-3)^2}{4}} - \frac{1}{5}e^{-(9x-4)^2-(9y-7)^2}$$

on $[0,1]\times[0,1]$ and transform the $N = 5151$ Padua points from $[-1,+1]^2$ to $[0,1]^2$. These transformed points are denoted by $\boldsymbol{\zeta}_i$. Then we compute the right-hand side $\mathbf{f} = F(\boldsymbol{\zeta}_i)$ and the least squares solution coefficients $\Phi^H W^H W\mathbf{f} = \mathbf{c}$. In Figure 4.3 the absolute values of the solution coefficients $\mathbf{c}_k$ are plotted for all k. It is easy to see that from $k \approx 3000$ they are of machine-precision size.

On Figures 4.4 and 4.5 we present surface plots of the relative LS solution error, approximating with the polynomial of length 1000 and of length 3000, correspondingly. (The concept of length is defined in Section 4.1.) When we approximate with the polynomial of length more than 3000, such a surface plot remains basically the same as for 3000. This corresponds to the data of Figure 4.3: all the basis polynomials of length more than approximately 3000 are taken into the solution with the coefficients of machine-precision size and thus basically do not change the solution.

The polynomial of length 5151 is a polynomial of degree 100, one of length 3000 is of degree 76 and one of length 1000 is of degree 44.

Using the same polynomial basis as for the Franke function, we solved the LS problem with the function

$$G(x,y) = \exp\{(\sqrt{x^2+y^2})^3\}.$$

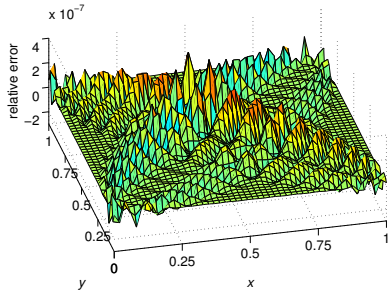


Figure 4.4: Max LS error when approximating with the polynomial of length 1000, $N = 5151$ points

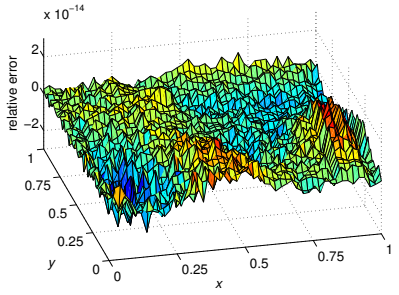


Figure 4.5: Max LS error when approximating with the polynomial of length 3000, $N = 5151$ points

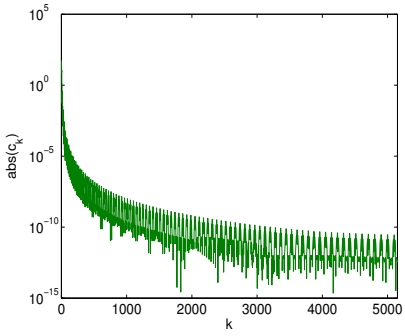


Figure 4.6: LS solution coefficients for $G(x, y)$, $N = 5151$ points

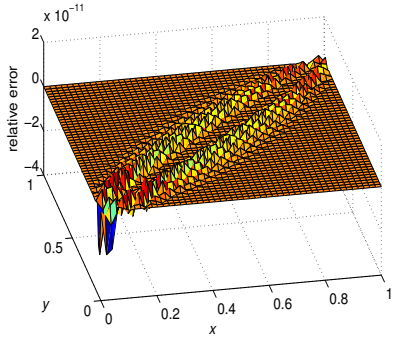


Figure 4.7: Max LS error when approximating with the polynomial of length 4000, $N = 5151$ points

On Figure 4.6 we plot the LS solution coefficients for $G(x, y)$. It is clear that they decay slower than the ones for the Franke function, so we can expect a less accurate approximation. On Figure 4.7 we plot the relative error for the approximant of length 4000 (degree 88). Again, the size of the error is in correspondence with the size of the LS coefficients.

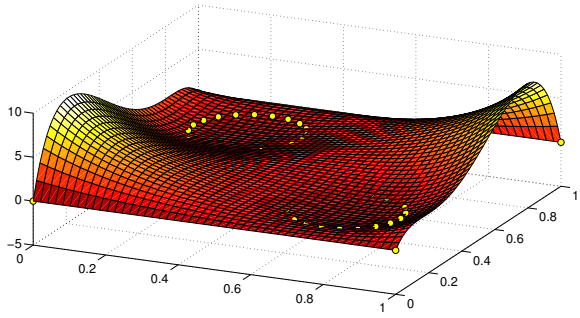


Figure 4.8: Surface plot of an interpolating polynomial

Example 3: Polynomial that goes through some points

Consider the square $[0,1] \times [0,1]$. As points, we take 20 equidistant points on the circle with center $(0.25;\ 0.25)$ and radius 0.15. The next 20 points are taken similarly on a circle with center $(0.75;\ 0.75)$ and radius 0.15. The last 4 points are the 4 edges of the square. We look for the polynomial having "least degree" that has zero value in the given points. This situation corresponds to the first zero pivot appearing in the recurrence relation. It happens for the 28th orthogonal polynomial. It is the polynomial of degree 6, which corresponds with the theoretical estimate (degree 2+2+1+1, two circles and two lines). Figure 4.8 shows the surface plot of this polynomial.

Example 4: Comparison with 2D Lanczos method

This example compares the accuracy of our new method and the two dimensional Hermitian Lanczos method, by Huhtanen and Larsen, [88] (further referred as the HL-method). Both methods were implemented in Matlab and executed with standard Matlab precision (32 digits) and in variable precision arithmetic (vpa) with 128 digits. Because vpa computations are very slow, we used it only for small examples.

Consider the square $[-1,1] \times [-1,1]$ and three data sets in it: (a) 15 Padua points, (b) 10 points such that x and y coordinates are independently random and equally distributed, and (c) 10 points lying on a circle, perturbed with noise of different magnitudes (*amp* in the following table) to make the problem more ill-conditioned or less ill-conditioned.

	(a) Padua	(b) random	(c) ill-cond, $amp = 10^{-4}$	(c) ill-cond, $amp = 10^{-10}$
$Q - U$	1e-15	2e-15	2e-12	3e-6
$U_s - U$	4e-16	6e-16	3e-12	1e-6
$U_s - Q$	1e-15	1e-15	2e-12	3e-6
$H_x - A$	7e-16	6e-16	1e-12	8e-7
$A_s - A$	2e-16	2e-16	4e-13	8e-8
$A_s - H_x$	6e-16	6e-16	7e-13	5e-7
$Q^*Q - I$	1e-15	9e-16	1e-15	7e-16
$U^*U - I$	5e-16	3e-16	3e-16	5e-16

Table 4.1: Comparison of methods: some relative errors

For different points and methods we compared the transformation matrices Q and the recurrence relation matrices H_x and H_y. We use the following notation: matrices given by our method we denote by Q, H_x and H_y as earlier, matrices coming from the HL-method we denote as U, A, B correspondingly. Since both methods have shown exactly the same behavior in vpa arithmetic, we use one set of vpa results to compare with and denote them by U_s, A_s and B_s. It was seen that the matrices H_x and H_y, as well as A and B for the selected ordering scheme do not differ from the accuracy point of view, so we also give the results only for one of them. The numerical results are presented in Table 4.1. For each table cell with random input data we performed 10 runs and averaged the results.

As the last test, we sampled different polynomials represented as linear combinations of the basis polynomials in the data points and recovered them as LS solutions of problem (4.2). We denote as vector $\mathbf{c}$ the linear combination coefficients in Table 4.2 and the following possibilities were investigated: all-ones vector, random vector uniformly distributed on $[0, 1]$ and random vector uniformly distributed on $[0, 100]$. Since we did not mention any significant difference between all these, results only for all-ones vector are given. We computed relative errors $\|\mathbf{c}_{comp} - \mathbf{c}\|/\|\mathbf{c}\|$, where $\mathbf{c}_{comp}$ is the recovered LS solution, coming from our method ($norm1$) and from the HL-method ($norm2$). The numerical results are presented in Table 4.2.

We can conclude from these examples that the Q, H_x and H_y matrices themselves have on average a half-digit better precision in the HL-method than in the new method, which leads to the whole digit difference in the recovery problem. However, both methods seem to treat equally well-conditioned and ill-conditioned problems, giving good accuracy. Slightly lower accuracy of the new method is compensated by the fact that it is able to treat data points one after another and store the intermediate results. Such updating procedure is

data set	$norm1$	$norm2$
496 Padua pts	6e-13	1e-14
861 Padua pts	7e-13	2e-14
1326 Padua pts	8e-13	4e-14
105 random pts	1e-10	3e-11
496 random pts	5e-13	5e-14
36 pts on circle, noise amp 1e-2	6e-6	5e-7

Table 4.2: Comparison of methods: polynomial recovery

useful in many applications.

4.5 Conclusion

We presented an algorithm computing the recurrence relation coefficients for bivariate polynomials, orthonormal with respect to a discrete inner product. To do so, we transformed the original problem to an inverse eigenvalue problem and solved it by applying a sequence of specially built Givens rotations. The algorithm is a basis tool in solving different problems of numerical mathematics, such as the polynomial interpolation problem or the discrete least squares problem. Numerical examples show that the algorithm could indeed be efficiently applied to such problems and also illustrate its good numerical stability.

Chapter 5

Structured matrices: facts

In this chapter we bring into a common framework some known facts and concepts involving structured matrices. Though not new themselves, they are necessary for the understanding of algorithms that will be presented later in Chapter 6. We begin with defining different structure types matrices and give several technical results in Section 5.1. Later in Section 5.2 we discuss a homotopy approach, which becomes an important tool to work out two of the problems in the next chapter.

5.1 Basic concepts

In this section we will define different types of structured matrices, including Toeplitz matrices, semiseparable matrices and general low displacement rank matrices. Later we will discuss several auxiliary theorems and techniques, that will be used in the next chapter.

5.1.1 Types of matrix structure

Toeplitz structure

A matrix $T \in \mathbb{R}^{n\times n}$ is called Toeplitz, if $T = (t_{ij}) = (a_{j-i})$:

$$T = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \\ a_{-1} & a_0 & a_1 & \ddots & \vdots \\ a_{-2} & a_{-1} & a_0 & \ddots & a_2 \\ \vdots & \ddots & \ddots & \ddots & a_1 \\ a_{-(n-1)} & \cdots & a_{-2} & a_{-1} & a_0 \end{pmatrix}. \tag{5.1}$$

A Toeplitz matrix is determined only by $2n-1$ parameters. This has led to several fast and superfast algorithms for the solution of linear Toeplitz systems $Tx = b$, utilizing the matrix structure. The two types of direct fast solvers that require $\mathcal{O}(n^2)$ operations are Levinson-type and Schur-type solvers. For more references and information about these algorithms, we refer the reader to [93].

One of the good ways to define, identify and exploit the matrix structure such as the Toeplitz one is the concept of *displacement ranks*. This idea was first defined in the paper [92]. We say that a matrix M has *low displacement rank*, if for some matrices A and B the rank of the matrix $X = M - AMB$ is low. For an appropriate choice of A and B a Toeplitz matrix T can be converted to a rank two matrix X. Then X can be decomposed as $X = GH^*$ with $G, H \in \mathbb{R}^{n\times 2}$ (or $G, H \in \mathbb{C}^{n\times 2}$), * denotes Hermitian conjugate. Matrices G and H are called *displacement generators* of T. Based on G and H it is possible to construct generators for T^{-1}. These generators are also of size $n \times 2$.

We study the theory of displacement ranks in more detail in Subsection 5.1.2. This theory, together with a homotopy approach, serves as an important tool while developing a superfast solver for linear systems with Toeplitz matrices in Section 6.1. Moreover, our algorithm also allows to construct a compact representation of the inverse of a given symmetric Toeplitz matrix.

In some applications, like numerical solution of multidimensional integral equations and image deblurring, the so-called two-level Toeplitz matrices occur, which are block Toeplitz matrices with blocks of Toeplitz structure. Despite their seeming similarity to dense Toeplitz matrices, all the attempts to build a superfast direct solver for such matrices to our knowledge did not succeed up till now.

In Section 6.3 we construct a direct solver for a class of two-level Toeplitz matrices, where the outer Toeplitz structure is banded.

Diagonal-plus-semiseparable structure

A matrix A is called a *symmetric semiseparable matrix* if all submatrices taken out of the lower and upper triangular part of the matrix are of rank 1 and the matrix is symmetric. Here the lower (upper) parts include the diagonal of the matrix.

A matrix A is called a *symmetric generator representable semiseparable matrix* if the lower triangular part of the matrix is coming from a rank 1 matrix and the matrix A is symmetric.

A matrix A is called a *symmetric diagonal-plus-semiseparable matrix* if it can be written as the sum of a diagonal and a symmetric semiseparable matrix. We will further often shorten the "diagonal-plus-semiseparable" to D+SS.

A matrix A is called a *symmetric generator-representable diagonal-plus-semiseparable matrix* if it can be written as the sum of a diagonal and a symmetric generator-representable semiseparable matrix.

Within this research we will work with symmetric generator-representable D+SS matrices of the form

$$A = \begin{pmatrix} d_1 & u_1v_2 & u_1v_3 & \cdots & u_1v_{N-1} & u_1v_N \\ v_2u_1 & d_2 & u_2v_3 & u_2v_4 & \cdots & u_2v_N \\ v_3u_1 & v_3u_2 & \ddots & \cdots & \cdots & \cdots \\ \vdots & v_4u_2 & \vdots & \ddots & \cdots & \cdots \\ v_{N-1}u_1 & \vdots & \vdots & \vdots & d_{N-1} & u_{N-1}v_N \\ v_Nu_1 & v_Nu_2 & \vdots & \vdots & v_Nu_{N-1} & d_N \end{pmatrix}. \tag{5.2}$$

Let $D = \operatorname{diag}(d_1, \ldots, d_N)$, $\mathbf{u} = (u_1, u_2, \ldots, u_{N-1}, u_N)^T$, $\mathbf{v} = (v_1, v_2, v_3, \ldots, v_N)^T$. A can be written as the sum of three matrices

$$A = D + \operatorname{triu}(\mathbf{u}\mathbf{v}^T) + \operatorname{triu}(\mathbf{u}\mathbf{v}^T)^T,$$

where $\operatorname{triu}(M)$ denotes the strictly upper triangular part of the matrix M.

In what follows we will assume that $u_1 \neq 0$. This is not a restriction for an eigensolver, because if $u_1 = 0$, then d_1 is an eigenvalue and the matrix can be reduced to a matrix with the first component of $\mathbf{u}$ different from 0.

Such symmetric diagonal-plus-semiseparable matrices are determined by an amount of parameters linear in their size, namely, by $3N - 2$ parameters. This type of structure allows even more effective algorithms, compared to Toeplitz structure. More specific, a QR-type, as well as Levinson and Schur-type solvers have been developed for D+SS matrices. Their complexity is linear in N, as shown in [174, Ch. 5 and Ch. 6]. For more references and information about these algorithms, we refer the reader to this book, which gives a very complete survey of recent achievements in the theory of semiseparable matrices.

It is also shown in [174], that in some applications the generator representation (5.2) does not allow to construct stable algorithms (for example, it causes problems for a QR-solver, see [165]). As an alternative, the so-called Givens-weight representation [174, Ch. 2] of semiseparable matrices should be preferred. Other definitions of semiseparability are also discussed within this reference.

In Section 6.2 we derive a divide-and-conquer algorithm to compute the eigenvalues and eigenvectors of a symmetric generator-representable diagonal-plus-semiseparable matrix. As shown by Van Camp [165], the generator representation does not cause much trouble for divide-and-conquer algorithms, but allows much simpler formulas for getting the insight of the algorithm. So the restriction to the representation (5.2) is well-founded.

The importance of a semiseparable class of structure is based on the fact that arbitrary symmetric matrices could be efficiently transformed to a semiseparable form with a finite number of orthogonal similarity tranformations, see [175, Ch. 2]. These reduction processes often have better stability and finely tunable convergence behavior, as described in detail in [175, Ch. 3], compared to a reduction to a tridiagonal form. Existence and uniqueness of the reduction to generator-representable semiseparable form are investigated in detail in [12]. Thus, D+SS matrices are excellent candidates for an intermediate step while trying to solve linear algebra problems with general matrices.

5.1.2 Low displacement rank matrices

Toeplitz matrices represent an important class of low displacement rank matrices. Computations with low displacement rank matrices, if performed properly, require much less computer time and memory space, compared to general matrices. In this subsection we will give definitions for different types of displacement ranks and also provide some technical results, required in the next subsections.

General definitions

Formally, we associate real $m \times n$ matrices with linear operators $L : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ of *Sylvester* type, $L = \nabla_{A,B}$:

$$L(M) = \nabla_{A,B}(M) = AM - MB, \tag{5.3}$$

and *Stein* type, $L = \Delta_{A,B}$:

$$L(M) = \Delta_{A,B}(M) = M - AMB, \tag{5.4}$$

for fixed pairs $\{A, B\}$ of *operator matrices.*

According to the following proposition, these two types of displacement operators are closely related:

Proposition 8 (see [119])**.** *Let $\nabla_{A,B}$ and $\Delta_{A,B}$ be the displacement operators, defined by (5.3) and (5.4) respectively. Then there exists the following connection between them: $\nabla_{A,B} = A\Delta_{A^{-1},B}$ if the operator matrix A is non-singular, and $\nabla_{A,B} = -\Delta_{A,B^{-1}}B$ if the operator matrix B is non-singular.*

Frequently used operator matrices are the matrices Z_f, Z_f^* and $D(\mathbf{v})$. Here Z_f is the unit f-circulant matrix, where f is an arbitrary scalar:

$$Z_f = \begin{pmatrix} 0 & & & f \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix}, \tag{5.5}$$

$D(\mathbf{v}) = \operatorname{diag}(v_i)_{i=1}^n$ and $\mathbf{v}$ is a vector of length n.

Let us fix matrices A and B. Let $L = \nabla_{A,B}$ or $L = \Delta_{A,B}$. The rank r of the displacement $L(M)$ is called the *displacement rank* of the matrix M. If A and B are chosen properly, the displacement rank of a structured matrix $M \in \mathbb{R}^{m\times n}$ can be very small (for instance, $r = \mathcal{O}(1)$ or $r = o(\min(m, n))$). Table 5.1 shows some specific choices of operator matrices for several well known classes of matrix structure.

Assume that the operator matrices A and B are chosen such that the displacement rank r of the matrix M is small. Then we can write a skeleton decomposition of $L(M)$:

$$L(M) = \Delta_{A,B} = GH^*, \tag{5.6}$$

operator matrices A	operator matrices B	class of structured matrices M	rank of $\nabla_{A,B}(M)$
Z_1	Z_0	Toeplitz and its inverse	$\leqslant 2$
Z_1	Z_0^*	Hankel and its inverse	$\leqslant 2$
$D(\mathbf{t})$	Z_0	Vandermonde	$\leqslant 1$
Z_0	$D(\mathbf{t})$	inverse of Vandermonde	$\leqslant 1$

Table 5.1: Displacement ranks for some structured matrices

where G and H are $n \times r$ matrices. These matrices are called $\{A, B\}$-*displacement generators*. The symbol $\{A, B\}$ can be omitted if the corresponding matrices are clear from the context. Such pair of matrices G and H is often called a *generator representation* of a given matrix M.

Since any Toeplitz matrix has displacement rank $\leqslant 2$ (see e.g. [119]), its generators are of size $n \times 2$. Thus to store the generators of an $n \times n$ Toeplitz matrix T we need only $\mathcal{O}(n)$ storage space, compared to $\mathcal{O}(n^2)$ for a whole dense matrix. Furthermore, under certain general conditions on A and B the matrix T can be fully reconstructed from its generators.

There exists a connection between the displacements of a nonsingular matrix and its inverse, as shown by the following proposition.

Proposition 9 ([156, 119])**.** *Let M be a nonsingular matrix. Then the displacements of M and its inverse M^{-1} are connected:*

$$\nabla_{B,A}(M^{-1}) = -M^{-1}\nabla_{A,B}(M)M^{-1}. \tag{5.7}$$

A similar proposition holds for the operator $\Delta_{A,B}$.

It is well known that the rank of a matrix is invariant under its multiplication by nonsingular matrices. Hence from Proposition 9 it immediately follows that the inverse T^{-1} of a nonsingular Toeplitz matrix T has displacement rank 2 and thus can be represented by generators of size $\mathcal{O}(n)$, similarly to T.

For the theory of displacement ranks and Toeplitz matrices, we refer the reader to [119, 93, 79].

Useful facts

In Section 6.1 we develop an algorithm that solves linear systems of equations with Toeplitz matrices. Within this algorithm we use a displacement generator

representation of Toeplitz matrices. We give now several technical details on this representation, that are needed by the algorithm.

For Toeplitz matrices we operate with the displacement operator $\Delta_{X,Y}$, where $X = Z_1$ and $Y = Z_{-1}^*$, defined by (5.5). Let us denote the $\{X, Y\}$-generators of T as G and H:

$$T - Z_1 T Z_{-1}^* = GH^*. \tag{5.8}$$

Theorem 2 ([156, 119]). *Let $\{Z_1, Z_{-1}^*\}$-generators be known for a Toeplitz matrix T. Then, T can be reconstructed by the formula*

$$T = \frac{1}{2}\left(GH^* + Z_1 GH^* Z_{-1}^* + \cdots + Z_1^{n-1} GH^* (Z_{-1}^*)^{n-1}\right). \tag{5.9}$$

Proof. Let us rewrite (5.8) as

$$T = GH^* + Z_1 T Z_{-1}^*. \tag{5.10}$$

Then let us substitute T on the right-hand side of (5.10) by its expression via (5.10) itself:

$$T = GH^* + Z_1 GH^* Z_{-1}^* + Z_1^2 T (Z_{-1}^*)^2.$$

Repeat this substitution again $(n-2)$ times. Then take into account that $Z_1^n = I_n$ and $(Z_{-1}^*)^n = -I_n$, where I_n is the identity matrix. Thus $Z_1^n T (Z_{-1}^*)^n = -T$. Let us move it to the left-hand side and divide the resulting equation by two. All these transformations result in (5.9). □

Let us show how to use formula (5.9) for computations of matrix-by-vector products. Let us slightly transform this formula and multiply it by a given vector v:

$$2Tv = [G|Z_1 G|\ldots|Z_1^{n-1}G][H|Z_{-1}H|\ldots|Z_{-1}^{n-1}H]^* v = XY^* v.$$

Let us recall that G and H are $n \times 2$ matrices, consequently, matrices X and Y, formed of the block columns, are of size $n \times (2n)$. We distinguish the submatrices X_1, X_2 and Y_1, Y_2 in X and Y, respectively, by incorporating into X_1 (respectively, Y_1) all the columns of X (respectively, Y) having even indices. The remaining columns constitute the submatrix X_2 (respectively Y_2). It is easy to see that X_1 and X_2 are circulant matrices while Y_1 and Y_2 are Toeplitz matrices. It is possible to multiply the $n \times n$ circulant matrix by a vector using only $O(n \log n)$ operations by means of FFT (see [93]). It is also possible to multiply the $n \times n$ Toeplitz matrix by a vector also using $O(n \log n)$ operations by embedding it into a bigger circulant matrix (the details can be found in [93]).

