Denis Chaykin

Verification of Semidefinite Optimization Problems with Application to Variational Electronic Structure Calculation

Cuvillier Verlag Göttingen
Internationaler wissenschaftlicher Fachverlag

Verification of Semidefinite Optimization Problems with Application to Variational Electronic Structure Calculation

Vom Promotionsausschuss der
Technischen Universität Hamburg-Harburg
zur Erlangung des akademischen Grades
Doktor-Ingenieur
genehmigte Dissertation

von
Denis Chaykin

aus
Krasnogorsk

2009

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.
1. Aufl. - Göttingen : Cuvillier, 2009
Zugl.: (TU) Hamburg-Harburg, Univ., Diss., 2009

978-3-86955-066-4

1. Gutachter: Prof. Dr. Dr. h.c. Frerich J. Keil

2. Gutachter: PD Dr. Christian Jansson

Tag der mündlichen Prüfung: 19.06.2009

© CUVILLIER VERLAG, Göttingen 2009
Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.
1. Auflage, 2009
Gedruckt auf säurefreiem Papier

978-3-86955-066-4

Acknowledgements

Naturally, my greatest appreciation goes to my advisors Christian Jansson and Frerich Keil. Christian Jansson's dedicated involvement has actually made this thesis possible. His knowledge, experience and insight are directly and indirectly reflected in this work. He was always the first person to consult with and to get advise from whenever I had to face any difficulties. I will be forever grateful for the given chance.

I want to thank Prof. Frerich Keil for the provided opportunity to continue my research, for his inspiring supervision, patience and understanding. Without his encouragement, it would be difficult to bring this thesis to its successful completion.

To all my colleagues, both in the Institute for Reliable Computing and in the Institute of Chemical Reaction Engineering, thank you for your help, support and for just making it a nice time.

Finally, on a more personal note, I would like to express my deepest gratitude and appreciation to my family for their understanding and encouragement. Especially I thank my wife Pei-Chi for many years of unconditional support and love. Without her by my side, this thesis could never be brought out.

Abstract

In this thesis we develop ideas of rigorous verification in optimization. Semidefinite programming (SDP) is reviewed as one of the fundamental types of convex optimization with a variety of applications in control theory, quantum chemistry, combinatorial optimization as well as many others. We show, how rigorous error bounds for the optimal value can be computed by carefully postprocessing the output of a semidefinite programming solver. All the errors due to the floating point arithmetic or ill-conditioning of the problems are considered. We also use interval arithmetic as a powerful tool to model uncertainties in the input data.

In the context of this thesis a software package implementing the verification algorithms was developed. We provide detailed explanations and show how efficient routines can be designed to manage real life problems. Criteria for detecting infeasible semidefinite programs and issuing certificates of infeasibility are formulated. Examples and results for benchmark problems are included.

Another important contribution is the verification of the electronic structure problems. There large semidefinite programs represent a reduced density matrix variational method. Our algorithms allow the calculation of a rigorous lower bound for the ground state energy. The obtained results and modified algorithms are also of importance because they show how much we can benefit in terms of problem complexity from exploiting the specific problem structure.

Contents

Contents **vii**

List of Tables **ix**

1 Introduction **1**

1.1 Overview and motivation 1

1.2 Outline 4

2 Theory and algorithms **6**

2.1 Semidefinite programming 6

2.2 Notation 10

2.3 Interval arithmetic 10

2.4 Rigorous error bounds in semidefinite programming 13

2.4.1 Rigorous lower bound 14

2.4.2 Rigorous upper bound 19

2.4.3 Verification of ill-posed problems 23

2.5 Certificates of infeasibility 24

3 Rigorous error bounds for RDM variational problems **27**

3.1 Introduction 27

3.2 Reduced density matrices 29

3.3 N-representability 31

3.4 Eigenvalue bounds of the problem structures 34

3.4.1 Maximal eigenvalues of 1-RDMs and 2-RDMs 35

3.4.2 Maximal eigenvalues of other matrices 38

3.4.3 Considerations 40

3.5 SDP implementations 43

3.5.1 Rigorous error bounds for the RDM method in “dual” SDP formulation 45
3.5.2 Numerical results 51
3.6 Discussion 58

4 Rigorous error bounds with verifiedSDP 59
4.1 Implementation of algorithms computing rigorous results 60
4.1.1 Calculating a lower eigenvalue bound of an interval matrix . . 60
4.1.2 Solving linear systems rigorously 62
4.1.3 Checking SDP infeasibility 62
4.2 Numerical results for the SDPLIB 63

5 Conclusions 70

A Phase I methods in semidefinite programming 72
A.1 Dual feasibility 72
A.2 Primal feasibility 74

B verifiedSDP quick reference 77
B.1 Installation 77
B.2 Examples of use 79
B.2.1 Stand-alone executable binary 79
B.2.2 Using verifiedSDP library 79

Bibliography 81

Lebenslauf 89

List of Tables

3.1 Rigorous bounds for the electronic structure benchmark problems. . . 51

3.2 Accuracy of the rigorous bounds for the electronic structure problems. 53

3.3 Computational efforts for the electronic structure benchmark problems. 54

3.4 Ratios of maximal eigenvalues and upper bounds. 57

4.1 Approximate optimal values and rigorous bounds for the SDPLIB problems. 63

4.2 Accuracy and computational effort for the SDPLIB problems. 66

Chapter 1

Introduction

1.1 Overview and motivation

A wide variety of problems in global optimization, combinatorial optimization, as well as systems and control theory can be solved by using semidefinite programming. Sometimes, due to the use of floating point arithmetic in combination with ill-conditioning and degeneracy, erroneous results may be produced. The purpose of this work is to show how rigorous error bounds for the optimal value can be computed by carefully postprocessing the output of a semidefinite programming solver. It turns out that in many cases the computational costs for postprocessing are small and reasonable compared to the effort required by the solver.

Semidefinite programming together with linear programming and second order cone programming is a special case of conic optimization. It is documented by many applications and a number of survey papers (see, for example, Skelton and Iwasaki [70], Balakrishnan and Feron [5], and Vandenberghe and Boyd [74]). In the context of linear matrix inequalities (LMI) it is well known since more than 100 years. In his work *Concerning the constant rotational motion of rigid bodies in fluids* (1890) Lyapunov showed that the differential equation

$$\frac{d}{dt}x(t) = Ax(t) \tag{1.1}$$

is stable (i.e., all trajectories converge to zero) if and only if there exists a positive definite matrix P such that

$$A^T P + PA \prec 0. \tag{1.2}$$

The requirement $P \succ 0, A^T P + PA \prec 0$ is what we now call a Lyapunov inequality on P, which is a special form of an LMI (Boyd et al. [8]). System and control theory is in general a very wide and well studied application domain of semidefinite programming. Thus Lyapunov ideas found their further development in works of Lur'e, Postnikov and, later, Yakubovich (see e.g. [45], [77], [76]). More recent results can be found in the works of Kamenetskii [38] and [39], and Pyatnitskii [62] published in the 1980's and early 1990's.

Nevertheless, the use of LMIs in control is not limited to the Lyapunov stability theory alone. Many interesting results have been achieved by using LMIs for pole placement (Chilali, Gahinet and Apkarian [10]) and shaping stability regions for systems with saturation (Paré et al. [60], and Hindi and Boyd [28]). Different LMI based observer design techniques for state or input estimation can be found among others in Ha and Trinh [24], and Arcak and Kokotović [4]. A good overview of different aspects of semidefinite programming in control can be found in the work of Parrilo and Lall [61], and, of course in the book by Boyd, Ghaoui, Feron and Balakrishnan [8]. Finally, the article by Henrion and Šebek [27] illustrates the use of LMIs for solving control problems with polynomial methods.

Control theory is, without a doubt, one of the main applications of semidefinite programming, but by far not the only one. Other important application fields include global optimization problems, combinatorial optimization (see e.g. Jansson [34] or Helmberg [26]), integer programming problems, process engineering (see e.g. Balakrishnan et al. [6], Chmielewski et al. [11]), as well as eigenvalue problems in the form of minimizing the largest, or minimizing the sum of the first few largest eigenvalues of a symmetric matrix X subject to linear constraints on X. And, of course, we cannot omit the use of semidefinite programming in quantum chemistry.

In Chapter 3 of this thesis we discuss the application of SDP methods to quantum chemical problems in more details. There we use the reduced density matrix (RDM) formulation of the electronic structure problem to determine a ground state energy of a system of N electrons. The method is well known and its history can be traced back to the works of Coleman [12] and Garrod and Percus [21]. In our work we follow the approach described in the paper of Fukuda et al. [19] to get an SDP formulation of the problem. With the help of rigorously bounding algorithms from Chapter 2 we then can find an accurate and reliable lower bound for the ground state energy. The RDM based electronic structure problem was chosen as an application example for our verification algorithms not only because of its high practical importance, but also because by producing large and not necessarily well-conditioned SDP problems, it sends a real numerical challenge.

Semidefinite programs can be solved in polynomial time if an a priori bound for the size of their solution is known (see Grötschel, Lovász, and Schrijver [23]). This is a consequence of the ellipsoid method for convex programming. The ellipsoid method has not proven practical, and interior point methods turned out to be the methods of choice in semidefinite programming.

Conventionally, algorithms assume that the input data are given exactly, and they use floating point arithmetic for computing an approximate solution. Occasionally, wrong results may be produced, not solely but especially for ill-conditioned and ill-posed problems in the sense defined by Renegar [64]. He defines the condition number as the scale-invariant reciprocal of the smallest data perturbation that will render the perturbed data instance either primal or dual infeasible. It is set to ∞ if the distance to primal or dual infeasibility is 0, and in this case the problem is called ill-posed. Examples where commercial solvers fail to solve linear optimization problems can be found in Neumaier and Shcherbina [57] and in [33]. It cannot be answered how frequently such failures occur. Ill-conditioning is, however, frequently observed. In a paper by Ordóñez and Freund [58] it is stated that 71% of the LP instances in the NETLIB Linear Programming Library [53] are ill-posed; i.e., the problems have an infinite condition number. Recently, Freund, Ordóñez, and Toh [15] solved 85 out of the 92 problems of the SDPLIB [7] with SDPT3 [73] and investigated the interior point iteration counts with respect to different measures for semidefinite programming problems. They omitted the four infeasible problems and three very large problems where SDPT3 ran out of memory. Of the remaining 85 problems they have shown 32 to be ill-posed.

As pointed out in Neumaier and Shcherbina [57], ill-conditioning is also likely to take place in combinatorial optimization when branch-and-cut procedures sequentially generate linear or semidefinite programming relaxations. Therefore, the computation of rigorous error bounds, which take account of all rounding errors and of small errors in the input data, is valuable in practice.

The primary purpose of this thesis is to show that by properly postprocessing the output of a semidefinite solver, rigorous error bounds for the optimal value can be obtained. Moreover, existence of optimal solutions can be proved, or a certificate of infeasibility can be given. The input data are allowed to vary within small intervals. Our numerical experience with the SDPLIB demonstrates that, roughly speaking, rigorous lower and upper error bounds for the optimal value are computed even for ill-conditioned and degenerate problems. The quality of the error bounds depends on the quality of the computed approximations and the distances to dual and primal infeasibility. By comparing these bounds, one knows whether the computed results are good.

Furthermore, we apply the developed algorithms to the electronic structure problem and thus calculate a rigorous lower bound of the ground state energy of N-fermion atomic-molecular systems. We also show how one can use specific problem information to adapt the algorithms to improve their verification quality in terms of performance and precision.

The structure of this thesis is detailed in the next section.

1.2 Outline

The thesis is organized as follows.

- Chapter 2 contains basics of semidefinite programming and interval arithmetic which are followed by our verification methods. We introduce an algorithm for computing a rigorous lower bound of the global minimum value, and present a rigorous upper bound of the optimal value together with a certificate of existence of optimal solutions. Later we show how these rigorous bounds can be used for obtaining certificates of infeasibility. The content of Chapter 2 reflects, for the most part, our results published in [37].

- Optimal value bounding algorithms are then applied to the electronic structure problem in Chapter 3. For a N-electron system in RDM formulation different available SDP representations are considered. We obtain eigenvalue bounds of reduced density matrices and other problem structures present in SDP formulations. Later we introduce two different approaches for calculating a rigorous lower bound to the ground state energy of such electron systems and compare them. Finally, numerical results for sample problems are presented.

- In Chapter 4 we describe verifiedSDP, a software package for rigorous verification of semidefinite programming problems developed in the context of our work. We explain some implementation details and specify functionality of the software. The chapter also contains numerical results for problems of the SDPLIB library.

- Chapter 5 concludes the thesis with a short summary.

- Methods of finding a starting feasible point or proving infeasibility of semidefinite programming problems, known as phase I methods, are described in detail in Appendix A. We elaborately discuss both primal and dual semidefinite programs and give examples. Special attention is payed to possible difficulties that can arise when solving derived phase I problems.

- Finally, in Appendix B a short verifiedSDP instruction manual is given. There we explain the package structure as well as introduce some parameters and methods that can be of interest to the potential users of the software. Some examples are also presented.

Chapter 2

Theory and algorithms

In this chapter the ideas behind rigorous verification will be presented. For a start, we give a detailed introduction into semidefinite programming. Notations and techniques of interval arithmetic are provided later. This allows for a more general approach to the rigorous verification (uncertainties in the input data can thus be involved in the computation). The actual theorems and algorithms for rigorously bounding the optimal value are presented in the subsequent sections. Finally, an important case of infeasible problems is examined in detail.

2.1 Semidefinite programming

Let us first define a *semidefinite program* in its primal form

$$\begin{aligned} p^* := \min \langle C, X\rangle \quad \text{s.t.} \quad & \langle A_i, X\rangle = b_i \quad \text{for } i = 1, \ldots, m, \\ & X \succeq 0, \end{aligned} \tag{2.1}$$

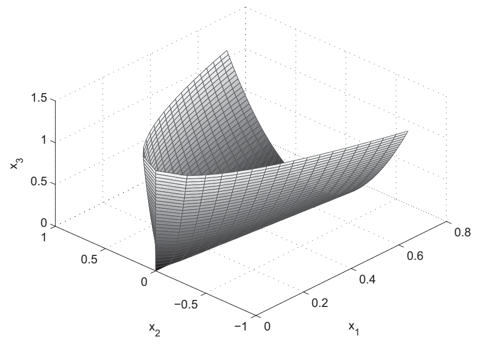
where $C \in \mathbb{S}^s$, $A_i \in \mathbb{S}^s$ and $b \in \mathbb{R}^m$ are given problem parameters, and $X \in \mathbb{S}^s$ is the optimization variable. Here, $\mathbb{S}^s$ denotes the space of real symmetric matrices of order s.

$$\langle C, X\rangle = \text{ trace } (C^T X) \tag{2.2}$$

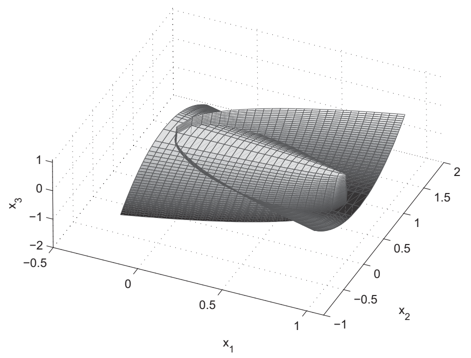
in its turn denotes the *inner product* over $\mathbb{S}^s$. Moreover, $\succeq$ is the *Löwner partial order*, that is $X \succeq Y$ iff $X - Y$ is positive semidefinite.

The *Lagrangian dual* of (2.1) is

$$d^* := \max b^T y \quad \text{s.t.} \quad \sum_{i=1}^{m} y_i A_i \preceq C, \tag{2.3}$$



(a)



(b)

Figure 2.1: Semidefinite cones for $s = 2$: $X \succeq 0$ (left) and $0 \preceq X \preceq I$ (right).

where $y \in \mathbb{R}^m$. The constraints $\sum_{i=1}^m y_i A_i \preceq C$ are called *linear matrix inequalities (LMI)*. We use the convention that $p^* = -\infty$ if (2.1) is unbounded and $p^* = \infty$ if (2.1) is infeasible. The analogous convention is used for (2.3).

Since we have formulated semidefinite programming in its standard form, it is easy to see, that the optimization domain is the intersection of the cone of positive semidefinite matrices with an affine space. The objective function is linear. The introduced problem can thus be seen as a subclass of cone programming and also as a generalization of linear programming. Indeed, if we demand all symmetric matrices to be diagonal, (2.1) will define a standard linear programming problem.

Example 2.1. *To get basic ideas about the geometry of the problem, let us consider the simplest case of $s = 2$ and a single equality constraint ($m = 1$) in Figures 2.1 and 2.2. The positive semidefiniteness condition*

$$X = \begin{pmatrix} x_1 & x_2 \\ x_2 & x_3 \end{pmatrix} \succeq 0 \tag{2.4}$$

is fulfilled for all the points in the interior and on the boundary of the cone in Figure 2.1a. The intersection of two semidefinite cones, shown in Figure 2.1b, is also a typical configuration. We have this situation, for example, in Chapter 3, where the condition on one-particle reduced density matrix is exactly $0 \preceq$ 1RDM $\preceq I$.

For the semidefinite optimization example in Figure 2.2 we use the following data:

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} 2 & 0.5 \\ 0.5 & 1 \end{pmatrix} \quad \textit{and} \quad b = 1. \tag{2.5}$$

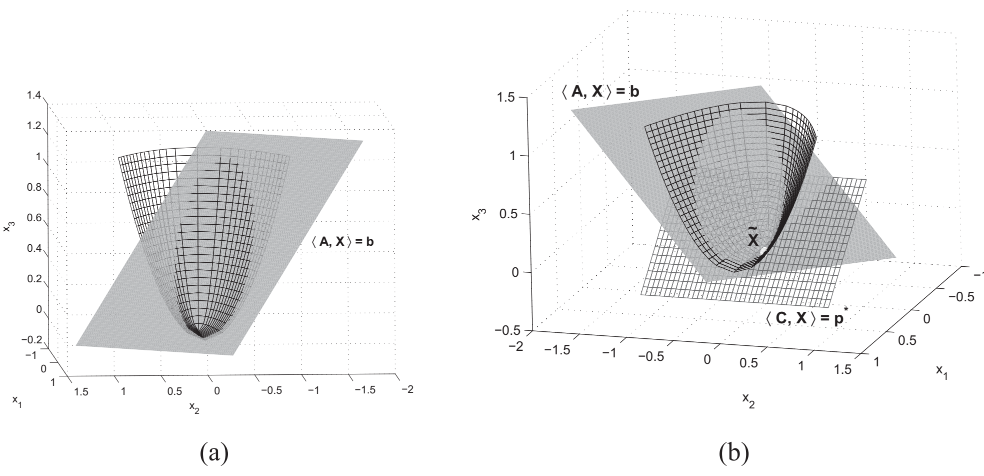


Figure 2.2: Intersection of the semidefinite cone from Figure 2.1a with the linear constraint plane (left) and the objective function plane passing through the optimal solution $\tilde{X}$ (right).

The solution of the obtained semidefinite optimization problem is

$$\tilde{X} = \begin{pmatrix} 0.3867 & 0.1602 \\ 0.1602 & 0.0664 \end{pmatrix} \tag{2.6}$$

and the objective value $p^* = \langle C, \tilde{X} \rangle = 0.4531$. *When trying to graphically solve the problem, the idea is again similar to that in LP. We shift the plane* $\langle C, X \rangle = const$ *downwards until we reach the edge of the feasibility region. In our example we have to stop at* $const = 0.4531$. $\tilde{X}$ *is then the only intersection point of the cone* $X \succeq 0$, *the affine constraint* $\langle A, X \rangle = b$ *and the objective function plane* $\langle C, X \rangle = 0.4531$.

As a standard solution algorithm, interior point methods proved to be an approach of choice. They have their roots in Karmarkar's work [40], where he introduced an algorithm to solve an LP with polynomial iteration complexity[1]. As applied to SDPs, the idea found its development in the works of Nesterov and Nemirovski (see for example [51] and [52]) and Alizadeh [2]. The concept, similarly to any other barrier function method, would be to substitute the initial problem with an optimization problem (more precisely a sequence of them) without the semidefiniteness constraint

$$p^* := \min \langle C, X \rangle + \mu \phi(X) \quad \text{s.t.} \quad \langle A_i, X \rangle = b_i \quad \text{for } i = 1, \ldots, m, \tag{2.7}$$

[1]However the development of the idea can be tracked back to the works of Frisch [16] on logarithmic barrier functions and Huard [30] on the method of centers.

where $\phi(X)$ is the *barrier function* and $\mu > 0$ is the *barrier parameter*. Standard barrier function would be a logarithmic function of the type

$$\begin{aligned} &\phi(X) = -\ln\det X = \ln(\det X)^{-1} \quad \text{if} \quad X \succ 0, \\ &\phi(X) = +\infty \quad \text{otherwise.} \end{aligned} \tag{2.8}$$

Later on by sequentially decreasing μ towards 0, we solve the initial problem.

The duality theory of semidefinite programming is a bit more subtle compared to linear programming. The programs satisfy the *weak duality* condition

$$d^* \leq p^*, \tag{2.9}$$

but strong duality requires in contrast to linear programming additional conditions (see for example Nemirovski [50], Ramana, Tunçel, and Wolkowicz [63] and Vandenberghe and Boyd [74]).

Theorem 2.1 (Strong Duality Theorem)**.**

a) If (2.1) is strictly feasible (i.e. there exists a feasible positive definite matrix X) and p^* is finite, then $p^* = d^*$ and the dual supremum is attained.

b) If (2.3) is strictly feasible (i.e. there exists some $y \in \mathbb{R}^m$ such that $C - \sum_{i=1}^m y_i A_i$ is positive definite) and d^* is finite, then $p^* = d^*$, and the primal infimum is attained.

In general, one of the problems (2.1) and (2.3) may have optimal solutions while its dual is infeasible, or the duality gap may be positive at optimality. The strict feasibility assumptions in Theorem 2.1 are called *Slater's constraint qualifications*.

As a matter of convenience, in the rest of the thesis semidefinite programs will be considered in more general block diagonal form reflecting the sparsity of the problem. The primal problem (2.1) becomes then

$$\begin{aligned} p^* := \min \sum_{j=1}^n \langle C_j, X_j \rangle \quad \text{s.t.} \quad &\sum_{j=1}^n \langle A_{ij}, X_j \rangle = b_i \quad \text{for } i = 1, \ldots, m, \\ &X_j \succeq 0 \quad \text{for } j = 1, \ldots, n, \end{aligned} \tag{2.10}$$

where $C_j \in \mathbb{S}^{s_j}$, $A_{ij} \in \mathbb{S}^{s_j}$ and $X_j \in \mathbb{S}^{s_j}$. Finally, instead of the dual problem (2.3) we now have

$$d^* := \max b^T y \quad \text{s.t.} \quad \sum_{i=1}^m y_i A_{ij} \preceq C_j \quad \text{for } j = 1, \ldots, n. \tag{2.11}$$

2.2 Notation

Throughout this thesis we use the following notation. $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^n_+$, and $\mathbb{R}^{m\times n}$ denote the sets of real numbers, real vectors, real nonnegative vectors, and real $m \times n$ matrices, respectively. $\mathbb{S}^n$, in its turn, stands for the set of real symmetric matrices. Comparisons $\leq$, absolute value $|\cdot|$, $\min$, $\max$, $\inf$ and $\sup$ are used entrywise for vectors and matrices. The identity matrix is denoted by I.

For a symmetric matrix A the eigenvalues are sorted non-increasingly, $\lambda_{\max}(A) = \lambda_1(A) \geq \lambda_2(A) \geq \ldots \geq \lambda_{\min}(A)$.

For $\mu \in \mathbb{R}$ the operator

$$\text{svec}(A, \mu) := (A_{11}, \mu A_{21}, \ldots, \mu A_{n1}, A_{22}, \mu A_{32}, \ldots, \mu A_{n\,n-1}, A_{nn})^T, \tag{2.12}$$

transforms symmetric $n \times n$ matrices into $((n+1)n/2)$-dimensional vectors with the property that the inner product of two symmetric matrices A, B is

$$\langle A, B\rangle = \text{svec}(A, 2)^T \text{svec}(B, 1) = \text{svec}(A, \sqrt{2})^T \text{svec}(B, \sqrt{2}), \tag{2.13}$$

and $\text{svec}(A, \sqrt{2})$ is the customary svec operator. We prefer the first representation of the inner product, since this avoids conversion errors of the input data of semidefinite programs in its vector representation form. The inverse operator of svec is denoted by $\text{smat}(a, \mu)$, where a is the vector representation (2.12).

For block matrices with blocks A_j for $j = 1, \ldots, n$ we define the concatenated vector

$$\text{svec}((A_j), \mu) := (\text{svec}(A_1, \mu); \ldots; \text{svec}(A_n, \mu)). \tag{2.14}$$

A block diagonal matrix with blocks $B_1, \ldots, B_n$ will be written as

$$\text{Diag}(B_1, \ldots, B_n). \tag{2.15}$$

Other necessary notation concerning, for example, interval arithmetic, will be introduced in the corresponding sections.

