Sparse Semidefinite Programming Relaxations for Large Scale Polynomial Optimization and Their Applications to Differential Equations

Martin Mevissen

Submitted in partial fulfillment of
the requirements for the degree of
DOCTOR OF SCIENCE

Department of Mathematical and Computing Sciences
Tokyo Institute of Technology

August 2010

Acknowledgements

The endeavor for the PhD in Tokyo and to complete this thesis would not have been possible without the support of a number of great people.

My greatest appreciation goes to my advisor Masakazu Kojima for his encouragement, guidance and continued support. I am grateful for his interest in my studies and the insight I gained by collaborating with him. I am indebted to him for providing me with an environment to enjoy research to the fullest ever since I joined his group for writing my Diploma thesis back in 2006.

Moreover, I would like to express my gratitude to Nobuki Takayama, who hosted me twice at Kobe university. I am thankful for his sincere interest and his engagement in our joint work with Kosuke Yokoyama.

My gratitude extends to Yasumasa Nishiura and Takashi Teramoto for inviting me to Hokkaido University in 2008. They offered me a great environment to learn more about reaction-diffusion equations. I am also thankful to Jean Lasserre and Didier Henrion for hosting me at LAAS in 2009, and I am looking forward to continuing our joint work in the future. I would like to thank Sunyoung Kim, Makoto Yamashita and Jiawang Nie for our fruitful collaborations and exciting discussions. In particular, I would like to express my gratitude to Makoto for his constant and patient technical support. Many thanks go to Hans-Jakob Lüthi for supporting this venture in Japan from early on and his encouraging advice. Also, I would like to thank the German Academic Exchange Service for enabling me to pursue this journey with its Doctoral Scholarship for three years.

The stay at Tokyo Institute of Technology would have been inconceivable without the people I shared this time, many thoughts and the bond of friendship. In particular, I would like to thank Paul Sheridan, for his unshakable optimism, Ken Shackleton, for his large-heartedness, Matthias Hietland Heie, for his open mind, Hiroshi Sugimoto, for his noble heart, Kojiro Akiba, for all our conversation, Mikael Onsjö, for being a great host, and Tomohiko Mizutani, for helping me out with a lot of things, when I was a newcomer.

In my life in Japan, I was glad to find many friends who I can count on and who gave me the chance to call this place home. I enjoyed greatly everything I shared with them. Thank you so much, Shuji, Yoko, Jif, Azra, Masa, Mari, Moe, Hiroshi, Shota, Chiaki, Soji, Naomi, Tomoko.

Finally, my deepest gratefulness goes to my parents. Their encouragement and love have been with me all the time. They stood by me every day of my life. I dedicate this thesis to them.

To my parents

Contents

Notation

$\mathbb{N}$	natural numbers
$\mathbb{N}^n$	n- dimensional vector space with entries in $\mathbb{N}$
$\mathbb{Z}$	integers
$\mathbb{R}$	real numbers
$\mathbb{R}^n$	n-dimensional vector space with real entries
$\mathbb{R}^{n\times m}$	vector space of n by m matrices with real entries
$\mathbb{S}^n$	vector of symmetric matrices in $\mathbb{R}^{n\times n}$
$\mathbb{S}^n_+$	cone of symmetric, positive semidefinite matrices in $\mathbb{R}^{n\times n}$
$\mathbb{S}^n(E,?)$	partial symmetric matrices with entries specified in E
$\mathbb{S}^n_+(E,?)$	matrices in $\mathbb{S}^n(E,?)$ that can be completed to positive semidefinite matrices
$\mathbb{S}^n(E,0)$	symmetric matrices with nonzero entries on the diagonal and in E
$\mathbb{S}^n_+(E,0)$	positive semidefinite matrices in $\mathbb{S}^n(E,0)$
$\mathbb{R}[x]$	ring of multivariate polynomials in the n dimensional variable x with coefficients in $\mathbb{R}$
$\mathbb{R}[x]_d$	set of polynomials of degree less or equal d
$\mathbb{R}[x,\mathcal{A}]$	set of polynomials supported on $\mathcal{A}\subset\mathbb{N}^n$, $\mathbb{R}[x,\mathcal{A}]=\{p\in\mathbb{R}[x] \mid \operatorname{supp}(p)\subset\mathcal{A}\}$
$\sum\mathbb{R}[x]^2$	set of sums of squares polynomials, $\sum\mathbb{R}[x]^2 := \left\{p\in\mathbb{R}[x] \mid p=\sum_{i=1}^r p_i^2,\ p_i\in\mathbb{R}[x] \text{ for some } r\in\mathbb{N}\right\}$
$\Lambda(d)$	set of multivariate indices of degree less or equal d, $\Lambda(d)=\{\alpha\in\mathbb{N}^n : \mid \alpha \mid\leq d\}$
$u(x,\mathcal{A})$	monomial vector for $\mathcal{A}\subset\mathbb{N}^n$, $u(x,\mathcal{A})=(x^\alpha \mid \alpha\in\mathcal{A})$
$G(N,E)$	graph with vertex set N and edge set E

$\succcurlyeq$	positive semidefinite matrix
$\succ$	positive definite matrix
$\bullet$	interior product on $\mathbb{S}^n$, $A\bullet B=\sum_{i=1}^n\sum_{j=1}^n A_{i,j}B_{i,j}$
$\det(\cdot)$	determinant of a matrix
$\operatorname{rank}(\cdot)$	rank of a matrix
$\operatorname{Tr}(\cdot)$	trace of a matrix, $\operatorname{Tr}(A):=\sum_{i=1}^n A_{i,i}$
$\deg(\cdot)$	degree of a polynomial
$\operatorname{supp}(\cdot)$	support of a polynomial, $\operatorname{supp}(p):=\{\alpha\in\mathbb{N}^n \mid p_\alpha\neq 0\}$

I	imaginary unit, i.e., $I^2 = -1$
$K(g_1, \ldots, g_m)$, K	basic, closed semialgebraic set generated by $g_1, \ldots, g_m \in \mathbb{R}[x]$, $K(g_1, \ldots, g_m) := \{x \in \mathbb{R}^n \mid g_1(x) \geq 0, \ldots, g_m(x) \geq 0\}$
$M(K)$	quadratic module defined by $g_1, \ldots, g_m$, $M(K) = \sum \mathbb{R}[x]^2 + g_1 \sum \mathbb{R}[x]^2 + \ldots + g_m \mathbb{R}[x]^2$
$M_d(K)$	approximation of order d for $M(K)$ $M_d(K) = \left\{\sum_{i=0}^m \sigma_i g_i \mid \sigma_i \in \sum \mathbb{R}[x]^2, \deg(\sigma_i g_i) \leq 2d\right\}$
$\sum^2\langle g_1, \ldots, g_m\rangle$	multiplicative convex cone generated by $\mathbb{R}[x]^2$ and $g_1, \ldots, g_m$, $\sum^2\langle g_1, \ldots, g_m\rangle = M(K) + g_1 g_2 \sum \mathbb{R}[x]^2 + \ldots + g_1 g_2 \cdots g_m \sum \mathbb{R}[x]^2$
$O(g_1, \ldots, g_m)$	multiplicative monoid generated by $g_1, \ldots, g_m$, $O(g_1, \ldots, g_m) = \{\prod_{i=1}^r t_i \mid t_i \in \{g_1, \ldots, g_m\} \text{ for } r \in \mathbb{N}\}$
$I(g_1, \ldots, g_m)$	ideal generated by $g_1, \ldots, g_m$, $I(g_1, \ldots, g_m) = \mathbb{R}[x] + g_1 \mathbb{R}[x] + \ldots + g_m \mathbb{R}[x]$
$M_w(y)$	moment matrix of order w for the vector y
$M_w(y, I)$	partial moment matrix, contains only those components y_α of y with $\alpha \in I$
$M_w(y\, g)$	localizing matrix of order w for vector y and $g \in \mathbb{R}[x]$
$M_w(y\, g, I)$	partial localizing matrix
$\mathcal{M}_S^k$	higher monomial set of a POP
$\mathcal{M}^k$	higher monomial list of a POP
$k_{i,j}^\alpha$	dividing coefficient
k_0	number of substitutions required by an algorithm to transform a given POP into a QOP
t_C	total computation time of an algorithm
ϵ_{sc}	scaled feasibility error of numerical solution for a POP
ϵ_{obj}	optimality error of SDP relaxation solution for a POP
ω	relaxation order of dense or sparse SDP relaxation
m_e	maximum eigenvalue of a Jacobian of a system of polynomial equations
N_x	number of grid points in x-direction in a discretized domain
h_x	distance of two grid points in x-direction in a discretized domain
$u_{i,j}$	approximation of u at grid point (x_i, y_j) in a finite difference scheme
max	maximum
min	minimum
sup	supremum
inf	infimum
lbd	lower bound
ubd	upper bound
SDP	semidefinite program
POP	polynomial optimization problem
QOP	quadratic optimization problem
ODE	ordinary differential equation
PDE	partial differential equation
OCP	optimal control problem
FDM	finite difference method
FEM	finite element method
FVM	finite volume method
SQP	sequential quadratic programming
QSDP	quadratic semidefinite program

Chapter 1

Introduction

1.1 Motivation

A wide variety of problems arising in mathematics, physics, engineering, control and computer science can be formulated as optimization problems, where all functions in objective and constraints are multivariate polynomials over the field of real numbers - so called polynomial optimization problems (POP). In general, polynomial optimization problems are severely non-convex and NP-hard to solve. In recent years there has been active research in semidefinite programming (SDP) relaxation methods for POPs. That is, a general, non-convex POP is relaxed to a convex optimization problem, an SDP. Solving the SDP provides either a lower bound to the minimum, or approximations for minimum and global minimizers of the POP. First convexification techniques for general POPs have been proposed by Shor [91] and Nesterov [70]. Since the pioneering work of Shor, convexification and in particular SDP relaxation techniques have been used in an ever increasing number of applications and problems. One of the classical examples is the SDP relaxation for a non-convex quadratic programming formulation of the NP-hard max-cut problem [26]. Other NP-hard problems, that can be formulated as POPs are $\{0,1\}-$ linear programming, or testing whether a symmetric matrix is copositive [67]. A breakthrough in this field was Lasserre's seminal paper [51]. Given the feasible set of a general POP is a basic, compact semialgebraic set, a hierarchy of SDP relaxations can be constructed, which provides a monoton increasing sequence of lower bounds for the minimum of the POP. Lasserre showed this sequence converges to the minimum of the POP under a fairly general condition. Lasserre's relaxation and also the approach [79] by Parrilo do rely on the representation of nonnegative polynomials as sum of squares of polynomials and the dual theory of moments. Despite being a powerful theoretical tool to approximate minimum and minimizer of general polynomial optimization problems, the Lasserre relaxation is not practical even for medium scale POPs. In fact, the size of the matrix inequality constraints in the SDP relaxations grows rapidly for increasing order of the hierarchy. Thus, in the case of a medium or large scale POPs the SDP relaxations becomes intractable for current SDP solvers as SeDuMi [95] or SDPA [104], even for small choices of the relaxation order. However, large scale POP arise from challenging problems and efficient methods to solve them are in high demand. For instance one problem, which received lots of attention recently, is the sensor network localization problem [6, 71, 43].
A first approach to reduce the size of SDP relaxations for POPs has been the concept of correlative sparsity of a POP [47, 102, 52]. Exploiting structured sparsity enables to attempt POP of larger dimension by a hierarchy of sparse SDP relaxations. Still, the capacity of current SDP solvers is limiting the applicability of sparse SDP relaxations for large scale POP. As one way to take advantage of sparsity more efficiently, we develop a general notion of sparsity in linear and nonlinear matrix inequalities and show how to exploit this sparsity via positive semidefinite matrix completion. We demonstrate how so called domain-space and range-space sparsity can be used to decrease the size of SDP relaxations for large scale POPs substantially. Another technique to attempt large scale POPs is based on the idea to reduce the size of the sparse SDP relaxations by transforming a general POP into an equivalent quadratic optimization problem (QOP). For an important class of large scale POPs the size of sparse SDP relaxations for the equivalent QOPs is far

smaller than the size of sparse SDP relaxations for the original POPs.
The second topic of this thesis is to investigate how to efficiently apply sparse SDP relaxation techniques for an important class of challenging problems, the numerical analysis of differential equations. For most problems involving ordinary or partial differential equations it is not possible to find analytic solutions - in particular if the equations are nonlinear in the unknown function. Even the problem to find approximations to the solutions of ODEs or PDEs by numerical methods is well known to be a hard problem, which begins to attract attention by researchers in moment, SDP and numerical algebra techniques. On the one hand a moment based approach [5] has been proposed to find tight bounds for linear functionals defined on linear PDEs. On the other hand, a homotopy continuation based approach [2, 33, 34] has been proposed to find all solutions of a discretized, possibly nonlinear PDE. We will show how to transform a problem involving differential equations into a POP by using standard finite difference schemes. The dimension of these POPs is determined by the discretization of the domain of the differential equation. Thus, for fine discretizations we obtain a challenging class of large scale POPs. These POPs satisfy both, correlative and domain-space sparsity, which enables us to apply sparse SDP relaxation techniques efficiently. The sparse SDP relaxation method is of particular interest for PDE problems with several solutions. We propose different algorithms based on the sparse SDP relaxation method to detect several or even all solutions of a system of nonlinear PDE. It is a strength of this method, that a wide variety of nonlinear PDE problems can be solved: Nonlinear elliptic, parabolic and hyperbolic equations, reaction-diffusion equations, steady state Navier-Stokes equations in fluid dynamics, differential algebraic equations or nonlinear optimal control problems.

1.2 Contribution

This thesis is largely based on the content of prior publications of the author. Its contributions can be summarized as follows.

- We present a general framework to detect, characterize and exploit sparsity in an optimization problem with linear and nonlinear matrix inequality constraints via positive semidefinite matrix completion. We distinguish two types of sparsity, *domain-space sparsity* for the symmetric matrix variable in objective and constraint functions of the problem, and *range-space sparsity*. Four conversion methods are proposed to exploit these two types of sparsity. We demonstrate the efficiency of these conversion methods on SDP relaxations for sparse, large-scale POP derived from discretizing partial differential equations and the sensor network localization problem. This result is based on our work [42].

- Based on the observation dating back to Shor [92], that any POP can be written as an equivalent QOP, we develop four heuristics for transforming a POP into a QOP. We show, that sparsity of the POP is maintained under our transformation procedures, and propose different techniques to improve the quality of the sparse SDP relaxations for the QOP, which are weaker than the more expensive sparse SDP relaxations for the equivalent POP. This technique is shown to be very efficient for large-scale POP: We are able to obtain highly accurate approximations to the global optimizers of the POP by solving SDP relaxations of vastly reduced size. This work is presented in detail in [62].

- We are the first to introduce a method based on sparse SDP relaxations to solve systems of linear and partial differential equations [61, 63]. Unlike the approach [5] we are able to approximate the actual solutions of an ordinary or partial differential equation. Moreover, our approach is applicable to nonlinear differential equations, whereas the technique [5] is limited to linear PDEs. Furthermore, compared to the numerical algebraic approach [2, 33, 34], we can solve a system of polynomial equations derived from a PDE for a much finer discretization by exploiting sparsity. Also, we do not aim at finding all complex solutions, but we detect the real solutions to that system of equations one by one.

- Comparing the sparse SDP relaxation method to solve differential equations to existing PDE solvers, our approach has the following advantages: (a) We can add polynomial inequality constraints for the unknown solutions of the differential equations to the system of equations obtained by the finite

difference discretization of the PDE, which can be understood as restricting the space of functions we are searching for solutions. (b) We can detect particular solutions of a PDE, by choosing an appropriate objective function for the sparse POP derived from the PDE problem or by adding inequality constraints to that POP. (c) We are able to systematically enumerate all solutions of a discretized PDE problem by iteratively applying the SDP relaxation method. (d) We exploit the fact, that the sparse SDP relaxations provide an approximation to the global optimizer of a POP. Thus, even if the accuracy of the solution of the SDP relaxation is not high, it is a good initial guess for locally convergent solvers for many PDE problems. This fact is in particular interesting for PDE problems with many solutions. These results are based on our work in [61, 63, 62].

- The sparse SDP relaxation method for solving large scale POP derived from differential equations can be applied to solve nonlinear, optimal control problems. Unlike the moment method in [54] our method yields approximations to the optimal control, trajectories and value of a control problem in addition to provide lower bounds for the optimal value of the control problems.

1.3 Outline of the thesis

This thesis consists of two main parts. In the first part given by Chapter 2 we introduce the approaches to use methods from convex optimization to solve general, nonconvex polynomial optimization problems. We begin in 2.1 with introducing the historical background of characterizing positive polynomials, the problem of minimizing multivariate polynomials over basic, closed semialgebraic sets and the dense Lasserre relaxation, a sequence of semidefinite programs whose minima converge to the minimum of a polynomial optimization problem under fairly general conditions. Finally, we review the method of exploiting correlative sparsity of a POP to construct a sequence of sparse SDP relaxations. In 2.2 we present a general framework to exploit domain- and range space sparsity in problems involving linear or nonlinear matrix inequalities. This technique can be applied to the large scale SDP relaxations for large scale POP. In 2.3 we introduce the approach to reduce the size of dense or sparse SDP relaxations for large scale POP, which is based on the idea to transform a general POP into an equivalent QOP.
In the second part presented by Chapter 3 we show how to use the methods and techniques from Chapter 2 for the numerical analysis of ordinary and partial differential equations. First we give an overview over existing numerical methods for solving partial differential equations, in particular the two most common approaches, the finite difference method and the finite element method, in 3.1. In 3.2 we introduce our method to transform a problem involving partial differential equations into a POP via finite difference discretization, and to solve the resulting large scale POP by the SDP relaxation techniques from Chapter 2. In 3.3 we apply our SDP relaxation method to a variety of different PDE problems such as nonlinear elliptic, parabolic and hyperbolic equations, differential algebraic equations, reaction-diffusion equations, fluid dynamics and nonlinear optimal control.
Finally, we summarize the thesis in Chapter 4 with some concluding remarks and give an outlook on possible future research directions based on the methods and results presented here.

Chapter 2

Semidefinite Programming and Polynomial Optimization

2.1 Positive polynomials and polynomial optimization

Polynomial optimization and the problem of global nonnegativity of polynomials are active fields of research and remain in the focus of researchers from various areas as real algebra, semidefinite programming and operator theory. Shor [91] was the first who introduced the idea of applying a convex optimization technique to minimize an unconstrained multivariate polynomial. Also, Nesterov [70] was one of the first who discussed to exploit the duality of moment cones and cones of nonnegative polynomials in a convex optimization framework. He showed the characterization of a moment cone by linear matrix inequalities, i.e., semidefinite constraints, if the elements of the corresponding cone of nonnegative polynomials can be written as sum of squares. The next milestone in minimizing multivariate polynomials was given by Lasserre [51], who applied recent real algebraic results by Putinar [81] to construct a sequence of semidefinite program relaxations whose optima converge to the optimum of a polynomial optimization problem. Another approach to apply real algebraic results to attempt the problem of nonnegativity of polynomials was introduced by Parrilo [79]. We attempt to solve the following **polynomial optimization problem**:

$$\begin{array}{ll} \min & p(x) \\ \text{s.t.} & g_i(x) \geq 0 \quad \forall i = 1, \ldots, m \end{array} \tag{2.1}$$

where $p,\ g_1, \ldots,\ g_m \in \mathbb{R}[x]$. Problem (2.1) can also be written as

$$\min_{x \in K}\ p(x) \tag{2.2}$$

where K the basic, closed semialgebraic set that is defined by the polynomials $g_1, \ldots, g_m$. Let $p^\star$ denote the optimal value of problem (2.2) and $K^\star := \{x^\star \in K \mid \forall x \in K :\ p(x^\star) \leq p(x)\}$. In the case K compact, $K^\star \neq \emptyset$, if $K \neq \emptyset$.

2.1.1 Decomposition of globally nonnegative polynomials

The origin of research in characterizing nonnegative and positive polynomials lies in Hilbert's 17th problem, whether it is possible to express a nonnegative rational function as sum of squares of rational functions. This question was answered positively by Artin in 1927. Moreover, the question arises, whether it is possible to express any nonnegative polynomial as sum of squares of polynomials. In the case of univariate polynomials the answer to this question is yes, as stated in the following theorem.

Theorem 2.1 *Let $p \in \mathbb{R}[x]$, $x \in \mathbb{R}$. Then, $p(x) \geq 0$ for all $x \in \mathbb{R}$ if and only if $p \in \sum \mathbb{R}[x]^2$.*

Proof " $\Leftarrow$ " : Trivial.
" $\Rightarrow$ " : Let $p(x) \geq 0$ for all $x \in \mathbb{R}$. It is obvious that $\deg(p) = 2k$ for some $k \in \mathbb{N}$. Then, the real roots of of $p(x)$ should have even multiplicity, otherwise $p(x)$ would alter its sign in a neighborhood of a root. Let $\lambda_i,\ i = 1, \ldots, r$ be its real roots with corresponding multiplicity $2m_i$. Its complex roots can be arranged in conjugate pairs, $a_j + Ib_j,\ a_j - Ib_j,\ j = 1, \ldots, h$. Then,

$$p(x) = C \prod_{i=1}^{r} (x - \lambda_i)^{2m_i} \prod_{j=1}^{h} ((x - a_j)^2 + b_j^2).$$

Note that the leading coefficient C needs to be positive. Thus, by expanding the terms in the products, we see that $p(x)$ can be written as a sum of squares of polynomials, of the form

$$p(x) = \sum_{i=0}^{k} \left(\sum_{j=0}^{k} v_{ij} x^j \right)^2 . \qquad \square$$

However, Hilbert himself already noted that not every nonnegative polynomial can be written as sum of squares. For instance the Motzkin form M,

$$M(x, y, z) = x^4 y^2 + x^2 y^4 + z^6 - 3x^2 y^2 z^2$$

is nonnegative but not sum of squares. In fact Hilbert gave a complete characterization of the cases where nonnegativity and the existence of a sum of squares decomposition are equivalent.

Definition 2.1 *A **form** is a polynomial where all the monomials have the same total degree m. $P_{n,m}$ denotes the set of nonnegative forms of degree m in n variables, $\Sigma_{n,m}$ the set of forms p such that $p = \Sigma_k h_k^2$, where h_k are forms of degree $\frac{m}{2}$.*

There is a correspondance between forms in n with power m and polynomials in $n-1$ variables with degree less or equal to m. In fact, a form in n variables of degree m can be dehomogenized to a polynomial in $n-1$ variables by fixing any of the n variables to the constant value 1. Conversely, given a polynomial in $n-1$ variables in can be homogenized by multiplying each monomial by powers of a new variable such that the degree of all monomials equals m. Obviously, $\Sigma_{n,m} \subseteq P_{n,m}$ holds for all n and m. The following Theorem is due to Hilbert.

Theorem 2.2 *$\Sigma_{n,m} \subseteq P_{n,m}$ holds with equality only in the following cases:*

(i) Bivariate forms: $n = 2$,

(ii) Quadratic forms: $m = 2$,

(iii) Ternary quartic forms: $n = 3$, $m = 4$.

We interprete the three cases in Theorem 2.2 in terms of polynomials. The first one corresponds to the equivalence of nonnegativity and sum of squares condition in the univariate case as in Theorem (2.1). The second one is the case of quadratic polynomials, where the sum of squares decomposition follows from an eigenvalue/eigenvector factorization. The third case corresponds to quartic polynomials in two variables.

Relevance of sum of squares characterizations Recall that the constraints of our original polynomial optimization problem are nonnegativity constraints for polynomials of the type $g_i(x) \geq 0 \ \ (i = 1, \ldots, m)$. The question, whether a given polynomial is globally nonnegative is decidable, for instance by the Tarski-Seidenberg decision procedure [7]. Nonetheless, regarding complexity, the general problem of testing global nonnegativity of a polynomial function is NP-hard [67], if the degree of the polynomial is at least four. Therefore it is reasonable to substitute the nonnegativity constraints by expressions that can be decided easier. It was shown by Parrilo that the decision whether a polynomial is sum of squares is equivalent to a semidefinite program as stated in the following theorem.