For a Toeplitz matrix T the matrices G and H are $(n \times 2)$-matrices. There is no need to store matrices Z_f explicitly, which reduces the memory requirements to $O(n)$.

Reconstruction formulas of type (5.9) can be easily derived for T^*, T^{-1} and $(T^*)^{-1}$.

Suppose that the generators G and H from (5.8) are known for some matrix T. Let us now derive equations for the generators of the inverse T^{-1}.

Multiplying (5.8) by Z_{-1} on the right gives

$$TZ_{-1} - Z_1 T = GH^* Z_{-1}. \tag{5.11}$$

Multiplying (5.11) by T^{-1} on the left and on the right leads to

$$Z_{-1}T^{-1} - T^{-1}Z_1 = (T^{-1}G)(H^* Z_{-1}T^{-1}). \tag{5.12}$$

Multiplying (5.12) by Z_{-1}^*on the left results in

$$T^{-1} - Z_{-1}^* T^{-1} Z_1 = (Z_{-1}^* T^{-1} G)(H^* Z_{-1} T^{-1}) = \tilde{G}\tilde{H}^*.$$

Here $\tilde{G}$ and $\tilde{H}$ are $\{Z_{-1}^*, Z_1\}$-generators for the inverse matrix T^{-1} and these generators are the (unique) solutions of two linear systems:

$$T(Z_{-1}\tilde{G}) = G, \tag{5.13}$$

$$T^*\tilde{H} = Z_{-1}^* H. \tag{5.14}$$

Later in Section 6.1 we describe how to incorporate these equations into a specific version of an iterative improvement process.

5.1.3 Iterative improvement processes

Iterative improvement processes (further referred as IIP) are a suitable tool for improving a close initial approximation X_0 to the inverse M^{-1} of a nonsingular matrix M.

Such processes include Newton iteration [138]

$$X_{i+1} = 2X_i - X_i M X_i, \quad i = 0, 1, \ldots, \tag{5.15}$$

scaled Newton iteration [121]

$$X_{i+1} = a_{i+1}(2X_i - X_i M X_i), \quad i = 0, 1, \ldots, \tag{5.16}$$

an improvement formula based on a cubic polynomial [40]

$$X_{i+1} = aX_i(MX_i)^2 + bX_i(MX_i) + cX_i + dI, \quad i = 0, 1, \ldots, \tag{5.17}$$

iterative refinement [71] for the equation $XM = I$

$$R_i = I - X_iM, \quad D_i = R_iX_0, \quad X_{i+1} = X_i + D_i, \quad i = 0, 1, \ldots. \tag{5.18}$$

For all processes mentioned above let R_i denote $I - X_iM$. Then their convergence is determined [119, 40, 71, 120] by the 2-norm $\|R_0\| = \|I - X_0M\|$:

$$\|R_i\| \leqslant \|R_0\|^{p^i}, \quad p \text{ depends on the process.} \tag{5.19}$$

This norm $\|R_0\|_2$ is called a *convergence factor* for a given iterative improvement process.

Suppose we know a matrix M_0, its inverse M_0^{-1}, and a matrix M. We call M_0^{-1} an *initial approximation* to the inverse M^{-1}. Then we try to improve this approximation using one of the processes (5.15)–(5.18). However, they can diverge if $\|R_0\| \geqslant 1$ or, in other words, when the initial approximation is too crude (the matrix M_0 is too far from the matrix M). One of possible workarounds is to split the path between M_0 and M into several segments and apply an IIP on each of them. This idea leads to continuation methods, that are discussed in Section 5.2.

IIP's (5.15)–(5.17) can be modified for structured matrices in such a way that they operate only with displacement generators but not with full matrices. However, this modification usually requires complicated compression techniques, details are presented in [119, 40]. We will use a modification of the improvement process (5.18) to compute approximations to the inverses of Toeplitz matrices, as described in Section 6.1. Algorithm 5 represents such a modification.

5.2 Continuation methods

Continuation methods define an easy problem for which we know the solution, and a path between this easy problem and the hard problem that we actually wish to solve. The solution to the easy problem is gradually transformed to the solution of the hard problem by tracing this path. The path may be defined by introducing an additional scalar parameter into the problem.

In this section we give a general formulation of a continuation method and briefly specify it to a Toeplitz matrix inversion problem and a complete eigenvalue problem for symmetric matrices.

5.2.1 General formulation

Suppose we want to solve a problem $\mathbf{F}(\mathbf{x}) = 0$, $\mathbf{x} \in \mathcal{L}$, where $\mathcal{L}$ is some linear space and $\mathbf{F}$ is a mapping that we assume to be smooth. Certainly, if a good approximation $\mathbf{x}_0$ of a zero $\mathbf{x}^*$ of $\mathbf{F}$ is available, it is advisable to calculate $\mathbf{x}^*$ via a Newton-type algorithm defined by an iteration formula such as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - A_k^{-1}\mathbf{F}(\mathbf{x}_k), \tag{5.20}$$

where A_k is some reasonable approximation of the Jacobian of $\mathbf{F}(\mathbf{x}_k)$. However, if such a priori knowledge is not available, the Newton-type iteration (5.20) will often fail because of poor starting values.

As a possible remedy, one defines a *homotopy* or *continuation function* $\mathbf{H} : \mathcal{L} \times [0,1] \to \mathcal{L}$ such that

$$\mathbf{H}(\mathbf{x}, 0) = \mathbf{G}(\mathbf{x}), \quad \mathbf{H}(\mathbf{x}, 1) = \mathbf{F}(\mathbf{x}), \tag{5.21}$$

where $\mathbf{G} : \mathcal{L} \to \mathcal{L}$ is such that its zeros are readily available and $\mathbf{H}$ is smooth. Typically, one chooses a homotopy such as

$$\mathbf{H}(\mathbf{x}, t) = (1-t)\mathbf{G}(\mathbf{x}) + t\mathbf{F}(\mathbf{x}) \tag{5.22}$$

and attempts to trace an implicitly defined curve $c(t) = (\mathbf{x}(t), t)$ in the zero set of this function from a starting point $(\mathbf{x}_0, 0)$ to a solution point $(\mathbf{x}^*, 1)$. If this succeeds, then a zero point $\mathbf{x}^*$ of $\mathbf{F}$ is obtained.

As soon as one effectively desires to trace such curves, a few questions should be answered first:

1. When does such a curve in the zero set exist and when is it smooth?
2. How can we numerically trace such a curve at reasonable cost?

The first question is answered by the implicit function theorem. If $(\mathbf{x}_0, 0)$ is a regular zero point of $\mathbf{H}$, that is, the Jacobian $J_{\mathbf{H}}(\mathbf{x}_0, 0)$ has maximal column rank n, then a smooth curve exists at least locally.

The second question must be investigated in the context of a particular problem. The general idea is to select a (possibly nonregular) mesh on the curve $c(t)$ and update the intermediate solutions at each of the mesh points, using the solution from the previous mesh point as an initial guess. As an updating procedure a Newton-like scheme (5.20) may be used, or a problem-specific updating algorithm.

Another general updating procedure may be constructed, following Allgower and Georg [1], by regarding the curve $c(t)$ as the solution of an initial value problem, which is obtained by differentiating the equation

$$\mathbf{H}(c(t)) = 0 \tag{5.23}$$

with respect to t:

$$\mathbf{H}'(c(t))c'(t) = 0, \quad \|c(t)\| = 1, \quad c(0) = (x_0, 0). \tag{5.24}$$

It is now clear that methods for numerically solving initial value problems may be applied to (5.24). However, when integrating the differential equation (5.24) numerically, one makes some error which will become significant if one does not wish to use a small stepsize. In consideration of the special form of the problem it is possible to stabilize the integration procedure. This may be done by solving (5.23) with an iterative method. Thus one can consider the integration procedure as a *predictor step* and the Newton-type method as a *corrector step*. This is the basic idea of predictor-corrector continuation methods. A thorough discussion of such methods is given in [1].

Within this research we applied homotopy methods to two different problems. The first one is the inversion of symmetric indefinite Toeplitz matrices. It is briefly sketched in the next subsection and it is discussed in detail in Section 6.1. The second one is an adaptation of divide-and-conquer techniques to solve the complete eigenvalue problem for diagonal-plus-semiseparable matrices. We show how continuation techniques can be applied to a general symmetric eigenvalue problem in Subsection 5.2.3 and further specify it to diagonal-plus-semiseparable matrices in Section 6.2.

Continuation methods now have several applications, like generalized eigenvalue problems [53], solving systems of polynomial equations [1, 73], multiobjective optimization problems [139].

5.2.2 Inversion of a matrix

We will now portray briefly a continuation method for the matrix inversion. Let us denote by A_0 some matrix, which is readily invertible, by A_1 the original matrix to invert, and let us construct a continuation matrix function

$$M(t) = (1-t)A_0 + tA_1. \tag{5.25}$$

Suppose we have selected one of the IIP's (5.15)–(5.18). The algorithm can be sketched in the following way:

Algorithm 4: First version of the continuation algorithm

input : A_0 – starting matrix, A_1 – original matrix
output : A_1^{-1}
notation: $M(t) = (1-t)A_0 + tA_1$

begin
  $t = 0$; compute the inverse $A_0^{-1} = M(0)^{-1}$
  while $t < 1$ **do**
    increase t so that $t \leqslant 1$ and the selected IIP still converges
    improve the approximation for $M(t)^{-1}$ using the inverse
      from the previous step as an initial approximation for the IIP
  end
  return $M(1)^{-1} = A_1^{-1}$
end

This formulation is very general and could be applied to any matrix. In Section 6.1 we describe how to specify Algorithm 4 to Toeplitz matrices.

5.2.3 Symmetric eigenvalue problem

In this subsection we will first describe a homotopy method for the symmetric eigenvalue problem. Later, the required properties of the starting matrices will be investigated. Finally, we sketch a predictor-corrector method for path tracing.

General theory

Solving a symmetric eigenvalue problem

$$A\mathbf{x} = \lambda\mathbf{x}, \quad A = A^T, \tag{5.26}$$

can be thought of as solving a system of n nonlinear equations in $n+1$ unknowns. By augmenting the system with normalization condition $(\mathbf{x}^T\mathbf{x} - 1)/2 = 0$ we can write it as

$$\mathbf{F}(\mathbf{x}, \lambda) = \begin{pmatrix} A\mathbf{x} - \lambda\mathbf{x} \\ (\mathbf{x}^T\mathbf{x} - 1)/2 \end{pmatrix} = 0. \tag{5.27}$$

The homotopy approach may be applied to this problem in a natural way. The idea is to start with a problem $D\mathbf{x} = \lambda\mathbf{x}$ which is easier to solve and to

continuously transform its solutions to those of the original problem (5.26). These solutions are the continuous eigenpairs of a matrix family $A(t)$ (cf. (5.25)):

$$A(t) = (1-t)D + tA = D + t(A-D), \quad t \in [0,1], \tag{5.28}$$

with the symmetric starting matrix D. Following Chu [37], we can easily derive the homotopy equation on the basis of (5.27):

$$\mathbf{H}(\mathbf{x},\lambda,t) = \begin{pmatrix} A(t)\mathbf{x} - \lambda\mathbf{x} \\ (\mathbf{x}^T\mathbf{x} - 1)/2 \end{pmatrix} = 0. \tag{5.29}$$

The question remains whether this equation really defines smooth curves, which connect the eigenpairs of D with those of A. The question is answered by the following theorem from perturbation theory.

Theorem 9 (Kato [95]). *The eigenvalues $\lambda_i(t)$, $i = 1,\ldots,n$ of a real symmetric matrix family $A(t)$ are analytic functions of t and there also exist corresponding eigenvectors $\mathbf{x}_i(t)$, which are analytic functions of t.*

These continous eigenpairs of $A(t)$ are called *eigenpaths* and the $\lambda_i(t)$ *eigenvalue curves* or just eigenvalues.

In case that two eigenvalue curves $\lambda_1(t)$ and $\lambda_2(t)$ cross each other, the corresponding eigenvectors are not uniquely defined in the crossing point at say $t = \hat{t}$. But it is always possible to give two orthogonal eigenvectors $\mathbf{x}_1(\hat{t})$ and $\mathbf{x}_2(\hat{t})$ satisfying $A\mathbf{x}_1(\hat{t}) = \lambda_1(\hat{t})\mathbf{x}_1(\hat{t})$ and $A\mathbf{x}_2(\hat{t}) = \lambda_2(\hat{t})\mathbf{x}_2(\hat{t})$ respectively, such that $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ are analytic at $t = \hat{t}$.

Starting matrix

The starting matrix D should meet two requirements. First, its spectral decomposition should be easy to compute. This is obviously best met by a matrix whose spectral decomposition is already known. Apart from diagonal matrices in general there exist various matrices of other structure with known analytic eigenvalues and eigenvectors, see e.g. [85]. Second, the eigenpaths $(\mathbf{x}(t);\lambda(t))$ should be as straight as possible to ease numerical tracing.

Criteria to characterize the smoothness of the paths are given by Li and Rhee [105]. They use bounds on the derivatives $\lambda_k'(t)$, $\mathbf{x}_k'(t)$ to characterize the variation of the eigenpairs. In detail, the derivatives $\lambda_k'(t)$, $\mathbf{x}_k'(t)$ are obtained by differentiating $A(t)\mathbf{x}_k(t) = \lambda_k(t)\mathbf{x}_k(t)$ with respect to t. Then, they have shown that

$$\lambda_k'(t) = \mathbf{x}_l(t)^T A'(t)\mathbf{x}_k(t) \tag{5.30}$$

and

$$\mathbf{x}_k'(t) = \sum_{i \neq k} \frac{\mathbf{x}_i(t)^T A'(t) \mathbf{x}_k(t)}{(\lambda_k(t) - \lambda_i(t))} \mathbf{x}_i(t), \tag{5.31}$$

if $\lambda_k(t)$ is simple at t. Note that the eigenvectors $\mathbf{x}_k(t)$ satisfy $\mathbf{x}_k(t)^T \mathbf{x}_k(t) = 1$ due to the homotopy equation (5.29). Since $A'(t) = A - D$, one obtains with a little more manipulation the bounds

$$|\lambda_k'(t)| \leqslant \|A - D\|, \tag{5.32}$$

$$\|\mathbf{x}_k'(t)\| \leqslant \|A - D\| / d_k(t), \tag{5.33}$$

where $d_k(t) = \min\{|\lambda_k(t) - \mu|,\ \mu \in \sigma(A(t)),\ \mu \neq \lambda_k(t)\}$ denotes the distance from $\lambda_k(t)$ to the nearest eigenvalue of $A(t)$.

Inequalities (5.32) and (5.33) imply that the variation of the eigenvalues is determined by $\|A - D\|$ while the variation of the eigenvectors additionally depends on the separation of the eigenvalues of $A(t)$. This corresponds to a well-known fact that an eigenvector may vary arbitrarily strongly with its corresponding eigenvalue, unless this eigenvalue is simple and sufficiently separated from the other eigenvalues.

A simple starting matrix that worked well in our previous application, namely, the identity matrix for the inversion of a Toeplitz matrix, does not work for the problem under consideration. Consider $D = \alpha I$, then the eigenvalues $\lambda(t)$ are all straight lines emerging from the point α at $t = 0$. Their slope is determined by its corresponding constant eigenvector $\mathbf{x}(t)$. However, it's not possible to find these eigenvectors at $t = 0$ since any orthogonal matrix is a valid eigenvector matrix of D.

Oettli [114] has shown that for simultaneously diagonalizable D and A most of the eigenvectors $\mathbf{x}(t)$ are constant. He also showed that these simultaneously diagonalizable pairs of D and A include all matrices D which are determined by a matrix function of A, i.e.

$$D = f(A) = S \cdot \operatorname{diag}(f(\lambda_1), \ldots, f(\lambda_n)) S^T, \tag{5.34}$$

where S is a nonsingular matrix and $f(\cdot)$ is an arbitrary function like a polynomial, sine, square root, etc. Unfortunately, these matrices are useless in practice because they either require the spectral decomposition of A to evaluate the function or, in the case of structured A, the matrix $f(A)$ does not have the same structure.

So, it is good to keep some information specific to A in the starting matrix D. One of the possibilities is to create D as a direct sum of two smaller

submatrices D_1 and D_2, based on the matrix A, solve the corresponding eigenproblems of smaller size and then use the eigendecomposition of D to compute the eigendecomposition of A. This technique is known as a divide-and-conquer approach. Up till now many divide-and-conquer algorithms for certain classes of symmetric matrices have been developed, see, for example, [5, 6, 104, 110].

The procedure of constructing the matrices D_1 and D_2 depends on the structure of A, and one of the possible choices is a low-rank modification of A. On the one hand, $\|A - D\|$ is small, which has a strong effect on the smoothness of the eigenvalue curves, as follows from (5.32)-(5.33). On the other hand, the spectral decomposition of D can be found with fewer arithmetic operations than that of A, provided that matrices D_1 and D_2 keep the same type of structure as A itself.

Assume that the operation count for diagonalizing a matrix of order n is $w(n) = cn^k$. Assume that D_1 and D_2 are of order $n/2$, then exploiting this fact reduces the cost of diagonalization by a factor $\frac{w(n)}{2w(n/2)} = \frac{cn^k}{2cn^k/2^k} = 2^{k-1}$. This procedure may be repeated recursively with the matrices D_1 and D_2, leading thus to even more speed-up.

The operation count for updating the eigendecomposition of D to construct an eigendecomposition of A will also depend on the structure of A. Fortunately, for certain classes of structured matrices like tridiagonal ones and diagonal-plus-semiseparable ones it is possible to keep the operation count linear in n, which makes the approach viable.

The tridiagonal case is covered by works of Li and Li [104] and of Oettli [115]. The non-homotopy divide-and-conquer approach to diagonal-plus-semiseparable matrices is studied by Mastronardi, Van Barel and Van Camp [110] and the corresponding homotopy algorithm is discussed in Section 6.2. We will now present a general formulation of such an algorithm, following Oettli [114].

Eigenpath tracing

In Subsection 5.2.1 we introduced a general predictor-corrector method for homotopy problems. We will now adapt it to our specific problem of tracing an eigenpair $(\mathbf{x}(t), \lambda(t))$ of a symmetric matrix family $A(t)$. Differentiating $\mathbf{H}(\mathbf{x}(t), \lambda(t), t) = 0$ with respect to t gives

$$\begin{pmatrix} A(t) - \lambda(t)I \\ \mathbf{x}(t)^T \end{pmatrix} \mathbf{x}'(t) + \begin{pmatrix} -\mathbf{x}(t) \\ 0 \end{pmatrix} \lambda'(t) + \begin{pmatrix} A'(t)\mathbf{x}(t) \\ 0 \end{pmatrix} = 0. \tag{5.35}$$

Provided that $A(t)$ has only simple eigenvalues, the above linear system is nonsingular and can be solved explicitly for the derivatives $\mathbf{x}'(t)$ and $\lambda(t)$. As $A(t) = D + t(A - D)$, it follows that $A'(t) = (A - D)$. Plugging this into the solutions given by formulas (5.30)-(5.31), we get

$$\lambda_k'(t) = \mathbf{x}_l(t)^T (A - D)\mathbf{x}_k(t) \tag{5.36}$$

and

$$\mathbf{x}_k'(t) = \sum_{i \neq k} \frac{\mathbf{x}_i(t)^T (A - D)\mathbf{x}_k(t)}{(\lambda_k(t) - \lambda_i(t))} \mathbf{x}_i(t). \tag{5.37}$$

Given an eigenpair $(\mathbf{x}_k(t_i), \lambda_k(t_i))$, one may obtain a prediction of some $\lambda_k(t_{i+1})$ by interpolation or integration. An easy way is to use Euler's method:

$$\hat{\lambda}_k(t_{i+1}) = \lambda_k(t_i) + h\lambda_k'(t_i), \quad h = t_{i+1} - t_i. \tag{5.38}$$

To obtain a corresponding approximate eigenvector $\hat{\mathbf{x}}_k(t_{i+1})$ one may use inverse iteration with shift $\hat{\lambda}_k(t_{i+1})$:

$$\left(A(t_{i+1}) - \hat{\lambda}_k(t_{i+1})I\right)\mathbf{y}(t_{i+1}) = \mathbf{x}_k(t_i), \tag{5.39}$$

$$\hat{\mathbf{x}}_k(t_{i+1}) = \mathbf{y}(t_{i+1})/\|\mathbf{y}(t_{i+1})\|. \tag{5.40}$$

On this way no knowledge of all eigenvectors of $A(t_i)$ is required.

It was already mentioned that the eigenvalue curves can come very close. However, we wish to stay on a correct eigenpath while integrating only coarsely. This can be achieved by stabilizing the integration locally at t_{i+1}. A possible solution here is to apply Newton's method to the nonlinear system of equations

$$\mathbf{F}_t(\mathbf{x}, \lambda) = \begin{pmatrix} (A(t_{i+1}) - \lambda I)\mathbf{x} \\ (\mathbf{x}^T\mathbf{x} - 1)/2 \end{pmatrix} = 0, \tag{5.41}$$

using $(\hat{\mathbf{x}}_k(t_{i+1}), \hat{\lambda}_k(t_{i+1}))$ as starting value. If the stepsize $h = t_{i+1} - t_i$ has been chosen in a good way, the initial approximation should be good and convergence fast and to the correct eigenpair $(\mathbf{x}_k(t_{i+1}), \lambda_k(t_{i+1}))$.