2.3 Interval arithmetic

Rigorous verification requires to consider rounding errors of the floating point arithmetic. One needs tools to control machine rounding and to estimate error propagation. Interval arithmetic provides us with such tools. Besides that, in real life applications many values or model parameters are measurement results. Since no devices possess infinite precision, such values have to be considered with measurement errors. To cope with this, we allow interval input in all problems discussed in the thesis.

We require only some elementary facts about interval calculations, which are described here. There are a number of textbooks on interval arithmetic and self-validating methods that can be highly recommended to readers. These include Alefeld and Herzberger [1], Moore [49], and Neumaier [54], [55].

If $\mathbb{V}$ is one of the spaces $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$, and $\underline{v}, \overline{v} \in \mathbb{V}$, then the box

$$\mathbf{v} := [\underline{v}, \overline{v}] := \{v \in \mathbb{V} : \ \underline{v} \le v \le \overline{v}\} \tag{2.16}$$

is called an *interval quantity* in $\mathbb{IV}$ with *lower bound* $\underline{v}$ and *upper bound* $\overline{v}$. In particular, $\mathbb{IR}$, $\mathbb{IR}^n$, and $\mathbb{IR}^{m\times n}$ denote the set of real intervals $\mathbf{a} = [\underline{a}, \overline{a}]$, the set of real interval vectors $\mathbf{x} = [\underline{x}, \overline{x}]$, and the set of real interval matrices $\mathbf{A} = [\underline{A}, \overline{A}]$, respectively. The real operations $A \circ B$ with $\circ \in \{+, -, \cdot, /\}$ between real numbers, real vectors and real matrices can be generalized to *interval operations*. The result $\mathbf{A} \circ \mathbf{B}$ of an interval operation is defined as the interval hull of all possible real results, that is

$$\mathbf{A} \circ \mathbf{B} := \cap\{\mathbf{C} \in \mathbb{IV} : \ A \circ B \in \mathbf{C} \quad \text{for all} \quad A \in \mathbf{A}, B \in \mathbf{B}\}. \tag{2.17}$$

All interval operations can be easily executed by working appropriately with the lower and upper bounds of the interval quantities. In the simple cases of addition and subtraction, we obtain

$$\begin{aligned} \mathbf{A} + \mathbf{B} &= [\underline{A} + \underline{B}, \overline{A} + \overline{B}], \\ \mathbf{A} - \mathbf{B} &= [\underline{A} - \overline{B}, \overline{A} - \underline{B}]. \end{aligned} \tag{2.18}$$

Interval multiplications and divisions require a distinction of cases. Let $\mathbf{a} = [\underline{a}, \overline{a}] \in \mathbb{IR}$ and $\mathbf{b} = [\underline{b}, \overline{b}] \in \mathbb{IR}$, then

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} &:= [\min\{\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b}\}, \max\{\underline{a}\underline{b}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{a}\overline{b}\}], \\ \mathbf{a}/\mathbf{b} &:= [\underline{a}, \overline{a}] \cdot \left[\frac{1}{\overline{b}}, \frac{1}{\underline{b}}\right], \quad \text{assuming} \quad 0 \notin \mathbf{b}. \end{aligned} \tag{2.19}$$

The rules of commutativity and associativity remain valid also for operations on $\mathbb{IR}$. The sub-distributivity rule

$$\mathbf{a}(\mathbf{b} + \mathbf{c}) \subseteq \mathbf{ab} + \mathbf{ac} \tag{2.20}$$

with $\mathbf{c} \in \mathbb{IR}$ substitutes the distributivity from $\mathbb{R}$. $\mathbf{x} = [0, 0]$ and $\mathbf{y} = [1, 1]$ are the unique neutral elements with respect to addition and multiplication. A fundamental property of interval arithmetic is inclusion monotonicity:

$$\mathbf{a} \subseteq \mathbf{a}',\ \mathbf{b} \subseteq \mathbf{b}' \quad \Rightarrow \quad \mathbf{a} \circ \mathbf{b} \subseteq \mathbf{a}' \circ \mathbf{b}', \quad \circ \in \{+,-,\cdot,/\}. \tag{2.21}$$

This property follows directly from the set-theoretical definitions of the interval arithmetic operations (2.17). Thus rational interval functions are inclusion monotonic, as are natural interval extensions of all the standard functions used in computing. With proper rounding procedures, rounded interval arithmetic operations are also inclusion monotonic (Moore [49]).

Similarly all operations (2.17) between interval vectors and interval matrices can be executed by replacing every real operation by the corresponding interval operation. For example the i, j component of the product of two interval matrices $\mathbf{C}, \mathbf{X} \in \mathbb{IR}^{n\times n}$ is

$$(\mathbf{CX})_{ij} := \sum_{k=1}^{n} \mathbf{C}_{ik}\mathbf{X}_{kj}, \tag{2.22}$$

and the inner product

$$\langle \mathbf{C}, \mathbf{X} \rangle = \text{ trace } (\mathbf{C}^T\mathbf{X}) = \sum_{i,j=1}^{n} \mathbf{C}_{ij}\mathbf{X}_{ij}. \tag{2.23}$$

For interval quantities $\mathbf{A}, \mathbf{B} \in \mathbb{IV}$ we define

$$\text{mid}\mathbf{A} := (\underline{A} + \overline{A})/2 \quad \text{as the } \textit{midpoint}, \tag{2.24}$$

$$\text{rad}\mathbf{A} := (\overline{A} - \underline{A})/2 \quad \text{as the } \textit{radius,} \tag{2.25}$$

$$|\mathbf{A}| := \sup\{|A| : A \in \mathbf{A}\} \quad \text{as the } \textit{absolute value}, \tag{2.26}$$

$$\mathbf{A}^+ := \max\{0, \overline{A}\}, \tag{2.27}$$

$$\mathbf{A}^- := \min\{0, \underline{A}\}. \tag{2.28}$$

Moreover, the comparison in $\mathbb{IV}$ is defined by

$$\mathbf{A} \leq \mathbf{B} \quad \text{iff} \quad \overline{A} \leq \underline{B},$$

and other relations are defined analogously. Real quantities v are embedded in the interval quantities by identifying $v = \mathbf{v} = [v, v]$.

We call $\mathbf{A} \in \mathbb{IR}^{n\times n}$ *symmetric*, if $\mathbf{A}_{ij} = \mathbf{A}_{ji}$ for all i, j, and $\mathbf{A}$ is called positive semidefinite if all $A \in \mathbf{A}$ have this property.

For linear systems of equations with inexact input data, the aim frequently is to compute an interval vector $\mathbf{x} \in \mathbb{IR}^n$ containing the *solution set*

$$\Sigma(\mathbf{A}, \mathbf{b}) := \{x \in \mathbb{R}^n : \ Ax = b \text{ for some } (A, b) \in (\mathbf{A}, \mathbf{b})\}, \tag{2.29}$$

where $\mathbf{A} \in \mathbb{IR}^{n\times n}$, and $\mathbf{b} \in \mathbb{IR}^n$. This is an NP-hard problem, but there are several methods that compute enclosures $\mathbf{x}$. A precise description of such methods, required assumptions, and approximation properties can be found for example in Neumaier [54]. Roughly speaking, it turns out that for interval matrices with $\|I - R\mathbf{A}\| < 1$ (R is an approximate inverse of the midpoint mid$\mathbf{A}$) there are several methods which compute an enclosure $\mathbf{x}$ with $O(n^3)$ operations. The radius rad$\mathbf{x}$ decreases linearly with decreasing radii rad$\mathbf{A}$ and rad$\mathbf{b}$. For the computation of enclosures in the case of large-scale linear systems the reader is referred to Rump [66].

In interval arithmetic several methods for computing rigorous bounds for all or some eigenvalues of interval matrices were developed. Some important references are Floudas [14], Mayer [46], Neumaier [56], and Rump [66, 67]. We will describe the used algorithm for computing a lower eigenvalue bound of an interval matrix in detail in Section 4.1.1.

2.4 Rigorous error bounds in semidefinite programming

In this thesis we show that by properly postprocessing the output of a semidefinite solver, rigorous error bounds for the optimal value can be obtained. Moreover, existence of optimal solutions can be proved, or a certificate of infeasibility can be given. The quality of the error bounds depends on the quality of the computed approximations and the distances to dual and primal infeasibility. It is typical that either no finite rigorous bounds or bounds being not close-by are computed if the solver gives bad approximations. By comparing these bounds one knows whether the computed results are good. Our numerical experience demonstrates that, roughly speaking, rigorous lower and upper error bounds for the optimal value are computed even for ill-conditioned and degenerate problems.

In the following sections we present theoretical results on optimal value bounding as well as the corresponding algorithmic frameworks. The special case of ill-posed problems is dealt with in section 2.4.3.

2.4.1 Rigorous lower bound

In many applications some or all input data are uncertain. We model these uncertainties by intervals. In the case of semidefinite programming we assume that symmetric interval matrices $\mathbf{C}_j, \mathbf{A}_{ij} \in \mathbb{IR}^{s_j \times s_j}$, $i = 1, \ldots, m$, $j = 1, \ldots, n$, and an interval vector $\mathbf{b} \in \mathbb{IR}^m$ are given. This yields a family of semidefinite programs (2.10), where the input data $P = (A, b, C)$ are allowed to vary within interval bounds $\mathbf{P} := (\mathbf{A}, \mathbf{b}, \mathbf{C})$.

In order to indicate the dependency on the input data, we sometimes write $p^*(P)$, $d^*(P)$, $X^*(P)$, etc.

First, we state a lemma proving a lower bound for the inner product of two symmetric matrices.

Lemma 2.1. *Let D, X be symmetric matrices of dimension s that satisfy*

$$\underline{d} \leq \lambda_{\min}(D), \quad 0 \leq \lambda_{\min}(X), \quad \text{and} \quad \lambda_{\max}(X) \leq \overline{x}. \tag{2.30}$$

Then

$$\langle D, X \rangle \geq s \cdot \underline{d}^- \cdot \overline{x}, \tag{2.31}$$

where $\underline{d}^- := \min\{0, \underline{d}\}$.

Proof. Let D have the eigenvalue decomposition

$$D = Q\Lambda(D)Q^T, \quad QQ^T = I,$$

where $\Lambda(D)$ is the diagonal matrix with eigenvalues of D on the diagonal. Then

$$\begin{aligned} \langle D, X \rangle &= \text{trace}(Q\Lambda(D)Q^T X) \\ &= \text{trace}(\Lambda(D)Q^T XQ) \\ &= \sum_{k=1}^{s} \lambda_k(D) Q(:,k)^T X Q(:,k). \end{aligned}$$

Because of (2.30), we have $0 \leq Q(:,k)^T X Q(:,k) \leq \overline{x}$ yielding

$$\langle D, X \rangle \geq \sum_{k=1}^{s} \lambda_k(D)^- \cdot \overline{x} \geq s \cdot \underline{d}^- \cdot \overline{x}.$$

□

We are now ready to prove a rigorous lower bound for the optimal value p^*.

Theorem 2.2. *Let $\mathbf{P}$ define a family of semidefinite programs (2.10) with input data $P \in \mathbf{P}$, let $\tilde{y} \in \mathbb{R}^m$, set*

$$\mathbf{D}_j := \mathbf{C}_j - \sum_{i=1}^{m} \tilde{y}_i \mathbf{A}_{ij} \quad \textit{for } j = 1, \ldots, n, \tag{2.32}$$

and suppose that

$$\underline{d}_j \le \lambda_{\min}(\mathbf{D}_j) \quad \textit{for } j = 1, \ldots, n. \tag{2.33}$$

Assume further that upper bounds for the maximal eigenvalues of the primal feasible solution of (2.10)

$$\lambda_{\max}(X_j) \le \overline{x}_j, \quad \textit{for } j = 1, \ldots, n \tag{2.34}$$

are known, where $\overline{x}_j$ may be infinite. If

$$\underline{d}_j \ge 0 \quad \textit{for } \overline{x}_j = +\infty, \tag{2.35}$$

then for every $P \in \mathbf{P}$ the inequality

$$p^*(P) \ge \inf\{\mathbf{b}^T \tilde{y} + \sum_{j=1}^{n} s_j \cdot \underline{d}_j^- \cdot \overline{x}_j\} \tag{2.36}$$

is satisfied, and the right hand side of (2.36) is finite[2]*. Moreover, for every $P \in \mathbf{P}$ and every j with $\underline{d}_j \ge 0$ the LMI*

$$\sum_{i=1}^{m} y_i A_{ij} - C_j \preceq 0$$

is feasible with $y := \tilde{y}$.

Proof. Let $P = (A, b, C) \in \mathbf{P}$ be chosen fixed, and let $X_j = X_j(P)$ be primal feasible for P and $j = 1, \ldots, n$. Let

$$D_j = C_j - \sum_{i=1}^{n} \tilde{y}_i A_{ij} \quad \text{for } j = 1, \ldots, n,$$

[2]Notice that $\mathbf{b}^T y$ is an interval operation yielding an interval for the expression in the braces in (2.36). Hence, the infimum denotes the lower bound of this interval. This notation applies also for the supremum and subsequently.

then

$$\sum_{j=1}^{n}\langle C_j, X_j\rangle - b^T\tilde{y} = \sum_{j=1}^{n}\langle D_j, X_j\rangle.$$

Since $D_j \in \mathbf{D}_j$, Lemma 2.1 implies

$$\sum_{j=1}^{n}\langle D_j, X_j\rangle \geq \sum_{j=1}^{n} s_j \cdot \underline{d}_j^- \cdot \overline{x}_j,$$

which proves the inequality (2.36), and the assumption (2.35) yields a finite right hand side. The last statement is an immediate consequence of $D_j \in \mathbf{D}_j$ and $\lambda_{\min}(D_j) \geq \underline{d}_j \geq 0$. □

Observe that $\tilde{y}$ is dual feasible provided $\underline{d}_j \geq 0$ for $j = 1, \dots, n$. Hence in this case, (2.36) yields the lower bound $\inf\{\mathbf{b}^T\tilde{y}\}$ for the dual optimal value $d^*(P)$ for every $P \in \mathbf{P}$.

In order to judge the quality of the lower bound (2.36), we assume that

i) exact input data $P = \mathbf{P}$ are given,
ii) $D = \mathbf{D}$ is computed exactly, and
iii) Slater's constraint qualifications (see Theorem 2.1) are fulfilled.

Moreover, let $\tilde{y}$ be the optimal solution of the dual problem (2.11), and let $\underline{d}_j = \lambda_{\min}(D)$ for $j = 1, \dots, n$. Then $\underline{d}_j \geq 0$ for $j = 1, \dots, n$, and

$$p^*(P) = d^*(P) = b^T\tilde{y}.$$

Hence, no overestimation occurs, and it follows that the quality of this lower bound mainly depends on the quality of the $\underline{d}_j$ and on the computed approximation $\tilde{y}$.

An immediate consequence is the following error bound for linear programming problems

$$p^* := \min c^T x \quad \text{s.t. } Ax = b, x \geq 0, \tag{2.37}$$

which is proved in [33], and in [67] for finite bounds $\overline{x}_j$. The input data are $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $P = (A, b, c) \in \mathbb{R}^{m\times n+m+n}$.

Corollary 2.1. *Let* $\mathbf{P} = (\mathbf{A}, \mathbf{b}, \mathbf{c}) \in \mathbb{IR}^{m\times n+m+n}$, $\tilde{y} \in \mathbb{R}^m$, *and let*

$$\mathbf{d} := \mathbf{c} - \mathbf{A}^T\tilde{y}. \tag{2.38}$$

Assume further that upper bounds of the primal feasible solutions

$$x_j \leq \overline{x}_j \quad \textit{for } j = 1, \dots, n$$

are known for all $P \in \mathbf{P}$, which may also be infinite. If

$$\mathbf{d}_j \geq 0 \quad \textit{for} \quad \overline{x}_j = +\infty, \tag{2.39}$$

then for every $P \in \mathbf{P}$ the optimal value $p^*(P)$ satisfies the inequality

$$p^*(P) \geq \inf\{\mathbf{b}^T \tilde{y} + \sum_{j=1}^{n} \mathbf{d}_j^- \cdot \overline{x}_j\}. \tag{2.40}$$

Proof. Apply Theorem 2.2 to the semidefinite program where the symmetric matrices A_{ij}, C_j and X_j are one-dimensional. □

Next, we describe an algorithm for computing a lower bound of the optimal value, which is based on Theorem 2.2. We assume that an approximate dual optimal solution $\tilde{y} \in \mathbb{R}^m$ of the midpoint problem $\operatorname{mid}\mathbf{P}$ is known. If condition (2.35) is fulfilled, the only work is to compute the right hand side of (2.36). Otherwise, the idea is to perturb all constraints which violate condition (2.35); that is, we solve a perturbed midpoint problem $P = (\operatorname{mid}\mathbf{A}, \operatorname{mid}\mathbf{b}, C(\varepsilon))$ with

$$C_j(\varepsilon) = \operatorname{mid}\mathbf{C}_j - \varepsilon_j I, \ \varepsilon_j = \begin{cases} > 0 & \text{if } \underline{d}_j < 0 \text{ and } \overline{x}_j = +\infty \\ 0 & \text{otherwise.} \end{cases} \tag{2.41}$$

Then the dual optimal solution $y(\varepsilon)$ satisfies the constraints

$$\operatorname{mid}\mathbf{C}_j - \sum_{i=1}^{m} y_i(\varepsilon) \operatorname{mid}\mathbf{A}_{ij} \succeq \varepsilon_j I.$$

Hence, the minimal eigenvalues of the new defect

$$\mathbf{D}_j(\varepsilon) := \mathbf{C}_j - \sum_{i=1}^{m} y_i(\varepsilon)\mathbf{A}_{ij}$$

will increase. Choosing ε_j very large may imply dual infeasibility, choosing $\varepsilon_j > 0$ too small may not be sufficient for satisfying (2.35). Our current trade off is to solve repeatedly perturbed programs until either condition (2.35) is satisfied, or the dual is infeasible. The details are given in the following algorithm.

Algorithm 2.1. *Rigorous lower bound*

Given: *real or interval input data $\mathbf{P} = (\mathbf{A}, \mathbf{b}, \mathbf{c})$,*
upper bounds $\overline{x}_j$ for $j = 1, \ldots, n$,
approximate dual optimal solution $\tilde{y}$ for $\operatorname{mid}\mathbf{P}$,

```
    $\underline{p}^* := -\infty$,
    $\varepsilon, k$ are the n-dimensional zero vectors,
    maximal numbers of iterations $l_{\max}$,
    $l := 0$.

While perturbed problem $P(\varepsilon)$ is dual feasible and $l \le l_{\max}$
  1. Compute $\mathbf{D}_j = \mathbf{C}_j - \sum_{i=1}^{m} \tilde{y}_i \mathbf{A}_{ij},\ j = 1, \ldots, n$.
  2. Compute rigorous lower bounds $\underline{d}_j \le \lambda_{\min}(\mathbf{D}_j)$, for $j = 1, \ldots, n$.
  3. If $\underline{d}_j \ge 0$ for every $j$ with $\overline{x}_j = +\infty$ then compute
        $\underline{p}^* = \inf\{\mathbf{b}^T \tilde{y} + \sum_{j=1}^{n} s_j \cdot \underline{d}_j^- \cdot \overline{x}_j\}$,
     stop.
  4. Compute for $j = 1, \ldots, n$
        $k_j := \begin{cases} k_j + 1 & \text{if } \underline{d}_j < 0 \text{ and } \overline{x}_j = +\infty \\ k_j & \text{otherwise,} \end{cases}$
        $\varepsilon_j := \begin{cases} -2^{k_j} \underline{d}_j + \varepsilon_j & \text{if } \underline{d}_j < 0 \text{ and } \overline{x}_j = +\infty \\ \varepsilon_j & \text{otherwise.} \end{cases}$
  5. Solve the perturbed midpoint problem $P(\varepsilon) = (\operatorname{mid}\mathbf{A}, \operatorname{mid}\mathbf{b}, C(\varepsilon))$, where
     $C_j(\varepsilon) = \operatorname{mid}\mathbf{C}_j - \varepsilon_j I$ for $j = 1, \ldots, n$, and set $\tilde{y} := \tilde{y}(\varepsilon)$ (approximate dual
     optimal solution).
  6. $l := l + 1$.
end.
```

This algorithm requires interval arithmetic (or at least the monotonic rounding operations) for computing the defect matrices $\mathbf{D}$ and the lower bounds $\underline{d}_j$, and a semidefinite solver for computing approximate solutions of the perturbed problems.

The algorithm terminates during the first iteration in step 3 if all simple bounds $\overline{x}_j$ are finite or all $\underline{d}_j$ are nonnegative. In this case the computational costs are $O(m \cdot \sum_{j=1}^{n} s_j^2)$ for computing the $\mathbf{D}_j$'s, the lower bounds $\underline{d}_j$ require $O(\sum_{j=1}^{n} s_j^3)$ operations, and the bound $\underline{p}^*$ needs $O(m+n)$ operations. Hence the costs are negligible compared to the costs for approximately solving a semidefinite program.

In other cases, however, the computational costs may increase because perturbed semidefinite programs must be solved until either the semidefinite programming solver indicates dual infeasibility of the perturbed problem or the maximal number of iterations $l_{\max}$ is reached.

Several modifications of this algorithm are possible and may yield improvements. Here we have considered a simple choice of perturbations: In each step we add to ε_j the negative defects $-\underline{d}_j$ multiplied by a factor 2^{k_j}, where k_j counts the number of iterations that violated the inequality $\underline{d}_j \geq 0$.

In applications we recommend to use infinite bounds $\overline{x}_j$ instead of unreasonable large bounds, because otherwise the sum in (2.36) may yield an unnecessary overestimation.

If the upper bounds $\overline{x}_j = +\infty$ for $j = 1, \ldots, n$, and Algorithm 2.1 delivers a finite lower bound $\underline{p}^*$, then the lower eigenvalue bounds $\underline{d}_j$ must be nonnegative. This proves dual feasibility, and if $\underline{d}_j$ is positive for $j = 1, \ldots, n$ strict dual feasibility is verified.

2.4.2 Rigorous upper bound

In this section we investigate the computation of a rigorous upper bound for the optimal value of a semidefinite program together with a certificate of existence of primal feasible solutions. The basic idea is to compute interval matrices $\mathbf{X}_j$ for $j = 1, \ldots, n$ that contain for every semidefinite program $P \in \mathbf{P}$ a primal feasible solution. The desirable characteristics of the matrices $\mathbf{X}_j$ are given in the next theorem.

Theorem 2.3. *Let* $\mathbf{P}$ *define a family of semidefinite programs (2.10), and suppose that there exist interval matrices* $\mathbf{X}_j$ *for* $j = 1, \ldots, n$*, such that*

$$\begin{aligned} &\forall\, b \in \mathbf{b},\ \forall\, A_{ij} \in \mathbf{A}_{ij},\ i = 1, \ldots, m,\ j = 1, \ldots, n \\ &\qquad \exists \text{ symmetric } X_j \in \mathbf{X}_j :\ \sum_{j=1}^{n} \langle A_{ij}, X_j \rangle = b_i, \end{aligned} \tag{2.42}$$

and for $j = 1, \ldots, n$

$$X_j \succeq 0 \text{ for all symmetric } X_j \in \mathbf{X}_j. \tag{2.43}$$

Then, the optimal value is bounded from above by

$$p^*(P) \leq \sup\{\sum_{j=1}^{n} \langle \mathbf{C}_j, \mathbf{X}_j \rangle\} \tag{2.44}$$

Moreover, if all symmetric $X_j \in \mathbf{X}_j$ *are positive definite and* $p^*(P)$ *is bounded from below, then* $p^*(P) = d^*(P)$ *for every* $P \in \mathbf{P}$ *(no duality gap), and the dual supremum is attained.*

Proof. Let $P \in \mathbf{P}$ be a fixed chosen problem. Then the conditions (2.42) and (2.43) imply that there exists a primal feasible solution $X_j = X_j(P)$ for $j = 1, \ldots, n$. Hence, $\sum_{j=1}^{n} \langle C_j, X_j \rangle \geq p^*(P)$, and the inclusion property (2.17) yields (2.44). If all $X_j \in \mathbf{X}_j$ are positive definite, then (2.42) and (2.43) imply the existence of strictly primal feasible solutions, and hence Theorem 2.1 shows that the dual optimal solution is attained and strong duality holds valid. □

By weak duality the upper bound in (2.44) is also an upper bound of the dual optimal value. Moreover, if all $X_j \in \mathbf{X}_j$ are positive definite, then the Strong Duality Theorem 2.1 implies that the right hand side of (2.36) is also a lower bound of the dual optimal value for all $P \in \mathbf{P}$. Hence, in this case it is not necessary to assume $\underline{d}_j \geq 0$ for $j = 1, \ldots, n$.

In the following, we describe an algorithm for computing this rigorous upper bound. This algorithm must find appropriate interval matrices $\mathbf{X}_j$, and verify the conditions (2.42) and (2.43). We discuss these items below.