Theorem 2.3 *The existence of a sum of squares decomposition of a polynomial in n variables of degree 2d can be decided by solving a semidefinite programming feasibility problem [79]. If the polynomial is dense, the dimensions of the matrix inequality are equal to* $\begin{pmatrix} n+d \\ d \end{pmatrix} \times \begin{pmatrix} n+d \\ d \end{pmatrix}$.

Proof Let $p \in \mathbb{R}[x]$ with degree $2d$. Recall $u(x, \Lambda(d))$ denotes the ordered vector of monomials $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ with $\sum_{i=1}^n \alpha_i \leq d$. The length of $u(x, \Lambda(d))$ is $s := s(d) = \begin{pmatrix} n+d \\ d \end{pmatrix}$.
Claim: $p \in \sum \mathbb{R}[x]^2$ if and only if $\exists V \in \mathbb{S}_+^s$ such that $p = u(x, \Lambda(d))^T V u(x, \Lambda(d))$.
Pf: $\Rightarrow$: $p \in \sum \mathbb{R}[x]^2$, i.e.

$$p = \sum_{i=1}^r q_i^2 = \sum_{i=1}^r (w_i^T u(x, \Lambda(d)))^2 = u(x, \Lambda(d))^T \left(\sum_{i=1}^r w_i w_i^T \right) u(x, \Lambda(d)).$$

Thus, $V = \sum_{i=1}^r w_i w_i^T$ and $V \in \mathbb{S}_+^s$.
$\Leftarrow$: As $V \in \mathbb{S}_+^s$ there exists a Cholesky factorization $V = WW^T$, where $W \in \mathbb{R}^{s \times s}$ and let w_i denote the ith column of W. We have

$$p = u(x, \Lambda(d))^T V u(x, \Lambda(d)) = \sum_{i=1}^s w_i w_i^T u(x, \Lambda(d)) = \sum_{i=1}^s (w_i^T u(x, \Lambda(d)))^2,$$

i.e., $p \in \mathbb{R}[x]$. Thus, the claim follows.
Expanding the quadratic form gives $p = \sum_{i,j=1}^s V_{i,j} u(x, \Lambda(d))_i u(x, \Lambda(d))_j$. Equating the coefficients in this expression with the coefficients of the corresponding monomials in the original form for p generates a set of linear equalities for the variables $V_{i,j}$ $(i, j = 1, \ldots, s)$. Adding the constraint $V \in \mathbb{S}_+^s$ to those linear equality constraints, we obtain conditions for p which are equivalent to claiming $p \in \sum \mathbb{R}[x]^2$. Therefore, to decide whether $p \in \sum \mathbb{R}[x]^2$ is equivalent to solving a semidefinite programming feasibility problem. □

2.1.2 Decomposition of polynomials positive on closed semialgebraic sets

Real algebraic geometry deals with the analysis of the real solution set of a system of polynomial equations. The main difference to algebraic geometry in the complex case lies in the fact that $\mathbb{R}$ is not algebraically closed. One of the main results of real algebra are the **Positivstellensätze** which provide certificates in the case a semialgebraic set is empty. Improved versions of the Positivstellensätze can be obtained in case of compact semialgebraic sets.

General semialgebraic sets

The **Positivstellensatz** below is due to Stengle; a proof can be found in [7].

Theorem 2.4 (Stengle) *Let* $(f_j)_{j=1,\ldots,t}$, $(g_k)_{k=1,\ldots,m}$, $(h_l)_{l=1,\ldots,k}$ *be finite families of polynomials in* $\mathbb{R}[x]$. *The following properties are equivalent:*

(i) $$\left\{ x \in \mathbb{R}^n \mid \begin{array}{ll} g_j(x) \geq 0, & j = 1, \ldots, m \\ f_s(x) \neq 0, & s = 1, \ldots, t \\ h_i(x) = 0, & i = 1, \ldots, k \end{array} \right\} = \emptyset.$$

(ii) There exist $g \in \Sigma^2 \langle g_1, \ldots, g_m \rangle$, $f \in O(f_1, \ldots, f_t)$, *the multiplicative monoid generated by* $f_1, \ldots, f_t$, $h \in I(h_1, \ldots, h_k)$, *the ideal generated by* $h_1, \ldots, h_k$, *such that* $g + f^2 + h = 0$.

To understand the differences between the real and the complex case, and the use of the Positivstellensatz 2.4 consider the following example.

Example 2.1 *Consider the very simple quadratic equation*

$$x^2 + ax + b = 0.$$

By the fundamental theorem of algebra, the equation has always solutions in $\mathbb{C}$*. For the case when the solution is required to be real, the solution set will be empty if and only if the discriminant* D *satisfies*

$$D := b - \frac{a^2}{4} > 0.$$

In this case taking

$$g := \left(\frac{1}{\sqrt{D}}(x + \frac{a}{2})\right)^2, \quad f := 1, \quad h := -\frac{1}{D}(x^2 + ax + b),$$

the identity $g + f^2 + h = 0$ *is satisfied.*

It is to remark, the Positivstellensatz represents the most general deductive system for which inferences from the given equations can be made. It guarantees the existence of **infeasibility certificates** given by the polynomials f, g and h. For complexity reasons these certificates cannot be polynomial time checkable for every possible instance, unless NP=co-NP. Parrilo showed that it is possible that the problem of finding infeasibility certificates is equivalent to an semidefinite program, if the degree of the possible multipliers is restricted [79].

Theorem 2.5 *Consider a system of polynomial equalities and inequalities as in Theorem 2.4. Then, the search for bounded degree Positivstellensatz infeasibility certificates can be done using semidefinite programming. If the degree bound is chosen to be large enough , then the SDPs will be feasible, and the certificates are obtained from its solution.*

Proof: Consequence of the Positivstellensatz and Theorem 2.3, c.f. [79].

As the feasible set of (2.2) is a closed semialgebraic set, we are interested in characterizations for these sets and polynomials positive on semialgebraic sets. The Positivstellensatz allows to deduce conditions for the positivity or the nonnegativity of a polynomial over a semialgebraic set. A direct consequence of the Positivstellensatz is the following corollary [7], pp. 92.

Corollary 2.1 *Let* $g_1, \ldots, g_m \in \mathbb{R}[x]$,
$K = \{x \in \mathbb{R}^n \mid g_1(x) \geq 0, \ldots, g_m(x) \geq 0\}$ *and* $f \in \mathbb{R}[x]$*. Then:*

(i) $\forall x \in K \;\; f(x) \geq 0 \;\Leftrightarrow\; \exists\, s \in \mathbb{N}\, \exists\, g, h \in \Sigma^2\langle g_1, \ldots, g_m\rangle$ *s.t.* $fg = f^{2s} + h$.

(ii) $\forall x \in K \;\; f(x) > 0 \;\Leftrightarrow\; \exists\, g, h \in \Sigma^2\langle g_1, \ldots, g_m\rangle$ *s.t.* $fg = 1 + h$.

Proof

(i) Apply the Positivstellensatz to the set

$$\{x \in \mathbb{R}^n \mid g_1(x) \geq 0, \ldots, g_m(x) \geq 0, -f(x) \geq 0, f(x) \neq 0\}.$$

(ii) Apply the Positivstellensatz to the set

$$\{x \in \mathbb{R}^n \mid g_1(x) \geq 0, \ldots, g_m(x) \geq 0, -f(x) \geq 0\}. \qquad \square$$

These conditions for the nonnegativity and positivity of polynomials on semialgebraic sets can be improved under additional assumptions. We present these improved conditions for **compact** semi-algebraic sets in the following section.

Compact semialgebraic sets

It is our aim to characterize polynomials that are positive or nonnegative on compact semialgebraic sets. A first characterization is a theorem due to Schmüdgen [87]:

Theorem 2.6 (Schmüdgen) *Let $K = \{x \in \mathbb{R}^n;\ g_1(x) \geq 0, \ldots, g_m(x) \geq 0\}$ be a compact semialgebraic subset of $\mathbb{R}^n$ and let p be a positive polynomial on K. Then $p \in \Sigma^2\langle g_1, \ldots, g_m\rangle$.*

It was Putinar [81] who simplified this characterization under an additional assumption.

Definition 2.2 *A quadratic module $M(K)$ is called **archimedean** if $N - \sum_{i=1}^n x_i^2 \in M(K)$ for some $N \in \mathbb{N}$.*

Theorem 2.7 (Putinar) *Let p be a polynomial, positive on the compact semialgebraic set K and $M(K)$ archimedian, then $p \in M(K)$.*

Thus, under the additional assumption of $M(K)$ being archimedian, we obtain the stricter characterization $p \in M(K) \subseteq \Sigma^2\langle g_1, \ldots, g_m\rangle$ instead of $p \in \Sigma^2\langle g_1, \ldots, g_m\rangle$. The original proof of Theorem 2.7 is due to Putinar [81]. In this proof Putinar applies the separation theorem for convex sets and some arguments from functional analysis. A new proof due to Schweighofer [88] avoids the arguments from functional analysis and requires only results from elementary analysis. A further theorem by Schmüdgen [87] provides equivalent conditions for $M(K)$ being archimedian.

Theorem 2.8 *The following are equivalent:*

(i) There exist finitely many $t_1, \ldots, t_s \in M(K)$ such that the set

$$\{x \in \mathbb{R}^n \mid t_1(x) \geq 0, \ldots, t_s(x) \geq 0\}$$

(which contains K) is compact and $\prod_{i\in I} t_i \in M(K)$ for all $I \subset \{1, \ldots, s\}$.

(ii) There exists some $p \in M(K)$ such that $\{x \in \mathbb{R}^n \mid p(x) \geq 0\}$ is compact.

(iii) There exists an $N \in \mathbb{N}$ such that $N - \sum_{i=1}^n x_i^2 \in M(K)$, i.e., $M(K)$ is archimedian.

(iv) For all $p \in \mathbb{R}[x]$, there is some $N \in \mathbb{N}$ such that $N \pm p \in M(K)$.

Thus, for any polynomial p positive on K, $p \in M(K)$ holds, if one of the conditions in Theorem 2.8 is satisfied. Whether it is decidable that one of the equivalent conditions hold is not known and subject of current research. However, for a given polynomial optimization problem with compact feasible set K, it is easy to make the corresponding quadratic module $M(K)$ archimedian. We just need to add a redundant constraint $N - \sum_{i=1}^n x_i^2 \geq 0$ for a sufficiently large N.

Example 2.2 *Consider the compact semialgebraic set*

$$K = \left\{x \in \mathbb{R}^2 \mid g(x) = 1 - x_1^2 - x_2^2 \geq 0\right\}.$$

The quadratic module $M(K)$ is archimedian, as $1 - x_1^2 - x_2^2 = 0^2 + 1^2 \cdot g(x) \in M(K)$. The polynomials $f_1(x) := x_1 + 2$ and $x_1^3 + 2$ are positive on K. Thus $f_1, f_2 \in M(K)$ with Theorem 2.7. Their decomposition can be derived as

$$\begin{array}{lll} f_1(x) & = x_1 + 2 & = \frac{1}{2}(x_1+1)^2 + \frac{1}{2}x_2^2 + 1 + \frac{1}{2}(1 - x_1^2 - x_2^2), \\ f_2(x) & = 2x_1^3 + 3 & = (x_1^3+1)^2 + (x_1^2x_2)^2 + (x_1x_2)^2 + x_2^2 + 1 + (x_1^4 + x_1^2 + 1)\,(1 - x_1^2 - x_2^2). \end{array}$$

The next example demonstrates that in general not every polynomial nonnegative on a compact semialgebraic set K is contained in $M(K)$ even if $M(K)$ is archimedian.

Example 2.3 *Consider the compact semialgebraic set*

$$K = \left\{x \in \mathbb{R} \mid g_1(x) := x^2 \geq 0,\ g_2(x) := -x^2 \geq 0\right\}.$$

It is obvious that $M(K)$ is archimedian. Also, it is easy to see that there are no $q, r, s \in \sum \mathbb{R}[x]^2$ such that

$$p(x) := x = q(x) + r(x)\, x^2 + s(x)\, (-x^2),$$

although p is nonnegative on K. However, the polynomial $p_a \in \mathbb{R}[x]$ defined by $p_a(x) = x + a$ for $a > 0$ can be decomposed as

$$p_a(x) = x + a = \frac{1}{4a}(x + 2a)^2 - \frac{1}{4a}x^2.$$

Thus $p_a \in M(K)$ for all $a > 0$.

Remark 2.1 *Given a compact semialgebraic set K, it apparently holds, any positive polynomial on K belongs to the cone $M(K)$ if and only if $M(K)$ is archimedian.*

Theorem 2.7 is called **Putinar's Positivstellensatz**. Obviously, it does not really characterize the polynomials positive on K since the polynomials in $M(K)$ must only be nonnegative on K. Also, it does not fully describe the polynomials nonnegative on K since they are not always contained in $M(K)$. However, it is Theorem 2.7 that is exploited by Lasserre in order to attempt the polynomial optimization problem.

2.1.3 Dense SDP relaxations for polynomial optimization problems

The idea to apply convex optimization techniques to solve polynomial optimization problems was first proposed in the pioneering work of Shor [91]. Shor introduced lower bounds for the global minimum of a polynomial function p. These bounds are derived by minimizing a quadratic function subject to quadratic constraints. Also Nesterov discussed the minimizaion of univariate polynomials and mentioned the problem of minimizing multivariate polynomials in [70]. It was Lasserre [51] who first realized the possibility to apply Putinar's Positivstellensatz, Theorem 2.7, to solve a broader class of polynomial optimization problems, that goes beyond the case where $p - p^\star$ can be described as sum of squares of polynomials.
At first, we introduce Lasserre's approach to derive semidefinite relaxations for minimizing a polynomial over a semialgebraic set, as Putinar's theorem is applied directly there. At second, we present the unconstrained case. Since semialgebraic sets enter through the backdoor, in order to be able to apply Putinar's Positivstellensatz, we present it after the constrained case.

Lasserre's relaxation in the constrained case

After studying positivity and nonnegativity of polynomials and the related problem of moments, we attempt the inital polynomial optimization problem (2.2) over a compact semialgebraic set K,

$$\min_{x \in K} p(x).$$

One of the major obstacles for finding the optimum $p^\star$ is the fact that the set K and the function p are far from being convex. The basic idea of Lasserre' s approach [51] is to convexify problem (2.2). We outline this procedure of convexification. It has to be emphasized that Lasserre's approach is based on two assumptions. First we require the semi-algebraic set K to be **compact**, and second we assume $M(K)$ is **archimedian**. These assumptions imply, we are able to apply Putinar's Positivstellensatz to polynomials positive on K.

At first we note,

$$p^\star = \sup\{a \in \mathbb{R} \mid p - a \geq 0 \text{ on } K\} = \sup\{a \in \mathbb{R} \mid p - a > 0 \text{ on } K\}. \tag{2.3}$$

Since we assume that $M(K)$ archimedian, we apply Theorem 2.7 to (2.3). Thus

$$p^\star \leq \sup\{a \in \mathbb{R} \mid p - a \in M(K)\} \leq \sup\{a \in \mathbb{R} \mid p - a \geq 0 \text{ on } K\} = p^\star.$$

Finally we obtain

$$p^\star = \sup\{a \in \mathbb{R} \mid p - a \in M(K)\}. \tag{2.4}$$

As a second approach, we note for the minimum $p^\star$ of (2.1) holds

$$p^\star = \inf\left\{\int p\, d\mu \mid \mu \in \mathcal{M}_P(K)\right\}, \tag{2.5}$$

where $\mathcal{M}_P(K) \subseteq \mathcal{M}(K)$ denotes the set of all Borel measures on K which are also probability measures. $'\leq'$ holds since $p(x) \geq p^\star$ on K implies $\int p\, d\mu \geq p^\star$. And $'\geq'$ follows as each x feasible in (2.1) corresponds to a $\mu = \delta_x \in \mathcal{M}(K)$, where δ_x the Dirac measure at x.
In order to get rid of the set $\mathcal{M}(K)$ in (2.5) we exploit the following theorem by Putinar [81].

Theorem 2.9 *For any map $L: \mathbb{R}[x] \to \mathbb{R}$, the following are equivalent:*

(i) *L is linear, $L(1) = 1$ and $L(M(K)) \subseteq [0, \infty)$.*

(ii) *L is integration with respect to a probability measure μ on K, i.e.,*

$$\exists\, \mu \in \mathcal{M}_P(K):\ \forall\, p \in \mathbb{R}[x]:\ L(p) = \int p\, d\mu.$$

Proof C.f. [88], pp. 10.
This theorem does not really characterize $\mathcal{M}_P(K)$, but all real families $(y_\alpha)_{\alpha \in \mathbb{N}^n}$ that are sequences of moments of probability measures on K, i.e.,

$$y_\alpha = \int x^\alpha\, d\mu \qquad \forall\, \alpha \in \mathbb{N}^n,$$

where $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. This statement is true, as every linear map $L: \mathbb{R}[x] \to \mathbb{R}$ is given uniquely by its values $L(x^\alpha)$ on the basis $(x^\alpha)_{\alpha \in \mathbb{N}^n}$ of $\mathbb{R}[x]$. With Theorem 2.9 we obtain

$$p^\star = \inf\{L(f) \mid L: \mathbb{R}[x] \to \mathbb{R} \text{ is linear },\ L(1) = 1,\ L(M(K)) \subseteq [0, \infty)\}. \tag{2.6}$$

Recall (2.4) as

$$p^\star = \sup\{a \in \mathbb{R} \mid f - a \in M(K)\}.$$

Thus (2.6) can be understood as a primal approach to the original problem (2.1) and (2.4) as a dual approach. Due to complexity reasons it is necessary to introduce relaxations to these primal-dual pair of optimization problems, in order to solve the problem (2.1). Therefore we approximate $M(K)$ by the sets $M_\omega(K) \subseteq \mathbb{R}[x]$, where $M_\omega(K) := \left\{\sum_{i=0}^m \sigma_i g_i \mid \sigma_i \in \sum \mathbb{R}[x]^2, \deg(\sigma_i g_i) \leq 2\omega\right\}$ for an

$$\omega \in \mathcal{N} := \{s \in \mathbb{N} \mid s \geq \omega_{\max} := \max\{\omega_0, \omega_1, \ldots, \omega_m\}\},$$

$\omega_i := \lceil \frac{\deg g_i}{2} \rceil$ $(i = 1, \ldots, m)$, $\omega_0 := \lceil \frac{\deg p}{2} \rceil$. Replacing $M(K)$ by $M_\omega(K)$ motivates to consider the following pair of optimization problems for a $\omega \in \mathcal{N}$:

$$\begin{array}{llllll} (P_\omega) & \min & L(p) & \text{subject to} & L: \mathbb{R}[x]_{2\omega} \to \mathbb{R} \text{ is linear}, \\ & & & & L(1) = 1 \text{ and} \\ & & & & L(M_\omega(K)) \subseteq [0, \infty). \\ (D_\omega) & \max & a & \text{subject to} & a \in \mathbb{R} \text{ and} \\ & & & & p - a \in M_\omega(K). \end{array} \tag{2.7}$$

The optimal values of (P_ω) and (D_ω) are denoted by $P_\omega^\star$ and $D_\omega^\star$, respectively. The parameter $\omega \in \mathcal{N}$ is called the **relaxation order** of (2.7). It determines the size of the relaxations (P_ω) and (D_ω) to (2.2) and therefore also the numerical effort that is necessary to solve them.

Theorem 2.10 (Lasserre) *Assume $M(K)$ is archimedian. $(P_\omega^\star)_{\omega\in\mathcal{N}}$ and $(D_\omega^\star)_{\omega\in\mathcal{N}}$ are increasing sequences that converge to $p^\star$ and satisfy $D_\omega^\star \leq P_\omega^\star \leq p^\star$ for all $\omega \in \mathcal{N}$. Moreover, if $p - p^\star \in M(K)$, then $D_\omega^\star = P_\omega^\star = p^\star$ for a sufficiently large relaxation order ω, i.e. strong duality holds.*

Proof Since the feasible set of (2.6) is a subset of the feasible set of (P_ω), $P_\omega^\star \leq p^\star$. Moreover, if L feasible for (P_ω) and a for (D_ω), $L(p) \geq a$ holds since $p-a \in M_\omega(K)$ implies $L(p)-a = L(p)-aL(1) = L(p-a) \geq 0$. Thus $D_\omega^\star \leq P_\omega^\star$. Obviously, a feasible a for (D_ω) is feasible for $(D_{\omega+1})$, and every feasible L of $(P_{\omega+1})$ is feasible for (P_ω). This implies $(P_\omega^\star)_{\omega\in\mathcal{N}}$ and $(D_\omega^\star)_{\omega\in\mathcal{N}}$ are increasing. Furthermore, as for any $\epsilon > 0$ there exists a sufficiently large $\omega \in \mathcal{N}$ such that $p - p^\star + \epsilon \in M_\omega(K)$ by Theorem 2.7, i.e. $p^\star - \epsilon$ feasible for (D_ω), the convergence follows. If $p - p^\star \in M(K)$, $p - p^\star \in M_\omega(K)$ for ω sufficiently large. Thus $p^\star$ feasible for (D_ω) und therefore $D_\omega^\star = P_\omega^\star = p^\star$. □

If $M(K)$ not archimedian, we are still able to exploit Schmuedgen's Positivstellensatz to characterize $p - a$ in (D_ω). As a next step we follow Lasserre's observation and translate (D_ω) and (P_ω) into a pair of primal-dual semidefinite programs.

Definition 2.3 *Let $L : \mathbb{R}[x] \to \mathbb{R}$ be linear functional, a sequence $y = (y_\alpha)_{\alpha\in\mathbb{N}^n}$ be given by*

$$y_\alpha := L(x^\alpha),$$

and $d \in \mathbb{N}$ fixed. The moment matrix $M_d(y)$ of order d is the matrix with rows and columns indexed by $u(x, \Lambda(d))$, such that

$$M_d(y)_{\alpha,\beta} = L(x^\alpha x^\beta) = y_{\alpha+\beta} \quad \forall \alpha, \beta \in \mathbb{N}^n \text{ with } \mid \alpha \mid, \mid \beta \mid \leq d.$$

The size of $M_d(y)$ is given by the $\mid u(x, \Lambda(d)) \mid = \begin{pmatrix} n+d \\ d \end{pmatrix}$, the number of components of y needed for constructing $M_d(y)$ is given by

$$\mid (y_\alpha)_{\mid\alpha\mid\leq 2d} \mid = \begin{pmatrix} n+2d \\ 2d \end{pmatrix}.$$

In the case $n = d = 2$, the moment matrix is given by

$$M_2(y) \quad = \begin{pmatrix} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{pmatrix}.$$

Let $g \in \mathbb{R}[x]$ with $g(x) = \sum_{\alpha\in\mathbb{N}^n} g_\alpha x^\alpha$. The localizing matrix $M_d(y)$ of order d associated with g and y is the matrix with rows and columns indexed in $u_d(x)$, obtained from the moment matrix by

$$M_d(g\,y)_{\alpha,\beta} := L(g(x)x^\alpha x^\beta) = \sum_{\gamma\in\mathbb{N}^n} h_\gamma y_{\gamma+\alpha+\beta} \quad \forall \alpha, \beta \in \mathbb{N}^n, \mid \alpha \mid, \mid \beta \mid \leq d.$$

For $g(x) = x_1^2 + 2x_2 + 3$, $n = 2$ and $d = 1$, the localizing matrix is given by

$$M_1(g\,y) \quad = \begin{pmatrix} y_{20} + 2y_{01} + 3y_{00} & y_{30} + 2y_{11} + 3y_{10} & y_{21} + 2y_{02} + 3y_{01} \\ y_{30} + 2y_{11} + 3y_{10} & y_{40} + 2y_{21} + 3y_{20} & y_{31} + 2y_{12} + 3y_{11} \\ y_{21} + 2y_{02} + 3y_{01} & y_{31} + 2y_{12} + 3y_{11} & y_{22} + 2y_{03} + 3y_{02}. \end{pmatrix}$$

We will exploit the following key lemma [88].