As inspired by Huang and Li [87] and further investigated by Oettli [114], the Newton's method applied to (5.41) is essentially an inverse iteration with a variable shift:

$$\left(A(t_{i+1}) - \lambda^{(j)} I\right)\mathbf{y} = \mathbf{x}^{(j)} \tag{5.42}$$

with

$$\lambda^{(j+1)} = \lambda^{(j)} + \frac{1 + \mathbf{x}^{(j)T}\mathbf{x}^{(j)}}{2\mathbf{x}^{(j)T}\mathbf{y}}, \quad \mathbf{x}^{(j+1)} = (\lambda^{(j+1)} - \lambda^{(j)})\mathbf{y}. \tag{5.43}$$

The only remaining problem is to ensure that the method does not jump to another eigenpath nearby. Therefore some extra tests are necessary. This could be achieved, for example, by computing Sturm sequences of the matrix $A(t)$ and checking the number of the eigenpath.

In Section 6.2 we will give more details on a specific implementation of these general algorithms for diagonal-plus-semiseparable matrices.

Chapter 6

Structured matrices: algorithms

In this chapter we present three algorithms that solve problems for certain classes of structured matrices. An important tool to work out two of the problems is a homotopy approach, that we described in a previous chapter. We apply this approach to the inversion of an indefinite symmetric Toeplitz matrix in Section 6.1 and to computing the eigenvalues and eigenvectors of a symmetric diagonal-plus-semiseparable matrix in Section 6.2. In Section 6.3 we give a direct method for the solution of a banded block Toeplitz linear system with Toeplitz structure of the inner blocks. This chapter synthesizes the results of our papers [164, 34] and the report [35].

6.1 Continuation algorithm for Toeplitz systems

In this section we will combine different concepts and techniques like a continuation method, displacement ranks and iterative refinement, to devise a continuation algorithm for the inversion of Toeplitz matrices. This algorithm is later applied to solve systems of linear equations with Toeplitz matrices. We begin with necessary adaptations of Algorithm 4 to make it working with generator-represented Toeplitz matrices. Then we study its convergence

properties and give theoretical bounds on its complexity. Finally, we illustrate the algorithm with numerical experiments. Within this section we follow our work [164].

Basically, the algorithm can be applied to any low displacement rank matrices, allowing the fast reconstruction formulas of type (5.9) and having equations of type (5.13)-(5.14) for the generators of the inverse. The most obvious of more general cases are *Toeplitz-like* matrices, i.e., matrices that can be represented as (5.8) with G and H of size $n \times 2$. Required adaptations are not very complicated, so we speak further only about Toeplitz matrices, for the sake of simpler notation.

Within this section we use α as a continuation parameter instead of t to avoid possible collisions with Toeplitz matrix notation.

6.1.1 Constructing generators for a continuation matrix

Let A_0 and A_1 be Toeplitz matrices and consider again the continuation matrix function, defined by (5.25). Since a linear combination of Toeplitz matrices is again a Toeplitz matrix, it is generator representable. Thus, the matrix $M(\alpha)$ is a Toeplitz matrix and we can derive formulas for its generators.

Each $n \times n$ Toeplitz matrix T is determined by its first column p and first row q. These two vectors are easily combined into one $2n-1$ vector b:

$$b_i = p_{n-i+1}, \quad i = 1, \ldots, n,$$

$$b_i = q_{-n+i+1}, \quad i = n+1, \ldots, 2n-1.$$

Let us call this vector b a *base vector* of the Toeplitz matrix T and denote it by $b(T)$. It is obvious that

$$b(T_1 + T_2) = b(T_1) + b(T_2). \tag{6.1}$$

There is a straightforward connection between the base vector b of any Toeplitz matrix T and $\{Z_1, Z_{-1}^*\}$ generators of T (let us call them G and H; $G, H \in \mathbb{C}^{n\times 2}$ or $G, H \in \mathbb{R}^{n\times 2}$):

$$G_{j,1} = b_{n-j+1} + b_{n+n-j+1}, \quad j = 2, \ldots, n, \tag{6.2}$$

$$H_{j,2} = \overline{b}_{n+j-1} - \overline{b}_{j-1} \quad j = 2, \ldots, n, \tag{6.3}$$

$$G_{1,2} = 1, \quad H_{1,1} = 1, \tag{6.4}$$

all other entries in G and H are zero. (6.5)

On the basis of (6.1) and (6.2)-(6.5) the generators for $M(\alpha)$, defined by (5.25), are constructed easily for any α in two steps. Firstly, compute the new base vector $\hat{b} = b(M(\alpha))$. Secondly, compute the generators on the basis of the computed $\hat{b}$.

6.1.2 Formulation

We will adapt Algorithm 4 to work with Toeplitz matrices, represented by their displacement generators. To achieve this goal, we need to specify the following: a starting matrix, a method for enlarging a continuation parameter α and an iterative improvement process. We will now present such a specification.

Suppose that A_1 is the Toeplitz matrix to be inverted, A_0 is a readily-invertible starting Toeplitz matrix. Let us denote by $M_k = (1 - \alpha_k)A_0 + \alpha_k A_1$ the continuation matrix, then denote by G_k and H_k its $\{Z_1, Z_{-1}^*\}$-generators and by $\tilde{G}_k$, $\tilde{H}_k$ the $\{Z_{-1}^*, Z_1\}$-generators of the inverse M_k^{-1}.

Matrix A_0 should be chosen to simplify the first step of computing generators for M_0 and M_0^{-1}. In our particular implementation we choose the identity matrix as A_0.

For enlarging the continuation parameter α_k we use an adaptive algorithm: choose some initial increase, then test the convergence of the `IIP`. If it converges, we can enlarge the parameter α_k and go further, otherwise we should go back, decrease the parameter α_k and try again with smaller step $\alpha_k - \alpha_{k-1}$.

The use of a generator representation for Toeplitz matrices leads to the fact that the inverse is also represented by its generators. Equations (5.13)–(5.14) describe such generators. This makes it possible to avoid working with dense inverses by splitting the IIP method (5.18) into two IIP's, one per generator.

Algorithm 5 states the specific `IIP` for a general equation $Ax = b$:

Algorithm 5: Adapted version of an iterative refinement (used as IIP)

input : A – coefficient matrix, b – right-hand side, x_0 — initial approximation to the solution, Y_0 — approximation to A^{-1}
output: "good" approximation to $A^{-1}b$

begin
  $i = 0$
  while *approximation* x_i *is still not good enough* **do**
    $i = i + 1$
    compute $R_i = b - Ax_i$
    compute $z_i = Y_0 R_i$
    $x_{i+1} = x_i + z_i$
  end
  return x_{i+1}
end

Here x_0 is the initial approximation to the exact solution x, Y_0 is some approximation to A^{-1}. Further in Algorithm 6 we will use as Y_0 an approximation, created by a reconstruction formula of type (5.9) on the basis of the displacement generators, coming from the previous successful iteration of the continuation method. The order of `IIP` parameters in Algorithm 6 will correspond to the input of Algorithm 5.

The convergence of the iterative refinement depends only on the properties of this approximation Y_0 because

$$x - x_i = (I - Y_0 A)(x - x_{i-1}) = (I - Y_0 A)^i (x - x_0). \tag{6.6}$$

Algorithm 5 in the present form does not contain a stopping criterion. From formula (5.19) follows that the ratio $\|R_{i+1}\|/\|R_i\|$ would be always of order $1/\|R_0\|$. However, we cannot get more than machine precision, so after several iterations the ratio would be closer to one than to $1/\|R_0\|$. Thus as a stopping criterion we use a combination of two tests: an upper limit on the number of iterations and a closeness of a ratio $\|R_{i+1}\|/\|R_i\|$ to one. As soon as one of these tests is positive, the iterations stop.

Taking into account all these specifications and plugging them into Algorithm 4 yields the final version of a continuation algorithm for the inversion of Toeplitz

matrices.

Algorithm 6: Continuation algorithm for Toeplitz matrix inversion

input : A_0 – starting matrix, A_1 – original Toeplitz matrix
output : A_1^{-1}
notation: $M_k = (1-\alpha_k)A_0 + \alpha_k A_1$ – the continuation matrix,
G_k, H_k – displacement generators for M_k,
$\tilde{G}_k$, $\tilde{H}_k$ – displacement generators for M_k^{-1}

begin
  compute generators G_0, H_0, $\tilde{G}_0$, $\tilde{H}_0$
  $k = 0$; $\alpha_k = 0$
  while $\alpha_k < 1$ **do**
    $k = k + 1$; enlarge α_k such that $\alpha_k \leqslant 1$
    compute G_k and H_k using (6.2)–(6.5)
    $\tilde{G}_k$ = `IIP`$(\Delta^{-1}(G_k, H_k)$, G_k, $\tilde{G}_{k-1}$, $\Delta^{-1}(\tilde{G}_{k-1}, \tilde{H}_{k-1}))$
    $\tilde{H}_k$ = `IIP`$((\Delta^{-1}(G_k, H_k))^*, Z_{-1}^* H_k$, $\tilde{H}_{k-1}$, $\Delta^{-1}(\tilde{G}_{k-1}, \tilde{H}_{k-1}))$
    if *some of* `IIP`*'s diverged* **then**
      decrease α_k and repeat the iteration with new G_k and H_k
    end
  end
  return $\tilde{G}_k$, $\tilde{H}_k$
end

6.1.3 Convergence and complexity estimation

It is clear that the method being presented converges to the inverse of the original Toeplitz matrix T when the selected IIP converges on each continuation step. As it follows from (6.6) the convergence of the IIP is controlled by the convergence factor $\|R_0\|_2 = \|I - Y_0 A\|_2$. For the convergence of the IIP this positive factor should be less than one.

For the convergence of the IIP we should have nonsingular matrices M_k at every continuation step. Let us consider the eigenvalues of matrices M_k. Recall that we have chosen the identity matrix as A_0 and let us denote the eigenvalues of A_1 as $\hat{\lambda}_1, \ldots, \hat{\lambda}_n$. Then the eigenvalues $\mu_1, \mu_2, \ldots, \mu_n$ of M_k can be computed as

$$\mu_i = (1 - \alpha_k) + \alpha_k \cdot \hat{\lambda}_i. \tag{6.7}$$

It can be easily seen that when α_k would be near the value $\frac{1}{1-\hat{\lambda}_i}$, corresponding to some negative $\hat{\lambda}_i$, then `IIP` would be trying to work with an almost singular matrix.

There are several approaches to the solution of this problem. First, we can use standard symmetrization techniques [119] to reduce the inversion of a nonsingular matrix W to the Hermitian (real symmetric) case. We have

$$W^{-1} = (WW^*)^{-1}W = W^*(WW^*)^{-1},$$

W^*W and WW^* are Hermitian positive definite matrices. The condition number $\kappa(W)$ is squared in the transition to the matrices W^*W and WW^*, and this can lead to operations with more ill-conditioned matrices.

The displacement rank of a Toeplitz matrix W is roughly doubled by this symmetrization.

In the case of a symmetric but indefinite Toeplitz matrix T we can switch from the equation $Tx = b$ to the equation $(\mathbf{i}T)x = \mathbf{i}b$, where $\mathbf{i} = \sqrt{-1}$. The new matrix $T' = \mathbf{i}T$ would not have negative eigenvalues and thus the problems with nonsingularity would not appear. Since operations with complex numbers take approximately six times more flops than with real numbers, one should multiply the complexity estimate from the positive definite case approximately by six.

Thus in the further analysis we restrict ourselves to positive definite matrices.

Let us now derive a lower bound on the total number of continuation steps taking into account the convergence requirement from the beginning of this section.

Since the segment $[0, 1]$ is splitted into small pieces and on each of those we apply our IIP, we should estimate the sizes of these small steps. On one hand we should take the intermediate step $\Delta\alpha = \alpha_k - \alpha_{k-1}$ as large as possible to reduce the total number of continuation steps. On the other hand two neighbouring matrices $M(\alpha_k)$ and $M(\alpha_{k-1})$ should not differ too much to have an appropriate convergence factor.

Now we will switch back to the continous notation. Let us denote by R the *residual matrix* $R = I - M(\alpha + \Delta\alpha)M^{-1}(\alpha)$. Here $\Delta\alpha$ is the step size and $M^{-1}(\alpha)$ is the inverse coming from the previous continuation step. Let us denote the eigenvalues of the original Toeplitz matrix T by $\lambda_{T,i}$. Recalling that $M(\alpha) = I + \alpha(T - I)$ (see (5.25)), we get that the eigenvalues of $M(\alpha)$ are given by the formula

$$\lambda_{M,i} = 1 + \alpha(\lambda_{T,i} - 1). \tag{6.8}$$

Thus the eigenvalues $\lambda_{R,i}$ of the residual matrix R are represented by the formula

$$\lambda_{R,i} = 1 - \frac{1 + (\alpha + \Delta\alpha)(\lambda_{T,i} - 1)}{1 + \alpha(\lambda_{T,i} - 1)} = \frac{\Delta\alpha(1 - \lambda_{T,i})}{1 - \alpha(1 - \lambda_{T,i})}. \tag{6.9}$$

If $|\lambda_{R,i}| < 1$ then we have the convergence of our IIP (as well as of the other IIP's (5.15)–(5.17) – they are controlled by the same convergence factor, see (5.19)).

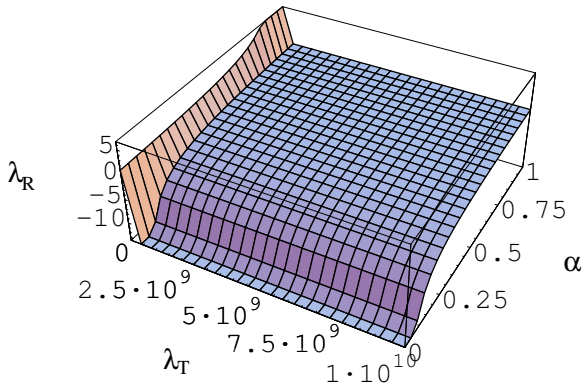


Figure 6.1: Surface plot of $f(\lambda_T, \alpha)$, $\lambda_T \in [0, 10^{10}]$, $\alpha \in [0, 1]$

Figures 6.1 and 6.2 show the surface plot of the function

$$f(\lambda_T, \alpha) = \frac{1 - \lambda_T}{1 - \alpha(1 - \lambda_T)} \tag{6.10}$$

for $\alpha \in [0, 1]$ and $\lambda_T \in [0, 10^{10}]$, $\lambda_T \in [0, 15]$, respectively. This function represents exactly all the possibilities for values of $\lambda_{R,i}$ without taking into account $\Delta\alpha$.

From the plots it is clear that $\max_i |\lambda_{R,i}|$ is controlled by the eigenvalue of T with maximum modulus when α is close to 0, and by the eigenvalue of T with minumum modulus when α is close to 1. In the further analysis we split the segment $[0, 1]$ into three parts: $[0, 0.1]$, $[0.1, 0.9]$ and $[0.9, 1]$ and perform the complexity analysis on each of these segments separately. In the case when the matrix does not have very large or very small eigenvalues one should extend the central segment to the left or to the right, respectively.

Suppose that $\alpha = 0$ and try to estimate $\Delta\alpha$ for the first step. Thus

$$\lambda_{R,i} = \Delta\alpha(1 - \lambda_{T,i})$$

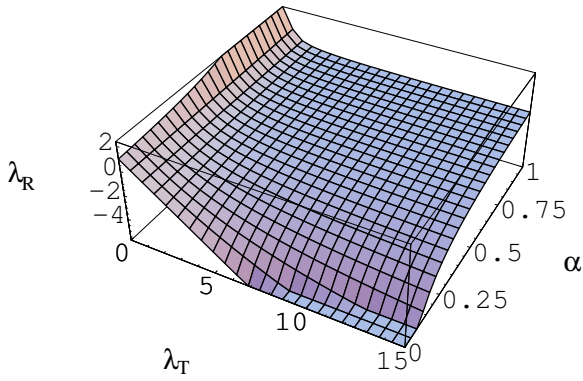


Figure 6.2: Surface plot of $f(\lambda_T, \alpha)$, $\lambda_T \in [0, 15]$, $\alpha \in [0; 1]$

It is obvious that $\lambda_{T,i} = \arg\max_i |\lambda_{R,i}|$ determines the step size. Suppose that $\max_i |\lambda_{T,i}| = 10^k \gg 1$. Then we should have

$$|\lambda_{R,i}| < \Delta\alpha|(1 - 10^k)| \approx \Delta\alpha \cdot 10^k.$$

Let us choose $\varepsilon = 10^c < 1$ as a convergence factor recalled in the beginning of this subsection. Finally we get from

$$\Delta\alpha \cdot 10^k = 10^c$$

that the initial step size should be of order 10^{c-k}.

Firstly, we would like to estimate the number of steps in the neighbourhood of $\alpha = 0$, say, in the segment $[0, 0.1]$. Suppose we have successfully performed all iterations till $\alpha = L$, $L < 0.1$, and the current step size is equal to $\Delta\alpha$. Let us determine when this $\Delta\alpha$ could be increased by factor F. Since convergence is controlled by the eigenvalues (6.9), such an increase is possible when the denominator in (6.9) would also increase by a factor F. Suppose we have to perform p continuation steps before this moment. Let us try to find p from the following equation:

$$F(1 - L(1 - 10^k)) = 1 - (L + p \cdot \Delta\alpha)(1 - 10^k). \tag{6.11}$$

Here the left side represents an old denominator multiplied by F and the right side represents the denominator in (6.9) after making p iterations with the step $\Delta\alpha$.

Let us simplify this equation:

$$F - FL + FL \cdot 10^k = 1 - L - p \cdot \Delta\alpha + L \cdot 10^k + p \cdot \Delta\alpha \cdot 10^k$$

We can throw away $-FL$ from the left side (it is small compared to F) as well as $-L$ and $-p \cdot \Delta\alpha$ from the right side (they are small compared to themselves multiplied by 10^k). This leads us to

$$\frac{(F-1)(1 + L \cdot 10^k)}{\Delta\alpha \cdot 10^k} \approx p.$$

Let us choose $F = 2$. This gives us

$$p \approx \frac{1 + L \cdot 10^k}{\Delta\alpha \cdot 10^k}. \tag{6.12}$$

Now we have to pass the segment $[0, D]$ with the bound $D = 0.1$, making small steps, increasing with the factor F. Let us find when α will leave this segment $[0, D]$.

Let us denote by p_i the number of iterations performed with the step $2^{i-1}\delta$, where $\delta = 10^{c-k}$ is the initial step. Now we want to determine the number of terms, i.e. $j + 1$ that should be taken in the sum

$$S = p_1 10^{c-k} + p_2 2 \cdot 10^{c-k} + \cdots + p_{j+1} 2^j \cdot 10^{c-k} \tag{6.13}$$

to exceed the bound $D = 0.1$.

We expand the expressions for p_i:

$$p_1 \approx \frac{1}{10^{c-k} \cdot 10^k} = \frac{1}{10^c}; \tag{6.14}$$

$$p_2 \approx \frac{1 + p_1 \cdot 10^{c-k} \cdot 10^k}{2 \cdot 10^{c-k} \cdot 10^k} = \frac{1 + p_1 \cdot 10^c}{2 \cdot 10^c} = \frac{1+1}{2 \cdot 10^c} = \frac{1}{10^c}; \tag{6.15}$$

$$p_3 \approx \frac{1 + p_1 \cdot 10^c + 2p_2 \cdot 10^c}{4 \cdot 10^c} = \frac{1+1+2}{4 \cdot 10^c} = \frac{1}{10^c} \tag{6.16}$$

$$p_i \approx \frac{1 + p_1 \cdot 10^c + \cdots + 2^{i-2} \cdot p_{i-1} \cdot 10^c}{2^{i-1} \cdot 10^c} = \frac{1 + 1 + 2 + 4 + \cdots + 2^{i-2}}{2^{i-1} \cdot 10^c} = \frac{1}{10^c}. \tag{6.17}$$

Taking into account (6.14)–(6.17) we have that S in (6.13) is approximated by

$$S \approx 10^{-k} \cdot (1 + 2 + \cdots + 2^j) = 10^{-k} \cdot (2^{j+1} - 1).$$

Thus

$$S > 10^{-k} \cdot 2^j > 0.1 = D.$$

The latter equation gives $j = \log_2 10 \cdot (k-1)$.

Finally, we need to estimate the sum

$$S' = p_1 + p_2 + \cdots + p_{j+1}$$

to get the total number of continuation steps. Since $p_i \approx \frac{1}{10^c}$ are almost constant, we easily get the following total upper bound on the number of continuation steps required to reach $\alpha = D = 0.1$:

$$S' \approx 10^{-c} \cdot (j+1) \approx 10^{-c} \cdot \log_2 10 \cdot (k-1). \tag{6.18}$$

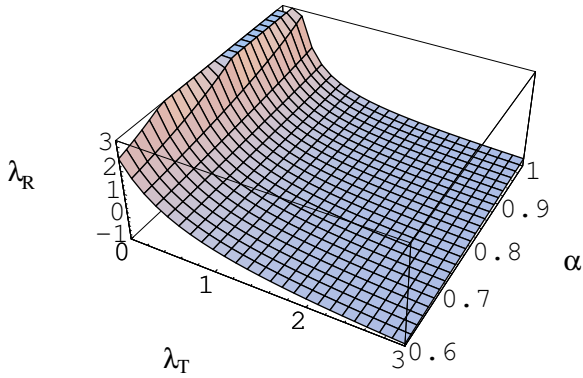


Figure 6.3: Surface plot of $f(\lambda_T, \alpha)$, $\lambda_T \in [0,3]$, $\alpha \in [0.6,1]$

A similar analysis could be performed when α is in the neighbourhood of 1 (say, $\alpha \in [0.9,1]$, see Figure 6.3) taking into account the eigenvalues with minimal modulus. The analysis yields the estimate $S'' \approx 10^{-c} \cdot \log_2 10 \cdot (m-1)$ on

the number of continuation steps. Here we assume that $\min_i |\lambda_{T,i}| = 10^{-m}$. As follows from figures 6.1–6.2, the interval $[0.1, 0.9]$ could be passed with a small constant number (say, s) of relatively big steps. From Figure 3 we can estimate s easily:

$$f(\lambda_T, \alpha) \cdot \Delta\alpha \approx 10^c,$$

taking into account that $f(\lambda_T, \alpha)$ is limited from above by 10 yields the following upper bound on $\Delta\alpha$:

$$\Delta\alpha \approx 10^{c-1}.$$

Thus

$$s \approx 0.8 \cdot 10^{1-c}.$$

This finally gives the estimate

$$N_{cont} < 4 \cdot 10^{-c}(k - 1 + m - 1) + s = 4 \cdot 10^{-c} \log_{10} \kappa(T) + s, \qquad (6.19)$$

where $\kappa(T)$ denotes the condition number of T, on the total number N_{cont} of continuation steps.