To make sure that the upper bound (2.44) is close to the optimal value, the interval matrices $\mathbf{X}_j$ must be close to optimality. In general the complementary slackness relations yield rank-deficient matrices that are not positive definite. Therefore, we solve the slightly perturbed midpoint problem

$$\min \sum_{j=1}^{n} \langle C_j, X_j \rangle \quad \text{s.t.} \quad \begin{array}{ll} \sum_{j=1}^{n} \langle A_{ij}, X_j \rangle = b_i & \text{for } i = 1, \ldots, m, \\ X_j \succeq \varepsilon_j \cdot I, & \text{for } j = 1, \ldots, n, \end{array} \tag{2.45}$$

where ε_j is positive and the input data $(A, b, c) = \operatorname{mid} \mathbf{P}$. Then for small ε_j the optimal solution $(X_j(\varepsilon_j))$ is positive definite and close to the optimal solution of the midpoint problem.

In the following we show how we can construct an appropriate interval matrix $(\mathbf{X}_j)$ by using an approximate optimal solution $(X_j(\varepsilon_j))$ of (2.45).

The semidefinite program (2.10) can be written in the equivalent vector representation form

$$\min c^T x \quad \text{s.t.} \quad A^{\text{mat}} x = b, \; X_j \succeq 0, \; \text{for } j = 1, \ldots, n, \tag{2.46}$$

where

$$c := \text{svec}((C_j), 2), \tag{2.47}$$

$$x := \text{svec}((X_j), 1), \tag{2.48}$$

and the i-th row of the $m \times \sum_{j=1}^{n} \frac{s_j(s_j+1)}{2}$ matrix A^{mat} is defined by

$$A^{\text{mat}}(i,:) = \text{svec}((A_{ij})_{j=1}^{n}, 2). \tag{2.49}$$

If interval input data $\mathbf{P}$ are given, then we denote by $\mathbf{A}^{\text{mat}}$, $\mathbf{b}$, and $\mathbf{c}$ the corresponding interval quantities. Thus condition (2.42) is equivalent to

$$\forall\, b \in \mathbf{b},\ \forall\, A^{\text{mat}} \in \mathbf{A}^{\text{mat}}\ \exists\, x \in \mathbf{x} \ \text{ such that } \ A^{\text{mat}} x = b, \tag{2.50}$$

which is an underdetermined system of linear equations with interval input data. Given an approximate optimal solution $(X_j(\varepsilon_j))_{j=1}^{n}$, it is straight forward to solve such a system.

We start by assuming that the $m \times m$ submatrix mid $\mathbf{A}_I^{\text{mat}}$ with the m columns mid $\mathbf{A}^{\text{mat}}(:, \beta_i)$ is nonsingular, where the index set $I := \{\beta_1, \ldots, \beta_m\}$. Let N denote all indices of columns of mid $\mathbf{A}^{\text{mat}}$ which are not in I, let $\mathbf{A}_N^{\text{mat}}$ be the matrix with columns corresponding to the indices of N, and let $\tilde{x} = \text{svec}((X_j(\varepsilon_j)), 1)$. In our algorithm we choose the index set I by performing an LU-decomposition on (mid $\mathbf{A}^{\text{mat}})^T$ and assembling the computed pivot columns to A_I^{mat}. Now we fix the variables $\tilde{x}_N$, and compute with some verification method for interval linear systems an enclosure $\mathbf{x}_I$ of the solution set

$$\Sigma_I := \{x_I \in \mathbb{R}^m : \ A_I^{\text{mat}} x_I = b - \sum_{\gamma \in N} A_N^{\text{mat}} \tilde{x}_N,\ A \in \mathbf{A}^{\text{mat}},\ b \in \mathbf{b}\}. \tag{2.51}$$

Then $\mathbf{x} := (\mathbf{x}_I; \tilde{x}_N)$ fulfills (2.50), and therefore $(\mathbf{X}_j) := \text{smat}(\mathbf{x}, 1)$ satisfies condition (2.42). Condition (2.43) must be verified by some method for computing a rigorous lower bound for the smallest eigenvalue of a symmetric interval matrix.

The following algorithm contains the details for computing a rigorous upper bound for the optimal value and for proving existence of primal feasible solutions. The algorithm needs verified solvers for interval linear systems and eigenvalue problems, and a semidefinite solver for computing approximations of the perturbed problems.

Algorithm 2.2. *Rigorous upper bound, certificate of feasibility*

Given: *real or interval input data* $\mathbf{P} = (\mathbf{A}, \mathbf{b}, \mathbf{c})$*,*
approximate primal optimal solution $(\tilde{X}_j)_{j=1}^{n}$ *of the midpoint problem,*
$\overline{p}^* := \infty$*,*
ε, k *are the n-dimensional zero vectors,*
maximal number of iterations $l_{\max}$*,*
$l := 0$.

```
Choose an index set I such that the submatrix $\operatorname{mid}\mathbf{A}_I^{\mathrm{mat}}$ is (at least numerically) nonsingular (for example, by performing an LU-decomposition on $(\operatorname{mid}\mathbf{A}^{\mathrm{mat}})^T$).

If there is no nonsingular submatrix then stop.

While perturbed problem $P(\varepsilon)$ is primal feasible and $l \leq l_{\max}$
    1. Compute an enclosure $\mathbf{x}_I$ of the solution set $\Sigma_I$, and set $\mathbf{x} := (\mathbf{x}_I; \tilde{x}_N)$.
    2. Set $(\mathbf{X}_j) = \operatorname{smat}(\mathbf{x}, 1)$, and compute rigorous bounds
           $\underline{\lambda}_j \leq \lambda_{\min}(\mathbf{X}_j)$ for $j = 1, \ldots, n$.
    3. If $\underline{\lambda}_j \geq 0$ for $j = 1, \ldots, n$ then compute
           $\overline{p}^* = \sup\{\mathbf{c}^T\mathbf{x}\}$,
       stop.
    4. Compute for $j = 1, \ldots, n$
           $k_j := \begin{cases} k_j + 1 & \text{if } \underline{\lambda}_j < 0 \\ k_j & \text{otherwise,} \end{cases}$
           $\varepsilon_j := \begin{cases} -2^{k_j}\underline{\lambda}_j + \varepsilon_j & \text{if } \underline{\lambda}_j < 0 \\ \varepsilon_j & \text{otherwise.} \end{cases}$
    5. Solve the perturbed problem (2.45), set $\tilde{X}_j := \tilde{X}_j(\varepsilon)$ for $j = 1, \ldots, n$ (approximate primal optimal solution), and set $\tilde{x} := \operatorname{svec}((\tilde{X}_j), 1)$.
    6. $l := l + 1$.
end.
```

If Algorithm 2.2 delivers a finite upper bound $\overline{p}^*$, then the lower eigenvalue bounds $\underline{\lambda}_j$ must be nonnegative. If $\underline{\lambda}_j > 0$ for $j = 1, ..., n$, then strict primal feasibility is verified.

Krawzcyk [44] was the first who solved non degenerate interval linear programming problems by using the technique of fixing appropriate variables (the nonbasic variables) and solving a remaining quadratic interval linear system for the basic variables. In [32] this technique was used to compute enclosures of all optimal vertices in the case of degeneration. Hansen [25] used this technique in order to prove existence of a feasible point for nonlinear equations within a bounded box. It was further modified and investigated numerically by Kearfott [42], and is also described in his book [41].

2.4.3 Verification of ill-posed problems

Ill-conditioned and ill-posed problems constitute a group of the most challenging tasks for any semidefinite solver. Thus it is all the more important to get verified error bounds for them. However, the bounding algorithms 2.1 and 2.2 given in the previous sections cannot be applied to the ill-posed cases. There perturbations of the initial problem, that underlie the iterations, might lead to infeasibility. In this section we will briefly introduce results found in Jansson [35] and [36]. The presented methods will also serve as a foundation for future modifications in the subsequent chapters.

It is useful to begin with the introduction of new boundedness qualifications, the so called *primal boundedness qualification* (PBQ) and *dual boundedness qualification* (DBQ). They are more appropriate for the ill-posed case than the Slater's constraint qualifications 2.1.

PBQ:

(i) Either the primal problem is infeasible,

(ii) or $\tilde{p}^*$ is finite, and there is a nonnegative number $\overline{x}$ such that for every $\varepsilon > 0$ there exists a primal feasible solution $X(\varepsilon) \le \overline{x} \cdot I$ and $\langle C, X(\varepsilon)\rangle - \tilde{p}^* \le \varepsilon$.

DBQ:

(i) Either the dual problem is infeasible,

(ii) or $\tilde{d}^*$ is finite, and there is a simple bound $\overline{y}$ such that for every $\varepsilon > 0$ there exists a dual feasible solution $y(\varepsilon)$ satisfying $|y(\varepsilon)| \le \overline{y}$ and $\tilde{d}^* - b^T y(\varepsilon) \le \varepsilon$.

Both conditions are rather weak compared to Slater's qualifications. Even the existence of optimal solutions is not demanded, and only simple bounds for ε-optimal solutions are required. Now we are ready to present the bounding theorems applicable also to ill-posed semidefinite programs.

Theorem 2.4. *Assume that PBQ holds, and let $\tilde{y} \in \mathbb{R}^m$. Let also*

$$D = C - \sum_{i=1}^{m} \tilde{y}_i A_i. \tag{2.52}$$

Suppose further that $\underline{d} = \lambda_{min}(D)$, and that D has at most l negative eigenvalues. Then

$$p^* \ge b^T \tilde{y} + l \cdot \underline{d}^- \cdot \overline{x} =: \underline{p}. \tag{2.53}$$

Moreover, if $\underline{d}^- = 0$ *then* $\tilde{y}$ *is dual feasible and* $d^* \geq \underline{p} = b^T\tilde{y}$*, and if* $\tilde{y}$ *is optimal, then* $d^* = \underline{p}$.

Here we used again the notation $\underline{d}^- := \min\{0, \underline{d}\}$ and $\overline{x}$ is a PBQ bound. The following theorem provides a finite upper bound $\overline{d}$ of the dual optimal value.

Theorem 2.5. *Assume that DBQ is fulfilled. Let* $\tilde{X} \in \mathbb{S}^s$, $\tilde{X} \succeq 0$. *Suppose further that*

$$|\langle A_i.\tilde{X}\rangle - b_i| \leq \overline{r}_i \qquad \textit{for} \quad i = 1, \ldots, m. \tag{2.54}$$

Then

$$d^* \leq \langle C, \tilde{X}\rangle + \overline{y}^T\overline{r} =: \overline{d}. \tag{2.55}$$

If $\overline{r} = 0$ *then* $\tilde{X}$ *is primal feasible and* $p^* \leq \overline{d} = \langle C, \tilde{X}\rangle$*, and if moreover* $\tilde{X}$ *is optimal, then* $p^* = \overline{d}$.

Should the computed approximation $\tilde{X}$ not be positive semidefinite, we could, for example, determine a shifted matrix with all eigenvalues above zero. Then we would use it instead of $\tilde{X}$ in (2.54) and (2.55). Another possibility would be to measure the deviation from positive semidefiniteness and include it directly in (2.55). For further details and proofs the reader is kindly referred to Jansson [35] and [36].

2.5 Certificates of infeasibility

In branch and bound algorithms a subproblem is discarded if the local nonlinear solver detects infeasibility. It is not a rare phenomenon that sometimes local solvers do not find feasible solutions of a subproblem, although they exist (see for example the comments for use of SDPT3 [73]). A consequence is that the global minimum solutions may be cut off.

To avoid this disadvantage we can apply the algorithms for computing rigorous bounds described in the previous sections to a phase I problem in order to verify infeasibility for primal and dual semidefinite problems. In the literature there are several variations of the phase I method. It is common, however, that the auxiliary objective function describes the infeasibility in the sense that the problem has no feasible solutions, provided the optimal value has specific sign. The latter property can be verified by the algorithms of the previous sections. A reference implementation of the phase I methods for primal and dual semidefinite programs can be found in Appendix A.

Another approach is based on certificates of infeasibility. For linear programs with bounded variables rigorous certificates of infeasibility are described in Neumaier and Shcherbina [57]. For infeasible semidefinite problems often (but not every time) certificates of infeasibility exposed by improving rays can also be obtained (see the discussion in Todd [72])

The primal problem (2.10) has a primal improving ray if there exists a block-diagonal matrix (X_j) such that for all $i = 1, \dots, m$ and $j = 1, \dots, n$

$$X_j \succeq 0, \quad \sum_{j=1}^{n} \langle A_{ij}, X_j \rangle = 0, \quad \text{and} \quad \sum_{j=1}^{n} \langle C_j, X_j \rangle < 0. \tag{2.56}$$

It is well-known and straightforward to show that the existence of a primal improving ray implies dual infeasibility. If interval input data $\mathbf{P}$ are given, and for the midpoint problem of $\mathbf{P}$ an approximate primal improving ray is known, then we can try to verify dual infeasibility for all problems with $P \in \mathbf{P}$ by using a similar approach as in the previous section. We assume that the semidefinite solver has computed an approximate primal improving ray $(\tilde{X}_j)$ for the midpoint problem. Let $\beta \approx \sum_{j=1}^{n} \langle \text{mid}\mathbf{C}_j, \tilde{X}_j \rangle$ be approximately calculated and assume that $\beta < 0$. Notice that for positive β the conditions (2.56) are in general not satisfied. Now, analogously to the previous section, we can compute enclosures $(\mathbf{X}_j)$ such that for every $A_{ij} \in \mathbf{A}_{ij}$ and for every $C_j \in \mathbf{C}_j$ there exist solutions $X_j \in \mathbf{X}_j$ of the underdetermined linear system

$$\sum_{j=1}^{n} \langle A_{ij}, X_j \rangle = 0 \quad \text{for} \quad i = 1, \dots, m, \qquad \sum_{j=1}^{n} \langle C_j, X_j \rangle = \beta. \tag{2.57}$$

If an enclosure $(\mathbf{X}_j)$ is computed, and if all minimal eigenvalues $\lambda_{min}(\mathbf{X}_j)$ are non-negative, then, because $\beta < 0$, it follows that for every $P \in \mathbf{P}$ there exists $X_j \in \mathbf{X}_j$ for $j = 1, \dots, n$ such that (2.56) is satisfied. Therefore, all problems with $P \in \mathbf{P}$ are dual infeasible, and the block-matrices $(\mathbf{X}_j)$ contain the primal improving rays.

The dual problem (2.11) has a dual improving ray if there is a vector $y \in \mathbb{R}^m$ such that

$$\sum_{i=1}^{m} y_i A_{ij} \preceq 0 \quad \text{for } j = 1, \dots, n, \quad \text{and} \quad b^T y > 0. \tag{2.58}$$

The existence of a dual improving ray implies primal infeasibility. If interval input data $\mathbf{P}$ are given and for some problem $P \in \mathbf{P}$ an approximate dual improving ray $\tilde{y}$ is known, then we can try to verify primal infeasibility for all problems with $P \in \mathbf{P}$ as follows: First we compute by using interval arithmetic upper bounds of the maximal eigenvalues of $\sum_{i=1}^{m} \tilde{y}_i \mathbf{A}_{ij}$ for $j = 1, \ldots, n$. If these upper bounds are nonpositive and if $\mathbf{b}^T \tilde{y} > 0$, then $\tilde{y}$ is a dual improving ray for all $P \in \mathbf{P}$. Hence, $\tilde{y}$ is a rigorous certificate of primal infeasibility for all $P \in \mathbf{P}$.

Chapter 3

Rigorous error bounds for RDM variational problems

In this chapter, the previously developed approaches for bounding optimal values of a semidefinite program will be applied to the electronic structure problem in the reduced density matrix (RDM) formulation. Verification algorithms adapted to utilize the specific problem structure will be proposed for computing a rigorous lower bound of the ground state energy. Different benchmark problems involving small molecules and ions will be rigorously solved.

3.1 Introduction

The problem of determining a ground state energy of a system of N electrons, known as the electronic structure problem, is one of the central challenges in quantum chemistry. A standard approach to cope with this task would be applying a variational principle. One tries to minimize the functional $E[\Psi] = \langle\Psi|\hat{H}|\Psi\rangle/\langle\Psi|\Psi\rangle$ to obtain an upper bound to the true ground state energy E_0. Here $\hat{H}$ is the Hamiltonian of the studied system, and Ψ is a normalizable trial wave function. Then the principle says that $E[\Psi] \geq E_0$ and the equality is attained if and only if Ψ is equal to the ground state wave function.

The RDM method allows to solve this problem without using wave functions which contain much more information than it's actually needed for the ground state calculation (see for example Coleman [12], Garrod and Percus [21]). One-electron and two-electron density matrices are used as trial functions instead.

The difficulty with RDMs is to assure that they describe some N-fermion system

(N-representability problem). An illustrative example showing the necessity of extra requirements together with the introduction of standard P, Q and G conditions can be found in [21]. Without imposing them, the minimization result appears deep below the exact value. The accuracy of the RDM method depends on how well we can restrict the trial density matrices to be N-representable. To approach the problem, either new N-representability conditions can be introduced (six-index matrices $T1$ and $T2$ in [79]), or higher order RDMs can be added to the model (see Mazziotti [47]). Unfortunately, the full set of sufficient conditions is not known.

Due to their nature, RDM problems can easily be represented as semidefinite programs. The Zhengji Zhao's dissertation [78] contains a detailed analysis of different possibilities to formulate such SDPs. A collection of benchmark problems in "dual" SDP formulation, used also by Zhao *et al.* in [79] can be found at [18]. The set contains data in SDPA [17] format for different molecules and ions with the biggest electron number $N = 16$ and basis sizes of 12, 14, 16 and 20 functions.

It is important to understand, that RDM problems are ill-posed due to the modeling. The antisymmetricity of two-electron density matrices assumes solutions only at the boundary of the semidefinite cone where Slater's constraint qualifications do not hold. Luckily, the corresponding SDP problems do not possess this property. Due to the redundancy elimination in "compacted" matrices, zero eigenvalues are no more an integral property of the solution. The conditions of the strong duality theorem 2.1 are thus fulfilled and the optimum is attained.

By solving an RDM problem, we get, in contrast to the wave function variation, a lower bound of the ground state energy[1]. The immediate cause of this is the unknown full N-representability condition. The variational minimization problem becomes relaxed and the optimum lies below the full configuration interaction (full CI) value. Another relaxation is hidden in the "dual" SDP formulation. The price for the relatively small problem size there is the necessity to replace all equality constraints by relaxed inequalities. The corresponding feasibility gap amounts to $2 \cdot 10^{-7}$ in the sample problems. Nevertheless, an approximate RDM solution cannot be taken as a ground state energy lower bound. As we show in due course, in many cases the obtained approximation is greater than the known full CI value. This happens if a semidefinite solver fails to find a good solution and the duality gap remains relatively big.

In this chapter we present a time efficient algorithm for calculating a rigorous lower

[1]It is necessary to note, that all the obtained values are valid within a fixed basis. Should we increase or change the one-particle basis set, the correspondences will remain, but the numerical values might change.

bound of the ground state energy. The obtained bound is accurate independent of the initial approximation quality (though a better approximation delivers a tighter bound). All rounding errors due to floating point arithmetic are rigorously estimated. Interval input data are also possible. Of course, an upper bound of the SDP optimal value can also be computed. However because of the relaxations described above, it cannot be taken as an upper bound of the ground state energy. To obtain one, a wave function (or a full density matrix) variational method like Hartree-Fock or similar should be used.

3.2 Reduced density matrices

For a system of N particles in a state with wave function $\Psi = \Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$, the corresponding p-body reduced density matrix (RDM) is defined as

$$\Gamma^p(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p; \mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_p) = C_N^p \int \Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p, \mathbf{x}_{p+1}, \ldots, \mathbf{x}_N) \times \Psi^*(\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_p, \mathbf{x}_{p+1}, \ldots, \mathbf{x}_N) d\mathbf{x}_{p+1} \cdots \mathbf{x}_N, \quad (3.1)$$

where $C_N^p = N!/(N-p)!p!$ is a binomial coefficient and $\mathbf{x}_i$ denotes full (spatial and spin) coordinate of the ith particle. For our work, of particular interest are the one-particle RDM (1-RDM)

$$\gamma(\mathbf{x}_1, \mathbf{x}'_1) = N \int \Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)\Psi^*(\mathbf{x}'_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) d\mathbf{x}_2 \cdots \mathbf{x}_N, \quad (3.2)$$

and the two-particle RDM (2-RDM)

$$\check{\Gamma}(\mathbf{x}_1, \mathbf{x}_2; \mathbf{x}'_1, \mathbf{x}'_2) = \frac{N(N-1)}{2} \int \Psi(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N)\Psi^*(\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}_3, \ldots, \mathbf{x}_N) d\mathbf{x}_3 \cdots \mathbf{x}_N. \quad (3.3)$$

We will, however, use the 2-RDM normalized by trace$(\Gamma) = N(N-1)$ as was done, for example, in [21]

$$\Gamma(\mathbf{x}_1, \mathbf{x}_2; \mathbf{x}'_1, \mathbf{x}'_2) = N(N-1) \int \Psi(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N)\Psi^*(\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}_3, \ldots, \mathbf{x}_N) d\mathbf{x}_3 \cdots \mathbf{x}_N. \quad (3.4)$$

The factor $\frac{1}{2}$, eliminating symmetric interactions, can be then embedded in the Hamiltonian.

The exact wave function of an N-electron system can be written as a linear combination of all possible N-electron Slater determinants formed from a complete set of spin orbitals [71]. These single electron wave functions, in there turn, can be represented in a discrete orthonormal finite basis $\{\psi_i\}$ of size r. The following expansion of our N-electron wave function Ψ is then valid:

$$\Psi(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \sum_{i_1,\ldots,i_N} c_{i_1,\ldots,i_N} \psi_{i_1}(\mathbf{x}_1) \cdots \psi_{i_N}(\mathbf{x}_N), \tag{3.5}$$

where i_j are independent indices taking values from 1 to r. Therefore, we can also switch to discrete representations of 1-RDM and 2-RDM in the orthonormal basis $\{\psi_i\}$, denoted by $\gamma(i, i')$ and $\Gamma(i, j; i', j')$ respectively. In case of $\gamma(i, i')$, for example, we obtain

$$\gamma(i, i') = \int \psi_i^*(\mathbf{x}_1)\gamma(\mathbf{x}_1, \mathbf{x}_1')\psi_{i'}(\mathbf{x}_1')d\mathbf{x}_1 d\mathbf{x}_1', \tag{3.6}$$

so that

$$\gamma(\mathbf{x}_1, \mathbf{x}_1') = \sum_{i,i'} \psi_i(\mathbf{x}_1)\gamma(i, i')\psi_{i'}^*(\mathbf{x}_1'). \tag{3.7}$$

In what follows, we keep to the assumption, that all entries of 1-RDM and 2-RDM are real. As described in [79], this is possible for the problem considered. Hence, both generally Hermitian γ and Γ are symmetric matrices. Furthermore, antisymmetry of the original N-electron wave function Ψ requires that Γ change its sign on interchange of unprimed or primed indices, thus

$$\Gamma(i, j; i', j') = -\Gamma(j, i; i', j') = -\Gamma(i, j; j', i'). \tag{3.8}$$

As mentioned before, the purpose of introducing reduced density matrices is to use them as variables in Rayleigh-Ritz minimization to find the ground state energy of an N-electron system. Originally, for a system with Hamiltonian $\hat{H}$ in the state Ψ we have

$$E[\Psi] = \frac{\langle\Psi|\hat{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle} \geq E_0, \tag{3.9}$$

where E_0 is the true ground-state energy. Full minimization of the functional $E[\Psi]$ with respect to all allowed N-electron wave functions will give the true ground state Ψ_0 and energy $E[\Psi_0] = E_0$:

$$E_0 = \min_{\Psi} E[\Psi]. \tag{3.10}$$

Reformulating the minimization problem (3.10) in terms of the discrete full density matrix Γ^N yields

$$E_0 = \min_{\Gamma^N} E[\Gamma^N] = \min_{\Gamma^N} \langle H, \Gamma^N \rangle, \tag{3.11}$$

where H is matrix representation of the Hamiltonian $\hat{H}$, and

$$\langle H, \Gamma^N \rangle = \text{trace}(H^T \Gamma^N) \tag{3.12}$$

denotes the usual inner product on the linear space of symmetric matrices.

Considering the fact, that the Hamiltonian of a N-electron system involves one-body and two-body interaction terms only, we can write

$$\langle H, \Gamma^N \rangle = \langle H_1, \gamma \rangle + \langle H_2, \Gamma \rangle, \tag{3.13}$$

where H_1 and H_2 are the one-body and two-body parts of the Hamiltonian. There is, however, a big obstacle to implementing this idea. Trial matrices γ and Γ must correspond to some antisymmetric Ψ, that is, for any guessed (γ, Γ) there must be a Ψ from which they come via (3.2) and (3.4). This is the reduced density matrices N-representability problem. Running minimization without imposing these extra conditions can give results far below the ground-state energy that are also in contradiction with the physics of the problem [21]. Unfortunately, complete N-representability condition is known only for γ, and for Γ a full set of constructive necessary and sufficient conditions is not available. Thus RDM method can give us only a lower bound for the ground-state energy, and the quality of this bound depends on how good the family of used conditions represents the initial electronic structure problem.