Lemma 2.1 *Suppose $L : \mathbb{R}[x] \to \mathbb{R}$ is a linear map. Then $L(M_\omega) \subseteq [0, \infty)$ if and only if the $m+1$ matrices*

$$M_{\omega-\omega_i}(y\,g_i) \succcurlyeq 0, \qquad \forall\, i \in \{0, \ldots, m\},$$

where $g_0 := 1$, $\omega_0 = 0$ *and the sequence* y *defined by* $y_\alpha := L(x^\alpha)$. *Moreover,*

$$M_\omega(K) = \left\{ \sum_{i=0}^{m} \langle M_{\omega-\omega_i}(u_{2\omega-2\omega_i}(x)\, g_i), G_i\rangle \mid G_0, \ldots, G_m \in \mathbb{S}_+^{s(\omega-\omega_i)} \right\}.$$

Proof C.f. [88], p. 19.
Using this Lemma, we reformulate (2.7) as

$$\begin{array}{lll} \mathrm{dSDP}_\omega & \min & \sum_{\alpha\in\Lambda(2\omega)} y_\alpha p_\alpha \\ & \text{s.t.} & y \in \mathbb{R}^{|\Lambda(2\omega)|},\ y_0 = 1 \text{ and} \\ & & M_{\omega-\omega_i}(y\, g_i) \succcurlyeq 0,\ i = 1, \ldots, m \\ & & M_\omega(y) \succcurlyeq 0, \\ \mathrm{dSDP}_\omega^\star & \max & a \\ & \text{s.t.} & a \in \mathbb{R},\ G_0 \in \mathbb{S}_+^{s(\omega)}, G_i \in \mathbb{S}_+^{s(\omega-\omega_i)} \text{ for } i = 1, \ldots, m \text{ and} \\ & & \sum_{i=0}^{m} \langle M_{\omega-\omega_i}(u(x, \Lambda(2\omega) g_i), G_i\rangle = p - a, \end{array} \tag{2.8}$$

where $p(x) = \sum_{\alpha\in\Lambda(2\omega)} p_\alpha x^\alpha$. We call dSDP_ω the *dense Lasserre relaxation* or the *dense SDP relaxation* of relaxation order ω for the polynomial optimization problem (2.1). By sorting the monomials in the moment and localizing matrix inequality constraints in (2.8), we can express $M_\omega(u(x, \Lambda(2\omega)))$ and $M_{\omega-\omega_i}(u(x, \Lambda(2\Omega)\, g_i))$ as

$$M_\omega(u(x, \Lambda(2\omega))) = \sum_{\alpha\in\Lambda(2\omega)} B_\alpha x^\alpha, \qquad M_{\omega-\omega_i}(u(x, \Lambda(2\omega - 2\omega_i))\, g_i) = \sum_{\alpha\in\Lambda(2\omega)} C_{\alpha i} x^\alpha,$$

for some matrices $B_\alpha \in \mathbb{S}^{s(\omega)}$ and $C_{\alpha i} \in \mathbb{S}^{s(\omega-\omega_i)}$. Thus we can rewrite the primal-dual pair of SDP (2.8) as the primal-dual pair of equivalent SDP in standard form

$$\begin{array}{lll} (P_\omega^{\mathrm{SDP}}) & \min & \sum_{\alpha\in\Lambda(2\omega)} p_\alpha y_\alpha \\ & \text{s.t.} & y \in \mathbb{R}^{|\Lambda(2\omega)|},\ y_0 = 1, \text{ and} \\ & & B_0 + \sum_{\alpha\in\Lambda(2\omega)\setminus\{0\}} y_\alpha B_\alpha \succcurlyeq 0, \\ & & C_{0i} + \sum_{\alpha\in\Lambda(2\omega)\setminus\{0\}} y_\alpha C_{\alpha i} \succcurlyeq 0,\ i = 1, \ldots, m \\ (D_\omega^{\mathrm{SDP}}) & \max & -G_0(1,1) - \sum_{i=1}^{m} \langle C_{0i}, G_i\rangle \\ & \text{s.t.} & a \in \mathbb{R},\ G_0 \in \mathbb{S}_+^{s(\omega)},\ G_i \in \mathbb{S}_+^{s(\omega-\omega_i)} \text{ for } i \in \{1, \ldots, m\} \text{ and} \\ & & \langle B_\alpha, G_0\rangle + \sum_{i=1}^{m} \langle C_{\alpha i}, G_i\rangle = p_\alpha,\ 0 \neq \alpha \in \Lambda(2\omega) \end{array} \tag{2.9}$$

In general SDP can be solved in polynomial time. Efficient solvers for the SDP (2.9) in standard form are provided by the software packages SeDuMi [95] and SDPA [104].

Lasserre's relaxation in the unconstrained case

The procedure to derive a sequence of convergent SDP relaxations in the case of an unconstrained polynomial optimization problem

$$\min_{x\in\mathbb{R}^n} p(x), \tag{2.10}$$

where $p \in \mathbb{R}[x]$ and $p^\star := \min_x p(x)$, is similar to the constrained case, which we discussed in the previous subsection. Let p be of even degree $2l$, otherwise $\inf p = -\infty$. Moreover, we will exploit the characterization of sum of squares decompositions by semidefinite matrices and Putinar's Positivstellensatz. In order to apply this theorem, it is necessary to construct an appropriate semialgebraic set.
First, we derive the following relaxation,

$$\begin{array}{ll} p^\star & = \inf \left\{ \int p\, d\mu \mid \mu \in \mathcal{M}_P(\mathbb{R}^n) \right\} \\ & \geq \inf \left\{ L(p) \mid L : \mathbb{R}[x] \to \mathbb{R},\ L(1) = 1,\ M_l(L(x)) \in \mathbb{S}_+^{s(l)} \right\}. \end{array} \tag{2.11}$$

We order the expression $M_l(L(x))$ and introduce symmetric matrices $B_\alpha \in \mathbb{S}^{s(l)}$ such that $M_l(L(x)) = \sum_{\alpha\in\Lambda(2l)} B_\alpha L(x^\alpha)$. Finally we identify $y_\alpha = L(x^\alpha)$ for $\alpha \in \Lambda(2l)\setminus\{0\}$ and $y_0 = 1$ to obtain the following relaxation for (2.10)

$$\begin{array}{lll} (P_l) & \min & \sum_\alpha p_\alpha y_\alpha \\ & \text{s.t.} & \sum_{\alpha\neq 0} y_\alpha B_\alpha \succcurlyeq -B_0. \end{array} \tag{2.12}$$

As in the constrained case we can apply a dual approach to (2.10),

$$\begin{array}{ll} p^\star & = \sup\{a\in\mathbb{R} \mid p(x)-a\geq 0\ \forall\, x\in\mathbb{R}^n\} \geq \sup\left\{a\in\mathbb{R} \mid p(x)-a\in\sum\mathbb{R}[x]^2\right\} \\ & = \sup\left\{a \mid p(x)-a = \langle M_l(x), G\rangle,\ G\in\mathbb{S}_+^{s(l)}\right\}. \end{array} \tag{2.13}$$

Thus, we derive another relaxation to problem (2.10),

$$\begin{array}{lll} (D_l) & \max & -G(1,1) \\ & \text{s.t.} & \langle B_\alpha, G\rangle = p_\alpha, \quad \alpha\neq 0 \\ & & G\succcurlyeq 0. \end{array} \tag{2.14}$$

With the duality theory of convex optimization it can be shown easily, that the two convex programs (2.12) and (2.14) are dual to each other. In the case (2.14) has an interior feasible solution, strong duality holds, that is

$$P_l^\star = D_l^\star.$$

The idea of the following theorem was proposed by Shor [91] first. The presented version is due to Lasserre [51].

Theorem 2.11 (Shor) *If the nonnegative polynomial $p-p^\star$ is a sum of squares of polynomials, then (2.10) is equivalent to (2.12). More precisely, $p^\star = Z_P$ and, if $x^\star$ is a global minimizer of (2.10), then*

$$y^\star := \left(x_1^\star,\ldots,x_n^\star,(x_1^\star)^2,x_1^\star x_2^\star,\ldots,(x_1^\star)^{2m},\ldots,(x_n^\star)^{2m}\right)$$

is a minimizer of (2.12).

Next, we treat the **general case**, that is, when $p-p^\star$ is not sum of squares. As mentioned at the beginning we have to construct a semialgebraic set in order to be able to apply Putinar's Positivstellensatz. Suppose we know that a global minimizer $x^\star$ of $p(x)$ has norm less than a for some $a>0$, that is, $p(x^\star)=p^\star$ and $||\, x^\star\,||_2 \leq a$. Then, with $x\to q_a(x) = a^2 - ||\, x\,||_2^2$, we have $p(x)-p^\star\geq 0$ on $K_a := \{q_a(x)\geq 0\}$. Obviously, $M(K_a)$ is archimedian, as the condition (iii) in Theorem 2.8 is satisfied for $N=a^2$. Now, we can use that every polynomial f, strictly positive on the semialgebraic set K_a is contained in the quadratic module $M(K_a)$.
For every $\omega\geq l$, consider the following semidefinite program

$$\begin{array}{lll} (P_\omega^a) & \min & \sum_\alpha p_\alpha y_\alpha, \\ & \text{s.t.} & M_\omega(y) \succcurlyeq 0, \\ & & M_{\omega-1}(q_a y) \geq 0. \end{array} \tag{2.15}$$

Writing $M_{\omega-1}(q_a y) = \sum_\alpha y_\alpha D_\alpha$, for appropriate matrices D_α ($|\,\alpha\,|\leq 2\omega$), the dual of (P_ω^a) is the semidefinite program

$$\begin{array}{lll} (D_\omega^a) & \max & -G(1,1) - a^2 H(1,1), \\ & \text{s.t.} & \langle G, B_\alpha\rangle + \langle H, D_\alpha\rangle = p_\alpha,\ \alpha\neq 0. \end{array} \tag{2.16}$$

Then, the following theorem is due to Lasserre [51].

Theorem 2.12 (Lasserre) *Given (P_ω^a) and (D_ω^a) for some $a>0$ such that $||\, x^\star\,||_2\leq a$ for some global minimizer $x^\star$. Then*

(a) as $\omega \to \infty$, one has

$$\inf(P_\omega^a) \uparrow p^\star.$$

Moreover, for ω sufficiently large, there is no duality gap between (P_ω^a) and its dual (D_ω^a), and (D_ω^a) is solvable.

(b) $\min(P_\omega^a) = p^\star$ *if and only if* $p - p^\star \in M_\omega(K_a)$. *In this case, the vector*

$$y^\star := \left(x_1^\star, \ldots, x_n^\star, (x_1^\star)^2, x_1^\star x_2^\star, \ldots, (x_1^\star)^{2\omega}, \ldots, (x_n^\star)^{2\omega}\right)$$

is a minimizer of (P_ω^a). In addition, $\max(P_\omega^a) = \min(D_\omega^a)$.

Proof

(a) From $x^\star \in K_a$ and with

$$y^\star := \left(x_1^\star, \ldots, x_n^\star, (x_1^\star)^2, x_1^\star x_2^\star, \ldots, (x_1^\star)^{2\omega}, \ldots, (x_n^\star)^{2\omega}\right)$$

it follows that $M_\omega(y^\star)$, $M_{\omega-1}(q_a y^\star) \succcurlyeq 0$ so that $y^\star$ is feasible for (P_ω^a) and thus $\inf(P_\omega^a) \leq p^\star$.
Now, fix $\epsilon > 0$ arbitrary. Then, $p - p^\star + \epsilon > 0$ and therefore, with Theorem 2.7 there is some N_0 such that

$$p - p^\star + \epsilon = \sum_{i=1}^{r_1} q_i(x)^2 + q(x) \sum_{j=1}^{r_2} t_j(x)^2$$

for some polynomials $q_i(x)$, $i = 1, \ldots, r_1$, of degree at most N_0, and some polynomials $t_j(x)$, $j = 1, \ldots, r_2$, of degree at most $N_0 - 1$. Let $q_i \in \mathbb{R}^{s(N_0)}, t_j \in \mathbb{R}^{s(N_0-1)}$ be the corresponding vectors of coefficients, and let

$$G := \sum_{i=1}^{r_1} q_i q_i^T, \quad Z := \sum_{j=1}^{r_2} t_j t_j^T$$

so that $G, H \succcurlyeq 0$. It is immediate to check that (G, H) feasible for (D_ω^a) with value $-G(1,1) - a^2 H(1,1) = (p^\star - \epsilon)$. From weak duality follows convergence as

$$p^\star - \epsilon \leq \inf(P_\omega^a) \leq p^\star.$$

For strong duality and for (b), c.f. [51]. $\square$
We needed to add the constraint $q_a(x) \geq 0$, in order to show convergence of the SDP relaxation (P_ω^a). For applications it is often sufficient to consider an SDP relaxation, which does not take into account this constraint. Thus, we denote the primal-dual pair of SDP

$$\begin{array}{lll} \text{dSDP}_\omega & \min & \sum_\alpha p_\alpha y_\alpha \\ & \text{s.t.} & M_\omega(y) \succcurlyeq 0, \\ \text{dSDP}_\omega^\star & \max & -G(1,1) \\ & \text{s.t.} & \langle G, B_\alpha \rangle = p_\alpha \quad \forall \alpha \neq 0, \end{array}$$

as the *dense SDP relaxation* of relaxation order ω for polynomial optimization problem (2.10), which is consistent with the dense SDP relaxation for the constrained case. This sequence of SDP is not guaranteed to converge to the minimum of (2.10) for $\omega \to \infty$. However, it provides a non-decreasing sequence of lower bounds to $p^\star$,

$$\min(\text{dSDP}_{\omega_{\max}}) \leq \min(\text{dSDP}_{\omega_{\max}+1}) \leq \ldots \leq p^\star.$$

Global minimizer

Usually one is not only interested in finding the minimum value $p^\star$ of p on K, but also in obtaining a global minimizer $x^\star \in K^\star$ with $p(x^\star) = p^\star$. It will be shown that in Lasserre's procedure not only $\min(\mathrm{dSDP}_\omega)$ converges to the infimum $p^\star$, but also a convergence to the minimizer $x^\star$ of (2.2) in the case it is unique.

Definition 2.4 *L_ω solves (P_ω) nearly to optimality $(\omega \in \mathcal{N})$ if L_ω is a feasible solution of (P_ω) $(\omega \in \mathcal{N})$ such that $\lim_{\omega\to\infty} L_\omega(p) = \lim_{\omega\to\infty} P_\omega^\star$.*

This notation is useful because (P_ω) might not possess an optimal solution, and even if it does, we might not be able to compute it exactly. For an example, c.f. [88], Example 22. Obviously L_ω solves (P_ω) nearly to optimality $(\omega \in \mathcal{N})$ if and only if $\lim_{\omega\to\infty} L_\omega(f) = p^\star$. The following theorem is the basis for the convergence to a minimizer in the case $K^\star$ is a singleton.

Theorem 2.13 *Suppose $K \neq \emptyset$ and L_ω solves (P_ω) nearly to optimality $(\omega \in \mathcal{N})$. Then*

$$\forall d \in \mathbb{N} : \ \forall \epsilon > 0 : \exists k_0 \in \mathcal{N} \cap [d, \infty) : \forall k \geq k_0 : \exists \mu \in \mathcal{M}(K^\star) : \left|\left| \left(L_\omega(x^\alpha) - \int x^\alpha \, d\mu \right)_{\alpha \in \Lambda(2d)} \right|\right| < \epsilon.$$

Proof [88], p. 11.
In the convenient case where $K^\star$ is a singleton it is possible to guarantee convergence of the minimizer:

Corollary 2.2 *$K^\star = \{x^\star\}$ is a singleton and L_ω solves (P_ω) nearly to optimality $(\omega \in \mathcal{N})$. Then*

$$\lim_{\omega\to\infty} (L_\omega(x_1), \ldots, L_\omega(x_n)) = x^\star.$$

Proof We set $d = 1$ in Theorem 2.13 and note that $\mathcal{M}(K^\star)$ contains only the Dirac measure $\delta_{x^\star}$ at the point $x^\star$. It is possible to apply Corollary 2.2 to certify that $p^\star$ has almost been reached after successively solving the relaxations (P_ω).

Corollary 2.3 *Suppose $M(K)$ is archimedian, p has a unique minimizer on the compact semialgebraic set K and L_ω solves (P_ω) nearly to optimality for all $\omega \in \mathcal{N}$. Then holds for all $\omega \in \mathcal{N}$,*

$$L_\omega(p) \leq p^\star \leq p(L_\omega(x_1), \ldots, L_\omega(x_n)),$$

and the lower and upper bounds for $p^\star$ converge to $p^\star$ for $\omega \to \infty$.

Proof $L_\omega(p) \leq p^\star$ follows from Theorem 2.10. The convergence of $p(L_\omega(x_1), \ldots, L_\omega(x_n))$ is a consequence of Corollary 2.2. To see that $p^\star$ is a lower bound, observe that

$$g_i\left(L_\omega(x_1), \ldots, L_\omega(x_n)\right) = L_\omega(g_i) \geq 0,$$

whence $(L_\omega(x_1), \ldots, L_\omega(x_n)) \in K$ for all $k \in \mathcal{N}$. $\square$
The case where several optimal solutions exist is more difficult to handle. In fact, as soon as there are two or more global minimizers, it often occurs that symmetry in the problem prevents the nearly optimal solutions of the SDP relaxations to converge to a particular minimizer.
Henrion and Lasserre established a sufficient condition for the dense SDP relaxations to detect all optimal solutions [35]. Given the dense SDP relaxation dSDP_ω for some order $\omega \geq \omega_{\max}$ and $y^\star$ an optimal solution of this semidefinite program. In the case

$$\mathrm{rank} M_\omega(y^\star) = \mathrm{rank} M_{\omega-\omega_{\max}}(y^\star) \tag{2.17}$$

holds, the SDP relaxation dSDP_ω is exact. That is, $\min(\mathrm{dSDP}_\omega) = p^\star$. Moreover, Henrion and Lasserre provided an algorithm for extracting all global optimal solutions of the POP (2.1), if (2.17) holds. See [35] for details. Note, (2.17) is not necessary, dSDP_ω may be exact already for some relaxation order ω with $\mathrm{rank} M_\omega(y_\omega^\star) > \mathrm{rank} M_{\omega-\omega_{\max}}(y_\omega^\star)$. For many POP it may not be practical to increase ω until (2.17) holds, as the size of the moment and localizing matrix constraints in dSDP_ω given by $\begin{pmatrix} n+\omega \\ n \end{pmatrix}$ grows rapidly for increasing ω.

2.1.4 Sparse SDP relaxations for polynomial optimization problems

The dense SDP relaxation (2.8) by Lasserre is a powerful theoretical result since it allows to approximate the solutions of polynomial optimization problems (2.1) as closely as desired by solving a finite sequene of SDP relaxations. However, since the size of the SDP relaxation grows as $\binom{n+\omega}{\omega}$, even for medium scale POPs the SDP relaxations become intractable for present SDP solvers in the case of small choices of the relaxation order ω. Therefore, it is crucial to reduce the size of the semidefinite programs to be solved, in order to be able to attempt large scale POPs. In this section we will review the approach [102] to exploit sparsity in a large scale POP by introducing a sequence of sparse SDP relaxations, which is of much smaller size than the dense SDP relaxations (2.8). A second method to exploit sparsity in a general optimization problem with linear and/ or nonlinear matrix inequality constraints is presented in [42].
In many problems of type (2.1), the involved polynomials $p,\ g_1, \ldots, g_m$ are sparse. Waki, Kojima, Kim and Muramatsu constructed a sequence of SDP relaxations which exploits the sparsity of such polynomial optimization problems [102]. This method shows strong numerical efforts in comparision to Lasserre's relaxations (2.8). The convergence of the sparse SDP relaxations to the optimum of the original problem (2.1) was shown by Lasserre [52] and Kojima and Muramatsu [49]. In the following, we give a review of the sparse SDP relaxations for POP with structured sparsity.

Let the polynomial optimization problem be given as in (2.2),

$$\min_{x \in K} p(x),$$

where K is a compact semialgebraic set defined by the m inequality constraints $g_1 \geq 0, \ldots, g_m \geq 0$. We characterize sparsity for a POP (2.2) with the following definition.

Definition 2.5 *Given a POP (2.2), we denote the $n \times n$ symbolic matrix R defined by*

$$R_{i,j} = \begin{cases} \star, & \text{if } x_i x_j \text{ occurs in some monomial of } p, \\ \star, & \text{if } x_i \text{ and } x_j \text{ occur in the same } g_l\ (l = 1, \ldots, m), \\ 0, & \text{else}, \end{cases}$$

*as the **correlative sparsity pattern matrix** of the POP. The graph $G = (V, N)$ with vertex set $V := \{1, \ldots, n\}$ and edge set*

$$N := \left\{ \{i, j\} \in V^2 \mid R_{i,j} = \star \right\},$$

*is called the corresponding **correlative sparsity pattern graph**. A POP is defined to be **correlatively sparse**, if R is sparse.*

We will construct a sequence of SDP relaxations to this polynomial optimization problem, which exploits the sparsity pattern characterized by the correlative sparsity pattern matrix R. Under a certain condition on the sparsity pattern of the problem, the optima of these SDP relaxations converge to the optimum of the polynomial optimization problem (2.2).
First, let $\{1, \ldots, n\}$ be the union $\cup_{k=1}^{q} I_k$ of subsets $I_k \subset \{1, \ldots, n\}$, such that every $g_j,\ j \in \{1, \ldots, m\}$ is only concerned with variables $\{x_i \mid i \in I_k\}$ for some k. And it is required the objective p can be written as $p = p_1 + \ldots + p_q$ where each p_k uses only variables $\{x_i \mid i \in I_k\}$. A possible choice for the sets $I_1, \ldots, I_p$ are the maximal cliques of the correlative sparsity pattern graph G. In order to tackle the sparse SDP relaxations we need some further definitions.

Definition 2.6 *Given a subset I of $\{1, \ldots, n\}$ we define the sets*

$$\begin{aligned} \mathcal{A}^I &= \{\alpha \in \mathbb{N}^n :\ \alpha_i = 0 \text{ if } i \notin I\}, \\ \mathcal{A}^I_\omega &= \left\{\alpha \in \mathbb{N}^n :\ \alpha_i = 0 \text{ if } i \notin I \text{ and } \textstyle\sum_{i \in I} \alpha_i \leq \omega\right\}. \end{aligned}$$

*Then, we define $\mathbb{R}[x, \mathcal{G}] := \{f \in \mathbb{R}[x] :\ supp(f) \subseteq \mathcal{G}\}$. Also, the **restricted moment matrix** $M_r(y, I)$ **and localizing matrices** $M_r(gy, I)$ are defined for $I \subseteq \{1, \ldots, n\}$, $r \in \mathbb{N}$ and $g \in \mathbb{R}[x]$. They are*

obtained from $M_r(y)$ and $M_r(gy)$ by retaining only those rows (and columns) $\alpha \in \mathbb{N}^n$ of $M_r(y)$ and $M_r(gy)$ with supp$(\alpha) \subseteq \mathcal{A}_r^I$. In doing so, $M_r(y, I)$ and $M_r(gy, I)$ can be interpreted as moment and localizing matrices with rows and columns indexed in the canonical basis $u(x, \mathcal{A}_r^I)$ of $\mathbb{R}\left[x, \mathcal{A}_r^I\right]$. Finally, we denote the set of sum of square polynomials in $\mathbb{R}[x, \mathcal{G}]$ as $\sum \mathbb{R}[x, \mathcal{G}]^2$. In analogy to Theorem 2.3, $\sum \mathbb{R}[x, \mathcal{G}]^2$ can be written as

$$\sum \mathbb{R}[x, \mathcal{G}]^2 = \left\{u(x, \mathcal{G})^T V u(x, \mathcal{G}) : V \succcurlyeq 0\right\}.$$

Let m' be the number of inequality constraints which define the basic, closed, semialgebraic set K. In our initial setting $m' = m$, but later on $m' > m$ may hold, in the case we add further inequality constraints to restrict the set K. We introduce a condition for the index sets $I_1, \ldots, I_q$.