Since we keep the convergence factor less than or equal to 10^c, our selected IIP will converge with a fixed number $N_{iter-per-step}$ of iterations (in our specific implementation we use six iterations on the intermediate steps and fifteen iterations on the last steps to get full convergence). Each iteration of our new IIP (Algorithm 5) consists of two $\mathcal{O}(n)$ additions and two matrix-by-vector multiplications. Since we use (5.9) for these multiplications, it takes only $N_{step} = \mathcal{O}(n \log n)$ operations (this follows from the complexity of FFT and its relation to our $M \times v$ multiplication, as shown in Theorem 2 and the discussion thereafter). So the total number of operations in our method is

$$N_{op} = N_{cont} \cdot N_{iter-per-step} \cdot N_{step} =$$
$$\mathcal{O}(\log_{10} \kappa(T)) \cdot \mathcal{O}(1) \cdot \mathcal{O}(n \log n) = \mathcal{O}(\log_{10} \kappa(T) \cdot n \log n). \qquad (6.20)$$

6.1.4 Numerical Experiments

The method was implemented in Matlab ver. 7.2.0.294 on a workstation (2Gb RAM, Intel Core 2 Duo E6400 processor) running Debian Linux (kernel 2.6.18-SMP).

In formula (6.20) we have to optimize the product of the first two factors $P = N_{cont} \cdot N_{iter-per-step} = \mathcal{O}(\log_{10} \kappa(T)) \cdot \mathcal{O}(1)$. The last one $N_{step} = \mathcal{O}(n \log n)$ is strictly defined by the choice of the IIP. Since the IIP consists only of two matrix-by-vector multiplications and several vector additions, where matrices are Toeplitz, we use existing and well-developed methods like FFTW for a good implementation of the IIP (see [93, 63]).

The size of P is controlled by the following equilibrium. We can decrease the number of iterations in each continuation step thus having more crude approximations to the intermediate inverses. This leads to the increase of the total number of continuation steps. Doing more iterations within one continuation step we get better approximation for the inverses, thus getting less continuation steps.

The choice between these variants at the present time is empirical and we base it on the extensive computer testing of the algorithm. (We should mention here that authors of other continuation algorithms (see e.g. [120]) also solve this problem empirically.) Good theoretical estimation may become a subject of future research.

We have chosen 6 iterations for the IIP in the middle part of the segment $[0, 1]$ and 15 iterations in the critical zones near the ends of the segment (see the theoretical investigation leading to (6.18)). Another empirical parameter is the modified convergence factor $\hat{c}$: we say that the IIP in Algorithm 3 diverges if the size of the last correction z_i (see Algorithm 5) exceeds $\hat{c}$. This parameter $\hat{c}$ is closely related to the convergence factor (5.19) by means of Theorem 4.1 in [40]. This theorem gives an estimate on the norm of a full matrix T when its generators are given. However in the present implementation we do not use this theorem explicitly.

Finally, it was dicovered that factor F defined near (6.11) has some influence on the number of continuation steps. Setting $F = 2$, we observe a very slow acceleration and slowdown near the critical points $\alpha = 0$ and $\alpha = 1$. Setting $F = 8$ leads to numerous unjustified increases in the step size (which means that an increase of a current step size results in the divergence of the IIP). The best choice turned out to be $F = 4$. The difference in complexity for different F is less than 7%.

Another parameter that influences the speed is a good prescaling of the matrix. The main idea of the prescaling should be to make a balance between very small and very large eigenvalues of T, say, spread values of continuation parameter α corresponding to them in almost equal portions between the segments $[0, 0.1]$ and $[0.9, 1]$. This makes possible to get rid of very small steps $\Delta\alpha$.

Speaking about the results of numerical experiments we would like to mention

order/$\kappa(T)$	10^2	10^4	10^8
10^3	47	59	90
10^4	50	55	81
10^5	53	61	84

Table 6.1: Average number of continuation steps depending on size and condition number

Number of continuation steps			
$\kappa(T)$	$\hat{c} = 1 \cdot 10^{-4}$	$\hat{c} = 1 \cdot 10^{-2}$	$\hat{c} = 1 \cdot 10^{0}$
10^2	66	55	47
10^4	90	68	59
10^8	115	105	90

Table 6.2: Influence of a modified convergence factor on N_{cont}

that the number of continuation steps N_{cont} depends only on the condition number but not on the order of the matrix. In Table 6.1 we present the rounded average number of N_{cont} among the ten runs of the program for each cell.

The average value of small parameter s (6.19) among all these runs is 7.

In the second table we show the influence of $\hat{c}$ to the value N_{cont} for matrices of order 10^3. Increasing this value reduces N_{cont} but may give divergent procedures for large $\kappa(T)$.

To obtain Toeplitz matrices with various condition numbers, we take a random base vector (6.1) and scale its central components, reducing them and thus increasing the condition number. The components corresponding to the first row of a Toeplitz matrix T are modified according to the formulas

$$v_i = \frac{v_i}{(n-i+1)^l}, \quad i = 1, \ldots, n,$$

where n is the order of T and the parameter l varies from 0 to 8. Since we only deal with symmetric matrices, our Toeplitz matrix is completely determined by the set v_i, $i = 1, \ldots, n$. The Toeplitz matrix obtained in this way is almost always not positive definite.

To test the accuracy of the computed inverse we applied it to solve a linear system with T as a coefficient matrix and a random right-hand side. In all the cases described above we were able to achieve a relative error of the solution of order $\varepsilon_{mach} \cdot \kappa(T)$, ε_{mach} denoting the machine precision. So, the amount of

continuation steps presented in Tables 6.1 and 6.2 was sufficient to reach this precision.

Since most of the test matrices were not positive definite, to solve the linear system with T as the coefficient matrix, we used the transformation $T \to \mathbf{i}T$, as it was discussed at the beginning of Subsection 6.1.3.

6.2 Homotopy method applied to diagonal-plus-semiseparable eigenvalue problems

In this section we derive a divide-and-conquer algorithm to compute the eigenvalues and eigenvectors of a symmetric generator-representable diagonal-plus-semiseparable matrix. Computing the eigendecomposition of such a matrix is reduced first to computing the spectral decomposition of two smaller submatrices of the same structure (divide step). These decompositions are then joined (conquer step). The conquer step is performed by solving a certain diagonal plus rank-one eigenvalue problem with a homotopy method.

Subsection 6.2.2 covers the reduction to similar problems of smaller size, and also illustrates how a diagonal plus rank-one eigenproblem appears in the context. The idea comes from the work [110]. Subsection 6.2.3 describes the homotopy part of the method and follows our work [35]. The last Subsection 6.2.5 presents the results of several numerical experiments.

6.2.1 Preliminaries

In Subsection 5.2.3 we presented the general idea of a predictor-corrector path tracing method for equation (5.35). An eigenpath $(\mathbf{x}_k(t), \lambda_k(t))$ was considered as a solution curve of this ordinary differential equation, and the initial point was given by a known eigenpair of the starting matrix at $t = 0$.

Such a predictor-corrector method goes back to Li and Rhee [105]. It allows to track an eigenpair $(\mathbf{x}(t), \lambda(t))$ of a matrix family $A(t)$ (5.28) from $t = 0$ to $t = 1$ and may be sketched as follows (t_i is the value of the continuation parameter t

at iteration number i):

Algorithm 7: Basic predictor-corrector method

input : known eigendecomposition of some starting matrix D
output: eigendecomposition of the target matrix A

```
begin
    while t_i < 1 do
        predict eigenpair at t_i
        correct prediction
        check for path jumping
        select next stepsize
    end
end
```

Li and Rhee applied their algorithm to tridiagonal matrices. Their numerical results in [105] show that the method works well for matrices with well-separated eigenvalues, but it is very inefficient for close eigenvalues and may even fail. The orthogonality of the eigenvectors is also badly affected, cf. (5.31).

As a remedy, Li, Zhang and Sun [106] suggested tracing an invariant subspace, corresponding to a cluster of very close eigenvalues. This invariant subspace is well conditioned if the cluster is well separated from the rest of the spectrum [122]. This leads to substantial improvements, but the Rayleigh-Ritz procedure used to follow the subspace is very expensive if the size of the cluster is comparable with the size of the matrix.

Oettli [115] proposed an extensive use of deflation techniques instead of subspace iterations. These techniques are well-known from the theory of divide-and-conquer methods, see Cuppen [42]. Deflation dramatically reduces the size of clusters of close eigenvalues and speeds up the convergence for the remaining eigenvalues.

Starting with the method of Li and Rhee, we will construct a divide-and-conquer homotopy method for generator-representable diagonal-plus-semiseparable matrices. This restriction to the class of symmetric generator-representable D+SS matrices does not worsen the stability of the algorithm, as proved by Van Camp [165]. It has been shown by numerical experiments that when the matrix elements have small relative errors, the generator representation gives very accurate results for divide and conquer algorithms.

The algorithm presented can also be applied to a more general class of symmetric D+SS matrices, defined in Subsection 5.1.1. As proven in [174, Ch. 2], a symmetric D+SS matrix can always be written as a block-diagonal

matrix whose blocks are symmetric generator-representable D+SS matrices. Hence, two cases can occur: either the original D+SS matrix has zero-blocks and then its eigenproblem can be split up into smaller eigenproblems of generator-representable D+SS matrices, either the whole original symmetric D+SS matrix is generator-representable. In this sense, symmetric generator-representable D+SS matrices are an analogue of irreducible tridiagonal matrices.

In the next subsections we will gradually build up the required components for the method, presented in its final appearance as Algorithm 10. Following the general theory of homotopy methods from Subsection 5.2.3, we have to choose starting matrices and describe the path tracing method. To construct starting matrices that will lead to smooth eigenvalue curves, we introduce in Subsection 6.2.2 a divide-and-conquer approach. Subsection 6.2.3 shows how this approach could be plugged in a homotopy and also discusses some difficulties arising during path tracing and workarounds for them. Finally, the algorithm is illustrated with numerical experiments in Subsection 6.2.5.

6.2.2 Divide-and-conquer for D+SS matrices

We design here an algorithm and give a theorem in order to split the original matrix in two submatrices, keeping the diagonal-plus-semiseparable structure. Provided that eigendecompositions of these smaller matrices are given, a certain rank-one modification should be processed to get the eigendecomposition of the original matrix.

The divide step is based on Givens rotations. These rotations are simultaneously applied to the top-left and the bottom-right corners of the matrix in order to annihilate elements in the first rows and columns, respectively in the last rows and columns.

The conquer step constitutes of computing the spectral decomposition of a diagonal matrix plus a rank-one modification. Its connection to the original problem is straightforward.

Divide step

Let A be the diagonal-plus-semiseparable matrix given by (5.2). We will define first two sequences of Givens rotations G_k and H_l of a special form, the first sequence is to be applied to the top-left corner of the starting matrix and the second sequence to the bottom-right corner.

Recall that N is the order of the matrix A, then let $K = \lceil \frac{N}{2} \rceil - 1$, $L = K + 3$, $\tilde{u}_1 = u_1$, $\tilde{v}_N = v_N$. For every $k = 1, \ldots, K$ we define a number $\tilde{u}_{k+1}$ and a Givens rotation G_k as

$$G_k = \begin{pmatrix} c_k & -s_k \\ s_k & c_k \end{pmatrix}, \text{ where } c_k = \frac{u_{k+1}}{\sqrt{\tilde{u}_k^2 + u_{k+1}^2}}, s_k = \frac{\tilde{u}_k}{\sqrt{\tilde{u}_k^2 + u_{k+1}^2}}, \quad (6.21)$$

such that

$$\begin{pmatrix} 0 \\ \tilde{u}_{k+1} \end{pmatrix} = \begin{pmatrix} 0 \\ \sqrt{\tilde{u}_k^2 + u_{k+1}^2} \end{pmatrix} = \begin{pmatrix} c_k & -s_k \\ s_k & c_k \end{pmatrix} \begin{pmatrix} \tilde{u}_k \\ u_{k+1} \end{pmatrix}. \quad (6.22)$$

Similarly, for every $l = N, \ldots, L$ we define a number $\tilde{v}_{l-1}$ and a Givens rotation H_l as

$$H_l = \begin{pmatrix} c_l & s_l \\ -s_l & c_l \end{pmatrix}, \text{ where } c_l = \frac{v_{l-1}}{\sqrt{\tilde{v}_l^2 + v_{l-1}^2}}, s_l = \frac{\tilde{v}_l}{\sqrt{\tilde{v}_l^2 + v_{l-1}^2}} \quad (6.23)$$

such that

$$\begin{pmatrix} \tilde{v}_{l-1} \\ 0 \end{pmatrix} = \begin{pmatrix} \sqrt{\tilde{v}_l^2 + v_{l-1}^2} \\ 0 \end{pmatrix} = \begin{pmatrix} c_l & s_l \\ -s_l & c_l \end{pmatrix} \begin{pmatrix} v_{l-1} \\ \tilde{v}_l \end{pmatrix}. \quad (6.24)$$

These rotations allow us to formulate Algorithm 8, which divides a symmetric diagonal-plus-semiseparable matrix into two submatrices of the same D+SS structure and of about half the dimension, and some additional structure.

Algorithm 8: Divide step for D+SS matrices

input : A – starting matrix
output: $A^{(K+1)}$ – reduced matrix

begin
  $A^{(1)} = A$
  for $k = 1 : \lceil \frac{N+1}{2} \rceil - 2$ **do**
    $\tilde{G}_k = \mathrm{diag}(I_{k-1}, G_k, I_{N-2k-2}, H_{N-k+1}, I_{k-1})$
    $A^{(k+1)} = \tilde{G}_k A^{(k)} \tilde{G}_k^T$
  end
  if N *is odd* **then**
    $k = \lceil \frac{N+1}{2} \rceil - 1$
    $\tilde{G}_k = \mathrm{diag}(I_{k-1}, G_k, I_{N-k-1})$
    $A^{(k+1)} = \tilde{G}_k A^{(k)} \tilde{G}_k^T$
  end
end

Mastronardi et al. [110] have proven the following theorem.

Theorem 10. *Let A be the diagonal-plus-semiseparable matrix defined in (5.2). Algorithm 8 transforms the matrix A into $A^{(K+1)}$ having the following structure:*

$$\begin{array}{c} \\ K+1 \\ N-L+2 \end{array} \begin{array}{c} \begin{array}{cc} K+1 & N-L+2 \end{array} \\ \left(\begin{array}{c|c} C_1^{(K+1)} & \alpha^{(K+1)}\mathbf{e}_{K+1}\mathbf{e}_1^T \\ \hline \alpha^{(K+1)}\mathbf{e}_1\mathbf{e}_{K+1}^T & C_2^{(K+1)} \end{array}\right) \end{array} \tag{6.25}$$

where $C_1^{(K+1)}$ and $C_2^{(K+1)}$ are diagonal-plus-semiseparable matrices, $\mathbf{e}_{K+1}$ the $(K+1)$-th vector of the canonical basis of $\mathbb{R}^{K+1}$, $\mathbf{e}_1$ the first vector of the canonical basis of $\mathbb{R}^{N-L+2}$ and $\alpha^{(K+1)} = A_{K+1,K+2}^{(K+1)}$.

We would like to represent the matrix $A^{(K+1)}$ as a sum of a block-diagonal matrix $D_1 \oplus D_2$, where D_1 and D_2 are diagonal-plus-semiseparable, and a rank-one matrix. To create this rank-one modification one extra operation needs to be performed on (6.25).

Define $G^{(K)} = \tilde{G}_K \dots \tilde{G}_2 \tilde{G}_1$, where $\tilde{G}_i$ are coming from Algorithm 8. Then

$$\begin{aligned} A &= G^{(K)T} A^{(K+1)} G^{(K)} \\ &= G^{(K)T} \left(\begin{array}{c|c} C_1^{(K+1)} & \alpha^{(K+1)}\mathbf{e}_{K+1}\mathbf{e}_1^T \\ \hline \alpha^{(K+1)}\mathbf{e}_1\mathbf{e}_{K+1}^T & C_2^{(K+1)} \end{array}\right) G^{(K)} \\ &= G^{(K)T} \left(\left(\begin{array}{c|c} C_1^{(K+1)} - \alpha^{(K+1)}\mathbf{e}_{K+1}\mathbf{e}_{K+1}^T & 0 \\ \hline 0 & C_2^{(K+1)} - \alpha^{(K+1)}\mathbf{e}_1\mathbf{e}_1^T \end{array}\right) \right. \\ &\qquad \left. + \alpha^{(K+1)} \left(\begin{array}{c|c} \mathbf{e}_{K+1}\mathbf{e}_{K+1}^T & \mathbf{e}_{K+1}\mathbf{e}_1^T \\ \hline \mathbf{e}_1\mathbf{e}_{K+1}^T & \mathbf{e}_1\mathbf{e}_1^T \end{array}\right) \right) G^{(K)}. \end{aligned}$$

The subtraction of the element $\alpha^{(K+1)}$ from the last diagonal element of $C_1^{(K+1)}$ and from the first diagonal element of $C_2^{(K+1)}$ creates the desired rank-one modification and does not affect the diagonal-plus-semiseparable structure. Thus we may define diagonal-plus-semiseparable matrices $D_1 = C_1^{(K+1)} - \alpha^{(K+1)}\mathbf{e}_{K+1}\mathbf{e}_{K+1}^T$ and $D_2 = C_2^{(K+1)} - \alpha^{(K+1)}\mathbf{e}_1\mathbf{e}_1^T$. The original problem of computing the eigendecomposition of A is now split into similar problems for the matrices D_1 and D_2. To unite the solutions of these smaller subproblems, we need now to work out this rank-one modification.

Conquer step

In what follows we omit the superscript $(K+1)$ at α. Suppose that eigendecompositions of D_1 and D_2 are computed:

$$D_1 = Q_1 \Delta_1 Q_1^T, \tag{6.26}$$

$$D_2 = Q_2 \Delta_2 Q_2^T. \tag{6.27}$$

In order to know the eigendecomposition of the original matrix A it is enough to calculate the spectral decomposition of a diagonal matrix plus a rank-one modification and perform some orthogonal transformations, as we will show now.

The matrix A can be transformed into:

$$\begin{aligned} A &= {G^{(K)}}^T \begin{pmatrix} Q_1 & \\ & Q_2 \end{pmatrix} \\ &\quad \times \left[\begin{pmatrix} \Delta_1 & \\ & \Delta_2 \end{pmatrix} + \alpha \begin{pmatrix} Q_1^T \mathbf{e}_{K+1} \\ Q_2^T \mathbf{e}_1 \end{pmatrix} \begin{pmatrix} \mathbf{e}_{K+1}^T Q_1 & \mathbf{e}_1^T Q_2 \end{pmatrix} \right] \\ &\quad \times \begin{pmatrix} Q_1 & \\ & Q_2 \end{pmatrix}^T G^{(K)} \\ &= {G^{(K)}}^T \begin{pmatrix} Q_1 & \\ & Q_2 \end{pmatrix} \left[\begin{pmatrix} \Delta_1 & \\ & \Delta_2 \end{pmatrix} + \alpha \mathbf{y}\mathbf{y}^T \right] \begin{pmatrix} Q_1 & \\ & Q_2 \end{pmatrix}^T G^{(K)} \end{aligned} \tag{6.28}$$

with

$$\mathbf{y} = \begin{pmatrix} Q_1^T \mathbf{e}_{K+1} \\ Q_2^T \mathbf{e}_1 \end{pmatrix}. \tag{6.29}$$

Hence, the eigenproblem of A is reduced to computing the eigendecomposition of a rank-one modification of a diagonal matrix

$$\begin{pmatrix} \Delta_1 & \\ & \Delta_2 \end{pmatrix} + \alpha \mathbf{y}\mathbf{y}^T. \tag{6.30}$$

This latter problem can be solved by a continuation method, as described in the next subsection.

One may also apply the continuation method to step directly from the eigendecomposition of the matrix $D = D_1 \oplus D_2$ to the one of $A^{(K+1)}$. However, in this way it becomes impossible to deflate certain eigenpairs without

disrupting the D+SS structure of the matrices or blowing up the computational cost. As our numerical experiments have shown, the deflation is essential for stability and accuracy of the method. Compared to such a straightforward continuation, the main drawback of the diagonal plus rank-one approach is an additional transformation of the eigenvectors of the matrix (6.30).