3.3 N-representability

The first restriction on γ and Γ resulting directly from their definitions (3.2) and (3.4) is the dependency

$$\sum_j \Gamma(i, j; i', j) = (N-1)\gamma(i, i'). \tag{3.14}$$

Hence γ is a scaled partial trace of Γ and can thus be eliminated entirely from the problem. We will however follow the approach used for example by Fukuda in [19] and retain it. The trace condition (3.14) will then be used as a set of linear constraints on the pair (γ, Γ).

The corresponding trace conditions on γ and Γ themselves are

$$\sum_i \gamma(i,i) = N \tag{3.15}$$

and

$$\sum_{i,j} \Gamma(i,j;i,j) = N(N-1). \tag{3.16}$$

In case of the 1-RDM γ, the remaining necessary and sufficient N-representability conditions were defined in [12]:

$$\gamma \succeq 0, \qquad I - \gamma \succeq 0 \tag{3.17}$$

with I being the identity matrix.

As already mentioned before, the complete set of N-representability conditions for Γ is not known. In this work we will utilize the P, G, Q, $T1$ and $T2$ conditions as found in [19]. Namely the positive semidefiniteness relations

$$P \succeq 0, \quad G \succeq 0, \quad Q \succeq 0, \quad T1 \succeq 0, \quad T2 \succeq 0, \tag{3.18}$$

where the matrices P, G, Q, $T1$ and $T2$ are defined by linear combinations of the entries of γ and Γ.

$$P \equiv \Gamma. \tag{3.19}$$

$$G(i,j;i',j') \equiv \Gamma(i,j';j,i') + \delta(i,i')\gamma(j',j). \tag{3.20}$$

$$\begin{aligned} Q(i,j;i',j') \equiv & \Gamma(i,j;i',j') - \delta(i,i')\gamma(j,j') - \delta(j,j')\gamma(i,i') \\ & + \delta(i,j')\gamma(j,i') + \delta(j,i')\gamma(i,j') \\ & + \delta(i,i')\delta(j,j') - \delta(i,j')\delta(j,i'). \end{aligned} \tag{3.21}$$

$$\begin{aligned}
T1(i,j,k;i',j',k') \equiv{}& \delta(i,i')\Gamma(k',j';k,j) - \delta(i,j')\Gamma(k',i';k,j) \\
&+ \delta(i,k')\Gamma(j',i';k,j) - \delta(j,i')\Gamma(k',j';k,i) \\
&+ \delta(j,j')\Gamma(k',i';k,i) - \delta(j,k')\Gamma(j',i';k,i) \\
&+ \delta(k,i')\Gamma(k',j';j,i) - \delta(k,j')\Gamma(k',i';j,i) \\
&+ \delta(k,k')\Gamma(j',i';j,i) \\
&+ (\delta(j,k')\delta(k,j') - \delta(j,j')\delta(k,k'))\gamma(i',i) \\
&+ (\delta(i,j')\delta(k,k') - \delta(i,k')\delta(k,j'))\gamma(i',j) \\
&+ (\delta(i,k')\delta(j,j') - \delta(i,j')\delta(j,k'))\gamma(i',k) \\
&+ (\delta(j,i')\delta(k,k') - \delta(j,k')\delta(k,i'))\gamma(j',i) \\
&+ (\delta(i,k')\delta(k,i') - \delta(i,i')\delta(k,k'))\gamma(j',j) \\
&+ (\delta(i,i')\delta(j,k') - \delta(i,k')\delta(j,i'))\gamma(j',k) \\
&+ (\delta(j,j')\delta(k,i') - \delta(j,i')\delta(k,j'))\gamma(k',i) \\
&+ (\delta(i,i')\delta(k,j') - \delta(i,j')\delta(k,i'))\gamma(k',j) \\
&+ (\delta(i,j')\delta(j,i') - \delta(i,i')\delta(j,j'))\gamma(k',k) \\
&+ \delta(i,i')\delta(j,j')\delta(k,k') - \delta(i,j')\delta(j,i')\delta(k,k') \\
&- \delta(i,i')\delta(j,k')\delta(k,j') + \delta(i,j')\delta(j,k')\delta(k,i') \\
&+ \delta(i,k')\delta(j,i')\delta(k,j') - \delta(i,k')\delta(j,j')\delta(k,i').
\end{aligned} \tag{3.22}$$

$$\begin{aligned}
T2(i,j,k;i',j',k') \equiv{}& \delta(i,i')\Gamma(j',k';j,k) - \delta(j,j')\Gamma(k',i;k,i') \\
&- \delta(k,k')\Gamma(j',i;j,i') - \delta(j,k')\Gamma(j',i;k,i') \\
&+ \delta(k,j')\Gamma(k',i;j,i') \\
&+ \delta(k,k')\delta(j,j')\gamma(i,i') - \delta(j,k')\delta(k,j')\gamma(i,i').
\end{aligned} \tag{3.23}$$

In the above definitions all indices range over $1,\dots,r$ and δ is the Kronecker delta.

Another class of conditions comes from the spin symmetry of the N-electron system [19]. Exploiting these properties will let us introduce block diagonal structures into the above presented matrices. To do that, we first have to arrange our basis functions $\psi_i, i = 1,\dots,r$. Each of them will be a product of a spatial orbital and one of two spin states. Each index i stands therefore for a pair of indices n_i and σ_i. The spatial orbital index n_i can then take values $1,2,\dots,r/2$ while the spin states σ_i can be either $+1/2$ (α spin) or $-1/2$ (β spin).

Hence, we obtain

$$\gamma(n_i\sigma_i, n_{i'}\sigma_{i'}) = 0 \qquad \text{for} \quad \sigma_i \neq \sigma_{i'}, \tag{3.24}$$

$$\Gamma(n_i\sigma_i, n_j\sigma_j; n_{i'}\sigma_{i'}, n_{j'}\sigma_{j'}) = 0 \qquad \text{for} \quad \sigma_i + \sigma_j \neq \sigma_{i'} + \sigma_{j'}, \tag{3.25}$$

$$G(n_i\sigma_i, n_j\sigma_j; n_{i'}\sigma_{i'}, n_{j'}\sigma_{j'}) = 0 \qquad \text{for} \quad \sigma_i + \sigma_{j'} \neq \sigma_j + \sigma_{i'}, \tag{3.26}$$

$$Q(n_i\sigma_i, n_j\sigma_j; n_{i'}\sigma_{i'}, n_{j'}\sigma_{j'}) = 0 \qquad \text{for} \quad \sigma_i + \sigma_j \neq \sigma_{i'} + \sigma_{j'}, \tag{3.27}$$

$$T1(n_i\sigma_i, n_j\sigma_j, n_k\sigma_k; n_{i'}\sigma_{i'}, n_{j'}\sigma_{j'}, n_{k'}\sigma_{k'}) = 0 \\ \text{for} \quad \sigma_i + \sigma_j + \sigma_k \neq \sigma_{i'} + \sigma_{j'} + \sigma_{k'}, \tag{3.28}$$

$$T2(n_i\sigma_i, n_j\sigma_j, n_k\sigma_k; n_{i'}\sigma_{i'}, n_{j'}\sigma_{j'}, n_{k'}\sigma_{k'}) = 0 \\ \text{for} \quad \sigma_i + \sigma_{j'} + \sigma_{k'} \neq \sigma_j + \sigma_k + \sigma_{i'}. \tag{3.29}$$

Trace constraints corresponding to the number of electrons with α spin (N_α):

$$\sum_{n_i=1}^{r/2} \gamma(n_i\alpha, n_i\alpha) = N_\alpha, \tag{3.30}$$

$$\sum_{n_i,n_j=1}^{r/2} \Gamma(n_i\alpha, n_j\alpha; n_i\alpha, n_j\alpha) = N_\alpha(N_\alpha - 1). \tag{3.31}$$

Finally, there is a linear constraint for the given total spin S,

$$\sum_{n_i,n_j=1}^{r/2} \left(\Gamma(n_i\alpha, n_j\alpha; n_i\alpha, n_j\alpha) + \Gamma(n_i\beta, n_j\beta; n_i\beta, n_j\beta)\right) \\ - 2\sum_{n_i,n_j=1}^{r/2} \Gamma(n_i\alpha, n_j\beta; n_i\alpha, n_j\beta) - 4\sum_{n_i,n_j=1}^{r/2} \Gamma(n_i\alpha, n_j\beta; n_j\alpha, n_i\beta) \\ + 3N = 4S(S+1). \tag{3.32}$$

3.4 Eigenvalue bounds of the problem structures

In order to be able to efficiently apply our rigorous verification algorithms, we need to know the eigenvalue bounds of all the structures present in the SDP formulation of the problem. Those include $\gamma, \Gamma, Q, G, T1$ and $T2$. The lower eigenvalue bound of all the operators is 0, which follows immediately from their positive semidefiniteness requirement (3.18). The upper bounds for all the above mentioned operators will be sequentially obtained in the following sections.

3.4.1 Maximal eigenvalues of 1-RDMs and 2-RDMs

In case of γ the upper bound is also straightforward and comes from the N-representability condition (3.17):

$$\lambda_{max}(\gamma) \leq 1. \tag{3.33}$$

For Γ an immediate upper bound can be obtained from its trace. Since the operator is positive semidefinite and thus possesses only nonnegative eigenvalues,

$$\lambda_{max}(\Gamma) \leq \sum_i \lambda_i(\Gamma) = \text{trace}(\Gamma) = N(N-1). \tag{3.34}$$

Though a tighter bound can be found. In their work [21] C. Garrod and J. K. Percus show, that the largest possible eigenvalue of Γ is N. Following their derivation, we will now carry out the proof in our notations.

In our matrix representation, $\Gamma(i,j;i',j')$ is an operator acting on antisymmetric matrices. The Rayleigh-Ritz theorem says:

$$\lambda_{max}(\Gamma) = \max_{\langle F,F\rangle=1} F^*\Gamma F, \tag{3.35}$$

where in our case, we optimize over antisymmetric matrices F, satisfying the norm condition $\langle F,F\rangle = 1$. Thus we have to obtain an upper bound to

$$\Lambda = F^*\Gamma F = \sum_{\substack{i,j\\i',j'}} F^*(i,j)\Gamma(i,j;i',j')F(i',j'), \tag{3.36}$$

where

$$F(i,j) = -F(j,i) \tag{3.37}$$

and

$$\langle F,F\rangle = \sum_{i,j} |F(i,j)|^2 = 1. \tag{3.38}$$

Without loss in generality F may be assumed real. Then the matrix iF is Hermitian and can thus be diagonalized in terms of its eigenvectors v_k.

$$iF(i,j) = \sum_{k=1}^{r} \lambda_k v_k(i) v_k^*(j) = \sum_{k=1}^{r} \lambda_k V_k(i,j), \tag{3.39}$$

where λ_k are the real eigenvalues of iF and $V_k = v_k v_k^*$. Since iF is a normal matrix,

$$\|F\|_2^2 = \sum_{i,j} |F(i,j)|^2 = \sum_k |\lambda_k|^2 \tag{3.40}$$

and equation (3.38) can now be written

$$\sum_k |\lambda_k|^2 = 1. \tag{3.41}$$

Expressing Λ in terms of the eigenvalues and eigenvectors of iF we obtain

$$\begin{aligned} \Lambda &= F^*\Gamma F = (iF)^*\Gamma(iF) \\ &= \sum_{\substack{i,j \\ i',j'}} \left(\sum_k \lambda_k V_k(i,j)\right) \Gamma(i,j;i',j') \left(\sum_l \lambda_l V_l(i',j')\right) \\ &\qquad = \sum_{k,l} \lambda_k \lambda_l \sum_{\substack{i,j \\ i',j'}} V_k(i,j)\Gamma(i,j;i',j')V_l(i',j') = \sum_{k,l} \lambda_k \tilde{\Gamma}(k,l)\lambda_l, \end{aligned} \tag{3.42}$$

where

$$\tilde{\Gamma}(k,l) = \sum_{\substack{i,j \\ i',j'}} V_k(i,j)\Gamma(i,j;i',j')V_l(i',j'). \tag{3.43}$$

Using again the Rayleigh-Ritz theorem and keeping in mind the normalization condition (3.41) we get

$$\Lambda = \sum_{k,l} \lambda_k \tilde{\Gamma}(k,l)\lambda_l = \lambda^T \tilde{\Gamma} \lambda \leq \lambda_{max}(\tilde{\Gamma}). \tag{3.44}$$

The fact that Γ is positive semidefinite implies that $\Lambda \geq 0$ for any choice of λ in (3.44). Then we can conclude, that $\tilde{\Gamma}$ is also positive semidefinite and the following bound is valid

$$\Lambda \leq \text{trace}(\tilde{\Gamma}). \tag{3.45}$$

According to the N-representability condition (3.18), G is a nonnegative operator:

$$\sum_{\substack{i,j \\ i',j'}} v_k(i)v_k^*(j)G(i,j;i',j')v_k(i')v_k^*(j') \geq 0. \tag{3.46}$$

Using the definition of G, we get

$$\sum_{\substack{i,j\\ i',j'}} v_k(i)v_k^*(j)\Gamma(i,j';j,i')v_k(i')v_k^*(j') \\ + \sum_{\substack{i,j\\ i',j'}} v_k(i)v_k^*(j)\delta(i,i')\gamma(j',j)v_k(i')v_k^*(j') \geq 0. \quad (3.47)$$

But, notice that

$$\sum_{\substack{i,j\\ i',j'}} v_k(i)v_k^*(j)\Gamma(i,j';j,i')v_k(i')v_k^*(j') \\ = -\sum_{\substack{i,j\\ i',j'}} v_k(i)v_k^*(j')\Gamma(i,j';i',j)v_k(i')v_k^*(j) = -\tilde{\Gamma}(k,k) \quad (3.48)$$

and

$$\sum_{\substack{i,j\\ i',j'}} v_k(i)v_k^*(j)\delta(i,i')\gamma(j',j)v_k(i')v_k^*(j') \\ = \sum_i v_k^2(i) \sum_{j,j'} v_k^*(j)\gamma(j,j')v_k^*(j') = v_k^{*T}\gamma v_k^*, \quad (3.49)$$

where we have used the normalization condition on v_k from (3.39) and Hermiticity of γ. As Hermitian, γ, in analogy to (3.39), can be represented in terms of its eigenvalues and eigenvectors

$$\gamma = \sum_{l=1}^{r} \mu_l u_l u_l^*. \quad (3.50)$$

The eigenvectors u_l, in their turn, have a representation in the orthonormal basis v_k^*:

$$u_l = \sum_m \alpha_{lm} v_m^*, \quad (3.51)$$

with

$$\sum_m \alpha_{lm}^2 = 1. \quad (3.52)$$

Going back to equation (3.49) and taking sum over all k we obtain

$$\sum_k v_k^{*T}\gamma v_k^* = \sum_{k,l} \mu_l v_k^{*T} u_l u_l^* v_k^* = \sum_l \mu_l \sum_k v_k^{*T} u_l u_l^* v_k^* \\ = \sum_l \mu_l \sum_k \alpha_{lk}^2 = \sum_l \mu_l = \text{trace}(\gamma) = N. \quad (3.53)$$

Considering the results of (3.48), (3.49) and (3.53), (3.47) becomes

$$-\text{trace}(\tilde{\Gamma}) + N \geq 0. \tag{3.54}$$

This brings us to the conclusion $\Lambda \leq N$ and completes the proof.

3.4.2 Maximal eigenvalues of other matrices

To determine the maximal eigenvalue bounds for Q, G, $T1$ and $T2$ matrices, two different approaches were considered. Firstly, the Weyl theorem can be used. That is for two Hermitian matrices A and B we have

$$\lambda_{max}(A + B) \leq \lambda_{max}(A) + \lambda_{max}(B). \tag{3.55}$$

Secondly, the same idea as in (3.34) can be used, where we took the trace as an upper bound for positive semidefinite matrix eigenvalues.

Both strategies will now be compared by the example of the Q matrix (3.21). Following (3.55) we obtain the bound

$$\begin{aligned}\lambda_{max}(Q(i,j;i',j')) \leq & \lambda_{max}(\Gamma(i,j;i',j')) + \lambda_{max}(\delta(i,i')(-\gamma(j,j'))) \\ & + \lambda_{max}(\delta(j,j')(-\gamma(i,i'))) + \lambda_{max}(\delta(i,j')\gamma(j,i')) \\ & + \lambda_{max}(\delta(j,i')\gamma(i,j')) + \lambda_{max}(\delta(i,i')\delta(j,j')) \\ & + \lambda_{max}(\delta(i,j')(-\delta(j,i'))).\end{aligned} \tag{3.56}$$

To proceed, a matrix representation of the four and six index structures has to be introduced. We used the following mappings:

$$\begin{aligned}&A(i,j;i',j') = A_m((i-1)r + j,\ (i'-1)r + j'),\\ &B(i,j,k;i',j',k') = B_m((i-1)r^2 + (j-1)r + k,\ (i'-1)r^2 + (j'-1)r + k'),\end{aligned} \tag{3.57}$$

where $A_m \in \mathbb{R}^{r^2 \times r^2}$, $B_m \in \mathbb{R}^{r^3 \times r^3}$ and r is the basis size. In this representation $\delta(i,i')\gamma(j,j') = \gamma_{diag}$ is a block diagonal matrix with γ as each of r blocks. Then

$$\lambda_{max}(-\gamma_{diag}) = -\lambda_{min}(\gamma) = 0, \tag{3.58}$$

$$\rho(\gamma_{diag}) = 1, \tag{3.59}$$

where $\rho(A)$ denotes the spectral radius of a matrix. Other matrices of type $\delta\gamma$ are permutations of γ_{diag}. Some, as for example $\delta(i,j')\gamma(j,i')$, can be obtained with only column permutations. For others both column and row interchanges are necessary. In general case, any matrix of type $\delta\gamma$ present in (3.56) can be represented as $P_1\gamma_{diag}P_2$, where P_1 and P_2 are symmetric permutation matrices. Having said that, the upper limit

$$\lambda_{max}(\delta\gamma) \leq \|P_1\gamma_{diag}P_2\|_2 \leq \|P_1\|_2\|\gamma_{diag}\|_2\|P_2\|_2 = \rho(\gamma_{diag}) = 1 \tag{3.60}$$

follows immediately. Here we utilized the fact, that spectral norm of a symmetric real-valued matrix is equal to its spectral radius. In their turn, $\delta(i,i')\delta(j,j')$ is an identity matrix of size r^2 and $\delta(i,j')\delta(j,i')$ is its symmetric permutation (and thus a symmetric permutation matrix itself). Obviously

$$\lambda_{max}(-\delta(i,j')\delta(j,i')) \leq \lambda_{max}(\delta(i,j')\delta(j,i')) = 1. \tag{3.61}$$

Now we are ready to calculate the upper eigenvalue bound for Q by substituting the results of (3.58), (3.60), (3.61) into (3.56) and using the eigenvalue bound on Γ from the previous section.

$$\lambda_{max}(Q) \leq N + 0 + 1 + 1 + 1 + 1 + 1 = N + 5. \tag{3.62}$$

Another bound based on the matrix trace equals:

$$\begin{aligned} \text{trace}(Q) &= N(N-1) + \sum_{i,j} (-\delta(i,i)\gamma(j,j) - \delta(j,j)\gamma(i,i) \\ &\quad +\delta(i,j)\gamma(j,i) + \delta(j,i)\gamma(i,j) + \delta(i,i)\delta(j,j) - \delta(i,j)\delta(j,i)) \\ &= N(N-1) - rN - rN + N + N + r^2 - r \\ &= (r-N)(r-N-1). \end{aligned} \tag{3.63}$$

None of the results (3.62) or (3.63) delivers in the case of Q matrix an unambiguously better bound. The quality of the estimations depends on the electron number and the corresponding basis size. Concerning other matrices, one can notice that trace in average shows a tighter result compared to that from the Weyl theorem. Plus no assumptions about mappings like those in (3.57) or similar are necessary for trace calculation. It is also worth mentioning, that for our verification algorithm the difference between the obtained bounds is not crucial (they will be multiplied by feasibility defects of the calculated approximation, which are supposed to be close to zero). Therefore, we will be using traces to constrain maximal eigenvalues of positive semidefinite matrices.

Trace calculations for the remaining matrices yield

$$\text{trace}(G) = -N(N-1) + rN = N(r-N+1), \tag{3.64}$$

$$\begin{aligned}\text{trace}(T1) =& 3N(N-1)r - 6N(N-1) + 3N(r-r^2) + 6N(r-1)\\ &+ r^3 - 3r^2 + 2r = (r-2)(r(r-1) - 3N(r-N)),\end{aligned} \tag{3.65}$$

$$\text{trace}(T2) = -rN(N-1) + r^2N - rN = rN(r-N). \tag{3.66}$$

3.4.3 Considerations

Beside the obtained upper eigenvalue bounds

$$\begin{aligned}&\lambda_{max}(\gamma) \leq 1,\\ &\lambda_{max}(\Gamma) \leq N,\\ &\lambda_{max}(G) \leq N(r-N+1),\\ &\lambda_{max}(Q) \leq (r-N)(r-N-1),\\ &\lambda_{max}(T1) \leq (r-2)(r(r-1) - 3N(r-N)),\\ &\lambda_{max}(T2) \leq rN(r-N)\end{aligned} \tag{3.67}$$

themselves, of importance for us are some relations between the number of particles N and the basis size r, that can be derived from these inequalities. Indeed, if we require positive semidefiniteness of the matrices, that is nonnegativity of their traces, we immediately get

$$r \geq N+1 \tag{3.68}$$

from the condition on Q matrix, and

$$N \geq 3 \tag{3.69}$$

if we want trace$(T1)$ to be always nonnegative. Thus, the approach works only for systems with not less than 3 electrons and the minimum required basis size is 4. In addition the condition (3.68) must be satisfied.

All the bounds obtained so far were derived solely for matrices satisfying the N-representability conditions formulated in section 3.3 and are implementation independent. To proceed, we have to introduce “compacted” matrices, as found for example in

[79] and [78], and the respective maximal eigenvalue bounds. After performing a mapping similar to (3.57), we get symmetric matrices of dimensions r^2 and r^3. As a result of the antisymmetricity of operators $\Gamma, Q, T1$ and $T2$, the corresponding matrices have both dependent and zero rows and columns. This gives us a possibility to drastically reduce the matrix dimensions by omitting such entries. This elimination also helps to fight the ill-posedness of the initial problem. If the original matrices necessarily had zero eigenvalues, and were thus always on the boundary of the semidefinite cone, their "compacted" versions can lie in its interior. The new dimensions are: $\tilde{\Gamma}$, $\tilde{Q} \in \mathbb{S}^{C_r^2}$, $\tilde{T}1 \in \mathbb{S}^{C_r^3}$ and $\tilde{T}2 \in \mathbb{S}^{r \times C_r^2}$. Furthermore, by choosing an appropriate mapping (rearranging rows and columns), matrices γ, $\tilde{\Gamma}$, $\tilde{Q}$, G, $\tilde{T}1$ and $\tilde{T}2$ can be made block diagonal. This is possible due to the spin symmetries (3.24) - (3.29).

Since the "compacted" matrices also have their positive semidefiniteness required, traces still can be used to estimate their maximal eigenvalues. Unfortunately, an explicit trace calculation for $\tilde{Q}$, $\tilde{T}1$ and $\tilde{T}2$ is not trivial. Nevertheless, since all diagonal elements of a positive semidefinite matrix are nonnegative, transformation to "compacted" form can only reduce traces of the matrices involved: trace$(\tilde{M}) \leq$ trace(M). Thus, the corresponding eigenvalue bounds from (3.67) are still valid, though with a greater overestimation.

In the case of $\tilde{\Gamma}$ the trace bound would be

$$\text{trace}(\tilde{\Gamma}) = \frac{1}{2}\text{trace}(\Gamma) = \frac{1}{2}N(N-1). \tag{3.70}$$

Nevertheless, analogously to Γ, we can prove N to be a suitable eigenvalue bound for it. Note, that the following decomposition is valid:

$$\Gamma = \Gamma_{pos} + \Gamma_{neg}, \tag{3.71}$$

where Γ_{pos} contains elements corresponding to $\Gamma(i,j;i',j')$ for $i < j$, $i' < j'$ and $i > j$, $i' > j'$, and Γ_{neg} is a negative row *or* column permutation of Γ_{pos}.