Assumption 1: Let $K \subseteq \mathbb{R}^n$ as in (2.23). The index set $J = \{1, \ldots, m'\}$ is partitioned into q disjoint sets J_k, $k = 1, \ldots, q$, and the collections $\{I_k\}$ and $\{J_k\}$ satisfy:

1. For every $j \in J_k$, $g_j \in \mathbb{R}\left[x, \mathcal{A}^{I_k}\right]$, that is, for every $j \in J_k$, the constraint $g_j(x) \geq 0$ is only concerned with the variables $x(I_k)$. Equivalently, viewing g_j as a polynomial in $\mathbb{R}[x]$, $g_{j\alpha} \neq 0 \Rightarrow \text{supp}(\alpha) \in \mathcal{A}^{I_k}$.

2. The objective function $p \in \mathbb{R}[x]$ can be written

$$p = \sum_{k=1}^{q} p_k, \quad \text{with } p_k \in \mathbb{R}\left[x, \mathcal{A}^{I_k}\right], \quad k = 1, \ldots, q.$$

   Equivalently, $f_\alpha \neq 0 \Rightarrow \text{supp}(\alpha) \in \cup_{k=1}^{q} \mathcal{A}^{I_k}$.

Example 2.4 *For $n = 6$ and $m = 6$, let*

$$g_1(x) = x_1x_2 - 1, \quad g_2(x) = x_1^2 + x_2x_3 - 1, \quad g_3(x) = x_2 + x_3^2x_4,$$

and

$$g_4(x) = x_3 + x_5, \quad g_5(x) = x_3x_6, \quad g_6(x) = x_2x_3.$$

Then we can construct $\{I_k\}$ and $\{J_k\}$ for $q = 4$ with

$$\begin{array}{llll} I_1 = \{1, 2, 3\}, & I_2 = \{2, 3, 4\}, & I_3 = \{3, 5\}, & I_4 = \{3, 6\}, \\ J_1 = \{1, 2, 6\}, & J_2 = \{3\}, & J_3 = \{4\}, & J_4 = \{5\}. \end{array}$$

Now, we can construct sparse SDP relaxations in analogy to the dense SDP relaxations (2.8). For each $j = 1, \ldots, m'$ write $\omega_j = \lceil \frac{\deg g_j}{2} \rceil$. Then, for some $\omega \in \mathcal{N}$ define the following semidefinite program

$$\begin{array}{lll} (\text{sSDP}_\omega) \quad \inf_y & \sum_\alpha p_\alpha y_\alpha & \\ \text{s.t.} & M_\omega(y, I_k) \succcurlyeq 0, & k = 1, \ldots, q, \\ & M_{\omega - \omega_j}(g_j y, I_k) \succcurlyeq 0, & j \in J_k;\ k = 1, \ldots, q, \\ & y_0 = 1. & \end{array} \tag{2.18}$$

Program (2.18) is well defined under Assumption 3, and it is easy to see, that it is an SDP relaxation of problem (2.2). In fact, it is also easy to see, that sSDP$_\omega$ is a weaker relaxation for (2.2) than dSDP$_\omega$, as the partial moment and localizing matrices in the constraints of (2.18) are minors of the full moment and localizing matrices in the constraints of (2.8), i.e.,

$$\min(\text{sSDP}_\omega) \leq \min(\text{dSDP}_\omega) \leq \min(\text{POP}) \qquad \forall \omega \in \mathcal{N}.$$

We call (2.18) the **sparse SDP relaxations** or **sparse Lasserre relaxations** for polynomials optimization problems. There are symmetric matrices $\left\{B_\alpha^k\right\}$ and $\left\{C_\alpha^{jk}\right\}$ such that

$$\begin{array}{ll} M_\omega(y, I_k) & = \sum_{\alpha \in \mathbb{N}^n} y_\alpha B_\alpha^k, \quad k = 1, \ldots, q, \\ M_{\omega - \omega_j}(g_j y, I_k) & = \sum_{\alpha \in \mathbb{N}^n} y_\alpha C_\alpha^{jk}, \quad k = 1, \ldots, q,\ j \in J_k, \end{array} \tag{2.19}$$

with $B_\alpha^k = 0$ and $C_\alpha^{jk} = 0$ whenever $\mathrm{supp}(\alpha) \notin \mathcal{A}^{I_k}$. Then we can rewrite (2.18) as

$$\begin{array}{ll} \inf_y & \sum_\alpha p_\alpha y_\alpha \\ \text{s.t.} & \sum\limits_{0\neq\alpha\in\mathbb{N}^n} y_\alpha B_\alpha^k \succcurlyeq -B_0^k, \quad k=1,\ldots,q, \\ & \sum\limits_{0\neq\alpha\in\mathbb{N}^n} y_\alpha C_\alpha^{jk} \succcurlyeq -C_0^{jk}, \quad j\in J_k;\ k=1,\ldots,q, \end{array} \tag{2.20}$$

and we derive the dual of this semidefinite program as

$$\begin{array}{lll} (\mathrm{sSDP}_\omega^\star) & \sup\limits_{Y_k,Z_{jk},\lambda} & \lambda \\ & \sum\limits_{k:\alpha\in\mathcal{A}^{I_k}} & \left[\langle Y_k, B_\alpha^k\rangle + \sum_{j\in J_k}\langle Z_{jk}, C_\alpha^{jk}\rangle\right] + \lambda\delta_{\alpha 0} = p_\alpha \quad \forall\ \alpha\in\Gamma_\omega, \\ & & Y_k, Z_{jk} \succcurlyeq 0, \qquad\qquad j\in J_k,\ k=1,\ldots,q, \end{array} \tag{2.21}$$

where $\Gamma_\omega := \left\{\alpha\in\mathbb{N}^n :\ \alpha\in\bigcup_{k=1}^q \mathcal{A}^{I_k};\ |\ \alpha\ |\leq 2\omega\right\}$.
The main advantage of the sparse SDP relaxations is the reduction of the size of the matrix inequality constraints. In order to understand the improved efficiency, let us compare the computational complexity of the dense relaxation (P_ω^{SDP}) and the sparse relaxation (P_ω^{sp}). The number of variables in (P_ω^{sp}) is bounded by $\sum_{k=1}^q \binom{n_k+2\omega}{\omega}$. Supposed $n_k \approx \frac{n}{q}$ for all k, the number of variables is bounded by $O(q(\frac{n}{q})^{2\omega})$, a strong improvement compared with $O(n^{2\omega})$, the number of variables in (P_ω^{SDP}). Also in (P_ω^{sp}) there are p LMI constraints of size $O((\frac{n}{q})^\omega)$ and $m+q$ LMI constraints of size $O((\frac{n}{q})^{\omega-\omega_{\max}})$, to be compared with a single LMI constraint of size $O(n^\omega)$ and m LMI constraints of size $O(n^{\omega-\omega_{\max}})$ in (P_ω^{SDP}).
As pointed out, the sparse SDP relaxations are weaker than the dense ones. The question arises, whether we still have convergence to the minimum of the POP. This question was answered positively by Lasserre [52]. We need two further conditions to show convergence.

Assumption 2: For all $k=1,\ldots,q-1$, $\quad I_{k+1}\cap\bigcup_{j=1}^k I_j \subseteq I_s$ for some $s\leq k$. (2.22)

The property of Assumption 2 is called the **running intersection property**. Note that (2.22) is always satsfied for $q=2$. Since property (2.22) depends on the ordering, it can be satisfied possibly after some relabelling of the $\{I_k\}$. In the case of Example 2.4 it is easy to check Assumption 2 is satisfied, but in general it is not obvious. However, Waki et al. [102] presented a general procedure to guarantee Assumption 2 is satisfied. Given $G=(V,E)$ the correlative sparsity pattern graph, we denote by $\tilde{G}=(V,\tilde{E})$ its chordal extension. A graph is said to be **chordal** if every (simple) cycle of the graph with more than three edges has a chord. A graph $G(V,\overline{E})$ is a chordal extension of $G(V,E)$ if it is a chordal graph and $E\subseteq\overline{E}$. See [4] for basic properties on choral graphs. Then, the maximal cliques $C_1,\ldots,C_q$ of $\tilde{G}$ satisfy the running intersection property, and the number q of maximal cliques in a chordal graph is bounded by n. Furthermore, there are efficient algorithms to determine the maximal cliques of a chordal graph, whereas it is NP-hard to determine the maximal cliques of an arbitrary graph.
Assumption 3: Let $K\subseteq\mathbb{R}^n$ be a closed semialgebraic set. Then, there is $M>0$ such that $||\ x\ ||_\infty < M$ for all $x\in K$.
This assumption implies $||\ x(I_k)\ ||_\infty^2 < n_k M^2$, $k=1,\ldots,q$, where $x(I_k) := \{x_i\mid i\in I_k\}$, and therefore we add to K the q redundant quadratic constraints

$$g_{m+k}(x) := n_k M^2 - ||\ x(I_k)\ ||\geq 0,\ k=1,\ldots,q,$$

and set $m'=m+q$, so that K is now defined by:

$$K := \{x\in\mathbb{R}^n\mid g_j(x)\geq 0,\ j=1,\ldots,m'\}. \tag{2.23}$$

Notice that $g_{m+k}\in\mathbb{R}\left[x,\mathcal{A}_2^{I_k}\right]$ for every $k=1,\ldots,q$. With Assumption 3, K is a **compact semialgebraic set**. Moreover, Assumption 3 is needed to guarantee the quadratic module $M(K)$ is archimedian, the condition of Putinar's Positivstellensatz. Finally, we obtain the following convergence result.

Theorem 2.14 *Let $p^\star$ denote the global minimum of (2.2) and let Assumption 1-3 hold. Then:*

(a) $\inf(sSDP_\omega) \uparrow p^\star$ *as* $\omega \to \infty$.

(b) If K has nonempty interior, then strong duality holds and $(sSDP_\omega^\star)$ solvable for sufficiently large ω, i.e., $\inf(sSDP_\omega) = \max(sSDP_\omega^\star)$.

(c) Let y^ω be a nearly optimal solution of $(sSDP_\omega)$, with e.g.

$$\sum_\alpha p_\alpha y_\alpha \le \inf(sSDP_\omega) + \frac{1}{\omega}, \quad \forall \omega \ge \omega_0,$$

and let $\hat{y}^\omega := \{y_\alpha^\omega : \mid \alpha \mid = 1\}$. If (2.2) has a unique global minimizer $x^\star \in K$, then $\hat{y}^\omega \to x^\star$ as $\omega \to \infty$.

Proof C.f. [52].

As in the dense case, it is also possible to extract global minimizers of the POP from the sparse SDP relaxations in certain cases where the minimizer of the POP is not unique. In fact, Lasserre derived the following sparse version of condition (2.17): Given $y^\star$ is an optimal solution for the sparse SDP relaxation sSDP_ω for some order $\omega \ge \omega_{\max}$. If the rank conditions,

$$\begin{array}{lll} \text{rank}M_\omega(y^\star, I_h) & = \text{rank}M_{\omega - a_h}(y^\star, I_h) & \forall\, h \in \{1, \ldots, q\}, \\ \text{rank}M_\omega(y^\star, I_h \cap I_{h'}) & = 1 & \forall h \ne h' \text{ with } I_h \cap I_{h'} \ne \emptyset, \end{array} \tag{2.24}$$

with $a_h := \max_{j \in J_h} \omega_j$, hold, then sSDP_ω is exact and all global minimizers can be extracted. However, (2.24) are very restrictive sufficient conditions for the SDP relaxations to be exact, and it is not practical to apply them to large scale POP in most cases.

The software SparsePOP [103] is an implementation of the sparse SDP relaxations. The running intersection property is guaranteed by choosing the maximal cliques of the chordal extension of the correlative sparsity pattern graph as the index sets $I_1, \ldots, I_q$ in (2.18). Instead of imposing the additional constraints of Assumption 3, in SparsePOP linear box constraints are imposed for each component of $x \in \mathbb{R}^n$,

$$\text{lbd}_i \le x_i \le \text{ubd}_i \qquad \forall\, i \in \{1, \ldots, n\}. \tag{2.25}$$

Moreover, SparsePOP adds small linear perturbation terms to the objective function of the POP, in order to enforce the POP to have a unique global minimizer.

2.2 Exploiting sparsity in linear and nonlinear matrix inequalities

Optimization problems with nonlinear matrix inequalities, including quadratic and polynomial matrix inequalities, are known as hard problems. They frequently belong to large-scale optimization problems. We present a basic framework for exploiting the sparsity characterized in terms of a chordal graph structure via positive semidefinite matrix completion [28]. Depending on where the sparsity is observed, two types of sparsities are studied: *the domain-space sparsity (d-space sparsity)* for a symmetric matrix X that appears as a variable in objective and/or constraint functions of a given optimization problem and is required to be positive semidefinite, and *the range-space sparsity (r-space sparsity)* for a matrix inequality involved in the constraint of the problem.

The d-space sparsity is basically equivalent to the sparsity studied by Fukuda et. al [21, 68] for an equality standard form SDP. One of the two d-space conversion methods proposed in this section corresponds to an extension of their conversion method, and the other d-space conversion method is an extension of the method used for the sparse SDP relaxation of polynomial optimization problems in [102, 103] and for the sparse SDP relaxation of a sensor network localization problem in [43].

The r-space sparsity concerns with a matrix inequality

$$M(y) \succcurlyeq 0, \tag{2.26}$$

involved in a general nonlinear optimization problem. Here M denotes a mapping from $\mathbb{R}^s$ into $\mathbb{S}^n$. If M is linear, (2.26) is known as a linear matrix inequality (LMI), which appears in the constraint of a dual standard form of SDP. If each element of $M(y)$ is a multivariate polynomial function in $y \in \mathbb{R}^s$, (2.26) is called a polynomial matrix inequality and the SDP relaxation [36, 37, 46, 48, 49, 52], which is an extension of the SDP relaxation [51] for POP, can be applied to (2.26). We assume a similar chordal graph structured sparsity as the d-space sparsity on the set of row and column index pairs (i, j) of the mapping M such that M_{ij} is not identically zero, i.e., $M_{ij}(y) \neq 0$ for some $y \in \mathbb{R}^s$. A representative example satisfying the r-space sparsity can be found with tridiagonal M. We do not impose any additional assumption on (2.26) to derive a r-space conversion method. When M is polynomial in $y \in \mathbb{R}^s$, we can effectively combine it with the sparse SDP relaxation method [46, 49] for polynomial optimization problems over symmetric cones to solve (2.26).

We propose two methods to exploit the r-space sparsity. One may be regarded as a dual of the d-space conversion method by Fukuda et. al [21]. More precisely, it exploits the sparsity of the mapping M in the range space via a dual of the positive semidefinite matrix completion to transform the matrix inequality (2.26) to a system of multiple matrix inequalities with smaller sizes and an auxiliary vector variable $z \in \mathbb{R}^q$. The resulting matrix inequality system is of the form

$$\widetilde{M}^k(y) - \widetilde{L}^k(z) \succcurlyeq 0 \ (k = 1, 2, \ldots, p), \tag{2.27}$$

and $y \in \mathbb{R}^s$ is a solution of (2.26) if and only if it satisfies (2.27) for some z. Here $\widetilde{M}^k$ denotes a mapping from $\mathbb{R}^s$ into the space of symmetric matrices with some size and $\widetilde{L}^k$ a linear mapping from $\mathbb{R}^q$ into the space of symmetric matrices with the same size. The sizes of symmetric matrix valued mappings $\widetilde{M}^k$ $(k = 1, 2, \ldots, p)$ and the dimension q of the auxiliary variable vector z are determined by the r-space sparsity pattern of M. For example, if M is tridiagonal, the sizes of $\widetilde{M}^k$ are all 2×2 and $q = n - 2$. The other r-space conversion method corresponds to a dual of the second d-space conversion method mentioned previously. We discuss how the d-space and r-space conversion methods enhance the correlative sparsity for POP introduced in the previous section. Furthermore, we present numerical results to demonstrate how the size of problems involving large scale matrix inequalities is reduced under the four proposed conversion methods.

2.2.1 An SDP example

A simple SDP example is shown to illustrate the two types of sparsities considered in this paper, the d-space sparsity and the r-space sparsity, and compare it to the correlative sparsity from 2.1.4 that characterizes the sparsity of the Schur complement matrix.

Let A^0 be a tridiagonal matrix in $\mathbb{S}^n$ such that $A^0_{ij} = 0$ if $|i - j| > 1$, and define a mapping M from $\mathbb{S}^n$ into $\mathbb{S}^n$ by

$$M(X) = \begin{pmatrix} 1 - X_{11} & 0 & 0 & \ldots & 0 & X_{12} \\ 0 & 1 - X_{22} & 0 & \ldots & 0 & X_{23} \\ 0 & 0 & \ddots & & 0 & X_{34} \\ \ldots & \ldots & \ldots & \ddots & \ldots & \ldots \\ 0 & 0 & 0 & & 1 - X_{n-1,n-1} & X_{n-1,n} \\ X_{21} & X_{32} & X_{43} & \ldots & X_{n,n-1} & 1 - X_{nn} \end{pmatrix}$$

for every $X \in \mathbb{S}^n$. Consider an SDP

$$\text{minimize } A^0 \bullet X \text{ subject to } M(X) \succcurlyeq 0, \ X \succcurlyeq 0. \tag{2.28}$$

Among the elements X_{ij} $(i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, n)$ of the matrix variable $X \in \mathbb{S}^n$, the elements X_{ij} with $|i - j| \leq 1$ are relevant and all other elements X_{ij} with $|i - j| > 1$ are unnecessary in evaluating the objective function $A^0 \bullet X$ and the matrix inequality $M(X) \succcurlyeq 0$. Hence, we can describe the d-sparsity

pattern as a symbolic tridiagonal matrix with the nonzero symbol $\star$

$$\begin{pmatrix} \star & \star & 0 & \ldots & 0 & 0 \\ \star & \star & \star & \ldots & 0 & 0 \\ 0 & \star & \star & \ddots & 0 & 0 \\ \ldots & \ldots & \ddots & \ddots & \ddots & \ldots \\ 0 & 0 & \ldots & \ddots & \star & \star \\ 0 & 0 & \ldots & \ldots & \star & \star \end{pmatrix}.$$

On the other hand, the r-space sparsity pattern is described as

$$\begin{pmatrix} \star & 0 & \ldots & 0 & \star \\ 0 & \star & \ldots & 0 & \star \\ \ldots & \ldots & \ddots & \ldots & \ldots \\ 0 & 0 & \ldots & \star & \star \\ \star & \star & \ldots & \star & \star \end{pmatrix}.$$

Applying the d-space conversion method using basis representation described in 2.2.3, and the r-space conversion method using clique trees presented in 2.2.5, we can reduce the SDP (2.28) to

$$\left.\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{n-1} \left(A^0_{ii} X_{ii} + 2A^0_{i,i+1} X_{i,i+1}\right) + A^0_{nn} X_{nn} \\ \text{subject to} & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} X_{11} & -X_{12} \\ -X_{21} & -z_1 \end{pmatrix} \succcurlyeq 0, \\ & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} X_{ii} & -X_{i,i+1} \\ -X_{i+1,i} & z_{i-1} - z_i \end{pmatrix} \succcurlyeq 0 \ (i = 2, 3, \ldots, n-2), \\ & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} X_{n-1,n-1} & -X_{n-1,n} \\ -X_{n,n-1} & X_{n,n} + z_{n-2} \end{pmatrix} \succcurlyeq 0, \\ & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} -X_{ii} & -X_{i,i+1} \\ -X_{i+1,i} & -X_{i+1,i+1} \end{pmatrix} \succcurlyeq 0 \ (i = 1, 2, \ldots, n-1). \end{array}\right\} \tag{2.29}$$

This problem has $(3n-3)$ real variables X_{ii} $(i = 1, 2, \ldots, n)$, $X_{i,i+1}$ $(i = 1, 2, \ldots, n-1)$ and z_i $(i = 1, 2, \ldots, n-2)$, and $(2n-1)$ linear matrix inequalities with size 2×2. Since the original SDP (2.28) involves an $n \times n$ matrix variable X and an $n \times n$ matrix inequality $M(X) \succcurlyeq 0$, we can expect to solve the SDP (2.29) much more efficiently than the SDP (2.28) as n becomes larger.

We can formulate both SDPs in terms of a dual standard form for SeDuMi [95]:

$$\text{maximize } b^T y \text{ subject to } c - A^T y \succcurlyeq 0,$$

where $b \in \mathbb{R}^l$, $A \in \mathbb{R}^{l \times m}$ and $c \in \mathbb{R}^m$ for some positive integers l and m. Table 2.2.1 shows numerical results on the SDPs (2.28) and (2.29) solved by SeDuMi. We observe that the SDP (2.29) greatly reduces the size of the coefficient matrix A, the number of nonzeros in A and the maximum SDP block compared to the original SDP (2.28). In addition, it should be emphasized that the $l \times l$ Schur complement matrix is sparse in the SDP (2.29) while it is fully dense in the the original SDP (2.28). As shown in Figure 2.1, the Schur complement matrix in the SDP (2.29) allows a very sparse Cholesky factorization. The sparsity of the Schur complement matrix is characterized by the correlative sparsity from 2.1.4. Notice *a hidden correlative sparsity* in the SDP (2.28), that is, each element X_{ij} of the matrix variable X appears at most once in the elements of $M(X)$. This leads to the correlative sparsity when the SDP (2.28) is decomposed into the SDP (2.29). The sparsity of the Schur complement matrix and the reduction in the size of matrix variable from 10000 to 2 are the main reasons that SeDuMi can solve the largest SDP in Table 1 with a 29997×79992 coefficient matrix A in less than 100 seconds.

	SeDuMi CPU time in seconds (sizeA, nnzA, maxBl, nnzSchur)	
n	the SDP (2.28)	the SDP (2.29)
10	0.2 (55×200,128,10,3025)	0.1 (27×72, 80,2,161)
100	1091.4 (5050×20000,10298,100,25502500)	0.6 (297×792,890,2,1871)
1000	OOM	6.3 (2997×7992,8990,2,18971)
10000	OOM	99.2 (29997×79992,89990,2,189971)

Table 2.1: Numerical results on the SDPs (2.28) and (2.29). Here sizeA denotes the size of the coefficient matrix A, nnzA the number of nonzero elements in A, maxBl the maximum SDP block size, and nnzSchur the number of nonzeros in the Schur complement matrix. OOM means out of memory error.

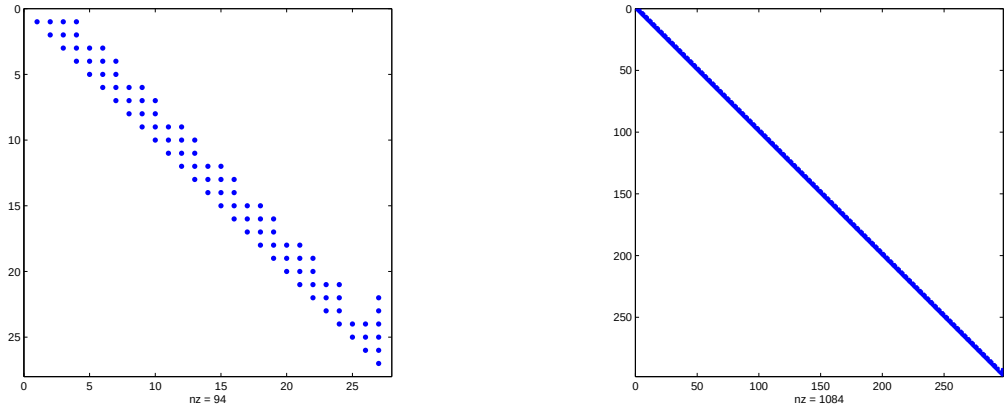


Figure 2.1: The sparsity pattern of the Cholesky factor of the Schur complement matrix for the SDP (2.29) with $n = 10$ and $n = 100$.