6.2.3 Homotopy within divide and-conquer

Rank-one modification as a starting matrix

As shown in the previous subsection, a symmetric diagonal-plus-semiseparable matrix A is split into two independent submatrices by a rank-one modification and some orthogonal transformation:

$$A = {G^{(K)}}^T A^{(K+1)} G^{(K)} = {G^{(K)}}^T Q(\Delta + \alpha \mathbf{y}\mathbf{y}^T) Q^T G^{(K)},$$

where $\Delta = \Delta_1 \oplus \Delta_2$, $Q = Q_1 \oplus Q_2$, $\mathbf{y} = Q\mathbf{v}$, where $\mathbf{v}^T = (\mathbf{e}_{K+1}^T, \mathbf{e}_1^T)$. Let us define the matrix family

$$A(t) = \Delta + t\alpha \mathbf{y}\mathbf{y}^T \tag{6.31}$$

and analyse how its eigenvalue curves $\lambda_i(t)$ are bounded depending on α and t. Analogously to [115], we can formulate and prove the following theorem:

Theorem 11. *Let Δ be a diagonal matrix and consider the continuation function (6.31). Then all the eigenvalue curves $\lambda_i(t)$ of $A(t)$ are monotonically increasing or decreasing with t, depending on the sign of α.*

Proof. Suppose $\alpha > 0$. Matrices $A(t)$ for different t's differ by a rank-one matrix: □

$$A(t_2) = A(t_1) + (t_2 - t_1)\alpha \mathbf{y}\mathbf{y}^T, \quad (0 \leqslant t_1 \leqslant t_2 \leqslant 1).$$

Because $\alpha > 0$, the difference is a positive semidefinite matrix of rank 1. If we number the eigenvalues $\lambda_i(t)$ in increasing order, then the following relations hold [179, p. 97]:

$$\lambda_i(t_2) - \lambda_i(t_1) = m_i(t_2 - t_1) \cdot 2\alpha, \qquad \text{where } 0 \leqslant m_i \leqslant 1, \sum_{i=1}^{n} m_i = 1.$$

Summation over i yields

$$\sum_{i=1}^{n} (\lambda_i(t_2) - \lambda_i(t_1)) = 2(t_2 - t_1)\alpha.$$

Hence when $(t_2-t_1)\alpha\mathbf{y}\mathbf{y}^T$ is added to $A(t_1)$, all eigenvalues of $A(t_1)$ are shifted by an amount not larger than $(t_2 - t_1) \cdot 2\alpha$. This means that all eigenvalues increase monotonically with t:

$$0 \leqslant \lambda_i(t_2) - \lambda_i(t_1) \leqslant 2 \cdot (t_2 - t_1)\alpha, \quad i = 1, \ldots, n.$$

The case of $\alpha < 0$ could be reduced to the already considered one by premultiplying matrix family (6.31) with -1. So in what follows we always assume that $\alpha > 0$. □

As shown above, the difference matrix $A(t_2)-A(t_1)$ $(0 \leqslant t_1 \leqslant t_2 \leqslant 1)$ is positive semidefinite, so by the interlacing theorem ([86, Ch. 4.3]) the eigenvalues of $A(t_1)$ and $A(t_2)$ interlace:

$$\lambda_1(t_1) \leqslant \lambda_1(t_2) \leqslant \lambda_2(t_1) \leqslant \lambda_2(t_2) \leqslant \cdots \leqslant \lambda_n(t_1) \leqslant \lambda_n(t_2), \quad t_1 \leqslant t_2. \tag{6.32}$$

If the matrix $A(1) = \Delta + \alpha\mathbf{y}\mathbf{y}^T$ has (almost) equal eigenvalues, they will be deflated, as shown in the next part of the theory. So, we may assume that the eigenvalues of $A(1)$ are mutually distinct. Oettli [114] extended the result of Parlett [122, Ch. 7] and has proven that if the eigenvalues of $A(1)$ are distinct, then the continuation matrices $A(t)$ defined by (6.31) have no multiple eigenvalues except possibly for $t = 0$. However, their eigenvalues may coincide up to the working precision.

Summarizing the facts given above, we conclude that if $\lambda_k(0)$ is the k-th smallest eigenvalue of D, then $\lambda_k(t)$ is also the k-th smallest eigenvalue of $A(t)$ for $t = (0, 1]$. Li and Rhee [105] call it the *order preserving property*. It is useful if only selected eigenvalues of a matrix are required. This property also allows parallel computation of the eigenpairs.

Deflation

Numerical experiments show that some eigenpairs $(\mathbf{q}_i, \delta_i)$ of D are often good approximations of eigenpairs of $A(1)$, if D is chosen as described above. The corresponding eigenpaths $\lambda_i(t)$ are almost straight lines. Such eigenpairs of $A(1)$ may be determined with little effort. The process of identifying such eigenpairs and eliminating them from the remaining problem is called *deflation*. It reduces the cost of solving the remaining problem and improves the stability as well as the accuracy of the algorithm, especially in the presence of close eigenvalues.

Our splitting method is close to the Cuppen-type divide-and-conquer algorithms [42, 25], so the deflation techniques are similar. It remains to answer when an approximate eigenpair $(\mathbf{q}_i, \delta_i)$ can be accepted as an eigenpair of $A(1)$ in finite precision machine arithmetic.

Since the spectral decomposition $D = Q\Delta Q^T$ is known, it was shown in the previous Subsection that matrices $A(t)$ represent a rank-one modification of a diagonal matrix (6.28):

$$A(t) = \Delta + t\alpha \mathbf{y}\mathbf{y}^T.$$

Cuppen [42] used this representation in his divide-and-conquer algorithm with $t = 1$ and Oettli [115] applied it with varying t for tridiagonal eigenproblems.

Cuppen has shown that zero components of $\mathbf{y}$ reveal the eigenvalues to be deflated. Zero components of $\mathbf{y}$ may correspond to two different situations. First, they may correspond to good approximations δ_i to the eigenvalues of $A^{(K+1)}$. Second, they may correspond to clusters of m almost equal eigenvalues, of which $m-1$ could be deflated. We refer here to his work [42] for details.

As a measure for being "close to zero enough to be deflated" for the components of $\mathbf{y}$ we choose some tolerance η depending on the machine epsilon ε and some norm $\|A\|$.

To perform deflation, the zero components of the vector $\mathbf{y}$ are permuted to the end of the matrix

$$A(t) = P \begin{pmatrix} \Delta_a + t\alpha \mathbf{y}_a \mathbf{y}_a^T & 0 \\ 0 & \Delta_b \end{pmatrix} P^T. \tag{6.33}$$

Such a permutation results in an irreducible problem of smaller size,

$$B(t) = \Delta_a + t\alpha \mathbf{y}_a \mathbf{y}_a^T, \tag{6.34}$$

for which the remaining eigenpairs can be computed by path tracing.

Say that ZHZ^T is the spectral decomposition of $B(1)$. Then, the desired eigenvalues of $A(1)$ are the diagonal elements of $\Lambda = H \oplus \Delta_b$ and the eigenvector matrix is $X = P^T[Z, I_b]P$.

Path tracing

We will describe now the path tracing algorithm for the matrix family $B(t)$, defined according to (6.34). Starting with the j-th eigenpair $(\mathbf{e}_j, \delta_j) =$

$(\mathbf{x}_j(0), \lambda_j(0))$ of Δ_a, Algorithm 9 traces the continuous eigenpath using the predictor-corrector method, applied to (5.35), see also the discussion thereafter.

Algorithm 9: Eigenpath tracing

input : $B(t)$ – continuation matrix, $(\mathbf{x}, \lambda)$ – k-th eigenpair of $B(0)$
output: $(\mathbf{x}, \lambda)$ – k-th eigenpair of $B(1)$

```
begin
    t = 0, h = 1
    while t < 1 do
        (x, λ) = predict (x, λ, t + h)
        (x, λ, success) = correct (x, λ, t + h)
        if success then
            t = t + h, h = min(2h, 1 − t)
        else
            h = h/2
            if h < ε then
                failure
            end
        end
    end
end
```

The $(i+1)$-th step consists of a prediction and a correction. For prediction at $t = 0$ we use Euler's method, as described in Subsection 5.2.3: some $\lambda_j(t_{i+1})$ is first approximated by interpolation through $\lambda_j(0)$ and its derivative $\lambda_j'(0)$:

$$\hat{\lambda}_j(h) = \lambda_j(0) + h\lambda_j'(0). \tag{6.35}$$

Otherwise, the eigenvalue and the derivative are known for two different t, allowing cubic interpolation. Using a Hermite interpolation scheme and the following substitutions

$$h = t_i - t_{i-1}, \quad t_n = (t_{i+1} - t_{i-1})/h,$$

$$y_1 = \lambda_j(t_{i-1}), \quad y_1' = \lambda_1'(t_{i-1}), \quad y_2 = \lambda_j(t_i), \quad y_2' = \lambda_j'(t_i)$$

the approximation is

$$\hat{\lambda}_j(t_{i+1}) = y_1 + (hy_1' + (y_2 - hy_1' - y_1 + (2y_1 - 2y_2 + hy_1' + hy_2')(t_n - 1))t_n)t_n.$$

Because of the special structure of the matrix family $B(t)$ the derivative $\lambda_j(t)$ is computed as

$$\lambda_j'(t) = \mathbf{x}_j(t)^T \left(B(1) - B(0)\right) \mathbf{x}_j(t)^T = \alpha \left(\mathbf{x}_j(t)^T \mathbf{y}_a\right)^2.$$

One step of inverse iteration with shift $\hat{\lambda}_j(t_{i+1})$ is used to obtain a prediction for the corresponding approximate eigenvector $\hat{\mathbf{x}}_k(t_{i+1})$. Such step of the iteration process consists of solving for $\mathbf{z}$ of

$$\left(H + \alpha \mathbf{y}_a \mathbf{y}_a^T\right) \mathbf{z} = \mathbf{x}, \tag{6.36}$$

where $H = \Delta_a - \sigma I$, σ is the spectral shift. Because of the special structure of the matrix, we can give an explicit expression for the solution $\mathbf{z}$.

First, solve (6.36) for $\mathbf{z}$ and subsitute $\beta = \mathbf{y}_a^T \mathbf{z}$:

$$\mathbf{z} = H^{-1}(\mathbf{x} - \alpha\beta \mathbf{y}_a). \tag{6.37}$$

Now determine β by substituting $\mathbf{z}$ in (6.36):

$$\beta = \left(\mathbf{y}_a^T H^{-1} \mathbf{x}\right) / \left(1 + \alpha \mathbf{y}_a^T H^{-1} \mathbf{y}_a\right). \tag{6.38}$$

Finally plug the expression (6.38) for β into (6.37), obtaining

$$\mathbf{z} = H^{-1} \left(\mathbf{x} - \alpha \left(\frac{\mathbf{y}_a^T H^{-1} \mathbf{x}}{1 + \alpha \mathbf{y}_a^T H^{-1} \mathbf{y}_a} \right) \mathbf{y}_a \right).$$

The correction step makes an extensive use of the interlacing property (6.32). Thanks to it an interval

$$I_j = \begin{cases} [\lambda_j(0), \lambda_{j+1}(0)], & j < m \\ [\lambda_j(0), \lambda_j(0) + 2\alpha], & j = m \end{cases} \tag{6.39}$$

is known, which contains the eigenvalue curve $\lambda_j(t)$ for $t \in [0, 1]$. Since $\alpha > 0$, all the eigenvalue curves monotonically nondecrease. This information allows to detect path jumping or to use bisection followed by inverse iteration to compute the corresponding eigenpair of $B(1)$ directly in case that path tracing fails.

The correction step is done by Rayleigh quotient iteration (further denoted as RQI). To be more precise, at $\left(\mathbf{x}^{(k-1)}(t_i), \lambda^{(k-1)}(t_i)\right)$, $k \geqslant 1$ we let

$$\lambda^{(k)}(t_i) = \mathbf{x}^{(k-1)}(t_i)^T B(t_i) \mathbf{x}^{(k-1)}(t_i),$$

and then solve

$$\left(B(t_i) - \lambda^{(k)}(t_i) I\right) \mathbf{y}^{(k)}(t_i) = \mathbf{x}^{(k-1)}(t_i)$$

and let

$$\mathbf{x}^{(k)}(t_i) = \mathbf{y}^{(k)}(t_i) / \|\mathbf{y}^{(k)}(t_i)\|.$$

The starting vector $\mathbf{x}^{(0)}(t_i)$ comes from the prediction step and an upper index $^{(k)}$ denotes k-th iteration within RQI.

These iterations are performed until either the eigenpair is converged, $\lambda^{(k)}$ left the interval I, or the number of iterations k has reached a limit. Although RQI is globally convergent, there is no guarantee that the sequence of Rayleigh quotient iterates will be restricted a priori to a given interval of the spectrum [122, Ch. 4]. So it is not possible to guarantee that the correction step succeeds and the stepsize may have to be reduced.

Close eigenvalue curves

As mentioned in the beginning of this section, the basic homotopy algorithm by Li and Rhee [105] only works satisfactorily for well-isolated eigenvalue curves. The problem arises as eigenvectors belonging to close eigenvalues tend to contaminate each other during inverse iteration. The fact that $A(t)$ has only simple eigenvalues (see the discussion following (6.32)) is of little help, because eigenvalue curves of $A(t)$ may coincide to working precision. A good example here is the Wilkinson matrix W^+_{n+1} ([179, p. 308]), as it also allows reduction to the similar diagonal plus rank-one eigenproblem as the one for (6.31).

The worst cases are however eliminated by deflation, but some extra work still needs to be done. Two different situations of close eigenvalue curves can be distinguished, deflection and clustering.

Deflection. In this situation the eigenvalue curves are close in just a small interval for $t \in [0, 1]$. As an example, Figure 6.4 shows the eigenvalue curves of matrix family (6.31) for a certain D+SS matrix. This a 16×16 matrix with the eigenvalues $\left((10^{-2})^{1/16}\right)^n$, $n = 1, \ldots, 16$. The eigenvalue curves look as some of them cross others, but in fact, the eigenvalue curves just come very close at some point, that we call a *pseudocrossing point*. Nevertheless, the two involved eigenpairs deflect each other, in other words, their values are exchanged.

However, from the interlacing property and monotonicity of the eigencurves follows that an eigenpair $(\mathbf{x}_i(0), \lambda_i(0))$ is usually a good approximation to $(\mathbf{x}_{i-1}(1), \lambda_{i-1}(1))$. As follows from our numerical experiments, such an approximation is often so good that the corresponding eigenpairs become subject to deflation. Figure 6.5 shows the eigenvalue curves for the same matrix as on Figure 6.4, but after deflation. So, only the last eigenpair in such a group of deflecting eigenpairs lacks a good approximation.

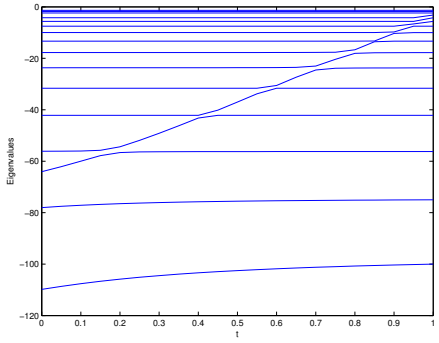


Figure 6.4: Eigenvalue curves before deflation

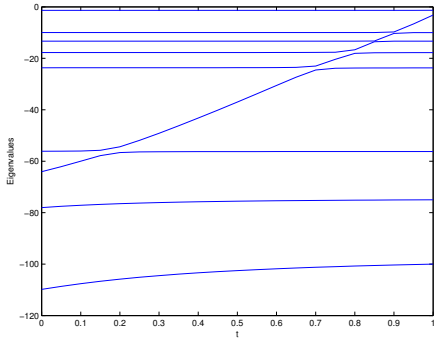


Figure 6.5: Eigenvalue curves after deflation

In case of deflecting eigenpairs we do not use homotopy path following. Instead, eigenpair $(\mathbf{x}_{i-1}(1), \lambda_{i-1}(1))$ is computed with RQI starting with the good approximation $(\mathbf{x}_i(0), \lambda_i(0))$. For the last eigenpair path tracing would be too slow until the pseudocrossing point would be passed. So, it is computed by bisection based on the interval I defined in (6.39), followed by inverse iteration.

We use the following criterion to detect the deflecting eigenvalues. Let $\mathrm{sturm}(\mu)$ denote the amount of eigenvalues of $A(1)$ that are smaller than μ. Then the next inequality serves as a criterion for eigenvalues δ_i and δ_{i+1} to deflect each other:

$$\mathrm{sturm}(\delta_{i+1} - \eta) < i, \quad \eta = 10^{-4}|\delta_{i+1} - \delta_i|.$$

This empiric formula could be easily derived from geometric considerations.

Clustered eigenvalues. If the eigenvalue curves are clustered in the whole interval or at least for $t = 1$, the situation was much worse for the basic homotopy algorithm of Li and Rhee. Luckily, clustering of eigenvalues at $t = 1$ together with the interlacing property (6.32) means that some clustering will be present also at $t = 0$. This means that the worst cases will be filtered out by deflation. In our numerical experiments the eigenvectors have shown reasonable orthogonality, but one may apply a partial reorthogonalization, if required by an application.

On the Figures 6.4-6.5 one may see that a cluster of close eigenvalues around 0 was mostly deflated. These figures represent a typical impact of deflation, as we have discovered while performing numerical experiments.

Final algorithm

We will incorporate now various concepts introduced above into a divide-and-conquer algorithm. Such algorithm solves the complete eigenvalue problem for symmetric irreducible diagonal-plus-semiseparable matrices.

Algorithm 10: HomDSS

input : A – D+SS matrix
output: (X, Λ) – eigenvectors and eigenvalues of A

begin
  if $n < n_0$ **then**
    solve $A = X\Lambda X^T$ by a conventional eigensolver
  else
    construct with Algorithm 8 smaller D+SS matrices D_1 and D_2
    $[Q_1, \Delta_1] = \text{HomDSS}(D_1)$
    $[Q_2, \Delta_2] = \text{HomDSS}(D_2)$
    construct matrix family $B(t)$ according to (6.34)
    deflate some eigenpairs according to (6.33)
    trace remaining paths of $B(t)$ with Algorithm 9
    transform the eigenvectors of $B(1)$ back to the ones of A with (6.28)
  end
end

6.2.4 Arithmetic complexity

Our complexity analysis is based on the number of floating-point operations (flop) performed. With the homotopy method, it is difficult to get an expression for the arithmetic complexity which is meaningful for a general matrix. The complexity not only depends on the matrix order n, but also on the spectrum that we would like to compute.

The divide step performed by Algorithm 8 requires $C_0 = \mathcal{O}(n)$ operations.

Consider now the complexity for one conquer step. Deflation of eigenpairs takes $\mathcal{O}(n)$ operations. The most expensive part in path tracing is a Rayleigh quotient iteration. Recall that matrices $B(t)$ (6.34), involved in RQI, are diagonal-plus-semiseparable. So, one step of RQI costs $\mathcal{O}(n)$ operations for matrices $B(t)$ as one may use the existing linear solver, see [174]. So, to trace nontrivial eigenpaths, one needs to perform

$$C_1 = \phi\omega_t\omega_{RQI}\mathcal{O}(n)$$

operations, where ϕ denotes the number of nontrivial eigenpaths, ω_t is an average number of time steps and ω_{RQI} denotes the average number of Rayleigh quotient iterations.

To obtain an upper bound on C_1 we assume that no deflation occurs (so $\phi = n$). The average numbers of iterations are coming from our numerical experiments and representative numbers are $\omega_t = 4/3$ and $\omega_{RQI} = 3$. This gives in total $C_1 = \mathcal{O}(n^2)$. Transformation of the eigenvectors of $A(1)$ back to the ones of A with (6.28) can be done effectively with an algorithm by Borges and Gragg [17], and costs $C_2 = \mathcal{O}(n \cdot n \log n) = \mathcal{O}(n^2 \log n)$ operations. The sum $C_0 + C_1 + C_2$ gives $\mathcal{O}(n^2 \log n)$ operations for one iteration of Algorithm 10.

The overall divide-an-conquer algorithm has the form of a binary tree of height $s = \log n$, and each node represents one iteration of Algorithm 10. So, the total complexity is $\mathcal{O}(n^2 \log^2 n)$ for the worst-case scenario (no deflation). However, deflation occurs for almost every matrix in general. Thus, the above asymptotics do not necessarily reflect what may be observed in an actual computation.

6.2.5 Numerical experiments

The numerical tests were performed on a PC with 2.93 GHz Intel Core 2 processor and 2 Gb of memory, running Debian Squeeze with 2.6.32 kernel and Matlab 7.9.0.529.

We built diagonal-plus-semiseparable matrices of dimension $N = 2^j$, $j = 7, \ldots, 10$ and for each dimension matrices which have condition number (defined as the product of the spectral norm of the matrix and that one of its inverse) equal to $10^3, 10^6, 10^9$ and 10^{12}. For each of these 16 classes of test matrices we took 10 samples.

The test matrices were built as follows: starting from a diagonal matrix $D = [\alpha, \alpha^2, \ldots, \alpha^N]$ with α the $(N-1)$th root of the requested condition number, we applied $(N-1)$ random Givens rotations G_i to the left, such that G_i works on the ith and $(i+1)$th row, and G_i^T to the right of D. Hence D was transformed into a matrix $A = GDG^T$. This matrix A is a diagonal-plus-semiseparable matrix because the ith Givens rotation G_i makes the elements of row i and $i+1$ proportional. The transpose G_i^T does the same with column i and $i+1$, so we created a semiseparable structure except on the diagonal.

Deflation

It is interesting to see that the deflation generally plays an important role in the algorithm. Table 6.3 shows the percentage of deflated, trivial eigenpaths on the last (highest) reassembly step observed in the homotopy algorithm for the test matrices. The given numbers show that, for example, for the matrix of order 2^{10} less than 6% curves have to be traced. The frequency of deflation will vary with the type of the matrix and with its order. For our series of test matrices one may see that the impact of the condition number is negligible, compared to that of the matrix size.

cond A	$i = 7$	$i = 8$	$i = 9$	$i = 10$
10^3	65	82	93	94
10^6	66	84	93	95
10^9	69	85	95	96
10^{12}	72	87	95	96

Table 6.3: Percentage of deflation, matrices of order 2^i

Cuppen showed for his divide-and-conquer algorithm, which has the similar deflation technique, that matrices with much deflation in general have an eigenvector matrix close to a band matrix [42].

Diagonal plus rank-one problem

The accuracy of an eigensolver is determined by the residual error $\mathcal{R}$ of the computed solution as well as the orthogonality $\mathcal{U}$ of the computed eigenvectors. The core part of the proposed algorithm consists of a homotopy algorithm for a diagonal plus rank-one problem. Therefore, we give several numerical results for this subproblem.