Example 3.1. *To illustrate the decomposition and the structure of* Γ*, consider a very simple case of* $r = 3$*. According to (3.68) and (3.69) this is not an appropriate basis size for the RDM problems considered, but for* Γ *alone it's acceptable and allows us to observe the properties on a matrix of reasonable size. We do not give attention to spin symmetries, and only symmetric/antisymmetric behavior of the operator is reflected.*

As a product of mappings (3.57) comes a 9×9 *matrix.*

$$\Gamma = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g_1 & g_4 & -g_1 & 0 & g_5 & -g_4 & -g_5 & 0 \\ 0 & g_4 & g_2 & -g_4 & 0 & g_6 & -g_2 & -g_6 & 0 \\ 0 & -g_1 & -g_4 & g_1 & 0 & -g_5 & g_4 & g_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g_5 & g_6 & -g_5 & 0 & g_3 & -g_6 & -g_3 & 0 \\ 0 & -g_4 & -g_2 & g_4 & 0 & -g_6 & g_2 & g_6 & 0 \\ 0 & -g_5 & -g_6 & g_5 & 0 & -g_3 & g_6 & g_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \Gamma_{pos} + \Gamma_{neg} \ , \quad (3.72)$$

where

$$\Gamma_{pos} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g_1 & g_4 & 0 & 0 & g_5 & 0 & 0 & 0 \\ 0 & g_4 & g_2 & 0 & 0 & g_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & g_1 & 0 & 0 & g_4 & g_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & g_5 & g_6 & 0 & 0 & g_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & g_4 & 0 & 0 & g_2 & g_6 & 0 \\ 0 & 0 & 0 & g_5 & 0 & 0 & g_6 & g_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.73)$$

and

$$\Gamma_{neg} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -g_1 & 0 & 0 & -g_4 & -g_5 & 0 \\ 0 & 0 & 0 & -g_4 & 0 & 0 & -g_2 & -g_6 & 0 \\ 0 & -g_1 & -g_4 & 0 & 0 & -g_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -g_5 & 0 & 0 & -g_6 & -g_3 & 0 \\ 0 & -g_4 & -g_2 & 0 & 0 & -g_6 & 0 & 0 & 0 \\ 0 & -g_5 & -g_6 & 0 & 0 & -g_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} . \quad (3.74)$$

$g_1 \ldots g_6$ *denote unique elements of* Γ *present also in* $\tilde{\Gamma}$*:*

$$\tilde{\Gamma} = \begin{pmatrix} g_1 & g_4 & g_5 \\ g_4 & g_2 & g_6 \\ g_5 & g_6 & g_3 \end{pmatrix} . \quad (3.75)$$

It can be seen, that

$$\Gamma_{neg} = -\Gamma_{pos} P \tag{3.76}$$

where

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.77}$$

is a symmetric permutation matrix with eigenvalues 1 *and* -1.

In the general case, Γ_{pos} has in addition to r zeros also C_r^2 distinct eigenvalues $\lambda_1, \dots, \lambda_{C_r^2}$ (those of $\tilde{\Gamma}$). They are nonnegative and have algebraic multiplicity 2. Γ_{neg}, in its turn, has $-\lambda_{C_r^2}, \dots, -\lambda_1, \lambda_1, \dots, \lambda_{C_r^2}$ plus r zeros as its eigenvalues. An important conclusion can be drawn

$$\lambda_{max}(\tilde{\Gamma}) = \lambda_{max}(\Gamma_{pos}) = \lambda_{max}(\Gamma_{neg}). \tag{3.78}$$

Putting now (3.78), (3.71), (3.67) together and keeping in mind that Γ_{pos} is positive semidefinite, we prove the suggested bound

$$\lambda_{max}(\tilde{\Gamma}) = \lambda_{max}(\Gamma_{neg}) \leq \lambda_{max}(\Gamma_{neg} + \Gamma_{pos}) = \lambda_{max}(\Gamma) \leq N. \tag{3.79}$$

Thus, the maximal eigenvalue bounds for “compacted” matrices are:

$$\begin{aligned} &\lambda_{max}(\tilde{\Gamma}) \leq N, \\ &\lambda_{max}(\tilde{Q}) \leq (r-N)(r-N-1), \\ &\lambda_{max}(\tilde{T}1) \leq (r-2)(r(r-1) - 3N(r-N)), \\ &\lambda_{max}(\tilde{T}2) \leq rN(r-N). \end{aligned} \tag{3.80}$$

3.5 SDP implementations

Owing to the character of the constraints, RDM optimization problems can be naturally represented as semidefinite programs. Obviously, there are numerous possibilities to formulate such an SDP. A very detailed study on it can be found in the work of Zhao [78]. Two main approaches, the so called “primal” and “dual” formulations, are presented there. The names tell us which part of an SDP carries the physical meaning of the initial electronic structure problem. They are not in any way SDP primal and dual of each other.

For the “primal” problem formulation the objective function is straightforward and appeared already in (3.11) and (3.13)

$$E = \min(\langle H_1, \gamma\rangle + \langle \tilde{H}_2, \tilde{\Gamma}\rangle). \tag{3.81}$$

As primal variable a block diagonal matrix

$$X = \text{Diag}(\gamma,\ I-\gamma,\ \tilde{\Gamma},\ \tilde{Q},\ G,\ \tilde{T}1,\ \tilde{T}2) \tag{3.82}$$

is taken. This guarantees us the fulfillment of N-representability conditions (3.17) and (3.18). With such choice of X, the objective C matrix has only 2 nonzero blocks:

$$C = \text{Diag}(H_1,\ O,\ \tilde{H}_2,\ O,\ O,\ O,\ O). \tag{3.83}$$

The fulfillment of equality constraints like (3.14) to (3.16), dependences (3.19) to (3.23) and relations (3.30) to (3.32) can also be secured by the appropriate choice of the A_i matrices and the b vector. For an extensive description and implementation details see [78].

To compute a fast lower bound to the ground state energy (primal optimal value) in this formulation, the Theorem 2.2 can be applied directly. For an approximate dual solution $\tilde{y}$ the residual matrix $D = C - \sum \tilde{y}_i A_i$ needs to be calculated. The lower eigenvalue bounds $\underline{d}_j$ to all of its n blocks can then be determined. Using them and the upper eigenvalue bounds $\overline{x}_j$ of the X blocks, given in the previous section, the right side of the inequality

$$E \geq \inf\{b^T\tilde{y} + \sum_{j=1}^{n} s_j \cdot \underline{d}_j^- \cdot \overline{x}_j\} \tag{3.84}$$

can be evaluated. Here s_j are the blocks sizes.

Yet because of its size, the “primal” SDP formulation described here is not computationally competitive with a less straightforward “dual” one [78], and thus the lower bound in (3.84) is rather of theoretical interest.

3.5.1 Rigorous error bounds for the RDM method in “dual” SDP formulation

In this approach the ground state energy is given by a dual SDP objective function. We define the dual variable

$$y = (\text{svec}(\gamma,\ \sqrt{2}); \text{svec}(\tilde{\Gamma},\ \sqrt{2})) \tag{3.85}$$

and the parameters vector

$$b = -(\text{svec}(H_1,\ \sqrt{2}); \text{svec}(\tilde{H}_2,\ \sqrt{2})). \tag{3.86}$$

Then the objective function turns to

$$E = -\max b^T y. \tag{3.87}$$

The difficulty with the “dual” formulation is that it does not allow any equality constraints. To overcome the problem, the equalities will be relaxed with a gap of $2\epsilon_d$. For example instead of

$$\text{trace}(\gamma) - N = 0 \tag{3.88}$$

we will use

$$\begin{aligned} \text{trace}(\gamma) - N + \epsilon_d \geq 0, \\ -\text{trace}(\gamma) + N + \epsilon_d \geq 0, \end{aligned} \tag{3.89}$$

where ϵ_d is a small positive number (in her work Z. Zhao used values 10^{-7} and 10^{-5}). Such relaxations introduce an additional positive semidefinite diagonal matrix D_a having entries similar to the left hand sides in (3.88) for all the equalities present in the RDM problem. Then, in analogy to the “primal” case, we define our dual matrix variable as

$$Z = C - \sum_i y_i A_i = \text{Diag}(\gamma,\ I - \gamma,\ \tilde{\Gamma},\ \tilde{Q},\ G,\ \tilde{T}1,\ \tilde{T}2,\ D_a). \tag{3.90}$$

The upper eigenvalue bound for D_a is obvious. In the worst case, one or several of the inequalities reach the gap edge, which means none of the entries can be greater than $2\epsilon_d$. Since it’s diagonal,

$$\lambda_{max}(D_a) \leq 2\epsilon_d. \tag{3.91}$$

The introduced relaxation also influences the upper eigenvalue bounds of the other blocks in Z. Since all the N-representability equalities (including traces of γ and Γ) are now fulfilled only with ϵ_d tolerance, the appropriate corrections have to be introduced. Only $\lambda_{max}(\gamma) \leq 1$ remains unchanged. As $\lambda_{max}(\tilde{\Gamma}) \leq \text{trace}(\gamma)$,

$$\lambda_{max}(\tilde{\Gamma}) \leq N + \epsilon_d \tag{3.92}$$

is the new valid bound for the 2-RDM matrix. For other matrices, ϵ_d has to appear in their trace calculations everywhere γ or Γ occurs, plus it should also be added to the traces at the end.

The matrices A_i and C in (3.90) are chosen to reflect the N-representability conditions and to relate all the blocks of Z to γ and $\tilde{\Gamma}$.

For obtaining a fast rigorous lower bound of the ground state energy, one approach would be to use Theorem 2.5 and to calculate the dual optimal value upper bound

$$\underline{E} = -\overline{f_d} = -\langle C, X\rangle - \overline{y}^T\overline{r} \tag{3.93}$$

knowing the bounds of the dual feasible variable vector $|y| \leq \overline{y}$. Another solution to this problem could be reformulating the problem so, that the dual matrix variable Z becomes a primal variable X' of a new problem. Then the eigenvalue bounds from Section 3.4 and the formula (3.84) can be applied. We will proceed as follows: first the latter approach will be discussed and the reformulation introduced; afterwards the algorithm involving the calculation of the dual optimal value upper bound will be presented.

Though problems in the “primal” SDP formulation are too big to consider solving them, their structures can be used to calculate a lower bound of the primal optimal value (ground state energy) if the solution is known. A high-level description of the algorithm follows:

Algorithm 3.1.

1. *Solve the RDM problem in “dual” formulation and get the primal and dual approximate solution matrices $\tilde{X}$ and $\tilde{Z}$.*
2. *Write down a new problem having $\tilde{Z}$ as a primal feasible solution and $\tilde{E} = \langle C_{new}, \tilde{Z}\rangle$ as its primal optimal value.*
3. *Express $\tilde{y}_{new}$ in terms of the original problem structures and its approximate solution.*

4. *Find the rigorous lower bound $\underline{E}$ using the formula (3.84) applied to the reformulated problem.*

Steps 2 and 3 of the Algorithm 3.1 require some explanations.

To tell apart old and new structures, the latter will be primed. We start by preserving the energy value.

$$E = -b^T y = \langle C',\ X' \rangle, \tag{3.94}$$

where

$$X' = Z = \text{Diag}(\gamma,\ I-\gamma,\ \tilde{\Gamma},\ \tilde{Q},\ G,\ \tilde{T}1,\ \tilde{T}2,\ D_a). \tag{3.95}$$

This gives us the C' matrix:

$$C' = \text{Diag}(-\text{smat}(b_1,\ \sqrt{2}),\ O,\ -\text{smat}(b_2,\ \sqrt{2}),\ O,\ O,\ O,\ O,\ O), \tag{3.96}$$

where b_1 and b_2 are two parts of the b vector in (3.86) of length $l_1 = s_1(s_1+1)/2$ and $l_2 = s_3(s_3+1)/2$, so that $b = (b_1;\ b_2)$ and $l_1 + l_2 = m$. Of course this relation is identical to (3.83), but our task is to express the primed matrices as functions of the "dual" formulation structures, and in that way, the formula (3.96) is preferred. The next step will be to use the equality constraints of the new problem to guarantee $X' = Z$. We have

$$X' = C - \sum_{i=1}^{m} y_i A_i \tag{3.97}$$

and

$$y = (\text{svec}(X_1',\ \sqrt{2}); \text{svec}(X_3',\ \sqrt{2})), \tag{3.98}$$

where X_1' and X_3' are the first and the third blocks of X' in (3.95). Thus we can reach the aim by having an equality for every unique element in X'. The total amount of equalities will then be

$$m' = \sum_{j=1}^{8} \frac{1}{2} s_j(s_j+1), \tag{3.99}$$

where s_j is again the size of the jth block. Let us now rewrite (3.97) with the substitution of (3.98). We have

$$\begin{pmatrix} x'_{11} & x'_{12} & \cdots & 0 \\ x'_{12} & x'_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x'_{pp} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \cdots & 0 \\ c_{12} & c_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{pp} \end{pmatrix} - A_1 \cdot x'_{11} - A_2 \cdot \sqrt{2} x'_{12} - \ldots$$
$$- A_{l_1} \cdot x'_{s_1 s_1} - A_{l_1+1} \cdot x'_{(s_1+s_2+1)\,(s_1+s_2+1)} - A_{l_1+2} \cdot \sqrt{2} x'_{(s_1+s_2+1)\,(s_1+s_2+2)} - \ldots$$
$$- A_m \cdot x'_{(s_1+s_2+s_3)\,(s_1+s_2+s_3)}, \quad (3.100)$$

where $p = \sum_{j=1}^{8} s_j$ is the total size of X'. Further

$$x'_{11} \left(\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} + A_1 \right) + x'_{12} \left(\begin{pmatrix} 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} + \sqrt{2} A_2 \right) + \ldots$$
$$+ x'_{pp} \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & \cdots & 0 \\ c_{12} & c_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{pp} \end{pmatrix}. \quad (3.101)$$

Now it is easy to see the structure of the A'_i matrices. For example

$$\langle A'_1,\ X' \rangle = \langle \begin{pmatrix} 1 + A_1(1,1) & \frac{1}{\sqrt{2}} A_2(1,1) & \cdots & 0 \\ \frac{1}{\sqrt{2}} A_2(1,1) & A_{s_1+1}(1,1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix},\ X' \rangle$$
$$= C(1,1) = b'_1; \quad (3.102)$$

$$\langle A'_2,\ X' \rangle = \langle \begin{pmatrix} A_1(1,2) & \frac{1}{2}(1 + \sqrt{2} A_2(1,2)) & \cdots & 0 \\ \frac{1}{2}(1 + \sqrt{2} A_2(1,2)) & A_{s_1+1}(1,2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix},\ X' \rangle$$
$$= C(1,2) = b'_2; \quad (3.103)$$

and

$$\langle A'_{m'},\ X'\rangle = \langle \begin{pmatrix} A_1(p,p) & \frac{1}{\sqrt{2}}A_2(p,p) & \cdots & 0 \\ \frac{1}{\sqrt{2}}A_2(p,p) & A_{s_1+1}(p,p) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix},\ X'\rangle$$

$$= C(p,p) = b'_{m'}. \quad (3.104)$$

The b' vector is then

$$b' = (\text{svec}(C_1,\ 1); \ldots;\ \text{svec}(C_8,\ 1)). \quad (3.105)$$

Finally, the dual variable vector y' is chosen so that

$$b'^T y' = -\langle C,\ X\rangle. \quad (3.106)$$

This yields

$$y' = (-\text{svec}(X_1,\ 2); \ldots;\ -\text{svec}(X_8,\ 2)). \quad (3.107)$$

At this point, the reformulation can be considered complete. C', A'_i, b', X' and y' of the new problem can now be found, given the original RDM problem in "dual" formulation. According to step 4 of the Algorithm 3.1, the ground state energy lower bound can be calculated now.

However, this approach did not turn out to be a good solution to the lower bound problem. The reformulated SDP is very similar to the "primal" formulation described before. The only significant difference is that we transfer the relaxed equalities from the "dual" formulation instead of using them directly. The new SDP is large. The problem has the same block structure, but the number of constraints grows drastically. For RDMs with basis size $r = 12$, for example, we get 121 063 instead of 948 constraints. For $r = 20$ the number goes from 7 230 up to 2 595 595. From the other side, the reformulated problem does not have to be solved and the sparsity of its matrices is much higher than in the "dual" problem itself.

The trouble is hidden in the residual matrix D'. Though all A'_i matrices themselves are extremely sparse (it's not obvious from (3.102), (3.103) and (3.104), but except for the blocks 1 and 3, A'_is have at most 2 nonzero entries), they "perfectly" complement each other. Pooled together in D', they produce a block diagonal matrix which, in general case, does not have a single nonzero element in its blocks. The time to calculate its eigenvalues, but, above all, to allocate a dense matrix of such size is no longer competitive. Based on the numerical experience, we can say, that calculating the energy lower bound according to the Algorithm 2.2 is in average faster, though it requires a rigorous solution of a liner system.

Now we will concentrate on the other approach, which was previously mentioned in (3.93). There the ground state energy lower bound is calculated as a negative SDP upper bound directly for the RDM in "dual" formulation. We start by posing an algorithm.

Algorithm 3.2.

1. *Solve the RDM problem in "dual" formulation and get the approximate primal solution* $\tilde{X}$.
2. *Compute a rigorous bound* $\underline{x} \leq \{\lambda_{min}(\tilde{X}),\ 0\}$. *Then shift the matrix* $\tilde{X}' = \tilde{X} - \underline{x}I$.
3. *Calculate rigorous upper bounds on the residuals* $\overline{r} \geq |\langle A_i,\ \tilde{X}'\rangle - b_i|$.
4. *Find the lower bound*
$$\underline{E} = -\langle C, \tilde{X}'\rangle - \overline{y}^T\overline{r}, \tag{3.108}$$
where $\overline{y}$ *is a bound on* $\tilde{y}$ *such that* $|\tilde{y}| \leq \overline{y}$.

Here step 2 is performed to compensate possible deviations from positive semidefiniteness in the computed approximation.

Essential for a successful use of the Algorithm 3.2 is the knowledge of the bound $\overline{y}$. If we look at (3.85), it becomes evident, that we can achieve this by bounding the elements of γ and $\tilde{\Gamma}$. Let $M = [m_{ij}]$ be a Hermitian positive semidefinite matrix. The diagonal entries then satisfy the inequality

$$0 \leq \lambda_{min} \leq m_{ii} \leq \lambda_{max}, \tag{3.109}$$

which can be seen as, for example, a corollary of the Rayleigh-Ritz theorem. Here λ_{min} and λ_{max} are again the minimal and the maximal eigenvalue of M. It is also known, that

$$|m_{ij}| \leq \frac{m_{ii} + m_{jj}}{2} \leq \lambda_{max}. \tag{3.110}$$

Thus, all the components of $\overline{y}$ are available, given the eigenvalue bounds (3.67) (for γ) and (3.92) (for $\tilde{\Gamma}$), and the Algorithm 3.2 can now be applied to calculate a rigorous lower bound of the ground state energy.

3.5.2 Numerical results

In this section the calculated optimal value bounds for the electronic structure benchmark problems are presented. The SDPs were taken from [18] and are those found in Tables 1 and 2 in [79]. As an approximate semidefinite solver, SDPA [17] was used in all calculations. Rigorous bounds were obtained with the help of verifiedSDP, a verification software package developed in the context of this thesis. This package is described in more detail in Chapter 4 and in Appendix B.

Results could be obtained for all the problems from the benchmark set with the only exception of Fluoromethylidyne (CF). The problem file in SDPA format appeared to be inconsistent (for example, the electron number equaled 12 instead of 15) resulting in incorrect output.

Detailed numerical results can be found in the Tables 3.1, 3.2 and 3.3. Table 3.1 contains the approximate value of the ground state energy $\tilde{E}^*$[2] and its rigorous bounds. The $\overline{E^*}$ and $\underline{E_1^*}$ are the upper and the lower optimal value bounds obtained according to the Algorithms 2.1 and 2.2 respectively. The value $\underline{E_2^*}$ is the lower energy bound calculated following the Algorithm 3.2 in the previous section. In Table 3.2 the accuracies of the computed approximations and bounds are shown. The quantity

$$\mu(a,b) := \frac{a-b}{\max\{1.0, (|a|+|b|)/2\}} \tag{3.111}$$

measures the relative accuracy. Table 3.3 reveals the time costs $\tilde{t}$, $\overline{t}$, $\underline{t}_1$ and $\underline{t}_2$ for calculating $\tilde{E}^*$, $\overline{E^*}$, $\underline{E_1^*}$ and $\underline{E_2^*}$ respectively. The ratio $\underline{t}_2/\underline{t}_1$ is also shown to indicate the time benefit of using the Algorithm 3.2 to compute the lower bound of the ground state energy.

Table 3.1:
Rigorous bounds for the electronic structure benchmark problems (all energies are in Hartree).

Problem	$\tilde{E}^*$	$\overline{E^*}$	$\underline{E_1^*}$	$\underline{E_2^*}$
LiH	-7.701590	-7.701346	-8.311430	-8.328970
BeH	-15.031051	-15.031020	-15.211742	-15.223065
BH^+	-24.711638	-24.711546	-24.908349	-24.926349
BH	-24.868066	-24.868023	-25.230332	-25.252726
CH^+	-37.791631	-37.791645	-37.990384	-38.007703
CH	-38.072447	-38.072351	-38.305066	-38.330361

continued. . .

[2] $\tilde{E}^*$ results after adding the nuclear repulsion energy to the negative dual optimal value of the semidefinite program.

Problem	$\tilde{E}^*$	$\overline{E^*}$	$\underline{E}_1^*$	$\underline{E}_2^*$
CH^-	-37.790264	-37.790147	-38.079407	-38.097093
NH^+	-54.259270	-54.259208	-54.530753	-54.553506
NH	-54.755845	∞	-54.863067	-54.895371
NH^-	-54.343880	-54.343584	-54.679564	-54.732619
OH^+	-74.699369	-74.699390	-74.824963	-74.854506
OH	-74.915322	-75.045130	-75.270221	-75.294777
OH^-	-74.690903	-74.690903	-74.965723	-74.986655
HF^+	-99.063292	∞	-99.179959	-99.207903
HF	-99.525649	-99.525649	-99.525654	-99.525655
BH_2	-25.599463	-25.481014	-25.795225	-25.837000
$CH_2(1)$	-38.565139	-38.565116	-39.001309	-39.101874
$CH_2(3)$	-38.759072	-38.759072	-38.925286	-38.957234
NH_2	-55.210014	-55.210014	-55.582395	-55.642632
H_2O^+	-75.319716	-75.192582	-75.603978	-75.558738
H_2O	-75.571667	-75.571528	-75.847698	-75.975893
NH_3	-56.014499	-56.014488	-56.014497	-56.014497
H_3O^+	-76.104886	-76.104873	-76.104889	-76.104889
O_2^+	-148.796142	-148.796108	-148.796142	-148.796153
O_2	-149.163951	-149.163935	-149.163955	-149.163955
SiH	-288.377609	-288.377588	-288.377631	-288.377631
SiH^-	-288.131908	∞	-288.131926	-288.131927
NO^-	-128.665758	-128.665739	-5356.284879	-128.665771
NF	-153.244885	-153.244876	-153.244892	-153.244893
HS^+	-396.498571	-396.498532	-396.498590	-396.498591
Li_2	-14.837748	-14.837743	-14.837748	-14.837748
B_2	-49.017651	-49.017610	-49.017658	-49.017702
C_2^+	-75.078961	-75.078938	-75.078958	-75.078966
C_2	-75.438209	-75.438194	-75.438217	-75.438211
C_2^-	-75.316191	-75.316172	-75.316195	-75.316205
LiF	-106.444036	-106.444025	-106.444034	-106.444035
BeO	-89.201540	-89.201526	-89.201543	-89.201550
NaH	-161.738049	-161.738037	-161.738059	-161.738059
BeF	-113.641045	-113.641029	-113.641048	-113.641049
BO	-99.259120	-99.259100	-99.259132	-99.259143
N_2^+	-108.224658	-108.224625	-108.224668	-108.224677
N_2	-108.701822	-108.701805	-108.701819	-108.701820
CO^+	-112.037934	-112.037906	-112.037947	-112.037959
CO	-112.443919	-112.443901	-112.443921	-112.443923
BF	-123.612505	-123.612495	-123.612516	-123.612517
AlH	-241.507340	-241.507326	-241.507340	-241.507342

Table 3.2:
Accuracy of the rigorous bounds for the electronic structure problems. Here $\tilde{E}_d^*$ is the approximate dual optimal value computed by SDPA.