2.2.2 Positive semidefinite matrix completion

A problem of *positive semidefinite matrix completion* is: Given an $n \times n$ partial symmetric matrix X with entries specified in a proper subset F of $N \times N$, where $N = \{1, \ldots, n\}$, find an $\overline{X} \in \mathbb{S}^n_+$ satisfying $\overline{X}_{ij} = X_{ij}$ $((i, j) \in F)$ if it exists. If $\overline{X}$ is a solution of this problem, we say that X *is completed to a positive semidefinite symmetric matrix* $\overline{X}$. For example, the following 3×3 partial symmetric matrix

$$X = \begin{pmatrix} 3 & 3 & \\ 3 & 3 & 2 \\ & 2 & 2 \end{pmatrix}$$

is completed to a 3×3 positive semidefinite symmetric matrix

$$\overline{X} = \begin{pmatrix} 3 & 3 & 2 \\ 3 & 3 & 2 \\ 2 & 2 & 2 \end{pmatrix}.$$

For a class of problems of positive semidefinite matrix completion, we discuss the existence of a solution and its characterization in this section. This provides a theoretical basis for both d- and r-space conversion methods.

Let us use a graph $G(N, E)$ with the node set N and an edge set $E \subseteq N \times N$ to describe a class of $n \times n$ partial symmetric matrices. We assume that $(i, i) \notin E$, *i.e.*, the graph $G(N, E)$ has no loop. We also assume that if $(i, j) \in E$, then $(j, i) \in E$, and (i, j) and (j, i) are interchangeably identified. Define

$$\begin{aligned} E^\bullet &= E \cup \{(i,i) : i \in N\}, \\ \mathbb{S}^n(E,?) &= \text{the set of } n \times n \text{ partial symmetric matrices with entries} \\ &\quad \text{specified in } E^\bullet, \\ \mathbb{S}^n_+(E,?) &= \{X \in \mathbb{S}^n(E,?) : \exists \overline{X} \in \mathbb{S}^n_+; \overline{X}_{ij} = X_{ij} \text{ if } (i,j) \in E^\bullet\} \\ &\quad \text{(the set of } n \times n \text{ partial symmetric matrices with entries} \\ &\quad \text{specified in } E^\bullet \text{ that can be completed to positive} \\ &\quad \text{semidefinite symmetric matrices).} \end{aligned}$$

For a graph $G(N, E)$ shown in Figure 2.2 as an illustrative example, we have

$$\mathbb{S}^6(E,?) = \left\{ \begin{pmatrix} X_{11} & & & & & X_{16} \\ & X_{22} & & & & X_{26} \\ & & X_{33} & X_{34} & & X_{36} \\ & & X_{43} & X_{44} & X_{45} & \\ & & & X_{54} & X_{55} & X_{56} \\ X_{61} & X_{62} & X_{63} & & X_{65} & X_{66} \end{pmatrix} : X_{ij} \in \mathbb{R} \ \ (i,j) \in E^\bullet \right\}. \tag{2.30}$$

Let

$$\begin{aligned} \#C &= \text{the number of elements in } C \text{ for every } C \subseteq N, \\ \mathbb{S}^C &= \{X \in \mathbb{S}^n : X_{ij} = 0 \text{ if } (i,j) \notin C \times C\} \text{ for every } C \subseteq N, \\ \mathbb{S}^C_+ &= \{X \in \mathbb{S}^C : X \succcurlyeq 0\} \text{ for every } C \subseteq N, \\ X(C) &= \widetilde{X} \in \mathbb{S}^C \text{ such that } \widetilde{X}_{ij} = X_{ij} \ ((i,j) \in C \times C) \\ &\quad \text{for every } X \in \mathbb{S}^n \text{ and every } C \subseteq N, \\ J(C) &= \{(i,j) \in C \times C : 1 \le i \le j \le n\} \text{ for every } C \subseteq N. \end{aligned}$$

Note that $X \in \mathbb{S}^C$ is an $n \times n$ matrix although $X_{ij} = 0$ for every $(i, j) \notin C \times C$. Thus, $X \in \mathbb{S}^C$ and $X' \in \mathbb{S}^{C'}$ can be added even when C and C' are distinct subsets of N. When all matrices involved in an equality or a matrix inequality belong to $\mathbb{S}^C$, matrices in $\mathbb{S}^C$ are frequently identified with the $\#C \times \#C$ matrix whose elements are indexed with $(i, j) \in C \times C$. If $N = \{1, 2, 3\}$ and $C = \{1, 3\}$, then a matrix variable $X \in \mathbb{S}^C \subset \mathbb{S}^n$ has full and compact representations as follows:

$$X = \begin{pmatrix} X_{11} & 0 & X_{13} \\ 0 & 0 & 0 \\ X_{31} & 0 & X_{33} \end{pmatrix} \text{ and } X = \begin{pmatrix} X_{11} & X_{13} \\ X_{31} & X_{33} \end{pmatrix}.$$

It should be noted that $X \in \mathbb{S}^C \subset \mathbb{S}^n$ has elements X_{ij} with $(i, j) \in C \times C$ in the 2×2 compact representation on the right. Let

$$\begin{aligned} E_{ij} &= \text{the } n \times n \text{ symmetric matrix with 1 in } (i,j)\text{th and } (j,i)\text{th} \\ &\quad \text{elements and 0 elsewhere} \end{aligned}$$

for every $(i, j) \in N \times N$. Then E_{ij} $(1 \le i \le j \le n)$ form a basis of $\mathbb{S}^n$. Obviously, if $i,\ j \in C \subseteq N$, then $E_{ij} \in \mathbb{S}^C$. We also observe the identity

$$X(C) = \sum_{(i,j) \in J(C)} E_{ij} X_{ij} \text{ for every } C \subseteq N. \tag{2.31}$$

This identity is utilized in 2.2.3.

With these notations we can now state the result from matrix completion which forms the basis for our d-space and r-space conversion techniques. Let $G(N,E)$ be a graph and C_k $(k=1,\ldots,p)$ be its maximal cliques. We assume that $X \in \mathbb{S}^n(E,?)$. The condition $X(C_k) \in \mathbb{S}^{C_k}_+$ $(k=1,2,\ldots,p)$ is necessary for $X \in \mathbb{S}^n_+(E,?)$. For the graph $G(N,E)$ shown in Figure 2.2, the maximal cliques are $C_1=\{1,6\}$, $C_2=\{2,6\}$, $C_3=\{3,4\}$, $C_4=\{3,6\}$, $C_5=\{4,5\}$ and $C_6=\{5,6\}$. Hence, the necessary condition for $X \in \mathbb{S}^6(E,?)$ to be completed to a positive semidefinite matrix is that its 6 principal submatrices $X(C_k)$ $(k=1,2,\ldots,6)$ are positive semidefinite. Although this condition is not sufficient in general, it is a sufficient condition for $X \in \mathbb{S}^n_+(E,?)$ when $G(N,E)$ is chordal. As stated in 2.1.4, in this case, the number of the maximal cliques is bounded by the number of nodes of $G(N,E)$, *i.e.,* $p \le n$. In general we have the following result.

Lemma 2.2 *Let C_k $(k=1,2,\ldots,p)$ be the maximal cliques of a chordal graph $G(N,E)$. Suppose that $X \in \mathbb{S}^n(E,?)$. Then $X \in \mathbb{S}^n_+(E,?)$ if and only if $X(C_k) \in \mathbb{S}^{C_k}_+$ $(k=1,2,\ldots,p)$.*

Proof: C.f. [28].

Since the graph $G(N,E)$ in Figure 2.2 is not a chordal graph, we can not apply Lemma 2.2 to determine whether $X \in \mathbb{S}^6(E,?)$ of the form (2.30) belongs to $\mathbb{S}^6_+(E,?)$. In such a case, we need to introduce a chordal extension of the graph $G(N,E)$ to use the lemma effectively. Figure 2.3 shows two chordal extensions. If we choose the left graph as a chordal extension $G(N,\overline{E})$ of $G(N,E)$, the maximal cliques are $C_1=\{3,4,6\}$, $C_2=\{4,5,6\}$, $C_3=\{1,6\}$ and $C_4=\{2,6\}$, consequently, $X \in \mathbb{S}^6_+(\overline{E},?)$ is characterized by $X(C_k) \in \mathbb{S}^{C_k}_+$ $(k=1,2,3,4)$.

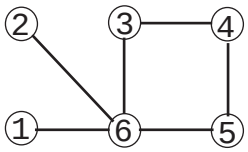


Figure 2.2: A graph $G(N,E)$ with $N=\{1,2,3,4,5,6\}$

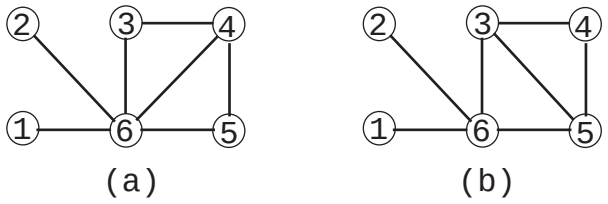


Figure 2.3: Chordal extensions of the graph $G(N,E)$ given in Figure 2.2. (a) The maximal cliques are $C_1=\{3,4,6\}$, $C_2=\{4,5,6\}$, $C_3=\{1,6\}$ and $C_4=\{2,6\}$. (b) The maximal cliques are $C_1=\{3,4,5\}$, $C_2=\{3,5,6\}$, $C_3=\{1,6\}$ and $C_4=\{2,6\}$.

Remark 2.2 *To compute the positive definite matrix completion of a matrix, we can recursively apply Lemma 2.6 of [21]. A numerical example is shown on page 657 of [21].*

2.2.3 Exploiting the domain-space sparsity

In this section, we consider a general nonlinear optimization problem involving a matrix variable $X \in \mathbb{S}^n$:

$$\text{minimize } f_0(x,X) \text{ subject to } f(x,X) \in \Omega \text{ and } X \in \mathbb{S}^n_+, \tag{2.32}$$

where $f_0 : \mathbb{R}^s \times \mathbb{S}^n \to \mathbb{R}$, $f : \mathbb{R}^s \times \mathbb{S}^n \to \mathbb{R}^m$ and $\Omega \subset \mathbb{R}^m$. Let E denote the set of distinct row and column index pairs (i,j) such that a value of X_{ij} is necessary to evaluate $f_0(x,X)$ and/or $f(x,X)$. More precisely, for $X^1_{kl} = X^2_{ij}$ $(k,l) \neq (i,j)$, $f_0(x,X^1) \neq f_0(x,X^2)$ and/or $f(x,X^1) \neq f(x,X^2)$ hold for some $x \in \mathbb{R}^s$, $X^1 \in \mathbb{S}^n$ and $X^2 \in \mathbb{S}^n$. Consider a graph $G(N,E)$. We call E *the d-space sparsity pattern* and $G(N,E)$ *the d-space sparsity pattern graph*. If $G(N,\overline{E})$ is an extension of $G(N,E)$, then we may replace the condition $X \in \mathbb{S}^n_+$ by $X \in \mathbb{S}^n_+(\overline{E},?)$. To apply Lemma 2.2, we choose a chordal extension $G(N,\overline{E})$ of $G(N,E)$. Let $C_1, C_2, \ldots, C_p$ be its maximal cliques. Then we may regard f_0 and f as functions in $x \in \mathbb{R}^s$ and $X(C_k)$ $(k = 1,2,\ldots,p)$, *i.e.,* there are functions $\tilde{f}_0$ and $\tilde{f}$ in the variables x and $X(C_k)$ $(k = 1,2,\ldots,p)$ such that

$$\left.\begin{array}{l} f_0(x,X) = \tilde{f}_0(x,X(C_1),X(C_2),\ldots,X(C_p)) \ \text{ for every } (x,X) \in \mathbb{R}^s \times \mathbb{S}^n, \\ f(x,X) = \tilde{f}(x,X(C_1),X(C_2),\ldots,X(C_p)) \ \text{ for every } (x,X) \in \mathbb{R}^s \times \mathbb{S}^n. \end{array}\right\} \tag{2.33}$$

Therefore, the problem (2.32) is equivalent to

$$\begin{array}{ll} \text{minimize} & \tilde{f}_0(x,X(C_1),X(C_2),\ldots,X(C_p)) \\ \text{subject to} & \tilde{f}(x,X(C_1),X(C_2),\ldots,X(C_p)) \in \Omega \text{ and} \\ & X(C_k) \in \mathbb{S}^{C_k}_+ \ (k = 1,2,\ldots,p). \end{array} \tag{2.34}$$

As an illustrative example, we consider the problem whose d-space sparsity pattern graph $G(N,E)$ is shown in Figure 2.2:

$$\left.\begin{array}{ll} \text{minimize} & -\displaystyle\sum_{(i,j)\in E,\ i<j} X_{ij} \\ \text{subject to} & \displaystyle\sum_{i=1}^{6}(X_{ii}-\alpha_i)^2 \le 6,\ X \in \mathbb{S}^6_+, \end{array}\right\} \tag{2.35}$$

where $\alpha_i > 0$ $(i = 1,2,\ldots,6)$. As a chordal extension, we choose the graph $G(N,\overline{E})$ in (a) of Figure 2.3. Then, the problem (2.34) becomes

$$\left.\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{4} \tilde{f}_{0k}(X(C_k)) \\ \text{subject to} & \displaystyle\sum_{k=1}^{4} \tilde{f}_k(X(C_k)) \le 6,\ X(C_k) \in \mathbb{S}^{C_k}_+ \ (k = 1,2,3,4), \end{array}\right\} \tag{2.36}$$

where

$$\left.\begin{array}{lcl} \tilde{f}_{01}(X(C_1)) &=& -X_{34} - X_{36},\ \tilde{f}_{02}(X(C_2)) = -X_{45} - X_{56}, \\ \tilde{f}_{03}(X(C_3)) &=& -X_{16},\ \tilde{f}_{04}(X(C_4)) = -X_{26}, \\ \tilde{f}_1(X(C_1)) &=& (X_{33}-\alpha_3)^2 + (X_{44}-\alpha_4)^2 + (X_{66}-\alpha_6)^2, \\ \tilde{f}_2(X(C_2)) &=& (X_{55}-\alpha_5)^2,\ \tilde{f}_3(X(C_3)) = (X_{11}-\alpha_1)^2, \\ \tilde{f}_4(X(C_4)) &=& (X_{22}-\alpha_2)^2. \end{array}\right\} \tag{2.37}$$

The positive semidefinite condition $X(C_k) \in \mathbb{S}^{C_k}_+$ $(k = 1,2,\ldots,p)$ in the problem (2.34) is not an ordinary positive semidefinite condition in the sense that overlapping variables X_{ij} $((i,j) \in C_k \cap C_l)$ exist in two distinct positive semidefinite constraints $X(C_k) \in \mathbb{S}^{C_k}_+$ and $X(C_l) \in \mathbb{S}^{C_l}_+$ if $C_k \cap C_l \neq \emptyset$. We describe two methods to transform the condition into an ordinary positive semidefinite condition. The first one was given in the papers [21, 68] where a d-space conversion method was proposed, and the second one was originally used for the sparse SDP relaxation of polynomial optimization problems [102, 103] and also in the paper [43] where a d-space conversion method was applied to an SDP relaxation of a sensor network localization problem. We call the first one *the d-space conversion method using clique trees* and the second one *the d-space conversion method using basis representation*.

The d-space conversion method using clique trees

We can replace $X(C_k)$ $(k = 1, 2, \ldots, p)$ by p independent matrix variables X^k $(k = 1, 2, \ldots, p)$ if we add all equality constraints $X^k_{ij} = X^l_{ij}$ for every $(i, j) \in C_k \cap C_l$ with $i \leq j$ and every pair of C_k and C_l such that $C_k \cap C_l \neq \emptyset$. For the chordal graph $G(N, \overline{E})$ given in (a) of Figure 2.3, those equalities turn out to be the 8 equalities

$$X^k_{66} - X^l_{66} = 0 \ (1 \leq k < l \leq 4),\ X^1_{44} = X^2_{44},\ X^1_{46} = X^2_{46}$$

These equalities are linearly dependent, and we can choose a maximal number of linearly independent equalities that are equivalent to the original equalities. For example, either of a set of 5 equalities

$$X^1_{44} - X^2_{44} = 0,\ X^1_{46} - X^2_{46} = 0,\ X^1_{66} - X^2_{66} = 0,\ X^1_{66} - X^3_{66} = 0,\ X^1_{66} - X^4_{66} = 0. \tag{2.38}$$

and a set of 5 equalities

$$X^1_{44} - X^2_{44} = 0,\ X^1_{46} - X^2_{46} = 0,\ X^1_{66} - X^2_{66} = 0,\ X^2_{66} - X^3_{66} = 0,\ X^3_{66} - X^4_{66} = 0 \tag{2.39}$$

is equivalent to the set of 8 equalities above.

In general, we use a clique tree $\mathcal{T}(\mathcal{K}, \mathcal{E})$ with $\mathcal{K} = \{C_1, C_2, \ldots, C_p\}$ and $\mathcal{E} \subseteq \mathcal{K} \times \mathcal{K}$ to consistently choose a set of maximal number of linearly independent equalities. Here $\mathcal{T}(\mathcal{K}, \mathcal{E})$ is called a clique tree if it satisfies the *clique-intersection property*, that is, for each pair of nodes $C_k \in \mathcal{K}$ and $C_l \in \mathcal{K}$, the set $C_k \cap C_l$ is contained in every node on the (unique) path connecting C_k and C_l. See [4] for basic properties on clique trees. We fix one clique for a root node of the tree $\mathcal{T}(\mathcal{K}, \mathcal{E})$, say C_1. For simplicity, we assume that the nodes $C_2, \ldots, C_p$ are indexed so that if a sequence of nodes $C_1, C_{l_2}, \ldots, C_{l_k}$ forms a path from the root node C_1 to a leaf node C_{l_k}, then $1 < l_2 < \cdots < l_k$, and each edge is directed from the node with a smaller index to the other node with a larger index. Thus, the clique tree $\mathcal{T}(\mathcal{K}, \mathcal{E})$ is directed from the root node C_1 to its leaf nodes. Each edge (C_k, C_l) of the clique tree $\mathcal{T}(\mathcal{K}, \mathcal{E})$ induces a set of equalities

$$X^k_{ij} - X^l_{ij} = 0 \ ((i, j) \in J(C_k \cap C_l)),$$

or equivalently,

$$E_{ij} \bullet X^k - E_{ij} \bullet X^l = 0 \ ((i, j) \in J(C_k \cap C_l)),$$

where $J(C) = \{(i, j) \in C \times C : i \leq j\}$ for every $C \subseteq N$. We add equalities of the form above for all $(C_k, C_l) \in \mathcal{E}$ when we replace $X(C_k)$ $(k = 1, 2, \ldots, p)$ by p independent matrix variables X^k $(k = 1, 2, \ldots, p)$. We thus obtain a problem

$$\left.\begin{array}{ll} \text{minimize} & \tilde{f}_0(x, X^1, X^2, \ldots, X^p) \\ \text{subject to} & \tilde{f}(x, X^1, X^2, \ldots, X^p) \in \Omega, \\ & E_{ij} \bullet X^k - E_{ij} \bullet X^l = 0 \ ((i, j, k, l) \in \Lambda), \\ & X^k \in \mathbb{S}^{C_k}_+ \ (k = 1, 2, \ldots, p), \end{array}\right\} \tag{2.40}$$

where

$$\Lambda = \{(g, h, k, l) : (g, h) \in J(C_k \cap C_l),\ (C_k, C_l) \in \mathcal{E}\}. \tag{2.41}$$

This is equivalent to the problem (2.34). See Section 4 of [68] for more details.

Now we illustrate the conversion process above by the simple example (2.35). Figure 2.4 shows two clique trees for the graph given in (a) of Figure 2.3. The left clique tree in Figure 2.4 leads to the 5 equalities in (2.38), while the right clique tree in Figure 2.4 induces the 5 equalities in (2.39). In both cases, the problem (2.40) has the following form

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{k=1}^{4} \hat{f}_{0k}(X^k) \\ \text{subject to} & \displaystyle\sum_{k=1}^{4} \hat{f}_k(X^k) \leq 6, \\ & \text{the 5 equalities in (2.38) or (2.39)}, \\ & X^k \in \mathbb{S}^{C_k}_+ \ (k = 1, 2, 3, 4), \end{array}$$

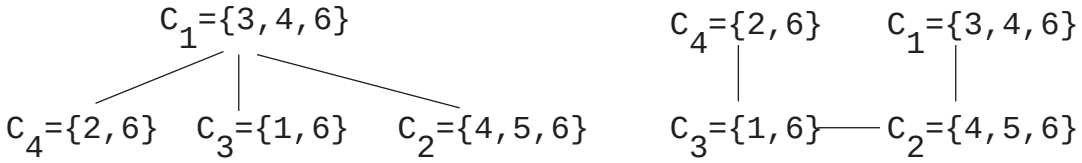


Figure 2.4: Two clique trees with $\mathcal{K} = \{C_1 = \{1,2\},\ C_2 = \{1,4\},\ C_3 = \{1,6\},\ C_4 = \{1,3,5\}\}$.

where

$$\begin{aligned}
\hat{f}_{01}(X^1) &= -X^1_{34} - X^1_{36},\ \hat{f}_{02}(X^2) = -X^2_{45} - X^2_{56},\\
\hat{f}_{03}(X^3) &= -X^3_{16},\ \hat{f}_{04}(X^4) = -X^4_{26},\\
\hat{f}_1(X^1) &= (X^1_{33} - \alpha_3)^2 + (X^1_{44} - \alpha_4) + (X^1_{66} - \alpha_6)^2,\\
\hat{f}_2(X^2) &= (X^2_{55} - \alpha_5)^2,\ \hat{f}_3(X^3) = (X^3_{11} - \alpha_1)^2,\ \hat{f}_4(X^4) = (X^4_{22} - \alpha_2)^2.
\end{aligned}$$

Remark 2.3 *The d-space conversion method using clique trees can be implemented in many different ways. The fact that the chordal extension $G(N, \overline{E})$ of $G(N, E)$ is not unique offers flexibility in constructing an optimization problem of the form (2.40). More precisely, a choice of chordal extension $G(N, \overline{E})$ of $G(N, E)$ decides how "small" and "sparse" an optimization problem of the form (2.40) is, which is an important issue for solving the problem more efficiently. For the size of the problem (2.40), we need to consider the sizes of the matrix variables X^k $(k = 1, 2, \ldots, p)$ and the number of equalities in (2.40). Note that the sizes of the matrix variables X^k $(k = 1, 2, \ldots, p)$ are determined by the sizes of the maximal cliques C_k $(k = 1, 2, \ldots, p)$. This indicates that a chordal extension $G(N, \overline{E})$ with smaller maximal cliques C_k $(k = 1, 2, \ldots, p)$ may be better theoretically. (In computation, however, this is not necessarily true because of overhead of processing too many small positive semidefinite matrix variables.) The number of equalities in (2.40) or the cardinality of Λ is also determined by the chordal extension $G(N, \overline{E})$ of $G(N, E)$. Choosing a chordal extension $G(N, \overline{E})$ with smaller maximal cliques increases the number of equalities. Balancing these two contradicting targets, decreasing the sizes of the matrix variables and decreasing the number of equalities was studied in the paper [68] by combining some adjacent cliques along the clique tree $\mathcal{T}(\mathcal{K}, \mathcal{E})$. See Section 4 of [68] for more details. In addition to the choice of a chordal extension $G(N, \overline{E})$ of $G(N, E)$, the representation of the functions and the choice of a clique tree add flexibilities in the construction of the problem (2.40). That is, the representation of the functions $f_0 : \mathbb{R}^s \times \mathbb{S}^n \to \mathbb{R}$ and $f : \mathbb{R}^s \times \mathbb{S}^n \to \mathbb{R}^m$ in the vector variable x and the matrix variables $X(C_k)$ $(k = 1, 2, \ldots, p)$ as in (2.33); for example, we could move the term $(X_{66} - \alpha_6)^2$ from $\tilde{f}_1(x, X(C_1))$ to either of $\tilde{f}_k(x, X(C_k))$ $(k = 2, 3, 4)$. These choices of the functions f_0, f and a clique tree affect the sparse structure of the resulting problem (2.40), which is also important for efficient computation.*