The computed eigenpairs of such a diagonal plus rank-one matrix $A(1)$ (6.33) divide into two classes. The first class includes those eigenpairs coming from deflation. By construction, deflated eigenpairs represent accurate approximations to eigenvalues and eigenvectors, and the eigenvectors are perfectly orthogonal (they are columns of the identity matrix). The second class represents those eigenpairs coming from Algorithm 9. Suppose that H is a diagonal matrix with the computed eigenvalues on the diagonal, and Z is the (orthogonal) matrix of the computed eigenvectors. On the last (highest)

reassembly step we look at the relative residual norm

$$\mathcal{R} = \frac{\|B(1)Z - ZH\|_2}{\varepsilon \|B(1)\|_2}$$

and at the orthogonality condition

$$\mathcal{U} = \frac{\|Z^T Z - I\|_2}{\varepsilon \|Z\|_2}.$$

Relative residuals for our test matrices are presented in Table 6.4 and the orthogonality is given in Table 6.5.

cond A	$i = 7$	$i = 8$	$i = 9$	$i = 10$
10^3	6.2	4.8	5.7	4.1
10^6	4.3	4.2	2.8	1.8
10^9	1.9	2.1	1.3	3.7
10^{12}	4.1	2.2	2.2	3.3

Table 6.4: residuals $\mathcal{R}$, matrices of order 2^i reduced to $B(1)$

cond A	$i = 7$	$i = 8$	$i = 9$	$i = 10$
10^3	26	41	150	45
10^6	102	200	90	84
10^9	410	700	590	420
10^{12}	1020	1100	10200	7820

Table 6.5: Orthogonality $\mathcal{U}$, matrices of order 2^i reduced to $B(1)$

As follows from these tables, the residuals are accurate up to machine precision, but the orthogonality of the eigenvectors is slightly worse. One may apply a partial reorthogonalization, if desired.

In Table 6.6 we represent an average number of bisection+inverse iteration calls. This method serves as a fallback solution if RQI does not converge to the right eigenpair, and also it is applied to find the largest eigenvalue in the group of deflecting eigenvalues. One may see that this rather slow method is called on average for just one or two eigenpairs even for large matrices.

cond A	$i = 7$	$i = 8$	$i = 9$	$i = 10$
10^3	1.1	1.3	1.0	1.33
10^6	1.7	1.1	1.0	1.3
10^9	0.7	1.0	1.1	1.9
10^{12}	2.0	2.0	1.9	1.4

Table 6.6: Average number of BiSect calls, matrices of order 2^i reduced to $B(1)$

Original D+SS problem

Compared to the method presented by Mastronardi et al. [110], our algorithm differs only in the method used to solve the diagonal plus rank-one eigenproblem. We have computed the same residuals $\mathcal{R}$ and orthogonality measures $\mathcal{U}$ for their method, and the data does not differ significantly from Tables 6.4 and 6.5. This means that both methods have the same precison. However, our new method could be parallelized more effectively. Our easy technique for deflation allows to exclude from the active processing most of the eigenvalues and thus leads to better performance.

We refer to [110] for residual plots for the original D+SS problem.

6.3 A direct method for BBBT matrices

In this section we present a new direct algorithm to solve linear systems, where the coefficient matrices are block banded block Toeplitz matrices with dense Toeplitz blocks (hereinafter referred as BBBT matrices). The method transforms the doubly Toeplitz structure of the original coefficient matrix to a block circulant structure with Toeplitz blocks (hereinafter referred as BCTB matrix). The corresponding linear system is solved in a fast and stable way, and its solution is finally transformed back to the solution of the original system.

In Subsection 6.3.1 the main algorithm is given. The details of its theoretical and practical complexity are studied in Subsection 6.3.2. Finally, we present several numerical results, also for certain ill-conditioned cases. Within this section we follow our paper [34].

6.3.1 Main formulation

The algorithm first transforms a given BBBT matrix to a block circulant one by adding the required Toeplitz blocks into the lower left and upper right corners of the matrix, replacing zero blocks. Later, the outer (circulant) and inner (Toeplitz) block structures are interchanged and the system is converted to a block Toeplitz system with circulant blocks. The circulant blocks are easily diagonalized by the Fourier transform, and the large block Toeplitz system decomposes into several systems of simple Toeplitz structure. They are, in turn, solved by a conventional Toeplitz solver, and the resulting vectors are transformed back to the solution of the block circulant system. Finally, the target solution of the BBBT system is recovered via the Sherman-Morrison formula.

Reduction to the block circulant case

Consider a block banded block Toeplitz system with dense Toeplitz blocks $Bx = b$. We suppose that inner blocks are $n \times n$-matrices and there are m block columns and m block rows. The bandwidth is equal to $2k + 1$.

A block banded block Toeplitz matrix could be easily transformed to a block circulant matrix with Toeplitz blocks (further referred as BCTB matrix) by adding corresponding blocks in its lower-left and upper-right corners. The matrix B in (6.40) is block Toeplitz and C in (6.41) is its corresponding block circulant.

$$B = \begin{pmatrix} A_0 & A_1 & \ddots & A_k & 0 & 0 & \cdots & 0 \\ A_{-1} & A_0 & A_1 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \ddots & A_{-1} & A_0 & A_1 & \ddots & \ddots & \ddots & 0 \\ A_{-k} & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & A_k \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & A_{-1} & A_0 & A_1 \\ 0 & \dots & 0 & 0 & A_{-k} & \ddots & A_{-1} & A_0 \end{pmatrix} \tag{6.40}$$

$$C = \begin{pmatrix} A_0 & A_1 & \ddots & A_k & 0 & A_{-k} & \cdots & A_{-1} \\ A_{-1} & A_0 & A_1 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \ddots & A_{-1} & A_0 & A_1 & \ddots & \ddots & \ddots & A_{-k} \\ A_{-k} & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & A_k \\ A_k & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & A_{-1} & A_0 & A_1 \\ A_1 & \dots & A_k & 0 & A_{-k} & \ddots & A_{-1} & A_0 \end{pmatrix} \tag{6.41}$$

We can write that $B = C - UV^T$, where

$$U = \begin{pmatrix} I_{kn} & 0 \\ 0 & 0 \\ 0 & P \end{pmatrix}, \quad V^T = \begin{pmatrix} 0 & 0 & Q \\ I_{kn} & 0 & 0 \end{pmatrix}. \tag{6.42}$$

Matrices P and Q are $kn \times kn$ block triangular matrices that reside in the lower left and upper right corners of C, respectively,

$$P = \begin{pmatrix} A_k & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ A_2 & \ddots & \ddots & 0 \\ A_1 & A_2 & \cdots & A_k \end{pmatrix}, \quad Q = \begin{pmatrix} A_{-k} & \cdots & A_{-2} & A_{-1} \\ 0 & \ddots & \ddots & A_{-2} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A_{-k} \end{pmatrix}. \tag{6.43}$$

The idea of interchanging the outer and inner structures is thoroughly studied in a general case in the book [177].

The inverses of B and C are then connected by means of the Sherman-Morrison-Woodbury formula [71]:

$$B^{-1} = (C - UV^T)^{-1} = C^{-1} + C^{-1}U(I - V^T C^{-1} U)^{-1} V^T C^{-1}. \tag{6.44}$$

Remember that we have to compute $x = B^{-1}b$. By means of the above formula this computation is partitioned into several steps.

At first, solve the BCTB system $Cw = b$ as described in the next section. Let us denote $w = C^{-1}b$. Then by means of a simple matrix-vector multiplication we compute the product $v = V^T w = V^T C^{-1} b$.

In the third step, compute the inner matrix $S = I - V^T C^{-1} U$ and solve the linear system $Sy = v$. To compute S directly we need to solve $2kn$ linear systems with C as the coefficient matrix and the columns of U as right-hand side vectors. The computation of the S matrix could be slightly optimized as follows.

$$V^T C^{-1} U = \begin{pmatrix} 0 & 0 & Q \\ I_{kn} & 0 & 0 \end{pmatrix} \begin{pmatrix} C_{UL} & \dots & C_{UR} \\ \dots & \dots & \dots \\ C_{DL} & \dots & C_{DR} \end{pmatrix} \begin{pmatrix} I_{kn} & 0 \\ 0 & 0 \\ 0 & P \end{pmatrix} = \begin{pmatrix} QC_{DL} & QC_{DR}P \\ C_{UL} & C_{UR}P \end{pmatrix}. \quad (6.45)$$

Here C_{DL}, C_{DR}, C_{UL} and C_{UR} are the corresponding $kn \times kn$ corner blocks of C^{-1}.

Since C is block-circulant, C^{-1} is also block circulant and thus determined by its first block column. This means that it's enough to solve only n linear systems with the first n columns of a $mn \times mn$ identity matrix as right-hand side vectors. The rest of C^{-1} is then constructed just by a block reordering of its first block column.

In the fourth and the last step, we perform the matrix-vector multiplication $z = Uy$ and finally solve the remaining BCTB system $Cf = z$. Addition of $C^{-1}b$ to $C^{-1}z$ yields $B^{-1}b$.

Block circulant Toeplitz block case

Let us consider a block circulant matrix with Toeplitz blocks. We suppose that the inner blocks are $n \times n$-matrices and there are m block columns and m block rows. Let P_1 denote a permutation matrix of size $mn \times mn$ such that it brings rows with numbers 1, $n+1$, $2n+1$, etc, together to the first rows, then all rows with numbers 2, $n+2$, $2n+2$, etc, together, and so on. The same should hold for the columns after the multiplication by P_1^T. An example of the P_1 matrix

for $n = m = 3$ is given in (6.46).

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.46}$$

In other words, if C was a BCTB matrix, then $K = P_1 C P_1^T$ would be a BTCB matrix: the inner structure becomes the outer and vice versa. Because the outer structure of K is Toeplitz, there are only $2n+1$ different circulant blocks of size $m \times m$. These blocks are easily diagonalized by means of a Fourier transform.

Let $F = \bigoplus F_m$ be a direct sum of n Fourier matrices of order m. Then $L = FKF^* = FP_1CP_1^TF^*$ is a block Toeplitz matrix with diagonal blocks. Let us denote by P_2 a permutation matrix which is like P_1, but brings first all rows with numbers 1, $m+1$, $2m+1$ together, then all rows with numbers 2, $m+2$, $2m+2$ together and so on. Then $M = P_2FP_1CP_1^TF^*P_2^T$ is a block diagonal matrix with Toeplitz blocks.

Remember that we have to solve a linear system $Cx = b$, where the inner blocks are $n \times n$-matrices and there are m block rows and m block columns. By means of the transforms described above we have converted the original BCTB problem to a solution of a block diagonal Toeplitz block system:

$$Cx = b \Leftrightarrow P_2FP_1Cx = P_2FP_1b \Leftrightarrow$$

$$(P_2FP_1CP_1^TF^*P_2^T)(P_2FP_1x) = P_2FP_1b \Leftrightarrow M\hat{x} = \hat{b}, \tag{6.47}$$

where

$$M = P_2FP_1CP_1^TF^*P_2^T, \quad \hat{x} = P_2FP_1x, \quad \hat{b} = P_2FP_1b. \tag{6.48}$$

This block Toeplitz system splits up into m $(n \times n)$-Toeplitz systems. These systems could be solved by means of any conventional Toeplitz solver, like fast and superfast solvers [30, 75, 98, 109, 155].

Transition from $\hat{x}$ to x is obvious and consists of two reorderings and n inverse FFT's.

6.3.2 Complexity

Theoretical estimation

Let us look at the formula (6.44) and consequently calculate its computational cost. Denote the total computational cost by $\xi(m, n, k)$.

1. Computing $w = C^{-1}b$ is exactly solving a corresponding linear system. Let us denote the complexity of the solution of a BCTB linear system with m block rows and m block columns, each block is of size $n \times n$, by $\nu(m, n)$. Thus we have $\nu(m, n)$.
2. Computing $v = V^T w$ is in fact one $kn \times kn$ matrix-vector product due to a special structure of V^T. Thus we have $O(k^2 n^2)$.
3. Computing $V^T C^{-1} U$ by means of a technique (6.45) involves n operations of complexity $\nu(m, n)$ and then four matrix-matrix products of order $kn \times kn$. Thus we have $n \cdot \nu(m, n) + O(k^3 n^3)$. Recalling that matrices P and Q are block Toeplitz can slightly reduce this estimate.
4. Solution of a linear system with the matrix $I - V^T C^{-1} U$ as a coefficient matrix and vector v as a right-hand side takes $O(k^3 n^3)$ operations since the matrix is unstructured.
5. Multiplication by matrix U on the left takes $O(k^2 n^2)$ operations and yields an $mn \times 1$ vector z.
6. And finally one has to solve the linear system with C matrix and vector z. It takes another $\nu(m, n)$ operations.

In total after gobbling up all low-order terms we get $\xi(m, n, k) = n \cdot \nu(m, n) + O(k^3 n^3)$.

Let us now estimate $\nu(m, n)$. On the basis of the formulas given in the last part of the previous subsection, we have the following essential steps.

1. Multiplications with P_1, P_2, P_1^T and P_2^T are in fact just reorderings.
2. There are n Fourier transforms of size m, which gives $nm \log m$ operations.
3. We have to solve m Toeplitz systems of order n. This could be done by any conventional method, like Gaussian elimination ($O(mn^3)$ in total) or existing fast methods ([75, 98, 155]) ($O(mn^2)$ in total), or even by superfast methods, like the one described in the previous section, see also [163].

In the case one uses fast Toeplitz solvers the total complexity will be $\xi(m,n,k) = O(mn^3)+O(k^3n^3)$ flops. For superfast solvers the total complexity will be equal to $\xi(m,n,k) = O(mn^2\log^2 n) + O(k^3n^3)$.

6.3.3 Numerical experiments

The experiments were performed on a machine with 2Gb memory and Xeon 2.3 GHz processor, running Kubuntu Linux, kernel 2.6.26, in Matlab 7.7.0.471. We have chosen as Toeplitz solvers (further referred as T-solvers) 1) the fast Hankel solver developed by Rodriguez and Arico [7], written in C and compiled as `mex`-file for Matlab and 2) the simple backslash operator for comparison. Another possibility for the fast solver with existing software is the one by Van Barel and Kravanja [98].

Well-conditioned matrices

Toeplitz matrices and right-hand side vectors were random.

The results presented in Table 6.7 are times in seconds. For each table cell there were three runs and the resulting time was averaged. The last column represents the time required to solve the dense original system directly with the backslash operator of Matlab. OoM stands for the Out Of Memory Matlab message.

For the complexity we can look at the ratios of the neighbouring values in each column. We can see that the results do agree with the estimate. The method is linear in m and is quadratic in n when the fast T-solver is applied, and is cubic in n when Gaussian elimination is applied to Toeplitz systems. For fixed m and $n \geqslant 256$ the fast solver beats both its competitors. It's also more efficient for $m = n = 128$ and the bandwidth is not too large. The bandwidth increase has impact on time only when $k^3 \geqslant m$.

The residual size was of the same order for the fast solver and for the backslash operator applied to full dense matrices. For large matrices where the OoM message appeared, the residual was of the order $10^{-10} - 10^{-13}$.

Ill-conditioned matrices

Note that we replace the BBBT matrix B with the BCTB matrix C in the solution process. This has positive and negative effects, as we will show. Usually in the case of an ill-conditioned B or C it is useful to perform one

k	m	n	$O(n^3)$ T-solver	$O(n^2)$ T-solver	\ operator
2	32	32	0.44	0.95	0.27
2	32	64	2.04	2.77	1.33
2	32	128	15.56	10.72	8.10
2	32	256	172.3	55.19	66.21
2	32	512	1996	366.66	OoM
2	64	32	0.73	1.90	1.37
2	128	32	1.46	3.79	9.14
2	256	32	2.87	7.52	64.81
2	512	32	5.79	15.08	OoM
2	1024	32	11.40	30.06	OoM
2	64	64	4.08	5.43	9.19
4	64	64	4.14	5.57	9.32
8	64	64	4.53	6.62	9.20
16	64	64	6.87	13.19	9.22
32	64	64	23.13	61.62	9.23
2	128	128	63.09	41.09	OoM
4	128	128	63.34	42.12	OoM
8	128	128	65.93	49.49	OoM
16	128	128	82.62	97.76	OoM

Table 6.7: Execution times for different m, n, k

or more steps of an iterative refinement (IR) process:

$$R_i = b - Bx_i, \quad d_i = B^{-1}R_i, \quad x_{i+1} = x_i + d_i, \quad i = 0, 1, \ldots, \tag{6.49}$$

where x_0 is some initial approximation. To perform one IR step after the original system was solved is cheaper than to solve it again: the most expensive step 3 (Subsection 6.3.2) of computing the inner matrix in the Sherman-Morrison-Woodbury formula does not have to be repeated. The total cost of an IR step is then $\nu(m, n) + O(k^3 n^3)$.

Ill-conditioned B

In this case the change to the circulant transformation C has a certain regularizing effect. We have constructed several ill-conditioned BBBT matrices and compared their condition numbers with the numbers of their circulant transformations. The results are given in Table 6.8. The residual $b - Bx$ was of the same order as the residual $b - B\hat{x}$, where $\hat{x}$ was the result of the Matlab

cond(B)	cond(C)
10^8	17
10^{11}	32
10^{14}	20
10^{18}	8

Table 6.8: Regularizing effect of circulant transformation

command $B\backslash b$. For the matrices with the condition numbers greater than 10^{10} we applied one IR step.

However in the last case ($\text{cond}(B) = 10^{18}$) we were not able to get any reasonable result – the matrix was numerically singular.

Ill-conditioned BBBT matrices were constructed in the following way. We took a well-conditioned matrix and then scaled its elements: the elements further away from the main diagonal had bigger values.

Ill-conditioned C

It could happen that even for a well-conditioned B the matrix C would be ill-conditioned or singular. This is illustrated by the following example:

$$B = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \tag{6.50}$$

We can handle this situation by adding a small block banded block Toeplitz noise to the matrix B before it is transformed into the BCTB matrix. Our experiments have shown that it's enough to add just block-diagonal Toeplitz noise and not to affect the strictly lower and upper block triangles of the BBBT matrix.

In Table 6.9 we illustrate this by the following example: matrix B was fixed in such a way that its circulant extension was singular, $\text{cond}(B) = 15$. We were adding some block-diagonal noise, the 2-norm of this noise is given in the second column. The condition number of (permuted) C matrix is given in the first column. The relative residual $(b - Bx)/\|b\|_2$ is given in the last column.

This example shows that small noise does not have enough regularizing effect and large noise moves the perturbed matrix too far from the original. After one IR step with the noise size $O(10^{-8})$ we got the machine precision for the residual.

cond(C)	noise size	rel. error
10^5	10^{-4}	10^{-4}
10^9	10^{-6}	10^{-6}
10^9	10^{-8}	10^{-8}
10^{10}	10^{-9}	10^{-6}
10^{13}	10^{-10}	10^{-4}
10^{13}	10^{-12}	10^{-3}
10^{15}	10^{-14}	10^{-1}
10^{18}	10^{-16}	100

Table 6.9: Effect of small noise on relative erros for singular circulant extensions

We have also mentioned that an ill-conditioned C gives the very ill-conditioned inner matrix in the Sherman-Morrison formula (step 3, Section 6.3.2), as well as ill-conditioned small Toeplitz systems.

We may use all these facts to build a generic approach.

- Start the algorithm as described.
- If one of the small Toeplitz systems of order n or the inner matrix in the Sherman-Morrison-Woodbury formula would appear to be ill-conditioned (this is reported by a conventional solver used), then we have ill-conditioned C. Return to the previous step and add some noise, then repeat the procedure.
- Compute the relative residual $(b - Bx)/\|b\|_2$. If it's not sufficiently small, then perform one or more steps of iterative refinement.

6.4 Conclusion

We presented three algorithms, solving linear-algebraic problems with structured matrices. Firstly, an iterative method for inverting Hermitian Toeplitz matrices by a continuation algorithm is proposed. The method can be used for solving systems of linear equations with matrices in this class as coefficient matrices. The algorithm gradually transforms the identity matrix to a target inverse, and has a complexity of $\mathcal{O}(n \log n)$ flops, where n is the order of a given matrix. Later, a similar continuation approach is applied to find all the eigenvalues and eigenvectors of a given diagonal-plus-semiseparable matrix. The goal is achieved by tracing a solution curve of a certain differential equation.

To construct good starting matrices, divide-and-conquer methods are used. Deflation techniques are implemented and lead to better speed and accuracy of the algorithm.

Finally, a fast algorithm to solve block banded block Toeplitz linear systems with non-banded Toeplitz blocks is developed. A circulant extension of a given Toeplitz system is constructed and then by means of the Sherman-Morrison-Woodbury formula its inverse is transformed to the inverse of the original matrix. In turn, the block circulant matrix with Toeplitz blocks is converted to a block diagonal matrix with Toeplitz blocks, and then the resulting Toeplitz systems are solved by means of a fast solver. The computational complexity in the case one uses fast Toeplitz solvers is equal to $\xi(m, n, k) = \mathcal{O}(mn^3)+\mathcal{O}(k^3n^3)$ flops, where m denotes the block order of the matrix, n denotes the order of the blocks, and $2k + 1$ is the bandwidth.

Several numerical experiments are presented for each of the methods and show the effectiveness of the algorithms.

Chapter 7

General conclusions and future perspectives

7.1 Conclusion

In this thesis eigenvalues, structured matrices and orthogonal functions were studied from a practical point of view. We exploited relations between any of these three objects to design algorithms in the context of five problems. Each problem is closely related to one of the basic linear algebra problems: solving a system of linear equations or an eigenvalue problem. The five corresponding problems are divided between different chapters.

Within Chapter 2, after defining the necessary concepts, we have shown that regularity of a graph cannot be deduced from its spectrum with respect to a certain generalized adjacency matrix. First, several small counterexamples were found by computer enumeration. Finally, a general procedure, allowing to construct more counterexamples, is described.

Chapter 3 was devoted to the convergence behavior of the rational Lanczos method. Again, we begun with necessary concepts from logarithmic potential theory and established some properties of a weighted logarithmic potential. Then we presented a novel method to numerically solve the constrained weighted energy problem, describing a distribution of converged rational Ritz

values. First, we formulated the continuous version of the algorithm, and then we discretized it. Compared with the continuous version, the discretized version has the advantage that the algorithm always stops, producing a solution which is accurate in comparison to the exact solution when known. Finally, we used the algorithm to predict the region of convergence of Ritz values obtained by applying the rational Lanczos method for symmetric eigenvalue problems. In all cases our algorithm estimated the region of convergence of Ritz values in an accurate way.