Problem	$\mu(\tilde{E}^*, \tilde{E}_d^*)$	$\mu(\overline{E^*}, \underline{E}_1^*)$	$\mu(\overline{E^*}, \underline{E}_2^*)$
LiH	$6.77495250e-02$	$6.77775244e-02$	$6.96582874e-02$
BeH	$1.08206339e-02$	$1.08225341e-02$	$1.14967558e-02$
BH^+	$7.28876060e-03$	$7.29217873e-03$	$7.95650788e-03$
BH	$1.33205011e-02$	$1.33220720e-02$	$1.41396765e-02$
CH^+	$4.88354276e-03$	$4.88320678e-03$	$5.30760949e-03$
CH	$5.67033815e-03$	$5.67267150e-03$	$6.28733079e-03$
CH^-	$7.07387615e-03$	$7.07674557e-03$	$7.50780808e-03$
NH^+	$4.69232879e-03$	$4.69339181e-03$	$5.08565993e-03$
NH	$1.83471686e-03$	NaN	NaN
NH^-	$5.78270177e-03$	$5.78782558e-03$	$6.69873214e-03$
OH^+	$1.59228232e-03$	$1.59200965e-03$	$1.96619123e-03$
OH	$4.46647622e-03$	$2.83050144e-03$	$3.13879648e-03$
OH^-	$3.47026390e-03$	$3.47026390e-03$	$3.73408862e-03$
HF^+	$1.11893784e-03$	NaN	NaN
HF	$5.16619753e-08$	$5.16619753e-08$	$5.20739583e-08$
BH_2	$6.43439292e-03$	$1.03477637e-02$	$1.17154792e-02$
$CH_2(1)$	$9.71668378e-03$	$9.73075965e-03$	$1.19607843e-02$
$CH_2(3)$	$3.69216243e-03$	$3.69216243e-03$	$4.40029311e-03$
NH_2	$5.91453905e-03$	$5.91453905e-03$	$6.86799895e-03$
H_2O^+	$2.18488454e-03$	$4.88617681e-03$	$4.35002510e-03$
H_2O	$3.25107803e-03$	$3.25271072e-03$	$4.75898359e-03$
NH_3	$-2.88982565e-08$	$1.33048997e-07$	$1.34286355e-07$
H_3O^+	$3.18258752e-08$	$1.79953070e-07$	$1.80871458e-07$
O_2^+	$-7.33315858e-10$	$1.86976863e-07$	$2.52641619e-07$
O_2	$2.26148021e-08$	$1.14560700e-07$	$1.15510751e-07$
SiH	$7.47490148e-08$	$1.45456042e-07$	$1.46151661e-07$
SiH^-	$6.19425910e-08$	NaN	NaN
NO^-	$8.74319781e-08$	$1.89009929e+00$	$2.07987491e-07$
NF	$4.28158379e-08$	$9.10670206e-08$	$9.20458830e-08$
HS^+	$4.70607402e-08$	$1.45426257e-07$	$1.45935459e-07$
Li_2	$-1.40985166e-08$	$2.82259208e-07$	$2.83821771e-07$
B_2	$-3.07284567e-07$	$8.30115744e-07$	$1.60930716e-06$
C_2^+	$-3.93277769e-08$	$2.26425697e-07$	$3.10565764e-07$
C_2	$-3.78755770e-08$	$2.51955508e-07$	$1.86173826e-07$
C_2^-	$-4.42974374e-08$	$2.49676326e-07$	$3.62031476e-07$
LiF	$-1.19280976e-08$	$7.90331413e-08$	$8.33049230e-08$
BeO	$3.31789806e-08$	$1.67925245e-07$	$2.29854405e-07$

continued...

Problem	$\mu(\tilde{E}^*, \tilde{E}^*_d)$	$\mu(\overline{E^*}, \underline{E}^*_1)$	$\mu(\overline{E^*}, \underline{E}^*_2)$
NaH	$5.91194074e-08$	$1.32461766e-07$	$1.33461373e-07$
BeF	$2.69829874e-08$	$1.50175519e-07$	$1.53876387e-07$
BO	$8.71624232e-08$	$2.73302338e-07$	$3.73791444e-07$
N_2^+	$7.77705073e-08$	$3.27831383e-07$	$3.96942882e-07$
N_2	$-2.66884648e-08$	$1.03767561e-07$	$1.10451066e-07$
CO^+	$9.67583873e-08$	$3.03668089e-07$	$3.91139291e-07$
CO	$1.40674927e-08$	$1.47084159e-07$	$1.65718055e-07$
BF	$8.06621960e-08$	$1.51589161e-07$	$1.56653663e-07$
AlH	$2.95436623e-09$	$5.77518147e-08$	$6.34182585e-08$

Table 3.3:
Computational efforts for the electronic structure benchmark problems (in seconds).

Problem	$\tilde{t}$	$\bar{t}$	$\underline{t}_1$	$\underline{t}_2$	$\underline{t}_2/\underline{t}_1$
LiH	184.30	215.01	87.12	4.74	$5.4408e-02$
BeH	207.63	243.27	88.59	5.08	$5.7343e-02$
BH^+	204.59	239.62	84.12	4.87	$5.7893e-02$
BH	181.54	218.13	85.01	5.08	$5.9758e-02$
CH^+	193.71	233.96	85.23	4.76	$5.5849e-02$
CH	197.69	232.01	89.16	4.94	$5.5406e-02$
CH^-	202.69	238.47	85.39	5.25	$6.1483e-02$
NH^+	194.10	238.90	87.48	5.10	$5.8299e-02$
NH	212.73	1597.55	373.85	5.09	$1.3615e-02$
NH^-	199.53	237.93	84.49	4.72	$5.5865e-02$
OH^+	204.93	251.74	92.20	5.02	$5.4447e-02$
OH	192.59	249.48	83.52	5.21	$6.2380e-02$
OH^-	201.25	18.09	84.24	4.99	$5.9236e-02$
HF^+	213.69	1595.88	366.64	5.24	$1.4292e-02$
HF	346.23	18.19	85.31	5.12	$6.0016e-02$
BH_2	1074.45	2372.17	536.77	24.57	$4.5774e-02$
$CH_2(1)$	915.89	1440.96	2007.14	29.21	$1.4553e-02$
$CH_2(3)$	1031.98	107.57	532.16	24.37	$4.5794e-02$
NH_2	1060.12	114.87	542.02	25.39	$4.6843e-02$
H_2O^+	1129.96	2668.34	2063.36	24.78	$1.2010e-02$
H_2O	1058.53	1537.46	549.85	24.89	$4.5267e-02$
NH_3	8871.36	10111.15	2715.03	102.48	$3.7745e-02$
H_3O^+	9359.03	11813.02	2622.24	114.75	$4.3760e-02$
O_2^+	151648.46	188182.83	210175.90	981.44	$4.6696e-03$
O_2	210506.71	250956.60	44206.35	1256.59	$2.8426e-02$
SiH	173374.23	216012.32	39694.06	1161.07	$2.9250e-02$
SiH^-	182323.98	1432016.55	35283.97	998.71	$2.8305e-02$

continued...

Problem	$\tilde{t}$	$\bar{t}$	$\underline{t}_1$	$\underline{t}_2$	$\underline{t}_2/\underline{t}_1$
NO^-	160852.13	200886.74	203252.95	1001.09	$4.9253e-03$
NF	213465.37	250389.31	38889.09	1199.99	$3.0857e-02$
HS^+	183277.95	220068.98	40911.00	1208.92	$2.9550e-02$
Li_2	184344.38	221733.11	42447.46	1032.78	$2.4331e-02$
B_2	177982.85	198597.99	688761.75	1013.14	$1.4710e-03$
C_2^+	175298.65	210272.63	261548.67	1133.67	$4.3345e-03$
C_2	166848.15	207500.37	421375.58	1115.64	$2.6476e-03$
C_2^-	157356.87	193556.12	437603.92	1019.05	$2.3287e-03$
LiF	204941.14	231529.60	45476.43	1451.44	$3.1916e-02$
BeO	170350.62	215724.36	246105.61	1092.81	$4.4404e-03$
NaH	180483.03	228871.70	41847.35	1441.32	$3.4442e-02$
BeF	189988.88	663130.02	43601.91	1329.82	$3.0499e-02$
BO	136933.08	182904.38	200500.61	999.61	$4.9856e-03$
N_2^+	133803.37	172144.76	33617.73	983.02	$2.9241e-02$
N_2	173932.10	210359.02	42531.46	1042.67	$2.4515e-02$
CO^+	132556.80	182132.18	360098.91	978.20	$2.7165e-03$
CO	168828.64	202342.14	36485.30	1035.51	$2.8382e-02$
BF	194003.70	237460.27	42328.54	1046.14	$2.4715e-02$
AlH	179670.64	230179.11	42028.82	1085.38	$2.5825e-02$

The approximate energies in the first column of the Table 3.1 reproduce, for the most part, the values found in [79]. Nevertheless, for some problems (mostly with basis sizes $r = 12$ and $r = 14$) the semidefinite solver (SDPA) delivered results a bit above those in Table 1 in [79]. The problem lies in SDPA convergence. The primal minimum is not attained due to the duality gap remaining relatively big (see the first column of the accuracy Table 3.2). Another indicator of the solver difficulty is the returned termination code. Instead of `pdOPT` expected in case of normal termination, `pFEAS` was returned, showing that the primal problem is feasible, but an optimal solution could not be found within the iteration cycle (see [17] for more details). The situation could probably be improved by optimizing the SDPA parameters or using another SDP solver. All our calculations were performed with the default parameter set.

In three problems, namely CH^+, OH^+ and OH, the calculated rigorous upper bound $\overline{E^*}$ appeared to be less then the approximation $\widetilde{E}^*$. This is normal and the upper bound is still valid and accurate. The reason for such behavior is that during an iteration of the Algorithm 2.1, SDPA found a better solution even though the admissible region is smaller for the perturbed problem. This could never be possible if the initial approximation was exactly the optimum point, but CH^+, OH^+ and OH belong to the group described in the previous paragraph, and the precision of $\tilde{E}^*$ is insufficient. The described phenomenon also becomes apparent in Table 3.2, where, for the above mentioned problems, we have $\mu(\overline{E^*}, \underline{E}_1^*) \leq \mu(\tilde{E}^*, \tilde{E}_d^*)$. That is the gap between the rigorous bounds becomes smaller than the gap between the approximate primal and dual optimal values. For three other problems, NH, HF^+ and SiH^- despite many iterations (note longer running times $\overline{t}$ in the Table 3.3), no verified upper bound could be found.

As can be seen in Table 3.2, some accuracies $\mu(\tilde{E}^*, \tilde{E}_d^*)$ are negative. This indicates a weak duality violation in the corresponding approximations. In such cases it is common to obtain a lower bound which will be greater than $\tilde{E}^*$. This, for example, happened for NH_3, Li_2, C_2^+ and LiF. Again, the lower bounds are accurate, and the corresponding approximate solutions $\tilde{E}^*$ should be discarded.

It can be observed, that $\underline{E}_1^*$ usually delivers a tighter lower bound compared to $\underline{E}_2^*$. Only for some problems like H_2O^+ and C_2 the fast bound $\underline{E}_2^*$ appeared to be better. The same is true in case of NO^-, but the value $\underline{E}_2^* = -5356.28$ with relative accuracy $\mu(\overline{E^*}, \underline{E}_1^*) = 1.89$ is a definite outlier and should not be considered. Nevertheless, no matter which value, $\underline{E}_1^*$ or $\underline{E}_2^*$, they both give a verified lower bound of the ground state energy.

The time win of using Algorithm 3.2 compared to the standard Algorithm 2.2 for computing a rigorous lower bound is significant. Depending on the necessity of extra iterations in Algorithm 2.2, the value of $\underline{t}_2/\underline{t}_1$ varies between $1.4710e-03$ and $6.2380e-02$. Owing to its non-iterative nature, the computational cost of applying Algorithm 3.2 remains relatively constant for different problems of the same size.

For the use of Algorithms 3.1 and 3.2, it would be interesting to evaluate the quality of the obtained eigenvalue bounds $\overline{\lambda}$ in (3.67). Though the numerical results in Tables 3.1 to 3.3 were obtained with the Algorithm 3.2, and upper eigenvalue bounds for G, Q, $T1$ and $T2$ are not used there, we were monitoring the maximal eigenvalues of all the blocks in the solution matrix. Table 3.4 shows the corresponding ratios of $\lambda_{max} \,/\, \overline{\lambda}$. We present our results as min-max intervals of all calculated ratios for each matrix. An ideal upper bound should have this interval narrow, and $\max \lambda_{max} \,/\, \overline{\lambda}$ should be approaching 1.

As you can see from the table, overestimation can be significant for all matrices except γ. Especially poor results were obtained for $T1$ and $T2$. There are several reasons for that. Firstly, we justified and used the eigenvalue bounds derived for full (not "compacted") matrices. Much better estimations could be received if we consid-

Table 3.4:
Ratios of maximal eigenvalues and upper bounds.

Matrix	min $\lambda_{max} / \overline{\lambda}$	max $\lambda_{max} / \overline{\lambda}$
γ	$9.819005e-01$	$1.000000e+00$
Γ	$6.252713e-02$	$2.415637e-01$
G	$6.348046e-02$	$3.315521e-01$
Q	$6.140103e-03$	$4.999994e-01$
$T1$	$2.957354e-04$	$3.879563e-03$
$T2$	$3.402340e-03$	$4.144608e-02$

ered the new matrix structure when calculating traces of "compacted" matrices. Since $T1$ and $T2$ undergo the biggest changes, their estimation results suffer the most. Secondly, different N-representability conditions influence each other. For example, for electronic structure problems without Q, $T1$ and $T2$ conditions, the following results were obtained over $100\,000$ randomly generated problems:

$$\begin{aligned} \max \frac{\lambda_{max}(\Gamma)}{N} &= 0.9936, \\ \max \frac{\lambda_{max}(G)}{N(r-N+1)} &= 0.9813. \end{aligned} \tag{3.112}$$

Adding the Q N-representability condition to the problem changes the situation for G considerably

$$\begin{aligned} \max \frac{\lambda_{max}(\Gamma)}{N} &= 0.9784, \\ \max \frac{\lambda_{max}(G)}{N(r-N+1)} &= 0.4918, \\ \max \frac{\lambda_{max}(Q)}{(r-N)(r-N-1)} &= 0.9992. \end{aligned} \tag{3.113}$$

Thus knowing the maximal eigenvalue bound of Q, we could possibly track it back and improve our bound for G. Using such dependencies between different N-representability conditions could help us to further improve our eigenvalue bounds. After all, trace is a fast to find, but not the most precise maximal eigenvalue bound of a positive semidefinite matrix.

The upper bounds of the residuals $\overline{r}$, present in Algorithm 3.2, were calculated as supremums of the actual residuals $|\langle A_i, \tilde{X}' \rangle - b_i|$ computed using interval arithmetic. For the benchmark problems in Table 3.1, these upper bounds lie between $8.018579e-11$ and $4.123265e-04$.

3.6 Discussion

The algorithm introduced in this chapter proved to be an accurate tool for determining a lower bound of the ground state energy. Nevertheless, a good approximate solution $(\gamma^*, \tilde{\Gamma}^*)$ is crucial for a tight lower bound. Does a corresponding semidefinite program always have a finite solution? The feasible region is evidently not empty. Any RDMs representing an N-electron wave function of type (3.5) will by definition give a feasible point (all N-representability requirements are necessary conditions). The dual variables y from (3.85) are bounded. This immediately follows from the relations

$$0 \preceq \gamma \preceq I \quad \text{and} \quad 0 \preceq \tilde{\Gamma} \preceq N \cdot I. \tag{3.114}$$

The optimization domain is therefore convex and compact. From the Weierstrass extreme value theorem we know, that the objective function attains its optimal value and is bounded. Since the conditions of the strong duality theorem (Theorem 2.1) are also fulfilled, the duality gap in the considered RDM problems is zero.

After obtaining an approximate solution point, the advisable step in computing a lower bound would be the Algorithm 3.2. Based on the results presented in the previous section, we can say that the proposed method proved to be a fast and reliable postprocessing instrument for the RDM problems. It does not only compute an accurate and reliable lower bound for the ground state energy, but also helps to verify the quality of the approximate solutions delivered by an SDP solver.

Though we have paid the most attention to RDM problems in “dual” SDP formulation in this work, the shown approach remains valid for other semidefinite implementations too. Given an approximate solution and eigenvalue bounds of the key structures derived in this chapter, it should be fairly easy to adapt the algorithms for a new formulation.

Chapter 4

Rigorous error bounds with verifiedSDP

In the context of our work a software package for rigorous verification in semidefinite programming verifiedSDP was developed. The program implements verification algorithms from Section 2.4 and feasibility tests described in Section 2.5. This chapter is devoted to a description of implementation details that can be interesting to the reader and go beyond brief formulations in Chapter 2. Numerical results for the benchmark problems from the SDPLIB [7] collection are also provided at the end.

To be able to calculate rigorous error bounds, an approximate solution of the considered semidefinite program is required. No further assumptions about the quality of this approximation are necessary. The SDPA [17] software package was extensively used to solve all emerging semidefinite optimization problems. The corresponding SDPA file format was also taken as standard for input data. verifiedSDP was tested to work with SDPA libraries up to version 6.2.1. All necessary interval functionality was provided by the PROFIL [43] library. PROFIL is based on BIAS (Basic Interval Arithmetic Subroutines) and supports all needed interval data structures and operations on them. Both full and sparse matrix formats are available. Other software packages used in verifiedSDP include linear algebra package LAPACK [3], sparse linear system solver UMFPACK [13], and sparse eigenvalue and eigenvector solver ARPACK++ [22].

Functionality similar to that of verifiedSDP can also be found in VSDP [31]. This MATLAB software package uses the INTLAB [69] toolbox and allows semidefinite programming problems to be solved rigorously. Please refer to corresponding papers for more details.

4.1 Implementation of algorithms computing rigorous results

Though the solution methods described in the next subsections are parts of an SDP verification process, they represent standalone problems of their own value. With maybe slight modifications, these approaches can find their application in diverse problems of various nature.

4.1.1 Calculating a lower eigenvalue bound of an interval matrix

Computation of verified eigenvalue bounds is an important component of many algorithms described in this thesis and has, without any doubt, crucial value for a successful SDP verification. In the following we present one of the possible approaches to this problem, implemented in verifiedSDP. The aim is to obtain a verified lower bound of the minimal eigenvalue of a symmetric interval matrix.

Let $\mathbf{A} \in \mathbb{IR}^{n\times n}$ be our matrix of interest and mid$\mathbf{A}$ and rad$\mathbf{A}$ will be its midpoint and radius matrices. Then for a lower bound $\underline{\lambda} \leq \lambda_{min}(\mathbf{A})$ the following inequality holds

$$\underline{\lambda} \geq \lambda_{min}(\text{mid}\mathbf{A}) - \rho(\text{rad}\mathbf{A}), \tag{4.1}$$

where $\rho(M)$ denotes the spectral radius of a matrix M. This result can be found, for example, in the work of Rohn [65] and in the book by Floudas [14]. For our calculation of a lower bound of the smallest eigenvalue of mid$\mathbf{A}$, we apply the algorithm proposed by Rump in [68]. There perturbation analysis based on the Weyl's theorem is used. First, an approximation t to the smallest eigenvalue of mid$\mathbf{A}$ is computed. Then we put

$$A_s = \text{mid}\mathbf{A} - sI, \tag{4.2}$$

where s is a somewhat reduced t (e.g. $s = 0.9t$ for positive t and $s = 1.1t$ for negative). Afterwards, we try to compute an approximate Cholesky decomposition $A_s \approx HH^T$. Should this not be possible (the matrix $(\text{mid}\mathbf{A} - sI)$ does not seem to be positive semidefinite), we have to further reduce s and try again. At the end we get a decomposition HH^T with all nonnegative eigenvalues. The Weyl's theorem states

$$|\lambda_i(A_s+E)-\lambda_i(A_s)| \leq \|E\|_2, \quad \text{where} \quad E = HH^T - A_s \quad \text{and} \quad 1 \leq i \leq n. \tag{4.3}$$

Hence,

$$0 \le \lambda_i(HH^T) = \lambda_i(A_s + E) \le \lambda_i(A_s) + \|E\|_2. \tag{4.4}$$

At the same time

$$\lambda_i(\text{mid}\mathbf{A} - sI) = \lambda_i(\text{mid}\mathbf{A}) - s. \tag{4.5}$$

Finally, combining (4.4) and (4.5) we get a lower bound to the smallest eigenvalue of mid$\mathbf{A}$:

$$\lambda_{min}(\text{mid}\mathbf{A}) \ge s - \|E\|_2. \tag{4.6}$$

To determine an upper bound of the spectral radius of rad$\mathbf{A}$, we use its properties as a nonnegative matrix and apply the Perron-Frobenius theorem. We know that spectral radius of such matrix is equal to its biggest eigenvalue: $\rho(\text{rad}\mathbf{A}) = r \ge 0$, and that there exists an eigenvector $x_r \ge 0$, such that rad$\mathbf{A}x_r = rx_r$. On the other hand, the inequality

$$\rho(\text{rad}\mathbf{A}) \le \max_{1\le i\le n} \frac{1}{x_i} \sum_{j=1}^{n} (\text{rad}\mathbf{A})_{ij} x_j \tag{4.7}$$

is valid for any positive vector $x \in \mathbb{R}^n$. Hence we compute an approximation $\tilde{x}$ of the eigenvector corresponding to the largest eigenvalue of rad$\mathbf{A}$ and put the upper bound of the spectral radius

$$\rho(\text{rad}\mathbf{A}) \le \overline{\rho} = \max_{1\le i\le n} \frac{1}{\tilde{x}_i} \sum_{j=1}^{n} (\text{rad}\mathbf{A})_{ij} \tilde{x}_j. \tag{4.8}$$

If an eigenvector estimation is not available, or the obtained one is not positive, an arbitrary positive vector (e.g. normed random or all-ones) can be used instead. This will impair the bound, but preserve its correctness. Finally, after substituting (4.6) and (4.8) into (4.1), we obtain the required lower bound to the minimal eigenvalue of the original interval matrix $\mathbf{A}$:

$$\lambda_{min}(\mathbf{A}) \ge s - \|E\|_2 - \overline{\rho}. \tag{4.9}$$

In the verifiedSDP implementation of this algorithm directed rounding is used throughout the calculations to guarantee a verified bound. This and general interval functionality are provided by PROFIL. Standard LAPACK routine `DSYEV` was used as an eigensolver for dense matrices, and methods of the `ARluSymStdEig` ARPACK++ class were used to deal with sparse eigenproblems. `DPOTRF` from LAPACK was used for an approximate Cholesky decomposition in both cases.

4.1.2 Solving linear systems rigorously

The Algorithm 2.2 in Section 2.4.2 describes a procedure of computing an upper bound for the optimal value of a semidefinite program. One of the key steps in this algorithm is the computation of an enclosure vector $\mathbf{x} \in \mathbb{IR}^N$ (step 1). There we have a rectangular interval matrix $\mathbf{A}^{\text{mat}} \in \mathbb{IR}^{m \times N}, m \leq N$, interval vector $\mathbf{b} \in \mathbb{IR}^m$ and vector $x = \text{svec}((X_j), 1) \in \mathbb{R}^n$, where $(X_j), j = 1, \ldots, n$ are blocks of an approximate primal optimal point. The desired interval vector $\mathbf{x}$ must possess the following property (2.50):

$$\forall\, b \in \mathbf{b},\ \forall\, A^{\text{mat}} \in \mathbf{A}^{\text{mat}}\ \exists\, x \in \mathbf{x}\ \text{ such that }\ A^{\text{mat}} x = b. \tag{4.10}$$

First, a nonsingular square submatrix has to be put together from the columns of $\mathbf{A}^{\text{mat}}$. As already mentioned in Section 2.4.2, an LU-decomposition of the midpoint matrix $(\text{mid}\,\mathbf{A}^{\text{mat}})^T$ does the job in our program. For dense matrices the standard `DGETRF` routine from LAPACK package is used. It computes a decomposition and returns a necessary pivoting vector. Sparse matrices of moderate size can also be treated as dense (in most cases the redundancy of memory usage is compensated by the speed of the dense algorithm). To be able to approach large and highly sparse matrices we, nevertheless, use routines of the UMFPACK package [13]. `umfpack_di_get_symbolic` in combination with `umfpack_di_symbolic` made it possible to verify problems for which dense algorithms would run out of memory. The choice of procedure for sparse matrices is implemented in verifiedSDP by the flag `SPARSE_LU` in `init.h`. `SPARSE_LU = 0` will initiate a call of `DGETRF`, whereas by setting `SPARSE_LU = 1` we trigger sparse routines of UMFPACK.

In case of success, the obtained pivoting vector is used to split the $\mathbf{A}^{\text{mat}}$ matrix. Hence we get a square interval linear system with a new right side in accordance with (2.51). Since no linear system solver for sparse interval matrices is available in PROFIL yet, a dense routine `ILSS` is used to solve the derived system regardless to the sparsity of the problem. Finally, we build the required enclosure by merging both parts of $\mathbf{x}$ and reversing the pivoting to restore the original order.

4.1.3 Checking SDP infeasibility

A few words should be said about the SDP infeasibility verification in verifiedSDP. Phase I methods, explicitly described in Appendix A, as well as the improving rays approach, discussed in Section 2.5, are implemented in our program. If a task is given to check either primal or dual infeasibility of a semidefinite program, and no approximate solution (an improving ray candidate) for this SDP is available, the corresponding phase I algorithm will be chosen. If, on the other hand, the infeasibility suspicion arises after we have already tried to solve the SDP, the obtained approximation will first be checked for improving ray properties in accordance with the algorithms in Section 2.5. Only if no infeasibility certificate can be issued this way, the phase I methods will be applied. This strategy can help us avoid unnecessary computational effort on solving an extra SDP and reduce verification time.

4.2 Numerical results for the SDPLIB

The SDPLIB library [7] is a collection of semidefinite programming test problems. In this section we present some numerical results obtained with verifiedSDP. Table 4.1 contains approximations of the primal and dual optimal values $\tilde{p}^*$ and $\tilde{d}^*$ computed using SDPA, as well as the corresponding rigorous bounds $\overline{p}^*$ and $\underline{p}^*$. The bounds are expected to be infinite if the problem is infeasible or very ill-conditioned.

verifiedSDP could handle 91 out of 92 SDPLIB problems with the only exception of `maxG60`. In that case we had to capitulate because of the problem's very large size and, accordingly, solver's extremely high memory demand. The largest problems which could be solved are `maxG55` with its $m = 5000$ constraints and 12497500 primal variables, and `thetaG51` with $m = 6910$ constraints and 500500 primal variables.

Table 4.1:
Calculated approximate optimal values and rigorous bounds for the SDPLIB problems.