The domain-space conversion method using basis representation

Define

$$\begin{aligned}
\bar{J} &= \bigcup_{k=1}^{p} J(C_k),\\
(X_{ij} : (i,j) \in \bar{J}) &= \text{the vector variable consisting of } X_{ij}\ ((i,j) \in \bar{J}),\\
\bar{f}_0(x, (X_{ij} : (i,j) \in \bar{J})) &= f_0(x, X) \text{ for every } (x, X) \in \mathbb{R}^s \times \mathbb{S}^n,\\
\bar{f}(x, (X_{ij} : (i,j) \in \bar{J})) &= f(x, X) \text{ for every } (x, X) \in \mathbb{R}^s \times \mathbb{S}^n.
\end{aligned}$$

We represent each $X(C_k)$ in terms of a linear combination of the basis E_{ij} $((i,j) \in J(C_k))$ of the space $\mathbb{S}^{C_k}$ as in (2.31) with $C = C_k$ $(k = 1, 2, \ldots, p)$. Substituting this basis representation into the problem (2.34), we obtain

$$\left.\begin{array}{ll} \text{minimize} & \bar{f}_0(x, (X_{ij} : (i,j) \in \bar{J})) \\ \text{subject to} & \bar{f}(x, (X_{ij} : (i,j) \in \bar{J}) \in \Omega, \\ & \displaystyle\sum_{(i,j)\in J(C_k)} E_{ij} X_{ij} \in \mathbb{S}^{C_k}_+ \ (k = 1, 2, \ldots, p). \end{array}\right\} \tag{2.42}$$

We observe that the illustrative example (2.35) is converted into the problem

$$\left.\begin{array}{ll} \text{minimize} & \displaystyle -\sum_{(i,j)\in E,\ i<j} X_{ij} \\ \text{subject to} & \displaystyle\sum_{i=1}^{6} (X_{ii} - \alpha_i)^2 \le 6, \\ & \displaystyle\sum_{(i,j)\in J(C_k)} E_{ij} X_{ij} \in \mathbb{S}^{C_k}_+ \ (k = 1, 2, 3, 4). \end{array}\right\} \tag{2.43}$$

Remark 2.4 *Compared to the d-space conversion method using clique trees, the d-space conversion method using basis representation described above provides limited flexibilities. To make the size of the problem (2.42) smaller, we need to select a chordal extension $G(N, \overline{E})$ of $G(N, E)$ with smaller maximal cliques C_k $(k = 1, 2, \ldots, p)$. As a result, the sizes of semidefinite constraints become smaller. As we mentioned in Remark 2.3, however, too many smaller positive semidefinite matrix variables may yield heavy overhead in computation.*

2.2.4 Duality in positive semidefinite matrix completion

In order to present the r-space conversion methods in the next section, we need to derive some results, which can be understood as a dual approach to the positive semidefinite matrix completion approach from the 2.2.2. Throughout this section, we assume that $G(N, E)$ denotes a chordal graph. In Lemma 2.2, we have described a necessary and sufficient condition for a partial symmetric matrix $X \in \mathbb{S}^n(E, ?)$ to be completed to a positive semidefinite symmetric matrix. Let

$$\begin{aligned} \mathbb{S}^n(E,0) &= \{A \in \mathbb{S}^n : A_{ij} = 0 \text{ if } (i,j) \notin E^\bullet\}, \\ \mathbb{S}^n_+(E,0) &= \{A \in \mathbb{S}^n(E,0) : A \succcurlyeq 0\}. \end{aligned}$$

In this section, we derive a necessary and sufficient condition for a symmetric matrix $A \in \mathbb{S}^n(E,0)$ to be positive semidefinite, *i.e.*, $A \in \mathbb{S}^n_+(E,0)$. This condition is used for the range-space conversion methods in Section 5. We note that these two issues have primal-dual relationship:

$$A \in \mathbb{S}^n_+(E,0) \text{ if and only if } \sum_{(i,j)\in E^\bullet} A_{ij} X_{ij} \ge 0 \text{ for every } X \in \mathbb{S}^n_+(E, ?). \tag{2.44}$$

Suppose $A \in \mathbb{S}^n(E,0)$. Let $C_1, C_2, \ldots, C_p$ be the maximal cliques of $G(N, E)$. Then, we can consistently decompose $A \in \mathbb{S}^n(E,0)$ into $\widetilde{A}^k \in \mathbb{S}^{C_k}$ $(k = 1, 2, \ldots, p)$ such that $A = \displaystyle\sum_{k=1}^{p} \widetilde{A}^k$. We know that A is positive semidefinite if and only if $A \bullet X \ge 0$ for every $X \in \mathbb{S}^n_+$. This relation and Lemma 2.2 are used in the following.

Since $A \in \mathbb{S}^n(E,0)$, this condition can be relaxed to the condition (2.44). Therefore, A is positive semidefinite if and only if the following SDP has the optimal value 0.

$$\text{minimize} \quad \sum_{(i,j)\in E^\bullet} \left[\sum_{k=1}^{p} \widetilde{A}^k\right]_{ij} X_{ij} \quad \text{subject to } X \in \mathbb{S}^n_+(E, ?). \tag{2.45}$$

We can rewrite the objective function as

$$\begin{aligned}\sum_{(i,j)\in E^\bullet}\left[\sum_{k=1}^{p}\widetilde{A}^k\right]_{ij}X_{ij} &= \sum_{k=1}^{p}\left[\sum_{(i,j)\in E^\bullet}\widetilde{A}^k_{ij}X_{ij}\right]\\ &= \sum_{k=1}^{p}\left(\widetilde{A}^k \bullet X(C_k)\right) \text{ for every } X\in\mathbb{S}^n_+(E,?).\end{aligned}$$

Note that the second equality follows from $\widetilde{A}^k \in \mathbb{S}^{C_k}$ $(k=1,2,\ldots,p)$. Applying Lemma 2.2 to the constraint $X\in\mathbb{S}^n_+(E,?)$ of the SDP (2.45), we obtain an SDP

$$\text{minimize} \quad \sum_{k=1}^{p}\left(\widetilde{A}^k \bullet X(C_k)\right) \quad \text{subject to } X(C_k)\in\mathbb{S}^{C_k}_+ \ (k=1,2,\ldots,p), \tag{2.46}$$

which is equivalent to the SDP (2.45).

The SDP (2.46) involves multiple positive semidefinite matrix variables with overlapping elements. We have described two methods to convert such multiple matrix variables into independent ones with no overlapping elements in Sections 2.2.3 and 2.2.3, respectively. We apply the method given in Section 2.2.3 to the SDP (2.46). Let $\mathcal{T}(\mathcal{K},\mathcal{E})$ be a clique tree with $\mathcal{K}=\{C_1,C_2,\ldots,C_p\}$ and $\mathcal{E}\subseteq\mathcal{K}\times\mathcal{K}$. Then, we obtain an SDP

$$\begin{array}{ll}\text{minimize} & \displaystyle\sum_{k=1}^{p}\left(\widetilde{A}^k \bullet X^k\right)\\ \text{subject to} & E_{ij}\bullet X^k - E_{ij}\bullet X^l = 0 \ ((i,j,k,l)\in\Lambda),\\ & X^k\in\mathbb{S}^{C_k}_+ \ (k=1,2,\ldots,p),\end{array} \tag{2.47}$$

which is equivalent to the SDP (2.46). Here Λ is given in (2.41).

Theorem 2.15 *$A\in\mathbb{S}^n(E,0)$ is positive semidefinite if and only if the system of LMIs*

$$\widetilde{A}^k - \widetilde{L}^k(z)\succcurlyeq 0 \quad (k=1,2,\ldots,p). \tag{2.48}$$

has a solution $v=(v_{ghkl}:(g,h,k,l)\in\Lambda)$. Here $z=(z_{ghkl}:(g,h,k,l)\in\Lambda)$ denotes a vector variable consisting of z_{ghkl} $((g,h,k,l)\in\Lambda)$, and

$$\begin{aligned}\widetilde{L}^k(z) = &-\sum_{(i,j,h);\ (i,j,h,k)\in\Lambda} E_{ij}z_{ijhk} + \sum_{(i,j,l);\ (i,j,k,l)\in\Lambda} E_{ij}z_{ijkl}\\ &\text{for every } z=(z_{ijkl}:(i,j,k,l)\in\Lambda)\ (k=1,2,\ldots,p).\end{aligned} \tag{2.49}$$

Proof: In the previous discussions, we have shown that $A\in\mathbb{S}^n(E,0)$ is positive semidefinite if and only if the SDP (2.47) has the optimal value 0. The dual of the SDP (2.47) is

$$\text{maximize } 0 \text{ subject to (2.48).} \tag{2.50}$$

The primal SDP (2.47) attains the objective value 0 at a trivial feasible solution $(X_1,X_2,\ldots,X_p)=(0,0,\ldots,0)$. If the dual SDP (2.50) is feasible or the system of LMIs (2.48) has a solution, then the primal SDP (2.47) has the optimal value 0 by the week duality theorem. Thus we have shown the "if part" of the theorem. Now suppose that the primal SDP (2.47) has the optimal value 0. The primal SDP (2.47) has an interior-feasible solution; for example, take X^k to be the $\#C_k\times\#C_k$ identity matrix in $\mathbb{S}^{C_k}$ $(k=1,2,\ldots,p)$. By the strong duality theorem (Theorem 4.2.1 of [69]), the optimal value of the dual SDP (2.50) is zero, which implies that (2.50) is feasible. □

As a corollary, we obtain the following (Theorem 2.3 of [1]).

Theorem 2.16 *$A \in \mathbb{S}^n(E,0)$ is positive semidefinite if and only if there exist $Y^k \in \mathbb{S}^{C_k}_+$ $(k = 1,2,\ldots,p)$ which decompose A as $A = \sum_{k=1}^{p} Y^k$.*

Proof: Since the "if part" is straightforward, we prove the "only if" part. Assume that A is positive semidefinite. By Theorem 2.15, the LMI (2.48) has a solution $\tilde{z}$. Let $Y^k = \widetilde{A}^k - \widetilde{L}^k(\tilde{z})$ $(k = 1,2,\ldots,p)$. Then $Y^k \in \mathbb{S}^{C_k}_+$ $(k = 1,2,\ldots,p)$. Since $\sum_{k=1}^{p} \widetilde{L}^k(\tilde{z}) = 0$ by construction, we obtain the desired result. □

Conversely, Theorem 2.15 can be derived from Theorem 2.16. In the paper [1], Theorem 2.16 was proved by Theorem 7 of Grone et al. [28] (Lemma 2.2 in this thesis).

We conclude this section by applying Theorem 2.15 to the case of the chordal graph $G(N,E)$ given in (a) of Figure 2.3. The maximal cliques are $C_1 = \{3,4,6\}$, $C_2 = \{4,5,6\}$, $C_3 = \{1,6\}$ and $C_4 = \{2,6\}$, so that $A \in \mathbb{S}^6(E,0)$ is decomposed into 4 matrices

$$\widetilde{A}^1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{33} & A_{34} & 0 & A_{36} \\ 0 & 0 & A_{43} & A_{44} & 0 & A_{46} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{63} & A_{64} & 0 & A_{66} \end{pmatrix} \in \mathbb{S}^{\{3,4,6\}},$$

$$\widetilde{A}^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{45} & 0 \\ 0 & 0 & 0 & A_{54} & A_{55} & A_{56} \\ 0 & 0 & 0 & 0 & A_{65} & 0 \end{pmatrix} \in \mathbb{S}^{\{4,5,6\}},$$

$$\widetilde{A}^3 = \begin{pmatrix} A_{11} & 0 & 0 & 0 & 0 & A_{16} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ A_{61} & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{S}^{\{1,6\}},$$

$$\widetilde{A}^4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{22} & 0 & 0 & 0 & A_{26} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{62} & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathbb{S}^{\{2,6\}},$$

or,

$$\left.\begin{array}{rcl} \widetilde{A}^1 &=& \begin{pmatrix} A_{33} & A_{34} & A_{36} \\ A_{43} & A_{44} & A_{46} \\ A_{63} & A_{64} & A_{66} \end{pmatrix} \in \mathbb{S}^{\{3,4,6\}},\ \widetilde{A}^2 = \begin{pmatrix} 0 & A_{45} & 0 \\ A_{54} & A_{55} & A_{56} \\ 0 & A_{65} & 0 \end{pmatrix} \in \mathbb{S}^{\{4,5,6\}}, \\ \widetilde{A}^3 &=& \begin{pmatrix} A_{11} & A_{16} \\ A_{61} & 0 \end{pmatrix} \in \mathbb{S}^{\{1,6\}},\ \widetilde{A}^4 = \begin{pmatrix} A_{22} & A_{26} \\ A_{62} & 0 \end{pmatrix} \in \mathbb{S}^{\{2,6\}} \end{array}\right\} \tag{2.51}$$

in the compact representation. We note that this decomposition is not unique. For example, we can move the $(6,6)$ element A_{66} from $\widetilde{A}^1$ to any other $\widetilde{A}^k$. We showed two clique trees with $\mathcal{K} = \{C_1, C_2, C_3, C_4\}$ in Figure 2.4. For the left clique tree, we have $\Lambda = \{(4,4,1,2),(4,6,1,2),(6,6,1,2),(6,6,1,3),(6,6,1,4))\}$.

Thus, the system of LMIs (2.48) becomes

$$\left.\begin{array}{l} \left(\begin{array}{ccc} A_{33} & A_{34} & A_{36} \\ A_{43} & A_{44}-z_{4412} & A_{46}-z_{4612} \\ A_{63} & A_{64}-z_{4412} & A_{66}-z_{6612}-z_{6613}-z_{6614} \end{array}\right) \succcurlyeq 0, \\ \left(\begin{array}{ccc} z_{4412} & A_{45} & z_{4612} \\ A_{54} & A_{55} & A_{56} \\ z_{4612} & A_{65} & z_{6612} \end{array}\right) \succcurlyeq 0, \\ \left(\begin{array}{cc} A_{11} & A_{16} \\ A_{61} & z_{6613} \end{array}\right) \succcurlyeq 0, \ \left(\begin{array}{cc} A_{22} & A_{26} \\ A_{62} & z_{6614} \end{array}\right) \succcurlyeq 0. \end{array}\right\} \tag{2.52}$$

For the right clique tree, we have $\Lambda = \{(4,4,1,2),(4,6,1,2),(6,6,1,2),(6,6,2,3),(6,6,3,4)\}$ and

$$\left.\begin{array}{l} \left(\begin{array}{ccc} A_{33} & A_{34} & A_{36} \\ A_{43} & A_{44}-z_{4412} & A_{46}-z_{4612} \\ A_{63} & A_{64}-z_{4412} & A_{66}-z_{6612} \end{array}\right) \succcurlyeq 0, \\ \left(\begin{array}{ccc} z_{4412} & A_{45} & z_{4612} \\ A_{54} & A_{55} & A_{56} \\ z_{4612} & A_{65} & z_{6612}-z_{6623} \end{array}\right) \succcurlyeq 0, \\ \left(\begin{array}{cc} A_{11} & A_{16} \\ A_{61} & z_{6623}-z_{6634} \end{array}\right) \succcurlyeq 0, \ \left(\begin{array}{cc} A_{22} & A_{26} \\ A_{62} & z_{6634} \end{array}\right) \succcurlyeq 0. \end{array}\right\} \tag{2.53}$$

2.2.5 Exploiting the range-space sparsity

In this section, we present two range-space conversion methods, *the r-space conversion method using clique trees* based on Theorem 2.15 and *the r-space conversion method using matrix decomposition* based on Theorem 2.16.

The range-space conversion method using clique trees

Let

$$F \ = \ \{(i,j) \in N \times N : M_{ij}(y) \neq 0 \ \text{ for some } y \in \mathbb{R}^s, \ i \neq j\}.$$

We call F *the r-space sparsity pattern* and $G(N,F)$ *the r-space sparsity pattern graph* of the mapping $M : \mathbb{R}^s \to \mathbb{S}^n$. Apparently, $M(y) \in \mathbb{S}^n(F,0)$ for every $y \in \mathbb{R}^s$, but the graph $G(N,F)$ may not be chordal. Let $G(N,E)$ be a chordal extension of $G(N,F)$. Then

$$M(y) \in \mathbb{S}^n(E,0) \text{ for every } y \in \mathbb{R}^s. \tag{2.54}$$

Let $C_1, C_2, \ldots, C_p$ be the maximal cliques of $G(N,E)$.

To apply Theorem 2.15, we choose mappings $\widetilde{M}^k$ $(k = 1,2,\ldots,p)$ to decompose the mapping $M : \mathbb{R}^s \to \mathbb{S}^n$ such that

$$M(y) \ = \ \sum_{k=1}^{p} \widetilde{M}^k(y) \text{ for every } y \in \mathbb{R}^s, \ \widetilde{M}^k : \mathbb{R}^s \to \mathbb{S}^{C_k} \ (k = 1,2,\ldots,p). \tag{2.55}$$

Let $\mathcal{T}(\mathcal{K},\mathcal{E})$ be a clique tree where $\mathcal{K} = \{C_1, C_2, \ldots, C_p\}$ and $\mathcal{E} \subset \mathcal{K} \times \mathcal{K}$. By Theorem 2.15, y is a solution of (2.26) if and only if it is a solution of

$$\widetilde{M}^k(y) - \widetilde{L}^k(z) \succcurlyeq 0 \ (k = 1,2,\ldots,p) \tag{2.56}$$

for some $z = (z_{ghkl} : (g,h,k,l) \in \Lambda)$, where Λ is given in (2.41) and $\widetilde{L}^k$ in (2.49).

We may regard the r-space conversion method using clique trees described above as a dual of the d-space conversion method using clique trees applied to the SDP

$$\text{minimize } M(y) \bullet X \text{ subject to } X \succcurlyeq 0, \tag{2.57}$$

where $X \in \mathbb{S}^n$ denotes a variable matrix and $y \in \mathbb{R}^s$ a fixed vector. We know that $M(y) \succcurlyeq 0$ if and only if the optimal value of the SDP (2.57) is zero, so that (2.57) serves as a dual of the matrix inequality $M(y) \succcurlyeq 0$. Each element z_{ijkl} of the vector variable z corresponds to a dual variable of the equality constraint $E_{ij} \bullet X^k - E_{ij} \bullet X^l = 0$ in the problem (2.40), while each matrix variable $X^k \in \mathbb{S}^{C_k}$ in the problem (2.40) corresponds to a dual matrix variable of the kth matrix inequality $\widetilde{M}^k(y) - \widetilde{L}^k(z) \succcurlyeq 0$.

Remark 2.5 *On the flexibilities in implementing the r-space conversion method using clique trees, the comments in Remark 2.3 are valid if we replace the sizes of the matrix variable X^k by the size of the mapping $\widetilde{M}^k : \mathbb{R}^s \to \mathbb{S}^{C_k}$ and the number of equalities by the number of elements z_{ijkl} of the vector variable z. The correlative sparsity of (2.56) depends on the choice of the clique tree and the decomposition (2.55).*

As an example, we consider the case where M is tridiagonal, *i.e.*, the (i,j)th element M_{ij} of M is zero if $|i-j| \geq 2$, to illustrate the range space conversion of the matrix inequality (2.26) into the system of matrix inequalities (2.56). By letting $E = \{(i,j) : |i-j| = 1\}$, we have a simple chordal graph $G(N,E)$ with no cycle satisfying (2.54), its maximal cliques $C_k = \{k, k+1\}$ $(k = 1, 2, \ldots, n-1)$, and a clique tree $\mathcal{T}(\mathcal{K}, \mathcal{E})$ with

$$\mathcal{K} = \{C_1, C_2, \ldots, C_{n-1}\} \text{ and } \mathcal{E} = \{(C_k, C_{k+1}) \in \mathcal{K} \times \mathcal{K} : k = 1, 2, \ldots, n-2\}.$$

For every $y \in \mathbb{R}^s$, let

$$\widetilde{M}^k(y) = \begin{cases} \begin{pmatrix} M_{kk}(y) & M_{k,k+1}(y) \\ M_{k+1,k}(y) & 0 \end{pmatrix} \in \mathbb{S}^{C_k} & \text{if } 1 \leq k \leq n-2, \\ \begin{pmatrix} M_{n-1,n-1}(y) & M_{n-1,n}(y) \\ M_{n,n-1}(y) & M_{nn}(y) \end{pmatrix} \in \mathbb{S}^{C_k} & \text{if } k = n-1. \end{cases}$$

Then, we can decompose $M : \mathbb{R}^s \to \mathbb{S}^n(E,0)$ into $\widetilde{M}^k : \mathbb{R}^s \to \mathbb{S}^{C_k}$ $(k = 1, 2, \ldots, n-1)$ as in (2.55) with $p = n-1$. We also see that

$$\begin{aligned} \Lambda &= \{(k+1, k+1, k, k+1) : k = 1, 2, \ldots, n-2\}, \\ \widetilde{L}^k(z) &= \begin{cases} E_{22}\ z_{2212} \in \mathbb{S}^{C_1} & \text{if } k = 1, \\ -E_{k,k}\ z_{k,k,k-1,k} + E_{k+1,k+1}\ z_{k+1,k+1,k,k+1} \in \mathbb{S}^{C_k} & \text{if } k = 2, 3, \ldots, n-2, \\ -E_{n-1,n-1}\ z_{n-1,n-1,n-2,n-1} \in \mathbb{S}^{C_{n-1}} & \text{if } k = n-1, \end{cases} \end{aligned}$$

Thus the resulting system of matrix inequalities (2.56) is

$$\left.\begin{array}{l} \begin{pmatrix} M_{11}(y) & M_{12}(y) \\ M_{21}(y) & -z_{2212} \end{pmatrix} \succcurlyeq 0, \\ \begin{pmatrix} M_{kk}(y) + z_{k,k,k-1,k} & M_{k,k+1}(y) \\ M_{k+1,k}(y) & -z_{k+1,k+1,k,k+1} \end{pmatrix} \succcurlyeq 0 \ (k = 2, 3, \ldots, n-2), \\ \begin{pmatrix} M_{n-1,n-1}(y) + z_{n-1,n-1,n-2,n-1} & M_{n-1,n}(y) \\ M_{n,n-1}(y) & M_{nn}(y) \end{pmatrix} \succcurlyeq 0. \end{array}\right\}$$

The range-space conversion method using matrix decomposition

By Theorem 2.16, we obtain that $y \in \mathbb{R}^s$ is a solution of the matrix inequality (2.26) if and only if there exist $Y^k \in \mathbb{S}^{C_k}$ $(k = 1, 2, \ldots, p)$ such that

$$\sum_{k=1}^{p} Y^k = M(y) \text{ and } Y^k \in \mathbb{S}^{C_k}_+ \ (k = 1, 2, \ldots, p).$$

Let $\overline{J} = \cup_{k=1}^{p} J(C_k)$ and $\Gamma(i,j) = \{k : i \in C_k,\ j \in C_k\}$ $((i,j) \in \overline{J})$. Then we can rewrite the condition above as

$$\sum_{k \in \Gamma(i,j)} E_{ij} \bullet Y^k - E_{ij} \bullet M(y) = 0 \ ((i,j) \in \overline{J}) \text{ and } Y^k \in \mathbb{S}^{C_k}_+ \ (k = 1, 2, \ldots, p). \tag{2.58}$$

We may regard the r-space conversion method using matrix decomposition as a dual of the d-space conversion method using basis representation applied to the SDP (2.57) with a fixed $y \in \mathbb{R}^s$. Each variable X_{ij} ($(i,j) \in \overline{J}$) in the problem (2.42) corresponds to a dual real variable of the (i,j)th equality constraint of the problem (2.58), while each matrix variable Y^k in the problem (2.58) corresponds to a dual matrix variable of the constraint $\sum_{(i,j)\in J(C_k)} E_{ij}X_{ij} \in \mathbb{S}^{C_k}_+$.