In Chapter 4 we developed an algorithm to compute recurrence relation coefficients for bivariate polynomials, orthonormal with respect to a discrete inner product. We started with an application, namely, with the discrete least squares approximation problem. We generalized several ideas from the theory of univariate polynomials to the multivariate case and posed a pair of coupled inverse eigenvalue problems. These inverse eigenvalue problems describe the recurrence relation coefficients of the target polynomials. Later, such coupled problems were solved by means of a novel updating algorithm. The algorithm essentially represents a reduction to generalized Hessenberg form with a sequence of Givens rotations. Finally, the algorithm was tested for several numerical examples. Because of the orthogonal nature of rotations, the algorithm has shown a stable behavior. Within this work we consider the constructed algorithm as the most promising one from the applications point of view. Several possibilities to improve the algorithm and some potential applications thereof are described in the next section.

Three algorithms for three classes of structured matrices were studied in Chapters 5 and 6. First, a homotopy approach was applied to solve a linear system with a Toeplitz coefficient matrix. The compact representation of the inverse of the coefficient matrix comes for free while executing the method. A displacement rank representation of the matrices involved helped to reduce memory requirements and allowed fast matrix-by-vector multiplication techniques. Second, all the eigenvalues and eigenvectors of a symmetric diagonal-plus-semiseparable matrix were computed by another version of a homotopy algorithm. The goal was achieved by tracing a solution curve of a certain differential equation. To construct good starting matrices, divide-and-conquer methods were used. Deflation techniques were implemented and led to good accuracy of the algorithm. Finally, we derived in this chapter a direct method to solve a two-level Toeplitz linear system with banded outer structure.

In general, it is possible to conclude that the main objective of this work is achieved. In fact, five problems coming from the different fields were studied, and the relations between eigenvalues, orthogonal functions and structured matrices played an important role while designing the corresponding algorithms

to solve these problems. All developed methods performed well in numerical experiments.

7.2 Further research

The following paragraphs list ideas for further investigation, which came up during the development process of the algorithms and the comparison with other algorithms.

Number of graphs with cospectral mates

As we have mentioned in Chapter 2, the percentage of graphs on n vertices, determined by their spectrum, is still unknown. Only some asymptotic lower bounds for this number are present. It may be interesting to extend the computer enumeration results of [76, 22]. This problem by itself is challenging because the total amount of graphs on 12 vertices already is more than 165 billion. So, clever optimization has to be done to reduce the amount of work. Even more challenging is the theoretical investigation.

Best grid for computing multivariate discrete orthogonal polynomials

The problem of selecting the proper points to prescribe the inner product for computing discrete orthogonal polynomials is much more complex in the multivariate case, compared to the univariate case. The key issue is that the points have to be sufficiently independent, so that no algebraic curve of a small degree goes through all the points, thus leading to proper interpolation and preventing the method from finding further linearly independent basis polynomials. Within this research we used Padua points, and the achieved accuracy of the algorithm was sufficient for practical purposes. Further, different generalizations of a Chebyshev grid to more dimensions may be also studied. The complete answer to the question "which grid leads to the most stable computations with the presented updating algorithm" is still unknown.

Downdating data for inner products

The proposed method for constructing 2D-orthogonal polynomials has the feature that the points where the inner product is prescribed, could be added one-by-one, thus updating the inner product in every step. As shown by Bultheel and Van Barel in [160], a 1D-downdating scheme may also be designed, allowing deletion of certain points. However, as in the 1D-case, special precautions should be taken to obtain a stable algorithm to solve this (inverse) problem. So, designing and implementing a 2D-downdating algorithm requires further investigation.

Multivariate orthogonal polynomials as a tool

The proposed algorithm that constructs a multivariate polynomial basis may serve as a tool in many applications, similarly as Chebyshev polynomials work within the `chebfun` project [8] to speed up the computations with functions of different nature. The applications include, for example, solving ODEs and expressing the solutions in the constructed polynomial basis. We consider designing and implementing such a tool as an interesting project.

Efficient parallel implementation of one of the continuation methods

The present implementation of our continuation method for eigenvalue problems with D+SS matrices is sequential. An accurate implementation for some parallel computer is nontrivial since many aspects have to be taken into account: memory management, processor communication, distribution of data between different processors and so on.

Bibliography

[1] E. L. Allgower and K. Georg. *Introduction to numerical continuation methods*, volume 45 of *Classics in applied mathematics*. siam, 2003. pages 14, 15, 99

[2] G. S. Ammar and W. B. Gragg. The generalized Schur algorithm for the superfast solution of Toeplitz systems. In J. Gilewicz, M. Pindor, and W. Siemaszko, editors, *Rational Approximation and its Applications in Mathematics and Physics*, volume 1237 of *Lecture Notes in Mathematics*, pages 315–330. Springer-Verlag, Berlin, 1987. pages 11

[3] G. S. Ammar, W. B. Gragg, and L. Reichel. Constructing a unitary Hessenberg matrix from spectral data. In G. H. Golub and P. Van Dooren, editors, *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms*, volume 70 of *Computer and Systems Sciences*, pages 385–395. Springer-Verlag, Berlin, Germany, 1991. pages 9

[4] G. S. Ammar and C. He. On an inverse eigenvalue problem for unitary Hessenberg matrices. *Linear Algebra and its Applications*, 218:263–271, 1995. pages 9

[5] P. Arbenz. Divide and conquer algorithms for the band symmetric eigenvalue problem. *Parallel Computing*, 18:1105–1128, 1992. pages 103

[6] P. Arbenz and G. H. Golub. QR-like algorithms for the symmetric arrow matrices. *SIAM Journal on Matrix Analysis and Applications*, 13(2):655–658, April 1992. pages 103

[7] A. Arico and G. Rodriguez. A fast solver for linear systems with displacement structure. *Numerical Algorithms*, pages 1–27, 2010. pages 143

[8] Z. Battles and L. N. Trefethen. An extension of MATLAB to continuous functions and operators. *SIAM Journal on Scientific Computing*, 25(5):1743–1770, 2004. pages 152

[9] B. Beckermann. A note on the convergence of Ritz values for sequences of matrices. Technical Report ANO 408, Labo Paul Painlevé, Université de Lille I, France, 2000. pages 58

[10] B. Beckermann, S. Güttel, and R. Vandebril. On the convergence of rational Ritz values. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1740–1774, 2010. pages 6, 50, 54, 61

[11] B. Beckermann and A. B. J. Kuijlaars. Superlinear convergence of conjugate gradients. *SIAM Journal on Numerical Analysis*, 39:300–329, 2001. pages 56

[12] R. Bevilacqua and G. M. Del Corso. Structural properties of matrix unitary reduction to semiseparable form. *Calcolo*, 41(4):177–202, 2004. pages 92

[13] D. A. Bini and B. Meini. Improved cyclic reduction for solving queueing problems. *Numerical Algorithms*, 15(1):55–74, 1997. pages 12

[14] D. A. Bini and B. Meini. Effective methods for solving banded Toeplitz systems. *SIAM Journal on Matrix Analysis and Applications*, 20(3):700–719, July 1999. pages 12

[15] D. A. Bini and B. Meini. *Solving block banded block Toeplitz systems with structured blocks: algorithms and applications*, pages 21–41. Advances In Computation: Theory And Practice. Structured matrices: recent developments in theory and computation. Nova Science Publishers, Inc., Commack, NY, USA, 2001. pages 12

[16] A. Bjorck. *Numerical methods for least squares problems*. SIAM, 1996. pages 71

[17] C. F. Borges and W. B. Gragg. A parallel divide and conquer algorithm for the generalized real symmetric definite tridiagonal eigenproblem. In L. Reichel, A. Ruttan, and R. S. Varga, editors, *Numerical Linear Algebra and Scientific Computing*, pages 11–29. de Gruyter, Berlin, Germany, 1993. pages 134

[18] L. Bos, M. Caliari, S. De Marchi, M. Vianello, and Y. Xu. Bivariate Lagrange interpolation at the Padua points: the generating curve approach. *Journal of Approximation Theory*, 143:15–25, 2006. pages 9

[19] L. Bos, S. De Marchi, M. Vianello, and Y. Xu. Bivariate Lagrange interpolation at the Padua points: the ideal theory approach. *Numerische Mathematik*, 108:43–57, 2007. pages 9

[20] R. P. Brent, F. G. Gustavson, and D. Y. Y. Yun. Fast solution of Toeplitz systems of equations and computation of Padé approximants. *Journal of Algorithms*, 1(3):259–295, September 1980. pages 11

[21] A. E. Brouwer and W. H. Haemers. *Spectra of Graphs*. not published, 2011. pages 24, 29, 30

[22] A. E. Brouwer and E. Spence. Cospectral graphs on 12 vertices. *The Electronic Journal on Combinatorics*, 16(20), 2009. pages 29, 30, 151

[23] A. Bultheel and P. Dewilde. On the relation between Padé approximation algorithms and Levinson/Schur recursive algorithms. *Signal Processing: Theories and Applications*, pages 517–523, 1980. pages 11

[24] A. Bultheel and M. Van Barel. Vector orthogonal polynomials and least squares approximation. *SIAM Journal on Matrix Analysis and Applications*, 16(3):863–885, 1995. pages 9

[25] J. R. Bunch, C. P. Nielsen, and D. C. Sorensen. Rank-one modification of the symmetric eigenvalue problem. *Numerische Mathematik*, 31:31–48, 1978. pages 127

[26] M. Caliari, S. De Marchi, A. Sommariva, and M. Vianello. Padua2DM: fast interpolation and cubature at the Padua points in Matlab/Octave. *Numerical Algorithms*, 2009. Published online. pages 82

[27] M. Caliari, S. De Marchi, and M. Vianello. Bivariate polynomial interpolation on the square at new nodal sets. *Applied Mathematics and Computation*, 165(2):261–274, 2005. pages 9

[28] M. Cámara, J. Fàbrega, M. A. Fiol, and E. Garriga. Some families of orthogonal polynomials of a discrete variable and their applications to graphs and codes. *The Electronic Journal on Combinatorics*, 16(1), July 2009. pages 8

[29] R. H. Chan and M. K. Ng. Conjugate gradient methods for Toeplitz systems. *SIAM Review*, 38:427–482, 1996. pages 12

[30] T. F. Chan and P. Chr. Hansen. A stable Levinson algorithm for general Toeplitz systems. CAM Report 90-11, UCLA, May 1990. pages 141

[31] P. L. Chebyshev. Sur les fractions continues. *Journal de Mathématiques*, 3(2):289–323, 1858. pages 7

[32] A. A. Chesnokov, K. Deckers, and M. Van Barel. A numerical solution of the constrained weighted energy problem. *Journal of Computational and Applied Mathematics*, 235(4):950–965, December 2010. pages 6, 18, 40, 54

[33] A. A. Chesnokov and W. H. Haemers. Regularity and the generalized adjacency spectra of graphs. *Linear Algebra and its Applications*, 416(2-3):1033–1037, July 2006. pages 3, 18, 21

[34] A. A. Chesnokov and M. Van Barel. A direct method to solve block banded block toeplitz systems with non-banded toeplitz blocks. *Journal of Computational and Applied Mathematics*, pages 1485–1491, 2010. pages 12, 19, 107, 137

[35] A. A. Chesnokov, M. Van Barel, N. Mastronardi, and R. Vandebril. Homotopy algorithm for the symmetric diagonal-plus-semiseparable eigenvalue problem. Technical Report TW594, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium, May 2011. pages 15, 19, 107, 120

[36] T. S. Chihara. *An introduction to orthogonal polynomials*. Gordon & Breach, 1978. pages 7

[37] M. T. Chu. A simple application of the homotopy method to symmetric eigenvalue problems. *Linear Algebra and its Applications*, 59:85–90, 1984. pages 15, 101

[38] M. T. Chu and G. H. Golub. *Inverse Eigenvalue Problems: Theory, Algorithms and Applications*. Numerical Mathematics & Scientific Computations. Oxford University Press, New York, USA, 2005. pages 9

[39] F. Chung, N. M. Faber, and T. A. Manteuffel. An upper bound on the diameter of a graph from eigenvalues associated with its Laplacian. *SIAM Journal on Discrete Mathematics*, 7:443–457, 1994. pages 7

[40] G. Codevico, V. Y. Pan, M. Van Barel, X. Wang, and A. Zheng. Newton-like iteration for general and structured matrices. Technical Report TW372, Departement Computerwetenschappen, Katholieke Universiteit Leuven, November 2003. pages 97, 118

[41] L. Collatz and U. Sinogowitz. Spektren endlicher grafen. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 21:63–77, 1957. pages 2, 26

[42] J. J. M. Cuppen. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numerische Mathematik*, 36:177–195, 1981. pages 121, 127, 128, 135

[43] D. M. Cvetković. Graphs and their spectra. *Univ. Beograd. Publ. Elektrotehn. Fak. Ser. Mat. Fiz.*, 354-356:1–50, 1971. pages 26

[44] D. M. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs*. Johann Abrosius Barth Verlag, 1995. pages 2, 30

[45] F. R. de Hoog. A new algorithm for solving Toeplitz systems of equations. *Linear Algebra and its Applications*, 88/89:123–138, 1987. pages 11

[46] G. De Samblanx, K. Meerbergen, and A. Bultheel. The implicit application of a rational filter in the RKS method. Technical Report TW239, K.U.Leuven, Dept. of Computer Science, February 1996. pages 6

[47] K. Deckers and A. Bultheel. Rational Krylov sequences and orthogonal rational functions. Technical Report TW499, K.U.Leuven, Department of Computer Science, August 2007. pages 6, 60

[48] K. Deckers, J. Van Deun, and A. Bultheel. Rational Gauss-Chebyshev quadrature formulas for complex poles outside $[-1, 1]$. *Mathematics of Computation*, 77(262):967–983, 2008. pages 40

[49] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, Pennsylvania, USA, 1997. pages 55

[50] P. D. Dragnev and E. B. Saff. Constrained energy problems with applications to orthogonal polynomials of a discrete variable. *Journal d'Analyse Mathématique*, 72:223–259, 1997. pages 57

[51] T. A. Driscoll, K.-C. Toh, and L. N. Trefethen. From potential theory to matrix iterations in six steps. *SIAM Review*, 40(3):547–578, 1998. pages 5

[52] C. Dunkl and Y. Xu. *Orthogonal polynomials of several variables*. Cambridge University Press, Cambridge, UK, 2001. pages 8

[53] D. C. Dzeng and W.-W. Lin. Homotopy continuation method for the numerical solutions of generalised symmetric eigenvalue problems. *Journal of the Australian Mathematical Society*, 32(Ser. B):437–456, 1991. pages 99

[54] S. Elhay, G. H. Golub, and J. Kautsky. Updating and downdating of orthogonal polynomials with data fitting applications. *SIAM Journal on Matrix Analysis and Applications*, 12(2):327–353, 1991. pages 8, 9

[55] D. Fasino. Rational Krylov matrices and QR-steps on hermitian diagonal-plus-semiseparable matrices. *Numerical Linear Algebra with Applications*, 12(8):743–754, October 2005. pages 6

[56] D. Fasino and L. Gemignani. Direct and inverse eigenvalue problems, for diagonal-plus-semiseparable matrices. *Numerical Algorithms*, 34:313–324, 2003. pages 14

[57] H. Faßbender. On numerical methods for discrete least-squares approximation by trigonometric polynomials. *Mathematics of Computation*, 66(218):719–741, April 1997. pages 9

[58] G. Fiorentino and S. Serra-Capizzano. Multigrid methods for indefinite Toeplitz matrices. *Calcolo*, 33(3-4):223–236, September 1996. pages 12

[59] M. Fisher. On hearing the shape of a drum. *Journal of Combinatorial Theory, Series A*, 1:105–125, 1966. pages 2

[60] G. E. Forsythe. Generation and use of orthogonal polynomials for data-fitting with a digital computer. *Journal of the Society for Industrial and Applied Mathematics*, 5(2):74–88, January 1957. pages 8

[61] J. G. F. Francis. The QR transformation. A unitary analogue to the LR transformation – part 1. *The Computer Journal*, 4(3):265–271, 1961. pages 4

[62] J. G. F. Francis. The QR transformation – part 2. *The Computer Journal*, 4(4):332–345, 1962. pages 4

[63] M. Frigo and S. G. Johnson. FFTW: An adaptive software architecture for the FFT. In *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, volume 3, pages 1381–1384, Seattle, WA, May 1998. pages 118

[64] K. A. Gallivan, E. Grimme, and P. Van Dooren. A rational Lanczos algorithm for model reduction. *Numerical Algorithms*, 12:33–63, 1996. pages 6

[65] W. Gautschi. *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, New York, USA, 2004. pages 8

[66] L. Gemignani. Schur complements of Bezoutians and the inversion of block Hankel and block Toeplitz matrices. *Linear Algebra and its Applications*, 253(1–3):39–59, March 1997. pages 11

[67] C. D. Godsil. *Algebraic Combinatorics*. Chapman and Hall, 1993. pages 3, 7, 34, 35

[68] C. D. Godsil and B. D. McKay. Constructing cospectral graphs. *Aequationes Mathematicae*, 25:257–268, 1982. pages 26, 28, 29

[69] I. C. Gohberg and A. Semencul. On the inversion of finite Toeplitz matrices and their continuous analogs. *Matematicheskiie Issledovaniia*, 2:187–224, 1972. pages 12

[70] G. H. Golub and G. Meurant. *Matrices, Moments and Quadrature with Applications*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton (NJ), USA, 2009. pages 7

[71] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, USA, third edition, 1996. pages 4, 55, 97, 139

[72] G. H. Golub and J. H. Welsch. Calculation of gauss quadrature rules. *Mathematics of Computation*, 23:221–230, 1969. pages 8

[73] T. Gunji, B. K. Kim, A. Kojima, A. Takeda, K. Fujisawa, and T. Mizutani. PHoM – a polyhedral homotopy continuation method for polynomial systems. Research Report B-386, Dept. of Math. and Comp. Sciences, Tokyo Inst. of Tech., 2002. pages 14, 99

[74] Hs. H. Günthard and H. Primas. Zusammenhang von graphentheorie und MO-theorie von molekeln mit systemen konjugierter bindungen. *Helvetica Chimica Acta*, 39:1645–1653, 1956. pages 2

[75] M. H. Gutknecht and M. Hochbruck. Look-ahead Levinson and Schur algorithms for non-Hermitian Toeplitz systems. *Numerische Mathematik*, 70:181–227, 1995. pages 141, 142

[76] W. H. Haemers and E. Spence. Enumeration of cospectral graphs. *European Journal of Combinatorics*, 25(2):199–211, February 2004. pages 29, 30, 32, 151

[77] T. Hawkins. Cauchy and the spectral theory of matrices. *Historia Mathematica*, 2(1):1–29, February 1975. pages 1, 2

[78] G. Heinig. Inversion of generalized Cauchy matrices and other classes of structured matrices. In *Linear Algebra for Signal Processing*, volume 69 of *IMA Volumes in Mathematics and its Applications*, pages 95–114. Springer-Verlag, New York, 1994. pages 11

[79] G. Heinig and K. Rost. *Algebraic methods for Toeplitz-like matrices and operators*. Mathematical Research. Akademie Verlag-Berlin, 1984. pages 11, 12, 94

[80] S. Helsen, A. B. J. Kuijlaars, and M. Van Barel. Convergence of the isometric Arnoldi process. *SIAM Journal on Matrix Analysis and Applications*, 26(3):782–809, 2005. pages 61

[81] S. Helsen and M. Van Barel. A numerical solution of the constrained energy problem. *Journal of Computational and Applied Mathematics*, 189:442–452, 2006. pages 42

[82] G. M. Henry and R. A. van de Geijn. Parallelizing the QR algorithm for the unsymmetric algebraic eigenvalue problem: myths and reality. *SIAM Journal on Scientific Computing*, 17(4), July 1996. pages 14

[83] G. M. Henry, D. S. Watkins, and J. J. Dongarra. A parallel implementation of the nonsymmetric QR algorithm for distributed memory architectures. *SIAM Journal on Scientific Computing*, 24(1), 2002. pages 14

[84] D. J. Higham and N. J. Higham. *Matlab Guide*. SIAM, 2000. pages 14

[85] N. J. Higham. Algorithm 694: a collection of test matrices in MATLAB. *ACM Transactions on Mathematical Software*, 17(3), September 1991. pages 101

[86] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, 1985. pages 127

[87] L. J. Huang and T.-Y. Li. Parallel homotopy algorithm for symmetric large sparse eigenproblems. *Journal of Computational and Applied Mathematics*, 60:77–100, 1995. pages 15, 104

[88] M. Huhtanen and R. M. Larsen. On generating discrete orthogonal bivariate polynomials. *BIT*, 42(2):393–407, June 2002. pages 8, 9, 86

[89] S. Huss-Lederman, A. Tsao, and T. Turnbull. A parallelizable eigensolver for real diagonalizable matrices with real eigenvalues. *SIAM Journal on Scientific Computing*, 18(3):869–885, May 1997. pages 14

[90] C. R. Johnson and M. Newman. A note on cospectral graphs. *Journal of Combinatorial Theory, Series B*, 28(1):96–103, February 1980. pages 31

[91] M. Kac. Can one hear the shape of a drum? *American Mathematical Monthly*, 73:1–23, 1966. pages 2

[92] T. Kailath, S.-Y. Kung, and M. Morf. Displacement ranks of matrices and linear equations. *Journal of Mathematical Analysis and Applications*, 68(2):395–407, 1979. pages 11, 90

[93] T. Kailath and A. H. Sayed, editors. *Fast reliable algorithms for matrices with structure*. SIAM, Philadelphia, PA, USA, May 1999. pages 11, 90, 94, 95, 118

[94] T. Kailath, A. Vieira, and M. Morf. Inverses of Toeplitz operators, innovations and orthogonal polynomials. *SIAM Review*, 20(1):106–119, 1978. pages 11, 14

[95] T. Kato. *Perturbation theory for linear operators*. Classics in Mathematics. Springer, 1980. pages 101

[96] M. Kline. *Mathematical thought from ancient to modern times*. Oxford University Press, 1972. pages 1, 2

[97] M. A. Kowalski. Orthogonality and recursion formulas for polynomials in n variables. *SIAM Journal on Mathematical Analysis*, 13(2):316–323, 1982. pages 8