Problem	$\tilde{p}^*$	$\tilde{d}^*$	$\overline{p}^*$	$\underline{p}^*$
arch0	$-5.665172e-01$	$-5.665173e-01$	$-5.665170e-01$	$-5.665173e-01$
arch2	$-6.715153e-01$	$-6.715154e-01$	$-6.715153e-01$	$-6.715154e-01$
arch4	$-9.726273e-01$	$-9.726274e-01$	$-9.726272e-01$	$-9.726274e-01$
arch8	$-7.056975e+00$	$-7.056982e+00$	$-7.056969e+00$	$-7.056982e+00$
control1	$-1.778463e+01$	$-1.778463e+01$	$-1.778448e+01$	$-1.778463e+01$
control2	$-8.299999e+00$	$-8.300000e+00$	$-8.296941e+00$	$-8.300000e+00$
control3	$-1.363326e+01$	$-1.363327e+01$	$-1.341884e+01$	$-1.363327e+01$
control4	$-1.979423e+01$	$-1.979423e+01$	$-1.974714e+01$	$-1.979423e+01$
control5	$-1.688357e+01$	$-1.688361e+01$	$-1.678034e+01$	$-1.688361e+01$
control6	$-3.730434e+01$	$-3.730443e+01$	$-3.632569e+01$	$-3.730443e+01$
control7	$-2.062506e+01$	$-2.062508e+01$	$-1.975616e+01$	$-2.062508e+01$

continued...

Problem	$\tilde{p}^*$	$\tilde{d}^*$	$\overline{p}^*$	$\underline{p}^*$
control8	$-2.028633e+01$	$-2.028637e+01$	$-1.989140e+01$	$-2.028637e+01$
control9	$-1.467542e+01$	$-1.467543e+01$	$-1.432014e+01$	$-1.467543e+01$
control10	$-3.853284e+01$	$-3.853308e+01$	$-3.352668e+01$	$-3.853308e+01$
control11	$-3.195842e+01$	$-3.195871e+01$	$-2.915175e+01$	$-3.195871e+01$
equalG11	$-6.291546e+02$	$-6.291553e+02$	$-6.047805e+02$	$-6.291553e+02$
equalG51	$-4.005599e+03$	$-4.005601e+03$	$-3.527486e+03$	$-4.005601e+03$
gpp100	$4.494365e+01$	$4.494355e+01$	∞	$4.494355e+01$
gpp124-1	$7.344950e+00$	$7.343064e+00$	∞	$7.343064e+00$
gpp124-2	$4.686272e+01$	$4.686229e+01$	∞	$4.686229e+01$
gpp124-3	$1.530154e+02$	$1.530141e+02$	∞	$1.530141e+02$
gpp124-4	$4.189863e+02$	$4.189876e+02$	∞	$4.189876e+02$
gpp250-1	$1.544627e+01$	$1.544491e+01$	∞	$1.544491e+01$
gpp250-2	$8.187348e+01$	$8.186893e+01$	∞	$8.186893e+01$
gpp250-3	$3.035431e+02$	$3.035393e+02$	∞	$3.035393e+02$
gpp250-4	$7.473290e+02$	$7.473283e+02$	∞	$7.473283e+02$
gpp500-1	$2.532244e+01$	$2.532053e+01$	∞	$2.532053e+01$
gpp500-2	$1.560607e+02$	$1.560604e+02$	∞	$1.560604e+02$
gpp500-3	$5.130260e+02$	$5.130176e+02$	∞	$5.130176e+02$
gpp500-4	$1.567025e+03$	$1.567019e+03$	∞	$1.567019e+03$
hinf1	$-2.032623e+00$	$-2.032611e+00$	∞	$-2.032611e+00$
hinf2	$-1.096700e+01$	$-1.096709e+01$	$-1.447394e+00$	$-1.096709e+01$
hinf3	$-5.694556e+01$	$-5.694527e+01$	∞	$-5.694527e+01$
hinf4	$-2.747641e+02$	$-2.747641e+02$	∞	$-2.747641e+02$
hinf5	$-3.627368e+02$	$-3.626485e+02$	∞	$-3.626485e+02$
hinf6	$-4.489816e+02$	$-4.489603e+02$	∞	$-4.489603e+02$
hinf7	$-3.899108e+02$	$-3.912731e+02$	∞	$-3.912731e+02$
hinf8	$-1.161089e+02$	$-1.161626e+02$	∞	$-1.161626e+02$
hinf9	$-2.362492e+02$	$-2.362493e+02$	$-2.362492e+02$	$-2.362493e+02$
hinf10	$-1.088896e+02$	$-1.088007e+02$	∞	$-1.088007e+02$
hinf11	$-6.592865e+01$	$-6.589497e+01$	∞	$-6.589497e+01$
hinf12	$-4.535424e-01$	$-2.267973e-01$	∞	$-2.267973e-01$
hinf13	$-4.880138e+01$	$-4.728583e+01$	∞	$-4.728583e+01$
hinf14	$-1.298103e+01$	$-1.299714e+01$	∞	$-1.299714e+01$
hinf15	$-2.761076e+01$	$-2.662883e+01$	∞	$-2.662883e+01$
infd1	$-7.706246e+01$	$1.200141e+06$	∞	$1.200141e+06$
infd2	$6.334026e+00$	$2.806876e+06$	∞	$2.806876e+06$
infp1	$-8.190125e+07$	$-7.708220e+00$	$-8.190125e+07$	$-\infty$
infp2	$-1.337344e+08$	$-6.894643e+00$	$-1.337344e+08$	$-\infty$
maxG11	$-6.291648e+02$	$-6.291648e+02$	$-6.291648e+02$	$-6.291648e+02$
maxG32	$-1.567640e+03$	$-1.567640e+03$	$-1.567640e+03$	$-1.567640e+03$
maxG51	$-4.006255e+03$	$-4.006256e+03$	$-4.006255e+03$	$-4.006256e+03$

continued…

Problem	$\tilde{p}^*$	$\tilde{d}^*$	$\overline{p}^*$	$\underline{p}^*$
maxG55	$-1.286987e+04$	$-1.286987e+04$	$-1.286987e+04$	$-1.286987e+04$
mcp100	$-2.261573e+02$	$-2.261574e+02$	$-2.261573e+02$	$-2.261574e+02$
mcp124-1	$-1.419905e+02$	$-1.419905e+02$	$-1.419905e+02$	$-1.419905e+02$
mcp124-2	$-2.698802e+02$	$-2.698802e+02$	$-2.698802e+02$	$-2.698802e+02$
mcp124-3	$-4.677501e+02$	$-4.677501e+02$	$-4.677501e+02$	$-4.677501e+02$
mcp124-4	$-8.644118e+02$	$-8.644119e+02$	$-8.644118e+02$	$-8.644119e+02$
mcp250-1	$-3.172643e+02$	$-3.172643e+02$	$-3.172643e+02$	$-3.172643e+02$
mcp250-2	$-5.319300e+02$	$-5.319301e+02$	$-5.319300e+02$	$-5.319301e+02$
mcp250-3	$-9.811725e+02$	$-9.811726e+02$	$-9.811725e+02$	$-9.811726e+02$
mcp250-4	$-1.681960e+03$	$-1.681960e+03$	$-1.681960e+03$	$-1.681960e+03$
mcp500-1	$-5.981485e+02$	$-5.981485e+02$	$-5.981485e+02$	$-5.981485e+02$
mcp500-2	$-1.070057e+03$	$-1.070057e+03$	$-1.070057e+03$	$-1.070057e+03$
mcp500-3	$-1.847970e+03$	$-1.847970e+03$	$-1.847970e+03$	$-1.847970e+03$
mcp500-4	$-3.566738e+03$	$-3.566738e+03$	$-3.566738e+03$	$-3.566738e+03$
qap5	$4.360624e+02$	$4.359993e+02$	∞	$4.359993e+02$
qap6	$3.814358e+02$	$3.814370e+02$	∞	$3.814370e+02$
qap7	$4.248868e+02$	$4.247967e+02$	∞	$4.247967e+02$
qap8	$7.570976e+02$	$7.569443e+02$	∞	$7.569443e+02$
qap9	$1.410377e+03$	$1.409895e+03$	∞	$1.409895e+03$
qap10	$1.093368e+03$	$1.092526e+03$	∞	$1.092526e+03$
qpG11	$-2.448659e+03$	$-2.448659e+03$	$-2.448659e+03$	$-2.448659e+03$
qpG51	$-1.181800e+04$	$-1.181800e+04$	$-1.181800e+04$	$-1.181800e+04$
ss30	$-2.023950e+01$	$-2.023951e+01$	$-2.023948e+01$	$-2.023951e+01$
theta1	$-2.300000e+01$	$-2.300000e+01$	$-2.300000e+01$	$-2.300000e+01$
theta2	$-3.287917e+01$	$-3.287917e+01$	$-3.287917e+01$	$-3.287917e+01$
theta3	$-4.216698e+01$	$-4.216698e+01$	$-4.216698e+01$	$-4.216698e+01$
theta4	$-5.032122e+01$	$-5.032122e+01$	$-5.032122e+01$	$-5.032122e+01$
theta5	$-5.723231e+01$	$-5.723231e+01$	$-5.723231e+01$	$-5.723231e+01$
theta6	$-6.347709e+01$	$-6.347709e+01$	$-6.347709e+01$	$-6.347709e+01$
thetaG11	$-4.000000e+02$	$-4.000000e+02$	$-4.000000e+02$	$-4.000000e+02$
thetaG51	$-3.489997e+02$	$-3.490000e+02$	$-3.489882e+02$	$-3.490000e+02$
truss1	$8.999997e+00$	$8.999996e+00$	$8.999997e+00$	$8.999996e+00$
truss2	$1.233804e+02$	$1.233804e+02$	$1.233804e+02$	$1.233804e+02$
truss3	$9.109996e+00$	$9.109996e+00$	$9.109996e+00$	$9.109996e+00$
truss4	$9.009997e+00$	$9.009996e+00$	$9.009997e+00$	$9.009996e+00$
truss5	$1.326357e+02$	$1.326357e+02$	$1.326357e+02$	$1.326357e+02$
truss6	$9.010013e+02$	$9.010014e+02$	$9.010551e+02$	$9.010014e+02$
truss7	$9.000014e+02$	$9.000014e+02$	$9.000030e+02$	$9.000014e+02$
truss8	$1.331146e+02$	$1.331146e+02$	$1.331146e+02$	$1.331146e+02$

As can be seen from the table above, a rigorous lower bound $\underline{p}^*$ could be found for all problems except `infp1` and `infp2` which are dual infeasible. Thus verifiedSDP could compute for all 85 problems discussed in [15] a rigorous lower bound of the optimal value and verify the existence of strictly dual feasible solutions which implies a zero duality gap. A finite rigorous upper bound could be computed for all well-posed problems. For the 32 ill-posed problems (those from [15]) and two primal infeasible problems `infd1` and `infd2` verifiedSDP has computed $\overline{p}^* = +\infty$, which reflects that the distance to the next primal infeasible problem is zero as well as the infinite condition number.

The following Table 4.2 contains the approximate duality gap $\mu(\tilde{p}^*, \tilde{d}^*)$, the rigorous error $\mu(\overline{p}^*, \underline{p}^*)$, as well as computation times $\tilde{t}$, $\overline{t}$ and $\underline{t}$ for calculating an SDPA approximation and determining upper and lower rigorous bounds respectively. We measure the accuracy by the quantity

$$\mu(a,b) := \frac{a-b}{\max\{1.0, (|a|+|b|)/2\}}. \tag{4.11}$$

Notice that we do not use the absolute value of $a - b$. Hence, a negative sign implies that $a \leq b$. We have set $\mu(a, b) = \text{NaN}$ if one of the arguments is infinite.

Table 4.2:
Accuracy and computational effort (in seconds) for the SDPLIB problems.

Problem	$\mu(\tilde{p}^*, \tilde{d}^*)$	$\mu(\overline{p}^*, \underline{p}^*)$	$\tilde{t}$	$\overline{t}$	$\underline{t}$
arch0	$1.411e-07$	$2.564e-07$	4.32	6.94	0.50
arch2	$8.859e-08$	$9.344e-08$	4.48	7.10	0.50
arch4	$1.495e-07$	$2.097e-07$	4.80	7.62	0.50
arch8	$9.368e-07$	$1.754e-06$	4.66	7.48	0.50
control1	$7.138e-08$	$8.013e-06$	0.02	0.01	0.00
control2	$1.461e-07$	$3.686e-04$	0.11	0.12	0.01
control3	$4.470e-07$	$1.585e-02$	0.53	0.67	0.00
control4	$1.196e-07$	$2.382e-03$	1.73	2.46	0.01
control5	$1.997e-06$	$6.135e-03$	5.02	8.76	0.02
control6	$2.387e-06$	$2.659e-02$	14.68	26.26	0.05
control7	$1.051e-06$	$4.304e-02$	35.48	73.29	0.07
control8	$1.885e-06$	$1.966e-02$	67.95	150.99	0.12
control9	$7.376e-07$	$2.451e-02$	127.51	290.59	0.20
control10	$6.194e-06$	$1.390e-01$	200.99	539.09	0.29
control11	$9.082e-06$	$9.187e-02$	158.96	681.51	0.27
equalG11	$1.117e-06$	$3.951e-02$	290.45	969.05	1276.94
equalG51	$6.712e-07$	$1.269e-01$	731.19	4538.63	3024.85

continued...

Problem	$\mu(\tilde{p}^*, \tilde{d}^*)$	$\mu(\overline{p}^*, \underline{p}^*)$	$\tilde{t}$	$\bar{t}$	$\underline{t}$
gpp100	$2.254e-06$	NaN	0.32	3.62	0.19
gpp124-1	$2.568e-04$	NaN	0.64	13.86	0.44
gpp124-2	$9.050e-06$	NaN	0.72	6.99	0.43
gpp124-3	$8.233e-06$	NaN	0.60	6.84	0.44
gpp124-4	$-3.196e-06$	NaN	0.81	5.96	0.43
gpp250-1	$8.795e-05$	NaN	5.39	57.63	7.01
gpp250-2	$5.552e-05$	NaN	4.75	68.38	6.98
gpp250-3	$1.255e-05$	NaN	5.49	61.54	7.01
gpp250-4	$9.352e-07$	NaN	5.38	62.07	7.32
gpp500-1	$7.536e-05$	NaN	35.31	457.66	68.18
gpp500-2	$2.133e-06$	NaN	33.52	454.68	66.96
gpp500-3	$1.648e-05$	NaN	26.66	395.01	58.97
gpp500-4	$4.005e-06$	NaN	58.31	690.14	192.69
hinf1	$-6.054e-06$	NaN	0.00	0.05	0.00
hinf2	$8.622e-06$	$1.534e+00$	0.00	0.06	0.00
hinf3	$-5.034e-06$	NaN	0.00	0.03	0.00
hinf4	$-5.516e-08$	NaN	0.00	0.04	0.00
hinf5	$-2.435e-04$	NaN	0.00	0.04	0.00
hinf6	$-4.752e-05$	NaN	0.01	0.05	0.00
hinf7	$3.488e-03$	NaN	0.00	0.04	0.00
hinf8	$4.623e-04$	NaN	0.00	0.03	0.00
hinf9	$1.147e-07$	$1.147e-07$	0.00	0.01	0.00
hinf10	$-8.164e-04$	NaN	0.01	0.05	0.00
hinf11	$-5.109e-04$	NaN	0.03	0.08	0.01
hinf12	$-2.267e-01$	NaN	0.05	0.16	0.00
hinf13	$-3.155e-02$	NaN	0.02	0.14	0.00
hinf14	$1.241e-03$	NaN	0.03	0.28	0.00
hinf15	$-3.621e-02$	NaN	0.05	0.30	0.00
infd1	$-2.000e+00$	NaN	0.03	0.29	0.00
infd2	$-2.000e+00$	NaN	0.03	0.23	0.00
infp1	$-2.000e+00$	NaN	0.03	0.02	0.25
infp2	$-2.000e+00$	NaN	0.02	0.02	0.28
maxG11	$3.471e-08$	$3.471e-08$	73.39	217.11	21.49
maxG32	$1.967e-08$	$1.967e-08$	1124.21	5001.41	766.84
maxG51	$4.228e-08$	$4.228e-08$	131.34	303.47	1280.89
maxG55	$2.461e-08$	$2.461e-08$	16138.81	231628.62	267432.42
mcp100	$1.745e-08$	$1.745e-08$	0.20	0.13	0.06
mcp124-1	$1.137e-08$	$1.137e-08$	0.40	0.23	0.07
mcp124-2	$3.174e-08$	$3.174e-08$	0.38	0.24	0.13
mcp124-3	$2.803e-08$	$2.803e-08$	0.38	0.25	0.19
mcp124-4	$2.156e-08$	$2.156e-08$	0.37	0.25	0.24

continued. . .

Problem	$\mu(\tilde{p}^*, \tilde{d}^*)$	$\mu(\overline{p}^*, \underline{p}^*)$	$\tilde{t}$	$\overline{t}$	$\underline{t}$
mcp250-1	$6.006e-08$	$6.006e-08$	2.44	2.36	0.68
mcp250-2	$8.401e-08$	$8.401e-08$	2.25	2.36	1.38
mcp250-3	$4.582e-08$	$4.582e-08$	2.29	2.36	2.34
mcp250-4	$5.549e-08$	$5.549e-08$	2.27	2.38	3.39
mcp500-1	$1.583e-08$	$1.583e-08$	18.92	29.15	6.33
mcp500-2	$1.840e-08$	$1.840e-08$	18.02	29.44	18.07
mcp500-3	$2.242e-08$	$2.242e-08$	17.16	29.37	34.40
mcp500-4	$1.158e-08$	$1.158e-08$	17.01	29.54	53.56
qap5	$1.449e-04$	NaN	0.07	0.54	0.00
qap6	$-3.152e-06$	NaN	0.18	1.82	0.00
qap7	$2.121e-04$	NaN	0.42	8.66	0.02
qap8	$2.025e-04$	NaN	1.04	29.04	0.05
qap9	$3.424e-04$	NaN	2.21	80.88	0.11
qap10	$7.703e-04$	NaN	4.74	223.85	0.25
qpG11	$3.399e-08$	$3.399e-08$	569.47	2230.77	311.03
qpG51	$3.701e-08$	$3.701e-08$	1146.61	6938.11	10320.09
ss30	$7.125e-07$	$1.395e-06$	21.44	24.34	32.31
theta1	$5.411e-08$	$5.411e-08$	0.06	0.04	0.01
theta2	$1.475e-08$	$1.475e-08$	1.07	2.63	0.21
theta3	$1.581e-08$	$1.581e-08$	6.22	27.88	1.10
theta4	$2.270e-08$	$2.270e-08$	26.97	160.21	4.12
theta5	$2.479e-08$	$2.479e-08$	89.27	759.76	10.09
theta6	$2.483e-08$	$2.483e-08$	257.09	2374.57	22.60
thetaG11	$2.299e-08$	$2.299e-08$	76.38	250.55	17.15
thetaG51	$9.443e-07$	$3.387e-05$	1889.96	48907.86	1600.74
truss1	$2.785e-08$	$2.785e-08$	0.01	0.00	0.00
truss2	$3.714e-08$	$3.714e-08$	0.06	0.01	0.00
truss3	$4.141e-08$	$4.141e-08$	0.01	0.00	0.00
truss4	$5.485e-08$	$5.485e-08$	0.00	0.00	0.01
truss5	$3.348e-08$	$3.348e-08$	0.58	0.16	0.02
truss6	$-3.798e-08$	$5.959e-05$	0.48	2.83	0.04
truss7	$-3.428e-08$	$1.769e-06$	0.19	0.81	0.23
truss8	$1.868e-08$	$3.034e-07$	4.59	2.95	4.71

One of the first observations made upon this table would be the negative signs of $\mu(\tilde{p}^*, \tilde{d}^*)$ showing that the approximations do not satisfy weak duality. Therefore SDPA is not backward stable; i.e., $\tilde{p}^*$ and $\tilde{d}^*$ are not exact solutions of a slightly perturbed problem. This demonstrates that the measures for termination and accepting an approximation are not appropriate for ill-conditioned or ill-posed problems. The rigorous bounds $\overline{p}^*$ and $\underline{p}^*$ recognize difficult problems much better and overestimate the optimal value only slightly, and this overestimation depends on the quality of the computed approximations.

Looking at computation times of those problems for which no rigorous upper bound could be found ($\overline{p}^* = \infty$), one can notice that $\overline{t}/t$ ratio for them is much higher than that of problems with finite upper bounds. The reason lays in the big number of futile iterations of the algorithm 2.2 before it reaches $l_{\max}$. Otherwise the time overhead of computing rigorous results is fair compared to the time required to find an approximation.

Chapter 5

Conclusions

Finding rigorous error bounds for semidefinite optimization problems is a difficult computational problem, important for many real life applications. This thesis contributes to semidefinite programming verification by presenting algorithms for computing lower and upper bounds of its global optimal value, as well as enclosures of ε-optimal solutions. In Chapter 2 we have introduced a detailed investigation of the problem. Suggested verification methods for computation of rigorous error bounds can be viewed as a carefully postprocessing tool that uses only approximate solutions computed by an SDP solver. The numerical results show that such rigorous error bounds can be computed even for problems of large size. Moreover, we have shown how these verification algorithms can be used to check the possibility of issuing a certificate of infeasibility and thus to prove the existence of optimal solutions.

A key achievement of this thesis is the successful application of the suggested verification methods to the electronic structure problem. The proposed approach was thoroughly described in Chapter 3. There we took the corresponding atomic-molecular problem in reduced density matrix formulation (in its SDP representation). Then a rigorous lower bound of the resulting semidefinite program, and thus a lower bound of the ground state energy of the considered N-electron system, was computed.

To improve the algorithm's accuracy and performance, we have reformulated the respective semidefinite problem and estimated spectral properties of the involved operators. This gave us a considerable increase in performance and made the rigorous evaluation of large-scale problems possible in reasonable time.

In addition, we have developed a C++ software package which implements our verification algorithms and allows, in combination with an approximate semidefinite solver, to calculate rigorous optimal value bounds and to check feasibility of semidefinite programs. This package, called verifiedSDP, permits interval input and rigorously controls rounding errors. All numerical results in this thesis were obtained with verifiedSDP.

Based on our experience we can say, that at least for the problems considered in this study, our rigorous bounds reflect the problem's difficulty much better than warnings and termination codes of solvers. They provide safety, especially in the case where algorithms subsequently call other algorithms, as is done for example in branch-and-bound methods. The observed results show a strong correlation between the rigorous bounds and the difficulty of the problem. Moreover, since approximations often violate weak duality, our verification algorithms provide valuable means for optimal value assessment.

Appendix A

Phase I methods in semidefinite programming

For many semidefinite programming solvers a feasible starting point is necessary. If such a point is not known, a preliminary stage, called *phase I*, is used to find a feasible point or to prove the infeasibility of the problem. The feasible point found during phase I is then used as the starting point for the optimization algorithm (e.g. interior-point method), which is called the *phase II* stage [9].

We consider semidefinite programming problems in the following primal and dual block diagonal forms:

$$p^* := \min \sum_{j=1}^{n} \langle C_j, X_j \rangle \quad \text{s.t.} \quad \begin{array}{l} \sum_{j=1}^{n} \langle A_{ij}, X_j \rangle = b_i \quad \text{for } i = 1, \dots, m, \\ X_j \succeq 0 \quad \text{for } j = 1, \dots, n; \end{array} \tag{A.1}$$

$$d^* := \max b^T y \quad \text{s.t.} \quad \sum_{i=1}^{m} y_i A_{ij} \preceq C_j \quad \text{for } j = 1, \dots, n, \tag{A.2}$$

where $C_j, A_{ij}, X_j \in \mathbb{S}^{s_j}$ and $b, y \in \mathbb{R}^m$.

A.1 Dual feasibility

We start our description of the method with the dual SDP (A.2), since the implementation is more straightforward for checking feasibility of LMIs. The general idea is to modify the original problem to make it evidently feasible and to organize an optimization process to bound the maximum infeasibility of the original SDP. To check the feasibility of (A.2), we introduce a complementary variable y_{m+1}, put $b = (0, \dots, 0, 1)^T$, and set $A_{(m+1)j} = I_j,\ j = 1,\dots,n$, where I_j are identity matrices of appropriate size. The phase I dual problem then becomes

$$d^* := \max y_{m+1} \quad \text{s.t. } C_j - \sum_{i=1}^{m} y_i A_{ij} \succeq y_{m+1} I_j \quad \text{for } j = 1,\dots,n. \tag{A.3}$$

Indeed, there will always exist a y_{m+1} such that the inequality above holds for any values of $y_1,\dots,y_m$ and the optimization problem (A.3) is thus always feasible. Depending on the optimal value d^*, different cases can be distinguished.

1. $d^* > 0$ $(y_{m+1} > 0)$: the LMIs in (A.2) are strictly feasible.

2. $d^* < 0$ $(y_{m+1} < 0)$: the LMIs in (A.2) are infeasible.

3. $d^* = 0$ $(y_{m+1} = 0)$: the feasibility problem for LMIs in (A.2) is ill-posed. Arbitrary small perturbations yield infeasibility.