Remark 2.6 *On the flexibilities in implementing the r-space conversion method using matrix decomposition, the comments in Remsark 2.4 are valid if we replace the sizes of the semidefinite constraints by the sizes of the matrix variables Y^k $(k = 1, 2, \ldots, p)$.*

We illustrate the r-space conversion method using matrix decomposition with the same example where M is tridiagonal as in Section 2.2.3 In this case, we see that

$$\begin{aligned}
p &= n-1,\\
C_k &= \{k, k+1\}\ (k = 1, 2, \ldots, n-1),\\
J(C_k) &= \{(k,k),\ (k,k+1),\ (k+1,k+1)\}\ (k = 1, 2, \ldots, n-1),\\
\overline{J} &= \{(k,k) : k = 1, 2, \ldots, n\} \bigcup \{(k,k+1) : k = 1, 2, \ldots, n-1\},\\
\Gamma(i,j) &= \begin{cases} \{1\} & \text{if } i = j = 1,\\ \{k\} & \text{if } i = k,\ j = k+1 \text{ and } 1 \le k \le n-1,\\ \{k-1, k\} & \text{if } i = j = k \text{ and } 2 \le k \le n-1,\\ \{n-1\} & \text{if } i = j = n. \end{cases}
\end{aligned}$$

Hence, the matrix inequality (2.26) with the tridiagonal $M : \mathbb{R}^s \to \mathbb{S}^n$ is converted into

$$\begin{aligned}
&E_{11} \bullet Y^1 - E_{11} \bullet M(y) = 0,\\
&E_{k,k+1} \bullet Y^k - E_{k,k+1} M(y) = 0\ (k = 1, 2, \ldots, n-1),\\
&E_{kk} \bullet Y^{k-1} + E_{kk} \bullet Y^k - E_{kk} \bullet M(y) = 0\ (k = 2, \ldots, n-1),\\
&E_{nn} \bullet Y^{n-1} - E_{nn} \bullet M(y) = 0,\\
&Y^k = \begin{pmatrix} Y^k_{kk} & Y^k_{k,k+1} \\ Y^k_{k+1,k} & Y^k_{k+1,k+1} \end{pmatrix} \in \mathbb{S}^{C_k}_+ \quad (k = 1, 2, \ldots, n-1).
\end{aligned}$$

2.2.6 Enhancing the correlative sparsity

When we are concerned with the SDP relaxation of polynomial SDPs (including ordinary polynomial optimization problems) and linear SDPs, another type of sparsity called the correlative sparsity plays an important role in solving the SDPs efficiently. The correlative sparsity was dealt with extensively in the paper [45], and was introduced in 2.1.4 . It is known that the sparse SDP relaxation [51, 102] for a correlatively sparse polynomial optimization problem leads to an SDP that can maintain the sparsity for primal-dual interior-point methods. See Section 6 of [45]. In this section, we focus on how the d-space and r-space conversion methods enhance the correlative sparsity. We consider a polynomial SDP of the form

$$\text{maximize} \quad f_0(y) \quad \text{subject to} \quad F_k(y) \in \mathbb{S}^{m_k}_+\ (k = 1, \ldots, p). \tag{2.59}$$

Here $f_0 \in \mathbb{R}[y]$, $\mathcal{F}_k$ a mapping from $\mathbb{R}^n$ into $\mathbb{S}^{m_k}$ with all polynomial components in $y \in \mathbb{R}^n$. For simplicity, we assume that f_0 is a linear function of the form $f_0(y) = b^T y$ for some $b \in \mathbb{R}^n$. In this case, with the definition from 2.1.4 the correlative sparsity pattern graph is given by the graph $G(N, E)$ with the node set $N = \{1, 2, \ldots, n\}$ and the edge set

$$E = \left\{ (i,j) \in N \times N : \begin{array}{l} i \ne j,\ \text{both values } y_i \text{ and } y_j \text{ are necessary} \\ \text{to evaluate the value of } F_k(y) \text{ for some } k \end{array} \right\}.$$

When a chordal extension $G(N, \overline{E})$ of the correlative sparsity pattern graph $G(N, E)$ is sparse or all the maximal cliques of $G(N, \overline{E})$ are small-sized, we can effectively apply the sparse SDP relaxation [51, 102] to

the polynomial SDP (2.59). As a result, we have a linear SDP satisfying a correlative sparsity characterized by the same chordal graph structure as $G(N, \overline{E})$. More details can be found in Section 6 of [45]. Even when the correlative sparsity pattern graph $G(N, E)$ or its chordal extension $G(N, \overline{E})$ is not sparse, the polynomial SDP may have "a hidden correlative sparsity" that can be recognized by applying the d-space and/or r-space conversion methods to the problem to decompose a large size matrix variable (and/or inequality) into multiple smaller size matrix variables (and/or inequalities). To illustrate this, let us consider a polynomial SDP of the form

$$\text{minimize } b^T y \text{ subject to } F(y) \in \mathbb{S}^n_+,$$

where F denotes a mapping from $\mathbb{R}^n$ into $\mathbb{S}^n$ defined by

$$F(y) = \begin{pmatrix} 1-y_1^4 & 0 & 0 & \cdots & 0 & y_1y_2 \\ 0 & 1-y_2^4 & 0 & \cdots & 0 & y_2y_3 \\ 0 & 0 & \ddots & & 0 & y_3y_4 \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ 0 & 0 & 0 & & 1-y_{n-1}^4 & y_{n-1}y_n \\ y_1y_2 & y_2y_3 & y_3y_4 & \cdots & y_{n-1}y_n & 1-y_n^4 \end{pmatrix}.$$

This polynomial SDP is not correlatively sparse at all (*i.e.,* $G(N, E)$ becomes a complete graph) because all variables $y_1, y_2, \ldots, y_n$ are involved in the single matrix inequality $F(y) \in \mathbb{S}^n_+$. Hence, the sparse SDP relaxation (2.18) is not effective for this problem. Applying the r-space conversion method using clique trees to the polynomial SDP under consideration, we have a polynomial SDP

$$\left.\begin{array}{ll} \text{minimize} & b^T y \\ \text{subject to} & \begin{pmatrix} 1-y_1^4 & y_1y_2 \\ y_1y_2 & z_1 \end{pmatrix} \succcurlyeq 0, \\ & \begin{pmatrix} 1-y_i^4 & y_iy_{i+1} \\ y_iy_{i+1} & -z_{i-1}+z_i \end{pmatrix} \succcurlyeq 0 \ (i = 2, 3, \ldots, n-2), \\ & \begin{pmatrix} 1-y_{n-1}^4 & y_{n-1}y_n \\ y_{n-1}y_n & 1-y_n^4-z_{n-2} \end{pmatrix} \succcurlyeq 0, \end{array}\right\} \tag{2.60}$$

which is equivalent to the original polynomial SDP. The resulting polynomial SDP now satisfies the correlative sparsity as shown in Figure 2.5. Thus the sparse SDP relaxation (2.18) is efficient for solving (2.60).

The correlative sparsity is important in linear SDPs, too. We have seen such a case in Section 2.2.1. We can rewrite the SDP (2.28) as

$$\left.\begin{array}{ll} \text{maximize} & -\displaystyle\sum_{i=1}^{n-1} \left(A^0_{ii}X_{ii} + 2A^0_{i,i+1}X_{i,i+1}\right) - A^0_{nn}X_{nn} \\ \text{subject to} & I - \displaystyle\sum_{i=1}^{n} E_{ii}X_{ii} + \sum_{i=1}^{n-1} E_{in}X_{i,i+1} \succcurlyeq 0, \\ & \displaystyle\sum_{1\le i\le j\le n} E_{ij}X_{ij} \succcurlyeq 0, \end{array}\right\} \tag{2.61}$$

where I denotes the $n \times n$ identity matrix. Since the coefficient matrices of all real variables X_{ij} $(1 \le i \le j \le n)$ are nonzero in the last constraint, the correlative sparsity pattern graph $G(N, E)$ forms a complete graph. Applying the d-space conversion method using basis representation and the r-space conversion method using clique trees to the original SDP (2.28), we have reduced it to the SDP (2.29) in Section 2.1. We rewrite the constraints of the SDP (2.29) as an ordinary LMI form:

$$\left.\begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & A^k_0 - \displaystyle\sum_{h=1}^{s} A^k_h y_h \succcurlyeq 0 \ (k = 1, 2, \ldots, p). \end{array}\right\} \tag{2.62}$$

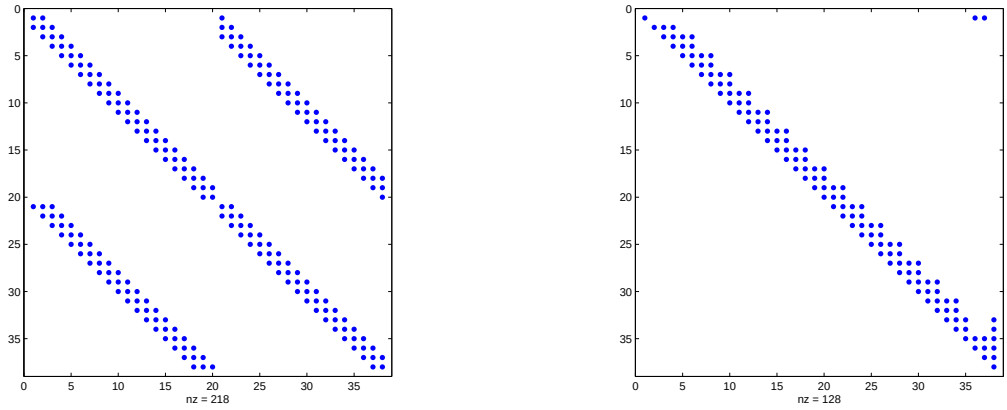


Figure 2.5: The correlative sparsity pattern of the polynomial SDP (2.60) with $n = 20$, and its Cholesky factor with a symmetric minimum degree ordering of its rows and columns.

Here $p = 2n - 2$, $s = 3n - 3$, each A_h^k is 2×2 matrix ($k = 1, 2, \ldots, p,\ h = 0, 1, \ldots, 3n - 3$), $b \in \mathbb{R}^{3n-3}$, $y \in \mathbb{R}^{3n-3}$, and each element y_h of y corresponds to some X_{ij} or some z_i. Comparing the SDP (2.61) with the SDP (2.62), we notice that the number of variables is reduced from $n(n + 1)/2$ to $3n - 3$, and the maximum size of the matrix inequality is reduced from n to 2. Furthermore, the correlative sparsity pattern graph becomes sparse. See Figure 2.6.

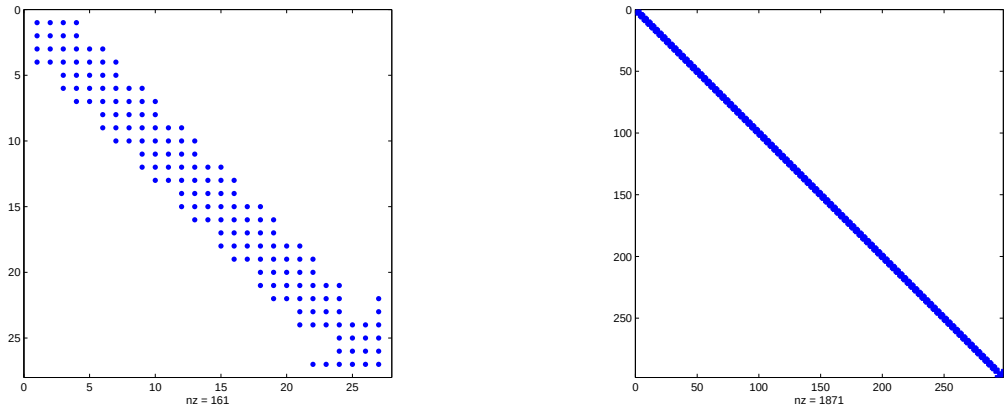


Figure 2.6: The correlative sparsity pattern of the SDP (2.62) induced from (2.29) with $n = 10$ and $n = 100$, and its Cholesky factor with a symmetric minimum degree ordering of its rows and columns.

Now we consider an SDP of the form (2.62) in general. The edge set E of the correlative sparsity pattern graph $G(N, E)$ is written as

$$E = \left\{ (g, h) \in N \times N : \ \ g \neq h,\ A_g^k \neq 0 \ \text{ and } A_h^k \neq 0 \text{ for some } k \ \right\},$$

where $N = \{1, 2, \ldots, s\}$. It is known that the graph $G(N, E)$ characterizes the sparsity pattern of the Schur

complement matrix of the SDP (2.62). More precisely, if R denotes the $s \times s$ sparsity pattern of the Schur complement matrix, then $R_{gh} = 0$ if $(g,h) \notin E^{\bullet}$. Furthermore, if the graph $G(N,E)$ is chordal, then there exists a perfect elimination ordering, a simultaneous row and column ordering of the Schur complement matrix that allows a Cholesky factorization with no fill-in. For the SDP induced from (2.29), we have seen the correlative sparsity pattern with a symmetric minimum degree ordering of its rows and columns in Figure 2.6, which coincides with the sparsity pattern of the Schur complement matrix whose symbolic Cholesky factorization is shown in Figure 2.1.

Remark 2.7 *As mentioned in Remark 2.5, the application of r-space conversion method using clique trees to reduce the SDP (2.28) to the SDP (2.29) can be implemented in many different ways. In practice, it should be implemented to have a better correlative sparsity in the resulting problem. For example, we can reduce the SDP (2.28) to*

$$\left.\begin{array}{ll} \textit{minimize} & \displaystyle\sum_{i=1}^{n-1} \left(A^0_{ii}X_{ii} + 2A^0_{i,i+1}X_{i,i+1}\right) + A^0_{nn}X_{nn} \\ \textit{subject to} & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} X_{11} & -X_{12} \\ -X_{21} & -z_1 \end{pmatrix} \succcurlyeq 0, \\ & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} X_{ii} & -X_{i,i+1} \\ -X_{i+1,i} & -z_i \end{pmatrix} \succcurlyeq 0 \ (i = 2,3,\ldots,n-2), \\ & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} X_{n-1,n-1} & -X_{n-1,n} \\ -X_{n,n-1} & X_{n,n} + \sum_{i=1}^{n-2} z_i \end{pmatrix} \succcurlyeq 0, \\ & \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} -X_{ii} & -X_{i,i+1} \\ -X_{i+1,i} & -X_{i+1,i+1} \end{pmatrix} \succcurlyeq 0 \ (i = 1,2,\ldots,n-1), \end{array}\right\} \tag{2.63}$$

which is different from the SDP (2.29). This is obtained by choosing a different clique tree in the r-space conversion method using clique trees for the SDP (2.28). In this case, all auxiliary variables z_i $(i = 1,2,\ldots,n-2)$ are contained in a single matrix inequality. This implies that the corresponding correlative sparsity pattern graph $G(N,E)$ involves a clique of size $n-2$. See Figure 2.7. Thus the correlative sparsity becomes worse than the previous conversion. Among various ways of implementing the d- and r-space conversion methods, determining which one is effective for a better correlative sparsity will be a subject which requires further study.

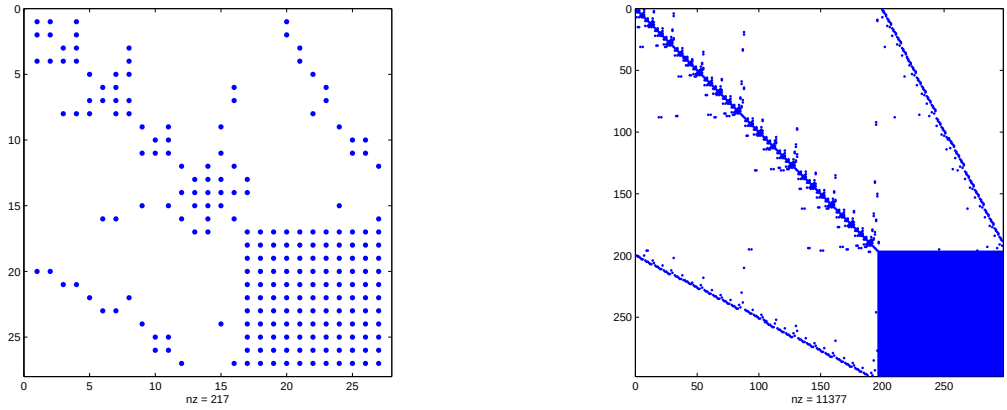


Figure 2.7: The correlative sparsity pattern of the SDP (2.62) induced from (2.63) with $n = 10$ and $n = 100$ where the rows and columns are simultaneously reordered by the Matlab function symamd (a symmetric minimum degree ordering).

2.2.7 Examples of d- and r-space sparsity in quadratic SDP

We present how to take advantage of the d- and r-space conversion methods introduced in the previous sections for the class of quadratic SDP, and demonstrate the effectiveness of these methods on examples of quadratic SDP. In the following we first consider a quadratic SDP of the form

$$\text{minimize } \sum_{i=1}^{s} c_i x_i \text{ subject to } M(x) \succcurlyeq 0, \tag{2.64}$$

where $c_i \in [0,1]$ $(i = 1, 2, \ldots, s)$, $M : \mathbb{R}^s \to \mathbb{S}^n$, and each non-zero element M_{ij} of the mapping $M : \mathbb{R}^s \to \mathbb{S}^n$ is a polynomial in $x = (x_1, x_2, \ldots, x_s) \in \mathbb{R}^s$ with degree at most 2.

We apply d- and r-space conversion methods to (2.64), get a new quadratic SDP with smaller size matrix inequality constraints and relax this quadratic SDP to obtain a linear SDP which can be solved by standard SDP solvers. The test problems of quadratic SDP we consider for numerical experiments are three max-cut problems, a Lovas theta problem, a box-constrained quadratic problem from SDPLIB [9], a sensor network localization problem and discretized partial differential equations (PDE) with Neumann and Dirichlet boundary conditions. In fact, a more detailed and systematic study of SDP relaxations exploiting d- and r-space sparsity in the case of quadratic optimization problems derived from PDEs compared to the hierarchy of sparse SDP relaxations (2.18) is presented in Chapter 3.

SDP relaxations of a quadratic SDP

In this subsection, we apply the d- and r-space conversion methods to the quadratic SDP (2.64), and derive four kinds of SDP relaxations:

(a) a dense SDP relaxation without exploiting any sparsity.

(b) a sparse SDP relaxation by applying the d-space conversion method using basis representation given in 2.2.3.

(c) a sparse SDP relaxation by applying the r-space conversion method using clique trees in 2.2.5.

(d) a sparse SDP relaxation by applying both, the d-space conversion method using basis representation and the r-space conversion method using clique trees.

We write each non-zero element $M_{ij}(x)$ as

$$M_{ij}(x) \;=\; Q_{ij} \bullet \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix} \text{ for every } x \in \mathbb{R}^s.$$

for some $Q_{ij} \in \mathbb{S}^{1+s}$. Assume that the rows and columns of each Q_{ij} are indexed from 0 to s. Let us introduce a linearization (or lifting) $\widehat{M}_{ij} : \mathbb{R}^s \times \mathbb{S}^s \to \mathbb{R}$ of the quadratic function $M_{ij} : \mathbb{R}^s \to \mathbb{R}$:

$$\widehat{M}_{ij}(x, X) \;=\; Q_{ij} \bullet \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \text{for every } x \in \mathbb{R}^s \text{ and } X \in \mathbb{S}^s,$$

which induces a linearization (or lifting) $\widehat{M} : \mathbb{R}^s \times \mathbb{S}^s \to \mathbb{S}^n$ of $M : \mathbb{R}^s \to \mathbb{S}^n$ whose (i,j)th element is $\widehat{M}_{ij}$. Then we can describe the dense SDP relaxation (a) for (2.64) as

$$\text{minimize} \sum_{i=1}^{n} c_i x_i \text{ subject to } \widehat{M}(x, X) \succcurlyeq 0 \text{ and } \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succcurlyeq 0.$$

For simplicity, we rewrite the dense SDP relaxation above as

(a) $$\text{minimize} \sum_{i=1}^{n} c_i W_{0i} \text{ subject to } \widehat{M}(W) \succcurlyeq 0,\ W_{00} = 1 \text{ and } W \succcurlyeq 0,$$

where

$$(W_{01}, W_{02}, \ldots, W_{0s}) \quad = \quad x^T \in \mathbb{R}^s \text{ and } W = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \in \mathbb{S}^{1+s}.$$

Let $G(N', F')$ be the d-space sparsity pattern graph for the SDP (a) with $N' = \{0, 1, \ldots, s\}$, and $F' =$ the set of distinct row and column index pairs (i, j) of W_{ij} that is necessary to evaluate the objective function $\sum_{i=1}^n c_i W_{0i}$ and/or the LMI $M(W) \succcurlyeq 0$. Let $G(N', E')$ be a chordal extension of $G(N', F')$, and $C'_1, C'_2, \ldots, C'_r$ be the maximal cliques of $G(N', E')$. Applying the *d-space conversion method using basis representation*, we obtain the SDP relaxation

$$\text{(b)} \qquad \begin{cases} \text{minimize} & \displaystyle\sum_{i=1}^n c_i W_{0i} \\ \text{subject to} & \widehat{M}((W_{ij} : (i,j) \in \overline{J})) \succcurlyeq 0, \ W_{00} = 1, \\ & \displaystyle\sum_{(i,j) \in J(C'_k)} E_{ij} W_{ij} \in \mathbb{S}_+^{C'_k} \ (k = 1, 2, \ldots, r). \end{cases}$$

Here $\overline{J} = \cup_{k=1}^r J(C'_k)$, $(W_{ij} : (i,j) \in \overline{J}) =$ the vector variable of the elements W_{ij} $((i,j) \in \overline{J})$ and

$$\widehat{M}((W_{ij} : (i,j) \in \overline{J})) \quad = \quad \widehat{M}(W) \text{ for every } W \in \mathbb{S}^s(E', 0).$$

To apply the *r-space conversion method using clique trees* to the quadratic SDP (2.64), we assume that $M : \mathbb{R}^s \to \mathbb{S}^n(E, 0)$ for some chordal graph $G(N, E)$ where $N = \{1, 2, \ldots, n\}$ and $E \subseteq N \times N$. Then, we convert the matrix inequality $M(x) \succcurlyeq 0$ in (2.64) into an equivalent system of matrix inequalities (2.56). The application of the LMI relaxation described above to (2.56) leads to the SDP relaxation

$$\text{(c)} \qquad \begin{cases} \text{minimize} & \displaystyle\sum_{i=1}^n c_i W_{0i} \\ \text{subject to} & \overline{M}^k(W) - \widetilde{L}^k(z) \succcurlyeq 0 \ (k = 1, 2, \ldots, p), \ W_{00} = 1, \ W \succcurlyeq 0, \end{cases}$$

where $\overline{M}^k : \mathbb{S}^{1+s} \to \mathbb{S}^{C_k}$ denotes a linearization (or lifting) of $\widetilde{M}^k : \mathbb{R}^s \to \mathbb{S}^{C_k}$. We may apply the linearization to (2.64) first to derive the dense SDP relaxation (a), and then apply the r-space conversion method using clique trees to (a). This results in the same sparse SDP relaxation (c) of (2.64). Note that both M and $\widehat{M}$ take values from $\mathbb{S}^n(E, 0)$. Thus, they provide the same r-space sparsity pattern characterized by the chordal graph $G(N, E)$.