[98] P. Kravanja and M. Van Barel. A fast Hankel solver based on an inversion formula for Loewner matrices. *Linear Algebra and its Applications*, 282(1):275–295, October 1998. pages 141, 142, 143

[99] V. N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *Computational Mathematics and Mathematical Physics*, 1(3):637–657, 1963. (received Feb 1961). pages 4

[100] A. B. J. Kuijlaars. Which eigenvalues are found by the Lanczos method? *SIAM Journal on Matrix Analysis and Applications*, 22(1):306–321, 2000. pages 5, 54, 56, 58, 61

[101] A. B. J. Kuijlaars. Convergence analysis of Krylov subspace iterations with methods from potential theory. *SIAM Review*, 48(1):3–40, 2006. pages 5, 6, 57, 58, 61, 65

[102] A. B. J. Kuijlaars and P. D. Dragnev. Equilibrium problems associated with fast decreasing polynomials. *Proceedings of the American Mathematical Society*, 127(4):1065–1074, April 1999. pages 41

[103] C. Lánczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950. pages 4

[104] K. Li and T.-Y. Li. An algorithm for symmetric tridiagonal eigenproblems: divide and conquer with homotopy continuation. *SIAM Journal on Scientific and Statistical Computation*, 14(3):735–751, May 1993. pages 103

[105] T. Y. Li and N. H. Rhee. Homotopy algorithm for symmetric eigenvalue problems. *Numerische Mathematik*, 55:265–280, 1989. pages 15, 101, 120, 121, 127, 131

[106] T.-Y. Li, H. Zhang, and X.-H. Sun. Parallel homotopy algorithm for the symmetric tridiagonal eigenvalue problem. *SIAM Journal on Scientific and Statistical Computation*, 12(3):469–487, May 1991. pages 121

[107] I. Lifanov and E. E. Tyrtyshnikov. Toeplitz matrices and singular integral equations. *Vychislitelnie processi i sistemi*, 7, 1989. (In Russian). pages 11

[108] A. A. Markov. *On some applications of algebraic continued fractions*. St. Petersburg University, 1884. (in Russian). pages 7

[109] P. G. Martinsson, V. Rokhlin, and M. Tygert. A fast algorithm for the inversion of general Toeplitz matrices. *Computers & Mathematics with Applications*, 50:741–752, 2005. pages 141

[110] N. Mastronardi, M. Van Barel, and E. Van Camp. Divide-and-conquer algorithms for computing the eigendecomposition of symmetric diagonal-plus-semiseparable matrices. *Numerical Algorithms*, 39(4):379–398, 2005. pages 13, 15, 103, 120, 124, 137

[111] B. D. McKay. `nauty` User's Guide (version 1.5). Technical Report TR-CS-90-02, Dept. Computer Science, Austral. Nat. Univ., 1990. pages 32

[112] K. Meerbergen. Changing poles in the rational Lanczos method for the Hermitian eigenvalue problem. *Numerical Linear Algebra with Applications*, 8:33–52, 2001. pages 6

[113] M. Morf. *Fast algorithms for multivariable systems*. PhD thesis, Department of Electrical Engineering, Stanford University, Stanford, 1974. pages 11

[114] M. H. Oettli. *The homotopy method applied to the symmetric eigenproblem*. Dissertation no. 11176, ETH Zürich, 1995. pages 102, 103, 104, 127

[115] M. H. Oettli. A robust, parallel homotopy algorithm for the symmetric tridiagonal eigenproblem. *SIAM Journal on Scientific Computing*, 20(3):1016–1032, 1999. pages 15, 103, 121, 126, 128

[116] V. Olshevsky, I. Oseledets, and E. E. Tyrtyshnikov. *Superfast inversion of two-level Toeplitz matrices using Newton iteration and tensor-displacement structure*, volume 179 of *Operator Theory: Advances and Applications. Recent Advances in Matrix and Operator Theory*, pages 1–12. Birkhauser Basel, 2008. pages 11

[117] C. C. Paige. *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*. PhD thesis, University of London, London, United Kingdom, 1971. pages 4

[118] C. C. Paige. Error analysis of the lanczos algorithm for tridiagonalizing a symmetric matrix. *Journal of the Institute of Mathematics and its Applications*, 18:341–349, June 1976. pages 4

[119] V. Y. Pan. *Structured matrices and polynomials. Unified superfast algorithms*. Birkhäuser Springer, 2001. pages 10, 93, 94, 95, 97, 112

[120] V. Y. Pan, M. Kunin, R. E. Rosholt, and H. Kodal. Homotopic residual correction processes. *Mathematics of Computation*, 75(253):345–368, 2006. pages 97, 118

[121] V. Y. Pan and R. Schreiber. An improved Newton iteration for the generalized inverse of a matrix, with applications. *SIAM Journal on Scientific and Statistical Computation*, 12(5):1109–1130, September 1991. pages 96

[122] B. N. Parlett. *The Symmetric Eigenvalue Problem*, volume 20 of *Classics in Applied Mathematics*. SIAM, Philadelphia, Pennsylvania, USA, 1998. pages 4, 55, 56, 121, 127, 131

[123] J. Petersen and A. C. M. Ran. LU- versus UL-factorization of integral operators with semiseparable kernel. *Integral Equations and Operator Theory*, 50:549–558, 2004. pages 13

[124] S. Prössdorf and B. Silbermann. *Numerical analysis for integral and related operator equations*. Akademie-Verlag, Berlin, 1991. pages 11

[125] E. A. Rakhmanov. Equilibrium measure and the distribution of zeros of the extremal polynomials of a discrete variable. *Mathematics Sbornik*, 187(8):1213–1228, 1996. pages 5, 57

[126] S. J. Reeves. Fast algorithm for solving block banded Toeplitz systems with banded Toeplitz blocks. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 3325–3328, May 2002. pages 12

[127] L. Reichel. Fast QR-decomposition of Vandermonde-like matrices and polynomial least squares approximation. *SIAM Journal on Matrix Analysis and Applications*, 12(3):552–564, July 1991. pages 8, 9

[128] L. Reichel, G. S. Ammar, and W. B. Gragg. Discrete least squares approximation by trigonometric polynomials. *Mathematics of Computation*, 57(195):273–289, July 1991. pages 8, 9

[129] A. Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Algebra and its Applications*, 58:391–405, 1984. pages 5, 59

[130] A. Ruhe. The rational Krylov algorithm for nonsymmetric eigenvalue problems. III: Complex shifts for real matrices. *BIT*, 34:165–176, 1994. pages 5

[131] A. Ruhe. Rational krylov algorithms for nonsymmetric eigenvalue problems, II: Matrix pairs. *Linear Algebra and its Applications*, 197/198:283–296, 1994. pages 5

[132] A. Ruhe. Rational Krylov: a practical algorithm for large sparse nonsymmetric matrix pencils. *SIAM Journal on Scientific Computing*, 19(5):1535–1551, September 1998. pages 6

[133] A. Ruhe and D. Skoogh. Rational krylov algorithms for eigenvalue computation and model reduction. In B. Kågström, J. J. Dongarra, E. Elmroth, and J. Wasniewski, editors, *Applied Parallel Computing. Large Scale Scientific and Industrial Problems*, volume 1541 of *Lecture Notes in Computer Science*, pages 491–502, 1998. pages 6

[134] *GAP – Groups, Algorithms and Programming, Version 4.4.12*, 2008. pages 32

[135] Y. Saad. On the rates of convergence of the Lanczos and the block-Lanczos methods. *SIAM Journal on Numerical Analysis*, 17(5):687–706, 1980. pages 6

[136] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, Philadelphia, Pennsylvania, USA, second edition, 2003. pages 55

[137] E. B. Saff and V. Totik. *Logarithmic potentials with external fields*. Springer, Berlin, 1997. pages 40, 41

[138] G. A. Schultz. Iterative berechnung der reziproken matrix. *Zeitschrift für Angewandte Mathematik und Mechanik*, 13:57–59, 1933. pages 96

[139] O. Schütze, A. Dell'Aere, and M. Dellnitz. On continuation methods for the numerical treatment of multi-objective optimization problems. In J. Branke, K. Deb, K. Miettinen, and R. Steuer, E., editors, *Practical approaches to multi-objective optimization*, number 04461 in Dagstuhl Seminar Proceedings, Schloss Dagstuhl, Germany, 2005. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI). pages 15, 99

[140] A. J. Schwenk. *Almost all trees are cospectral*, pages 275–307. New Directions in the Theory of Graphs. Academic Press, New York, 1973. pages 2, 26, 27

[141] A. J. Schwenk. Computing the characteristic polynomial of a graph. In *Graphs and combinatorics*, volume 406 of *Lecture Notes in Mathematics*, pages 153–172, 1974. pages 7

[142] J. Seidel. A survey of two-graphs. In *Teorie Combinatorie (Proc. Intern. Coll., Roma 1973)*, pages 481–511. Accad. Nac. Lincei, Roma, 1976. pages 27

[143] S. Serra-Capizzano. New PCG based algorithms for the solution of Hermitian Toeplitz systems. *Calcolo*, 32(3-4):153–176, December 1995. pages 12

[144] S. Serra-Capizzano. Asymptotic results on the spectra of block Toeplitz preconditioned matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(1):31–44, January 1999. pages 12

[145] G. L. G. Sleijpen and A. van der Sluis. Further results on the convergence behavior of conjugate-gradients and Ritz values. *Linear Algebra and its Applications*, 246:233–278, 1996. pages 6

[146] H. P. Jr. Starr. *On the numerical solution of one-dimensional integral and differential equations*. PhD thesis, Yale university, December 1992. pages 13

[147] E. Stiefel. Über einige methoden der relaxationsrechnung. *Zeitschrift für Angewandte Mathematik und Physik*, 3(1):1–33, 1952. pages 5

[148] T. J. Stieltjes. *Recherches sur les fractions continues*, volume 8 of *Annales de la Faculté des Sciences de Toulouse*. Toulouse University, 1894. pages 7

[149] P. K. Suetin. *Orthogonal polynomials in two variables*. Gordon & Breach, Abingdon, 1999. pages 8

[150] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa. A method for solving key equation for decoding Goppa codes. *Information and Control*, 27(1):87–99, 1975. pages 11

[151] G. Szegő. *Orthogonal Polynomials*. American Mathematical Society, Providence, Rhode Island, USA, fourth edition, 1975. pages 7, 14

[152] F. Tisseur and J. J. Dongarra. A parallel divide and conquer algorithm for the symmetric eigenvalue problem on distributed memory architectures. *SIAM Journal on Scientific Computing*, 20(6):2223–2236, November 1999. pages 14

[153] L. N. Trefethen and D. Bau. *Numerical Linear Algebra.* SIAM, Philadelphia, Pennsylvania, USA, 1997. pages 4, 5

[154] M. Tsuji. *Potential theory in modern function theory.* Maruzen, Tokyo, 1959. pages 40

[155] E. E. Tyrtyshnikov. New cost-effective and fast algorithms for special classes of Toeplitz systems. *Sov. J. Numer. Anal. Math. Modelling*, 3(1):63–76, 1988. pages 141, 142

[156] E. E. Tyrtyshnikov. *Toeplitz matrices, their analogs and applications.* Russian Academy of Sciences, Section for Computational Mathematics, Moscow, 1989. (In Russian). pages 11, 94, 95

[157] M. Van Barel and A. Bultheel. Discrete least squares approximation with polynomial vectors. Technical Report TW190, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, 1993. pages 8

[158] M. Van Barel and A. Bultheel. Orthonormal polynomial vectors and least squares approximation for a discrete inner product. *Electronic Transactions on Numerical Analysis*, 3:1–23, March 1995. pages 9

[159] M. Van Barel and A. Bultheel. Look-ahead schemes for block Toeplitz systems and formal orthogonal matrix polynomials. In A. Draux and V. Kaliaguine, editors, *Proceedings of the Workshop on orthogonal polynomials: the non-definite case, Rouen, France, April 24-26, 1995*, Actes de l' atelier de Rouen, pages 93–112, 1997. pages 14

[160] M. Van Barel and A. Bultheel. Updating and downdating of orthonormal polynomial vectors and some applications. In V. Olshevsky, editor, *Structured Matrices in Mathematics, Computer Science, and Engineering II*, volume 281 of *Contemporary Mathematics*, pages 145–162. American Mathematical Society, Providence, Rhode Island, USA, 2001. pages 8, 152

[161] M. Van Barel and A. A. Chesnokov. A method to compute recurrence relation coefficients for bivariate orthogonal polynomials by unitary matrix transformations. *Numerical Algorithms*, 55:383–402, 2010. pages 8, 19, 69

[162] M. Van Barel, D. Fasino, L. Gemignani, and N. Mastronardi. Orthogonal rational functions and diagonal plus semiseparable matrices. In F. T. Luk, editor, *Advanced Signal Processing Algorithms, Architectures, and Implementations XII*, volume 4791 of *Proceedings of SPIE, Bellingham, Washington, USA*, pages 167–170, 2002. pages 14

[163] M. Van Barel, G. Heinig, and P. Kravanja. A stabilized superfast solver for nonsymmetric Toeplitz systems. *SIAM Journal on Matrix Analysis and Applications*, 23(2):494–510, 2001. pages 11, 142

[164] M. Van Barel, Kh. D. Ikramov, and A. A. Chesnokov. A continuation method for solving symmetric Toeplitz systems. *Computational Mathematics and Mathematical Physics*, 48(12):2126–2139, 2008. pages 16, 19, 107, 108

[165] E. Van Camp. *Diagonal-Plus-Semiseparable Matrices and Their Use in Numerical Linear Algebra*. PhD thesis, Katholieke Universiteit Leuven, May 2005. pages 92, 121

[166] E. R. van Dam and W. H. Haemers. Eigenvalues and the diameter of graphs. *Linear and Multilinear Algebra*, 39:33–44, 1995. pages 7

[167] E. R. van Dam and W. H. Haemers. Which graphs are determined by their spectrum? *Linear Algebra and its Applications*, 373:241–272, November 2003. pages 3, 26, 31

[168] E. R. van Dam and W. H. Haemers. Developments on spectral characterizations of graphs. *Discrete Mathematics*, 309(3):576–586, February 2009. pages 3, 29

[169] E. R. van Dam, W. H. Haemers, and J. H. Koolen. Cospectral graphs and the generalized adjacency matrix. *Linear Algebra and its Applications*, 423(1), May 2007. pages 3, 31, 34, 37

[170] A. van der Sluis and H. A. van der Vorst. The rate of convergence of the conjugate gradients. *Numerische Mathematik*, 48(5):543–560, 1986. pages 6

[171] A. van der Sluis and H. A. van der Vorst. The convergence behavior of Ritz values in the presence of close eigenvalues. *Linear Algebra and its Applications*, 88/89:651–694, 1987. pages 6

[172] J. Van Deun, K. Deckers, A. Bultheel, and J. A. C. Weideman. Algorithm 882: Near-best fixed pole rational interpolation with applications in spectral methods. *ACM Transactions on Mathematical Software*, 35(2):1–20, July 2008. Article 14. pages 49

[173] J. H. van Lint and J. Seidel. Equilateral point sets in elliptic geometry. *Proceedings Nederlandse Akademie Wetenschappen*, A69:335–348, 1966. pages 26, 27

[174] R. Vandebril, M. Van Barel, and N. Mastronardi. *Matrix Computations and Semiseparable Matrices, Volume I: Linear Systems*. Johns Hopkins University Press, Baltimore, Maryland, USA, 2008. pages 10, 92, 121, 133

[175] R. Vandebril, M. Van Barel, and N. Mastronardi. *Matrix Computations and Semiseparable Matrices, Volume II: Eigenvalue and Singular Value Methods*. Johns Hopkins University Press, 2008. pages 13, 14, 92

[176] V. V. Voevodin and E. E. Tyrtyshnikov. Computations with Toeplitz matrices. *Vychislitelnie processi i sistemi*, 1:124–266, 1983. (In Russian). pages 11

[177] V. V. Voevodin and E. E. Tyrtyshnikov. *Computational processes with Toeplitz matrices*. Nauka, 1987. (In Russian). pages 139

[178] W. Wang and C.-X. Xu. A sufficient condition for a family of graphs being determined by their generalized spectra. *European Journal of Combinatorics*, 27(6):826–840, August 2006. pages 3

[179] J. H. Wilkinson. *The algebraic eigenvalue problem*. Oxford University Press, New York, USA, 1965. pages 126, 131

[180] Y. Xu. On multivariate orthogonal polynomials. *SIAM Journal on Mathematical Analysis*, 24:783–794, 1993. pages 8

[181] Y. Xu. On discrete orthogonal polynomials of several variables. *Advances in Applied Mathematics*, 33(3):615–632, 2004. pages 8

[182] A. E. Yagle. A fast algorithm for Toeplitz-block-Toeplitz linear systems. In *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1929–1932, 2001. pages 12

Curriculum Vitae

Date and place of birth: April 21, 1983, Moscow, USSR

Education:

2007-	PhD Study at Department of Computer Science, Katholieke Universiteit Leuven, Belgium; advisor – prof. Marc Van Barel
2005-2008	PhD Study at Moscow State University, Department of Computational Mathematics and Cybernetics, Moscow, Russia; advisor – prof. Khakim Ikramov
2005	Diploma degree (Teacher), summa cum laude; advisor – Elena Andreeva
2000-2005	Studies in Pedagogics at Moscow State University, Department of Pedagogical Education, Moscow, Russia
2005	Diploma degree (Applied Mathematics), magna cum laude; advisor – prof. Khakim Ikramov
2000-2005	Studies in Applied Mathematics at Moscow State University, Department of Computational Mathematics and Cybernetics, Moscow, Russia
1990-2000	School education, Moscow (Moscow State 57th School)

Publications:

- Van Barel, Marc; Chesnokov, Andrey. A method to compute recurrence relation coefficients for bivariate orthogonal polynomials by unitary matrix transformations, Numerical Algorithms, 55 (2010), 383-402.
- Chesnokov, Andrey; Van Barel, Marc. A direct method to solve block banded block Toeplitz systems with non-banded Toeplitz blocks, Journal of Computational and Applied Mathematics, 234:5 (2010), 1485-1491.

- Chesnokov, Andrey; Deckers, Karl; Van Barel, Marc. A numerical solution of the constrained weighted energy problem, Journal of Computational and Applied Mathematics, 235:4 (Dec 2010), 950-965.
- Van Barel, Marc; Ikramov, Khakim; Chesnokov, Andrey. *A continuation method for solving symmetric Toeplitz systems.* Zh. Vychisl. Mat. Mat. Fiz., 48:12 (2008), 2092-2106, in Russian, translation in Comput. Math. Math. Phys., 48:12 (2008), 2126-2139.
- Chesnokov, Andrey; Haemers, Willem. *Regularity and the generalized adjacency spectra of graphs*, Linear Algebra Appl. 416 (2006), 1033-1037.
- Ikramov, Khakim; Matin far, Mashallah; Chesnokov, Andrey. *Computer-Algebra Procedures for the Drazin Inversion of a Singular Matrix*, Zh. Vychisl. Mat. Mat. Fiz. 44:7 (2004), 1155-1163, in Russian; translation in Comput. Math. Math. Phys. 44:7 (2004), 1093-1101.
- Ikramov, Khakim; Chesnokov, Andrey. *On the Inverses of Brownian and Brownian-Like Matrices*, Mat. Zametki, 75:1 (2004), 89-99.

Conferences:

- Chesnokov, Andrey; Van Barel, Marc; Mastronardi, Nicola; Vandebril, Raf. Homotopy algorithm for symmetric diagonal-plus-semiseparable eigenvalue problems, ICCAM 2010, Leuven, Belgium, 5-9 July 2010.
- Chesnokov, Andrey; Van Barel, Marc; Mastronardi, Nicola; Vandebril, Raf. Homotopy algorithm for symmetric diagonal-plus-semiseparable eigenvalue problems, BIT 50 – Trends in Numerical Computing, Lund, Sweden, 17-20 June 2010.
- Van Barel, Marc; Chesnokov, Andrey. Multivariable orthogonal polynomials and structured matrix computations, Dolomites Workshop on Constructive Approximation and Applications, Alba di Canazei, Val di Fassa (Trento), Italy, 4-9 September 2009
- Chesnokov, Andrey; Deckers, Karl; Van Barel, Marc. A numerical solution of the constrained weighted energy problem, Dolomites Workshop on Constructive Approximation and Applications, Alba di Canazei, Val di Fassa (Trento), Italy, 4-9 September 2009
- Chesnokov, Andrey; Van Barel, Marc. A continuation method to solve symmetric indefinite Toeplitz systems, 2nd International

Conference on Matrix Methods and Operator Equations, Moscow, Russia, July 23-27, 2007

- Chesnokov, Andrey; Van Barel, Marc. A continuation method to solve symmetric indefinite Toeplitz systems, International Workshop: Numerical Linear Algebra, Internet and Large Scale Applications, Hotel porto Giardino, Monopoli (Bari), Italy, September 9-14, 2007

Summer Schools:

- The Fourth RISC/SCIEnce Training School in Symbolic Computation, Research Institute for Symbolic Computation, Johannes Kepler University of Linz, Castle of Hagenberg, Austria, June 29-July 10, 2009.
- Linear System Theory, Control, and Matrix Computations, International Summer School, Hotel Porto Giardino, Monopoli (Bari), Italy, September 8-12, 2008

Teaching experience:

- Teaching assistant, Numerical Mathematics, 2nd year Bachelor in Engineering, Katholieke Universiteit Leuven, Belgium (2010-)
- Teaching assistant, C++ for scientific programming (Athens interuniversity program), Katholieke Universiteit Leuven, Beglium (2008-)
- Teaching assistant, Numerical Linear Algebra, 2nd year Master in Mathematical Engineering, Katholieke Universiteit Leuven, Belgium (2007-)
- Teaching assistant Linear Algebra, 1st year Bachelor of Applied Mathematics, Moscow State University, Russia (2005-2007)
- Teacher and Senior Member of the Organizing committee, Summer school for children gifted in Physics and Mathematics, Keldysh Institute for Applied Mathematics, Moscow region, Russia (2002-2007).
- Teacher of Mathematics, School for gifted children “Intellectual”, Moscow, Russia (2003-2007)