Since any $d^* > 0$ proves feasibility, an artificial bound $y_{m+1} \le \overline{y}$, added to the problem (A.3), can help an SDP solver to find the optimum point. We recommend a simple bound of the type $\overline{y} \approx \alpha \cdot \|C\|$ with $\alpha \ge 10^3$. The necessary modifications of the feasibility problem are minimal. One extra block of size 1 has to be added, making the total number of blocks $j + 1$.

$$\begin{aligned} &A_{i(n+1)} = 0, \text{ for } i = 1,\dots,m,\\ &A_{(m+1)(n+1)} = 1,\\ &C_{n+1} = \overline{y}. \end{aligned} \tag{A.4}$$

As we have seen, the dual SDP (A.3) is always feasible. This fact does not, nevertheless, mean, that the corresponding primal problem is feasible too. In fact, the primal infeasibility of the phase I problem (A.3) is not a rare phenomenon at all, what makes the use of primal-dual interior-point methods in many cases inefficient. We will illustrate such behavior by a simple example.

Example A.1. *Let us take the problem discussed in example 2.1. There $m = 1$, $n = 1$ and*

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} 2 & 0.5 \\ 0.5 & 1 \end{pmatrix}, \quad b = \{1\}. \tag{A.5}$$

The primal variable

$$X := \begin{pmatrix} x_1 & x_2 \\ x_2 & x_3 \end{pmatrix} \tag{A.6}$$

yields the primal feasibility constraints

$$\begin{aligned} &2x_1 + x_2 + x_3 = 1, \\ &x_1 \geq 0, \\ &x_1x_3 - {x_2}^2 \geq 0. \end{aligned} \tag{A.7}$$

This set is not empty and contains strictly feasible points (e.g. $x_1 = 0.5$*,* $x_2 = -0.25$ *and* $x_3 = 0.25$*). This means, that the initial SDP is primal feasible.*

Let us now switch to the dual phase I problem of this SDP. The primal feasibility constraints then become

$$\begin{aligned} &\langle A, X\rangle = 0, \\ &\langle I, X\rangle = 1, \\ &X \succeq 0. \end{aligned} \tag{A.8}$$

From the first and the third conditions above we obtain

$$\begin{aligned} &x_3 = -2x_1 - x_2, \\ &x_1 \geq 0, \\ &-{x_2}^2 - x_1x_2 - 2{x_1}^2 \geq 0. \end{aligned} \tag{A.9}$$

This system has only one trivial solution $x_1 = x_2 = x_3 = 0$ *(follows from the last inequality), which is not compatible with the second equation in (A.8) (*$x_1 + x_3 = 1$*) and the phase I problem for the SDP (A.5) is thus proven to be primal infeasible.*

The purpose of this simple example was to show, that potential primal infeasibility of phase I problems should not be neglected, and that special care is needed when using primal-dual interior-point methods to solve them.

A.2 Primal feasibility

Phase I feasibility check of a primal SDP (A.1) is similar, but less evident. It is useful to begin with a formulation of the target feasibility problem.

$$\begin{aligned} p^* := \min -s \quad \text{s.t.} \quad &\sum_{j=1}^{n} \langle A_{ij}, X_j\rangle = b_i \quad \text{for } i = 1, \ldots, m, \\ &X_j \succeq sI_j \quad \text{for } j = 1, \ldots, n. \end{aligned} \tag{A.10}$$

This semidefinite program is again always feasible, and the optimization aim is to bring minimum feasibility of the original problem above zero. In the following we will show modifications of the original problem necessary to represent it in the form (A.10). First a change of variables $\hat{X}_j = X_j - sI_j$ is necessary. This yields

$$\sum_{j=1}^{n}\langle A_{ij}, X_j\rangle = \sum_{j=1}^{n}\langle A_{ij}, \hat{X}_j\rangle + s\sum_{j=1}^{n}\langle A_{ij}, I_j\rangle = \sum_{j=1}^{n}\langle A_{ij}, \hat{X}_j\rangle + s\cdot g_i = b_i, \quad \text{(A.11)}$$

where $g_i = \text{trace}(A_i) \in \mathbb{R}$. Nevertheless, simple inclusion of s as an extra block in the variable matrix is not acceptable. This will impose a constraint $s \geq 0$ which is inconsistent with the idea of phase I methods. To overcome this limitation, we introduce a decomposition $s = s^+ - s^-$. Having both s^+ and s^- positive, we avoid restrictions on s. The equation (A.11) then becomes

$$\sum_{j=1}^{n}\langle A_{ij}, \hat{X}_j\rangle + s^+\cdot g_i - s^-\cdot g_i = b_i, \quad \text{(A.12)}$$

and the new variable is the block diagonal matrix $X' = (\hat{X}_1; \ldots; \hat{X}_n; s^+; s^-)$. The corresponding constraint matrices become $A'_i = (A_{i1}; \ldots; A_{in}; g_i; -g_i)$, with $g_i = \sum_{j=1}^{n}\langle A_{ij}, I_j\rangle = \text{trace}(A_i)$. To obtain the desired objective function from (A.10), we put $C' = (0; \ldots; 0; -1; 1)$. The resulting semidefinite program

$$\begin{aligned} p^* := \min \sum_{j=1}^{n+2}\langle C'_j, X'_j\rangle \quad \text{s.t.} \quad & \sum_{j=1}^{n+2}\langle A'_{ij}, X'_j\rangle = b_i \quad \text{for } i = 1, \ldots, m, \\ & X'_j \succeq 0 \quad \text{for } j = 1, \ldots, n+2 \end{aligned} \quad \text{(A.13)}$$

is then identical to (A.10) and represents the phase I feasibility problem for a primal SDP (A.1). Depending on the optimal value of (A.13), three cases can again be segregated.

1. $p^* < 0$ $(s > 0)$: the semidefinite program (A.1) is strictly feasible.
2. $p^* > 0$ $(s < 0)$: the semidefinite program (A.1) is infeasible.
3. $p^* = 0$ $(s = 0)$: the feasibility problem for (A.1) is ill-posed. Arbitrary small perturbations yield infeasibility.

Of course, if the initial problem has design drawbacks and equations $\sum_{j=1}^{n}\langle A_{ij}, X_j\rangle = b_i$ in (A.1) represent incompatible constraints, both (A.1) and (A.13) will be infeasible. Otherwise, (A.13) will always have a solution. Furthermore, a situation similar to that in the Example A.1 is possible, when initially feasible dual SDP will become infeasible after the described phase I reformulation.

For both, primal and dual semidefinite programs, including problems with uncertain input, the bounding algorithms from Chapter 2 can be used for a verified infeasibility check. By calculating a rigorous upper or lower bound of the dual optimal value d^* in (A.3) or of the primal optimal value p^* in (A.13) respectively, one can either prove strict feasibility, alternatively infeasibility, of the corresponding problem, or illustrate the uncertainty of the case.

Appendix B

verifiedSDP quick reference

B.1 Installation

To be able to compile verifiedSDP, the following software has to be installed beforehand:

- SDPA (SemiDefinite Programming Algorithm) [17] is used as a semidefinite solver.

- LAPACK(CLAPACK) [3] and ATLAS (Automatically Tuned Linear Algebra Software) [75] provide verifiedSDP and SDPA with necessary linear algebra functionality.

- PROFIL/BIAS [43] interval libraries permit interval input data and contribute to verified calculations.

- UMFPACK [13] (optional) is necessary to enable sparse LU decompositions. Though it is possible to perform this operations with LAPACK routines, for large and highly sparse input data, using sparse methods can bring a considerable performance benefit and extend the scope of solvable problems.

- ARPACK/ARPACK++ [22] (optional) can be used for sparse eigenvalue calculations as an alternative to dense LAPACK functions.

The files delivered with the verifiedSDPpackage include:

Files		Description
verSDP.h, verSDP.cpp	—	Declaration and implementation of the verSDP class. Contains the verification algorithms and input/output methods.
verSDP_struct.h, verSDP_struct.cpp	—	Define different necessary structures (e.g. block diagonal matrices, rectangular matrices, etc.) and operations on them (such as svec, smat).
infeassdp.h, infeassdp.cpp	—	Contain the algorithms permitting infeasibility verification.
arpack_interface.h, arpack_interface.cpp	—	Wrapper for the ARPACK++ eigenvalue calculating routine.
init.h	—	Configuration file. Contains flags defining, for example, if sparse or dense methods should be taken to perform LU decompositions.
sdpl.cpp	—	This is a program for verified SDP solving (see section B.2.1).
test.cpp, test.dat-s	—	A sample program and a test SDP problem in SDPA format demonstrating the use of the libverSDP library.
Makefile	—	A makefile.

After adapting the makefile to the environment (external library paths, preferred compiler, etc.) it will be enough to call `make` in the directory where verifiedSDP is located to compile the package. After a successful compilation, the callable library `libverSDP.a`, an executable binary `sdpl` and an example program `test` will be generated.

To test the installation, execute `test` or `sdpl`. Make sure the programs work and that information similar to the output below is displayed:

```
$ ./test test.dat-s
The primal objective value :-1.7784338847274242e+01
The dual objective value   :-1.7784340068152051e+01
The upper bound            :-1.7784338847274238e+01
The lower bound            :-1.7784627305451828e+01
```

```
$ ./sdpl
Use sdpl -{v|i} input_file [output_file]
-v :  for a rigorous verification of an SDP problem
```

```
-i :  for infeasibility check of an SDP problem
```

B.2 Examples of use

B.2.1 Stand-alone executable binary

To rigorously solve semidefinite optimization problems or to verify their feasibility, the `sdpl` program can be used. It requires input problem to be in either dense (`.dat`) or sparse (`.dat-s`) SDPA format. To obtain rigorous upper and lower bounds of the primal optimal value, `sdpl` should be executed with `-v` option, input file name and an optional output file name. To check, if primal or dual infeasibility of a problem can be proven, run `sdpl` with the `-i` option:

```
$ ./sdpl -v test.dat-s
$ ./sdpl -i test.dat-s test.out
```

B.2.2 Using verifiedSDP library

The simplest program doing rigorous semidefinite verification using our `libverSDP.a` library will look similar to the example below (the provided source code was taken from `test.cpp`):

```
#include <stdio.h>
#include <verSDP.h>
int main (int argc, char *argv[])
{
  verSDP *sdp = new verSDP();
  sdp->readProblemFromFile(argv[1]);
  sdp->solveProblem();
  sdp->pointToFile();
  double lowerBound = sdp->sdpLow();
  double upperBound = sdp->sdpUp();
  fprintf(stdout, "p* :%.16e\n", -sdp->sdpProblem->getDualObj());
  fprintf(stdout, "d* :%.16e\n", -sdp->sdpProblem->getPrimalObj());
  fprintf(stdout, "p^ :%.16e\n", upperBound);
  fprintf(stdout, "p_ :%.16e\n", lowerBound);
  delete(sdp);
```

```
   exit(0);
}
```

The program reads an SDPA file whose name is passed as a command line argument, calculates an approximate optimal value, and then finds the rigorous bounds. The `pointToFile()` method is used to save an approximate solution of the semidefinite problem in a text file. It can then be used for the successive verification, or even kept for a later reuse.

The corresponding makefile could then be:

```
PROF = $(HOME)/Profil-2.0
SDPA = $(HOME)/sdpa
UMF = $(HOME)/umfpack/UMFPACK
VSDP = $(HOME)/verifiedSDP

CFLAGS = -Wno-deprecated -c -O3 -fomit-frame-pointer
LFLAGS = -z muldefs
CC = g++

test:  test.o
      $(CC) -o test $(LFLAGS) test.o \
      -L$(VSDP) -L$(SDPA) -L$(HOME)/lapack/lib -L$(PROF)/lib \
      -L$(UMF)/Lib -lverifiedSDP -lsdpa -lg2c -lf2c -llapack2 \
      -lcblaswr -lcblas -latlas -lProfilPackages -lProfil \
      -lBias -llr -lm -lblas -lumfpack -lgfortran
test.o:  test.cpp
      $(CC) $(CFLAGS) test.cpp -I$(VSDP) -I$(SDPA) \
      -I$(PROF)/include -I$(PROF)/src/base -I$(HOME)/lapack/include
clean:
      rm -f *.o *.a test
```

As already mentioned, verifiedSDP implements only import of semidefinite programs in SDPA format. Nevertheless, any other alternative importer (for example for interval problems) can be easily written. One just has to generate internal structures used in verifiedSDP. Those are elaborately described in `verSDP_struct.h`.

For more detailed information on the library interface, please refer to the definitions in the corresponding header files directly.

Bibliography

[1] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, New York, 1983.

[2] F. Alizadeh. *Combinatorial optimization with interior point methods and semidefinite matrices.* PhD thesis, Univ. of Minnesota, October 1991.

[3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A.Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[4] M. Arcak and P.V. Kokotović. Nonlinear observers: A circle criterion design and robustness analysis. *Automatica*, 37(12):1923–1930, 2001.

[5] V. Balakrishnan and E. Feron, editors. *Linear Matrix Inequalities in Control Theory and Applications,* special edition of *International Journal of Robust and Nonlinear Control*, volume 6, no. 9/10. 1996.

[6] V. Balakrishnan, F. Wang, and L. Vandenberghe. Applications of semidefinite programming in process control. In *Proceedings of the American Control Conference*, volume 5, pages 3219–3223, 2000.

[7] B. Borchers. SDPLIB 1.2, A Library of Semidefinite Programming Test Problems. *Optimization Methods and Software*, 11(1):683–690, 1999.

[8] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia, 1994.

[9] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[10] Mahmoud Chilali, Pascal Gahinet, and Pierre Apkarian. Robust pole placement in lmi regions. *IEEE Transactions on Automatic Control*, 44:2257–2270, 1999.

[11] D.J. Chmielewski, T. Palmer, and V. Manousiouthakis. On the theory of optimal sensor placement. *AlChE Journal*, 48(5):1001–1012, 2002.

[12] A. J. Coleman. Structure of fermion density matrices. *Rev. Mod. Phys.*, 35(3):668–686, Jul 1963.

[13] Timothy A. Davis. Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):196–199, 2004.

[14] C. A. Floudas. *Deterministic Global Optimization - Theory, Methods and Applications*, volume 37 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, Dordrecht, Boston, London, 2000.

[15] R. M. Freund, F. Ordóñez, and K. C. Toh. Behavioral measures and their correlation with ipm iteration counts on semi-definite programming problems. *Math. Program.*, 109(2):445–475, 2007.

[16] K. R. Frisch. The logarithmic potential method for convex programming. Memorandum, Institute of Economics, University of Oslo, Oslo, Norway, May 1955.

[17] K. Fujisawa, M. Kojima, K. Nakata, and M. Yamashita. *SDPA (SemiDefinite Programming Algorithm) User's Manual — Version 6.2.0*. Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, September 2004.

[18] Mituhiro Fukuda. SDP benchmark problems from electronic structure calculations. `http://www.is.titech.ac.jp/~mituhiro/`.

[19] Mituhiro Fukuda, Bastiaan J. Braams, Maho Nakata, Michael L. Overton, Jerome K. Percus, Makoto Yamashita, and Zhengji Zhao. Large-scale semidefinite programs in electronic structure calculation. *Math. Program.*, 109(2):553–580, 2007.

[20] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[21] Claude Garrod and Jerome K. Percus. Reduction of the n-particle variational problem. *Journal of Mathematical Physics*, 5(12):1756–1776, 1964.

[22] F. A. M Gomes and D. C. Sorensen. *ARPACK++: a C++ implementation of the ARPACK eigenvalue package.* Technical Report TR97729, CRPC, Rice University, Houston, TX, USA, 1997.

[23] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[24] Q. P. Ha and H. Trinh. State and input simultaneous estimation for a class of nonlinear systems. *Automatica*, 40(10):1779–1785, 2004.

[25] E. R. Hansen. *Global Optimization using Interval Analysis*. Marcel Dekker, New York, 1992.

[26] C. Helmberg. *Semidefinite Programming for Combinatorial Optimization*. Habilitation thesis, TU Berlin, 2000. ZIB-Report ZR-00-34, Konrad-Zuse-Zentrum Berlin.

[27] D. Henrion and M. Šebek. Lmis and polynomial methods in control: Illustrative examples. Research Report 00075, LAAS-CNRS, 2000.

[28] H. Hindi and S. Boyd. Analysis of linear systems with saturation using convex optimization. In *Proceedings of the 37th IEEE Conference on Decision & Control*, volume 1, pages 903–908, 1998.

[29] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[30] P. Huard. Resolution of mathematical programming with nonlinear constraints by the method of centers. In *Nonlinear Programming*, pages 207–219. North Holland, Amsterdam, The Netherlands, 1967.

[31] C. Jansson. VSDP: Verified SemiDefinite Programming, User's Guide. Beta Version 0.1, to appear.

[32] C. Jansson. A Self-Validating Method for Solving Linear Programming Problems with Interval Input Data. *Computing*, Suppl. 6:33–45, 1988.

[33] C. Jansson. Rigorous Lower and Upper Bounds in Linear Programming. *SIAM J. Optimization (SIOPT)*, 14(3):914–935, 2004.

[34] C. Jansson. On Verifed Computation in Combinatorial Optimization. *International Symposium on Nonlinear Theory and its Applications (NOLTA2005), Bruges, Belgium*, pages 714–717, 2005.

[35] C. Jansson. Termination and verification for ill-posed semidefinite programming problems. *Optimization online*, 2005.

[36] C. Jansson. Guaranteed accuracy for conic programming problems in vector lattices. *Optimization online*, 2007.

[37] C. Jansson, D. Chaykin, and C. Keil. Rigorous error bounds for the optimal value in semidefinite programming. *SIAM J. Numer. Anal.*, 46(1):180–200, 2007.

[38] V. A. Kamenetskii. Absolute stability and absolute instability of control systems with several nonlinear nonstationary elements. *Automation and Remote Control*, 44(12):1543–1552, 1983.

[39] V. A. Kamenetskii. Convolution method for matrix inequalities. *Automation and Remote Control*, 50(5):598–607, 1989.

[40] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[41] R. B. Kearfott. *Rigorous Global Search: Continuous Problems*. Kluwer Academic Publisher, Dordrecht, 1996.

[42] R. B. Kearfott. On proving existence of feasible points in equality constrained optimization problems. *Math. Program.*, 83(1):89–100, 1998.

[43] O. Knüppel. PROFIL/BIAS and extensions, Version 2.0. Technical report, Inst. f. Informatik III, Technische Universität Hamburg-Harburg, 1998.

[44] R. Krawczyk. Fehlerabschätzung bei linearer Optimierung. In K. Nickel, editor, *Interval Mathematics*, volume 29 of *Lecture Notes in Computer Science*, pages 215–222. Springer Verlag, Berlin, 1975.

[45] A. I. Lur'e. *Some Nonlinear Problems in the Theory of Automatic Control*. Gostechizdat, Moscow, 1951. In Russian.

[46] G. Mayer. Result verification for eigenvectors and eigenvalues. In J. Herzberger, editor, *Topics in validated computations. Proceedings of the IMACS-GAMM international workshop, Oldenburg, Germany, 30 August - 3 September 1993*, Stud. Comput. Math. 5, pages 209–276, Amsterdam, 1994. Elsevier.

[47] David A. Mazziotti. Variational minimization of atomic and molecular ground-state energies via the two-particle reduced density matrix. *Phys. Rev. A*, 65(6):062511, Jun 2002.

[48] C. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, Philadelphia, U.S.A., 2000.

[49] R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.

[50] A. Nemirovski. Lectures on Modern Convex Optimization, 2003.

[51] Y.E. Nesterov and A.S. Nemirovski. A general approach to polynomial-time algorithms design for convex programming. Technical report, Centr. Econ. & Math. Inst., USSR Acad. Sci., Moscow, USSR, 1988.

[52] Y.E. Nesterov and A.S. Nemirovski. Self-concordant functions and polynomial time methods in convex programming. Technical report, Centr. Econ. & Math. Inst., USSR Acad. Sci., Moscow, USSR, May 1990.

[53] Netlib. Netlib linear programming library. `http://www.netlib.org/lp`.

[54] A. Neumaier. *Interval Methods for Systems of Equations*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1990.

[55] A. Neumaier. *Introduction to Numerical Analysis*. Cambridge University Press, 2001.

[56] A. Neumaier. Complete Search in Continuous Global Optimization and Constraint Satisfaction. *Acta Numerica*, 13:271–369, 2004.

[57] A. Neumaier and O. Shcherbina. Safe bounds in linear and mixed-integer programming. *Mathematical Programming, Ser. A*, 99:283–296, 2004.

[58] F. Ordóñez and R. M. Freund. Computational experience and the explanatory value of condition measures for linear optimization. *SIAM J. Optimization*, 14(2):307–333, 2003.

[59] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization : Algorithms and Complexity*. Dover Publications, Mineola, NY, 1998.

[60] T.E. Paré, H. Hindi, J.P. How, and S.P. Boyd. Synthesizing stability regions for systems with saturating actuators. In *Proceedings of the 37th IEEE Conference on Decision & Control*, volume 2, pages 1981 – 1982, 1998.

[61] P.A. Parrilo and S. Lall. Semidefinite programming relaxations and algebraic optimization in control. *European Journal of Control*, 9:307–321, 2003.

[62] E. S. Pyatnitskii and L. B. Rapoport. Existence of periodic motions and test for absolute stability of nonlinear nonstationary systems in the three-dimensional case. *Automation and Remote Control*, 52(5):648–658, 1991.

[63] M. V. Ramana, L. Tunçel, and H. Wolkowicz. Strong duality for semidefinite programming. *SIAM J. on Optimization*, 7(3):641–662, 1997.

[64] J. Renegar. Linear programming, complexity theory and elementary functional analysis. *Math. Program.*, 70(3, Ser. A):279–351, 1995.

[65] J. Rohn. Bounds on eigenvalues of interval matrices. Technical Report 688, Institute of Computer Science, Academy of Sciences, Prague, 1996.

[66] S. M. Rump. Validated Solution of Large Linear Systems. In R. Albrecht, G. Alefeld, and H.J. Stetter, editors, *Validation numerics: theory and applications*, volume 9 of *Computing Supplementum*, pages 191–212. Springer, 1993.

[67] S. M. Rump. Verification Methods for Dense and Sparse Systems of Equations. In J. Herzberger, editor, *Topics in Validated Computations — Studies in Computational Mathematics*, pages 63–136, Elsevier, Amsterdam, 1994.

[68] S. M. Rump. Verified Solution of Large Linear and Nonlinear Systems. In H. Bulgak and C. Zenger, editors, *Error Control and adaptivity in Scientific Computing*, pages 279–298. Kluwer Academic Publishers, 1999.

[69] S. M. Rump. INTLAB - Interval Laboratory, the Matlab toolbox for verified computations, Version 5.3, 2006.

[70] R. E. Skelton and T. Iwasaki. Increased roles of linear algebra in control education. *IEEE Control Syst. Mag.*, 15(4):76–89, 1995.

[71] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry*. Dover Publications, New York, 1996.

[72] M. J. Todd. Detecting Infeasibility in Infeasible-Interior-Point Methods for Optimization. In F. Cucker, R. DeVore, P. Olver, and E. Süli, editors, *Foundations of Computational Mathematics, Minneapolis 2002*, number 312 in London Mathematical Society Lecture Note Series, pages 157–192. Cambridge University Press, 2004.

[73] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.*, 95B(2):189–217, 2003.

[74] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[75] R. Clint Whaley, Antoine Petitet, and Jack J. Dongarra. Automated empirical optimization of software and the ATLAS project. *Parallel Computing*, 27(1–2):3–35, 2001.

[76] V. A. Yakubovich. The method of matrix inequalities in the stability theory of nonlinear control systems, I, II, III. *Automation and Remote Control*, 7(1964), 4(1965), 5(1965):1017–1029, 577–590, 753–763. In Russian.

[77] V. A. Yakubovich. The solution of certain matrix inequalities in automatic control theory. *Soviet Math. Dokl.*, 5:620–623, 1961. In Russian.

[78] Zhengji Zhao. *The Reduced Density Matrix Method for Electronic Structure Calculations - Application of Semidefinite Programming to N-fermion Systems*. PhD thesis, Department of Physics, New York University, 2004.

[79] Zhengji Zhao, Bastiaan J. Braams, Mituhiro Fukuda, Michael L. Overton, and Jerome K. Percus. The reduced density matrix method for electronic structure calculations and the role of three-index representability conditions. *The Journal of Chemical Physics*, 120(5):2095–2104, 2004.

Lebenslauf

Persönliche Daten

Name	Denis Chaykin
Geburtsdatum	01.12.1977
Geburtsort	Krasnogorsk

Schulausbildung

09/1985 - 06/1995	Schule Nr. 30, Chernigov

Studium

09/1995 - 06/2001	Angewandte Mathematik an der Bauman Moskauer Staatliche Technische Universität
06/2001	Abschluss: Diplom-Ingenieur
10/2001 - 02/2004	Informatik an der TU Hamburg-Harburg
02/2004	Abschluss: Master of Science

Wissenschaftliche Tätigkeit

03/2004 - 07/2005	Wissenschaftlicher Mitarbeiter an der TU Hamburg-Harburg, Institut für Zuverlässiges Rechnen
08/2005 - 04/2008	Wissenschaftlicher Mitarbeiter an der TU Hamburg-Harburg, Institut für Chemische Reaktionstechnik