Finally, the sparse SDP relaxation (d) is derived by applying the *d-space conversion method using basis representation* to the the sparse LMI relaxation (c). We note that the d-space sparsity pattern graph for the SDP (c) with respect to the matrix variable $W \in \mathbb{S}^{1+s}$ is the same as the one for the SDP (a). Hence, the sparse SDP relaxation (d) is obtained in the same way as the SDP (b) is obtained from the SDP (a). Consequently, we have the sparse SDP relaxation

$$\text{(d)} \qquad \begin{cases} \text{minimize} & \displaystyle\sum_{i=1}^n c_i W_{0i} \\ \text{subject to} & \overline{M}^k((W_{ij} : (i,j) \in \overline{J})) - \widetilde{L}^k(z) \succcurlyeq 0 \ (k = 1, 2, \ldots, p), \ W_{00} = 1, \\ & \displaystyle\sum_{(\alpha,\beta) \in J(C'_k)} E_{\alpha\beta} W_{\alpha\beta} \in \mathbb{S}_+^{C'_j} \ (j = 1, 2, \ldots, r). \end{cases}$$

Here $\overline{J} = \cup_{k=1}^r J(C'_k)$, $(W_{ij} : (i,j) \in \overline{J}) =$ the vector variable of the elements W_{ij} $((i,j) \in \overline{J})$ and

$$\overline{M}^k((W_{ij} : (i,j) \in \overline{J})) \quad = \quad \overline{M}^k(W) \text{ for every } W \in \mathbb{S}^s(E', 0).$$

Quadratic SDPs with d- and r-sparsity from randomly generated sparse graphs

Quadratic SDP problems were constructed by first generating two graphs $G(N_d, E_d)$ with $N_s = \{1, 2, \ldots, 1+s\}$ and $G(N_r, E_r)$ with $N_r = \{1, 2, \ldots, n\}$ using the Matlab program generateProblem.m [44], which was

developed for sensor network localization problems. Sparse chordal extensions $G(N_d, \overline{E}_d)$ and $G(N_r, \overline{E}_r)$ were then obtained by the Matlab functions symamd.m and chol.m. Next, we generated data matrices $Q_{ij} \in \mathbb{S}^{1+s}$ $(i = 1, 2, \ldots, n, j = 1, 2, \ldots, n)$ and a data vector $c \in \mathbb{R}^s$ so that the d- and r-space pattern graphs of the resulting quadratic SDP coincide with $G(N_d, E_d)$ and $G(N_r, E_r)$, respectively. Some characteristics of the chordal extensions $G(N_d, \overline{E}_d)$ of $G(N_r, E_r)$ and $G(N_r, \overline{E}_r)$ of $G(N_r, E_r)$ used in the experiments are shown in Table 4.

For the problem with $s = 40$ and $n = 640$, the d- and r-space sparsity pattern obtained from the symmetric approximate minimum degree permutation of rows and columns by the Matlab function symamd.m is displayed in Figure 2.8.

		Domain space sparsity				Range space sparsity			
s	n	$\#\overline{E}_d$	NoC	Max	Min	$\#\overline{E}_d$	NoC	Max	Min
80	80	143	63	3	3	216	72	7	3
320	320	649	260	7	3	840	301	9	3
40	160	70	30	3	3	426	150	7	3
40	640	70	30	3	3	1732	616	13	3

Table 2.2: Some characteristics of d- and r-sparsities of the tested quadratic SDPs. $\#\overline{E}_d$ (or $\#\overline{E}_r$) denotes the number of edges of $G(N_d, \overline{E}_d)$ (or $G(N_r, \overline{E}_r)$), NoC the number of the maximal cliques of $G(N_d, \overline{E}_d)$ (or $G(N_r, \overline{E}_r)$), Max the maximum size of the maximal cliques of $G(N_d, \overline{E}_d)$ (or $G(N_r, \overline{E}_r)$), and Min the minimum size of the maximal cliques of $G(N_d, \overline{E}_d)$ (or $G(N_r, \overline{E}_r)$).

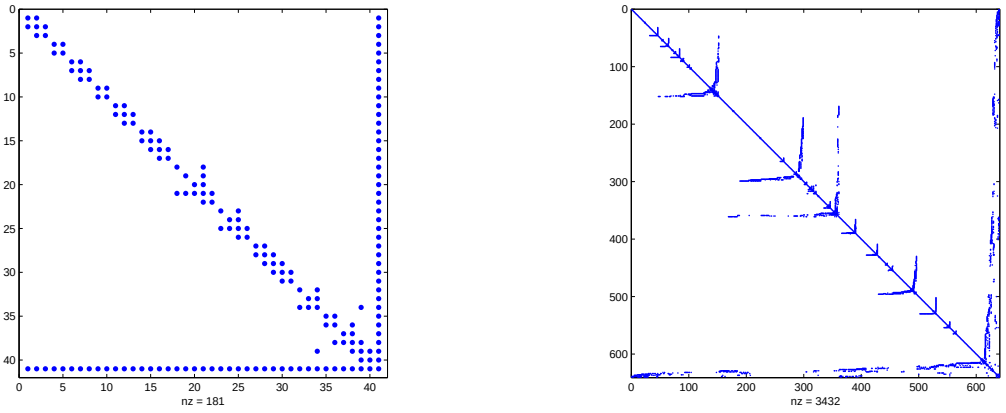


Figure 2.8: The d-space sparsity pattern of the quadratic SDP with $s = 40$ and $n = 640$ on the left and the r-space sparsity pattern on the right.

Table 5 shows numerical results on the quadratic SDPs whose d- and r- sparsity characteristics are given in Table 4. We observe that both the d-space conversion method using basis representation in (b) and the r-space conversion method using clique tree in (c) work effectively, and that their combination (d) results in the shortest CPU time among the four methods.

Quadratic SDPs arising from applications

For additional numerical experiments, we selected five SDP problems from SDPLIB [9], an quadratic SDP from a sensor network localization problem, and two quadratic SDP derived from PDE with Neumann and Dirichlet boundary condition. The test problems in Table 2.4 are

		SeDuMi CPU time in seconds (the size of the Schur complement matrix, the max. size of matrix variables)			
s	n	(a)	(b)	(c)	(d)
80	80	296.51 (3321, 81)	1.38 (224, 80)	1.58 (801, 81)	0.73 (252, 19)
320	320	OOM	74.19 (970, 320)	80.09 (322, 321)	35.20 (1216, 20)
40	160	6.70 (861, 160)	4.22 (111, 160)	2.91 (1626, 41)	0.74 (207, 21)
40	640	158.95 (861, 640)	151.20 (111, 640)	120.86 (6776, 41)	5.71 (772, 21)

Table 2.3: Numerical results on the quadratic SDPs with d- and r- sparsity from randomly generated sparse graphs. OOM indicates out of memory error in Matlab.

mcp500-1, maxG11, maxG32: An SDP relaxation of the max cut problem from SDPLIB.

thetaG11: An SDP relaxation of the Lovasz theta problem from SDPLIB.

qpG11: An SDP relaxation of the box constrained quadratic problem form SDPLIB.

d2n01s1000a100FSDP: A full SDP relaxation [6] of the sensor network localization problem with 1000 sensors, 100 anchors distributed in $[0,1]^2$, radio range = 0.1, and noise = 10%. The method (iii) in Table 2.4 for this problem is equivalent to the method used in SFSDP [43], a sparse version of full SDP relaxation.

ginzOrNeum(11): An SDP relaxation of the discretized nonlinear, elliptic PDE (4.4) (Case II, Neumann boundary condition) of [61]. We choose a 11×11 grid for the domain $[0,1]^2$ of the PDE.

pdeBifurcation(20): An SDP relaxation of the discretized nonlinear, elliptic PDE (4.5) (Dirichlet boundary condition) of [61]. We choose a 20×20 grid for the domain $[0,1]^2$ of the PDE.

The SDP relaxations (i), (ii) and (iii) in Table 2.4 indicate

(i) a dense SDP relaxation without exploiting any sparsity.

(ii) a sparse SDP relaxation by applying the d-space conversion method using clique trees given in Section 3.1.

(iii) a sparse SDP relaxation by applying the d-space conversion method using basis representation given in Section 3.2.

Table 2.4 shows that CPU time spent by (ii) is shorter than that by (i) and (iii) for all tested problems except for mcp500-1 and pdeEllipticNeum11. Notice that it took shorter CPU time to solve (iii) than (i) except for maxG32 and thetaG11. We confirm that applying at least one of the d-space conversion methods greatly reduces CPU time for the test problems. The d-space sparsity patterns for the test problems are displayed in Figures 2.9 and 2.10.

2.3 Reduction techniques for SDP relaxations for large scale POP

The global minimization of a multivariate polynomial over a semialgebraic set is a severely nonconvex, difficult optimization problem in general. In 2.1.3 a hierarchy of SDP relaxations has been proposed whose

Problem	SeDuMi CPU time (size.SC.mat., Max.size.mat.var.) (i)	(ii)	(iii)
mcp500-1	65.5 (500, 500)	94.5 (7222, 44)	15.9 (2878, 44)
maxG11	220.5 (800, 800)	12.1 (2432, 80)	26.8 (8333, 24)
maxG32	5373.8 (2000, 2000)	971.4 (13600, 210)	OOM
thetaG11	345.9 (2401, 801)	23.9 (4237, 81)	458.5 (9134, 25)
qpG11	2628.5 (800, 1600)	16.0 (2432, 80)	72.5 (9133, 24)
d2n01s1000a100FSDP	5193.5 (4949, 1002)	16.9 (7260, 45)	19.5 (15691, 17)
ginzOrNeum(11)	216.1 (1453, 485)	2.2 (1483, 17)	2.1 (1574, 4)
pdeBifurcation(20)	1120.4 (2401, 801)	4.3 (2451, 17)	5.3 (2001, 3)

Table 2.4: Numerical results on SDPs from some applications. size.SC.mat. denotes the size of the Schur complement matrix and Max.size.mat.var. the maximum size of matrix variables.

optima have been proven to converge to the optimum of a POP for increasing order of the relaxation. The practical use of this powerful theoretical result has been limited by the capacity of current SDP solvers, as the size of the SDP relaxations grows rapidly with increasing order. A first approach to attempt this problem has been the concept to exploit structured sparsity in a POP [47]. Whenever a POP satisfies a certain sparsity pattern, a convergent sequence of sparse SDP relaxations (2.18) of substantially smaller size can be constructed. Compared to the dense SDP relaxation (2.8), the sparse SDP relaxation (2.18) can be applied to POPs of larger scale.

Still, the size of the sparse SDP relaxation remains the major obstacle in order to solve large scale POPs, which contain polynomials of higher degree. We propose a substitution procedure to transform an arbitrary POP into an equivalent quadratic optimization problem (QOP). It is based on replacing quadratic terms in higher degree monomials by new variables successively, and adding the substitution relations as constraints to the optimization problem. The idea to transform a POP into an equivalent QOP can be traced back to Shor [92], who exploited it to derive dual lower bounds for the minimum of a polynomial function. As the substitution procedure is not unique, we introduce different heuristics which aim at deriving a QOP with as few additional variables as possible. Moreover, we show that sparsity of a POP is maintained under the substitution procedure. The main advantage of deriving an equivalent QOP for a POP is that the sparse SDP relaxation of first order can be applied to solve it approximately.

The substitution procedure and the considerations to minimize the number of additional variables while maintaining the sparsity are presented in 2.3.1. While a POP and the QOP derived from it are equivalent, we face the problem that the quality of the SDP relaxation for a QOP deteriorates in many cases. We discuss in 2.3.2 how to tighten the SDP relaxation for a QOP in order to achieve good approximations to the global minimum even for SDP relaxation of first or second order. For that purpose methods as choosing appropriate lower and upper bounds for the multivariate variables, Branch-and-Cut bounds to shrink the feasible region of the SDP relaxation and locally convergent optimization methods are proposed. Finally, the power of this technique is demonstrated in 2.3.3, where it is applied to solve various large scale POP of higher degree.

2.3.1 Transforming a POP into a QOP

The aim of 2.3 is to propose a technique to reduce the size of SDP relaxations for general POPs, which enables us to attempt large scale polynomial optimization efficiently. This technique, which transforms a POP into an equivalent QOP, reduces the size of the SDP relaxation by decreasing the minimum relaxation order $\omega_{\max}$, whereas the technique due to [102] presented in 2.1.4 aims at reducing the SDP relaxation by replacing matrix inequality constraints of size $\begin{pmatrix} n+\omega \\ \omega \end{pmatrix}$ through matrix inequality constraints of size

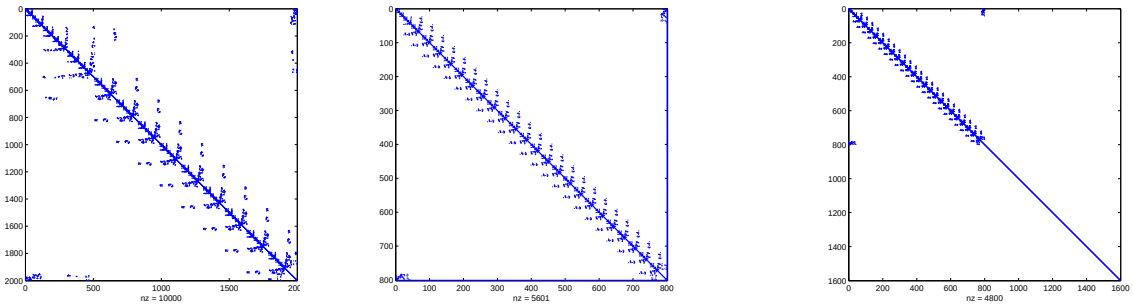


Figure 2.9: The d-space sparsity pattern with the symmetric approximate minimum degree permutation of rows and columns provided by the Matlab function symamd.m for maxG32(left), thetaG11(middle) and qpG11(right).

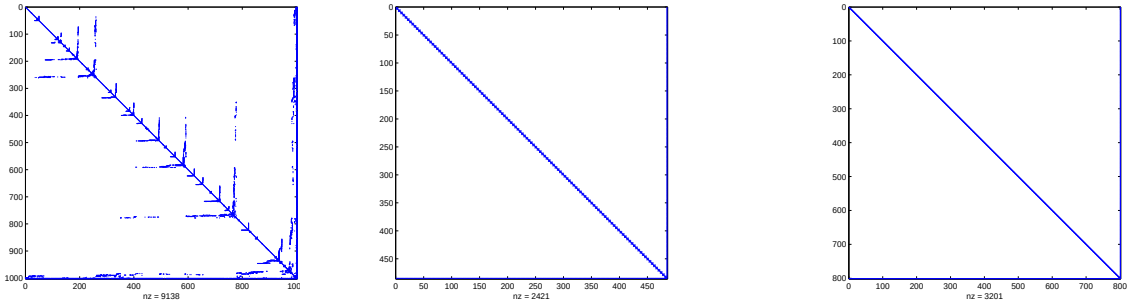


Figure 2.10: The d-space sparsity pattern with the symmetric approximate minimum degree permutation of rows and columns provided by the Matlab function symamd.m for d2n01s1000a100FSDP(left), pdeEllipticNeum11(middle) and pdeEllipticBifur20(right).

$p\begin{pmatrix} n_i+\omega \\ \omega \end{pmatrix}$. A general QOP is a special case of the POP (2.1), where the polynomials p, g_i $(i = 1, \ldots, m)$ are at most of degree 2. With respect to the definition of ω_k, the minimal relaxation order $\omega_{\max}$ of the sparse SDP relaxation (2.18) equals one. As pointed out in [102], the sparse SDP relaxation sSDP_1 and the dense SDP relaxation dSDP_1 of order one are equivalent for any QOP. The equivalence of a QOP and its SDP relaxation has been shown for a few restrictive classes of QOPs. For instance, if in a QOP p and $-g_i$ $(i = 1, \ldots, m)$ are convex quadratic polynomials, the QOP is equivalent to the corresponding SDP relaxation [56]. Also, equivalence of QOPs and their SDP relaxations was shown for the class of uniformly OD-nonpositive QOPs [41]. As shown in [52], $\min(\mathrm{sSDP}_\omega) \to \min(\mathrm{POP})$ for $\omega \to \infty$, but to the best of our knowledge there is no result for a rate of convergence or guaranteed approximation of the global minimum for a fixed relaxation order $\omega \geq \omega_{\max}$ in the case of a general POP.

To illustrate the idea of our transformation technique, consider the following example of a simple unconstrained POP, whose optimal value is $-\infty$:

$$\min\ 10x_1^3 - 10^2x_1^3x_2 + 10^3x_1^2x_2^2 - 10^4x_1x_2^3 + 10^5x_2^4 \tag{2.65}$$

It is straight forward that POP (2.65) is equivalent to

$$\begin{array}{ll} \min & 10x_1x_3 - 10^2x_3x_4 + 10^3x_4^2 - 10^4x_4x_5 + 10^5x_5^2 \\ \text{s.t.} & x_3 = x_1^2, \\ & x_4 = x_1x_2, \\ & x_5 = x_2^2, \end{array} \tag{2.66}$$

where we introduced three additional variables x_3, x_4 and x_5. Obviously QOP (2.66) is not the only QOP equivalent to POP (2.65): The QOP

$$\begin{array}{ll} \min & 10x_3 - 10^2x_2x_3 + 10^3x_5x_6 - 10^4x_1x_4 + 10^5x_2x_4 \\ \text{s.t.} & x_3 = x_1x_5, \\ & x_4 = x_2x_6, \\ & x_5 = x_1^2, \\ & x_6 = x_2^2, \end{array} \tag{2.67}$$

is equivalent to (2.65) as well. We notice the number of additional variables in QOP (2.66) equals three, whereas it equals four in QOP (2.67). Thus, there are numerous ways to transform a higher degree POP into a QOP in general. For the transformation procedures we are proposing, we consider 1) the number of additional variables should be as small as possible, in order to obtain a SDP relaxation of smaller size, 2) sparsity of a POP should be maintained under the transformation and 3) the quality of the SDP relaxation for the derived QOP should be as good as possible. How to deal with 3) is discussed in 2.3.2, 1) and 2) are discussed in the following.

Maintaining sparsity

The transformation proposed in the previous subsection raises the question, whether the correlative sparsity of a POP is preserved under the transformation, i.e., whether the resulting QOP is correlatively sparse as well.

Let POP* be a correlatively sparse POP of dimension n, $G(N, E')$ the chordal extension of its csp graph, $(C_1, \ldots, C_p)$ the maximal cliques of $G(N, E')$ and $n_{\max} = \max_{i=1,\ldots,p} \mid C_i \mid$. Let $x_{n+1} = x_ix_j$ be the substitution variable for some $i, j \in \{1, \ldots, n\}$. Let PÕP denote the POP derived after substituting $x_{n+1} = x_ix_j$ in POP*. Given the chordal extension $G(N, E')$ of the csp graph of POP*, a chordal extension of the csp graph of PÕP over the vertex set $\tilde{N} = N \cup \{n+1\}$ can be obtained by the extension: For a clique C_l with $\{i, j\} \subset C_l$ add the edges $\{v, n+1\}$ for all $v \in C_l$ and obtain the clique $\tilde{C}_l$. For each clique C_k not containing $\{i, j\}$, set $\tilde{C}_k = C_k$. In the end we obtain the graph $G(\tilde{N}, \tilde{E}')$ which is a chordal extension of the csp graph $G(\tilde{N}, \tilde{E})$ of PÕP. Note, $(\tilde{C}_1, \ldots, \tilde{C}_p)$ are maximal cliques for $G(\tilde{N}, \tilde{E}')$ and for all $\tilde{C}_l$ holds $\mid \tilde{C}_l \mid \leq \mid C_l \mid +1$, i.e. $\tilde{n}_{\max} \leq n_{\max} + 1$. Moreover, the number of maximal cliques p remains unchanged under the transformation. As pointed out, $G(\tilde{N}, \tilde{E}')$ is one possible chordal extension of $G(\tilde{N}, \tilde{E})$. It seems reasonable to expect that the heuristics we are using for the chordal extension, such as the reverse Cuthill-McKee and the symmetric minimum degree ordering, add less edges to $G(\tilde{N}, \tilde{E})$ than we did in constructing $G(\tilde{N}, \tilde{E}')$. Thus, we are able to apply the sparse SDP relaxations efficiently to the POPs derived after each iteration of the transformation algorithm. For illustration we consider Figure 2.11 and Figure 2.12, where the csp matrices of two POPs and their QOPs are pictured.

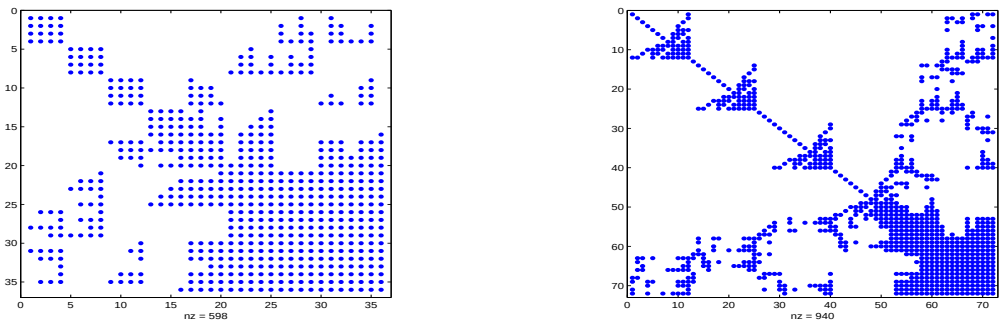


Figure 2.11: CSP matrix of the chordal extension of POP *pdeBifurcation(7)* (left) and its QOP (right) derived under strategy BI.

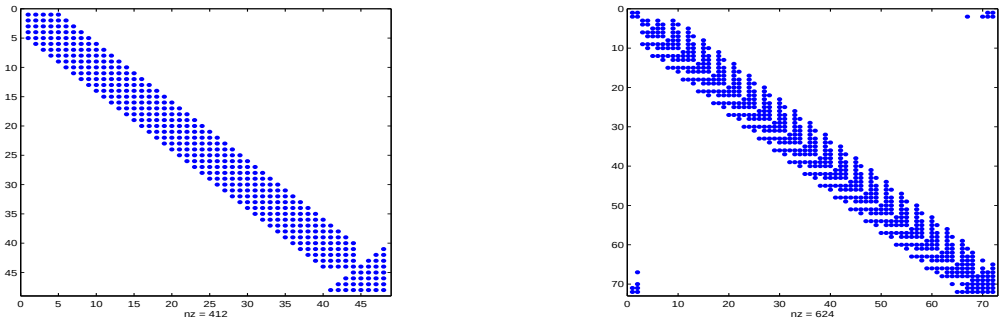


Figure 2.12: CSP matrix of the chordal extension of POP *Mimura(25)* (left) and its QOP (right) derived under strategy BI.

We observe that the sparsity pattern of the chordal extension of the csp graph is maintained under the substitution procedure. However, if the number of substitutions, which is required to transform a higher degree POP into a QOP, is far greater than the number of variables of the original POP, it may occur that we obtain a dense QOP under the transformation procedure. To illustrate this effect, consider the chordal extension of csp matrix of the QOP derived for the POP *randomEQ(7,3,5,8,0)* which is pictured in Figure 2.13. In that example, the number n of variables of the original POP equals seven, the number of additional variables equals 108.

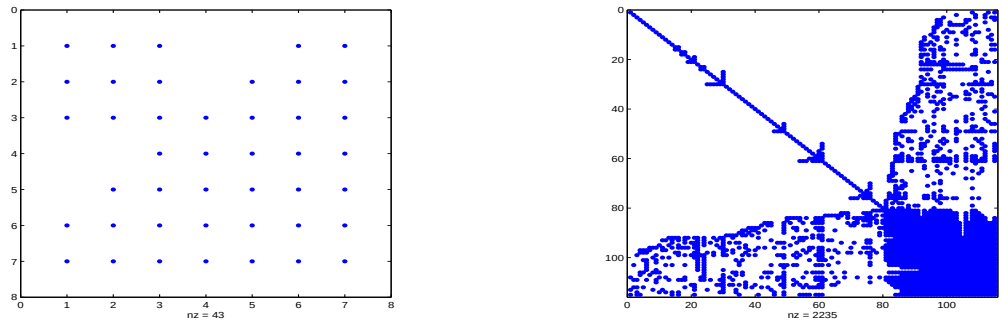


Figure 2.13: CSP matrix of the chordal extension of POP randomWithEQ(7,3,5,8,0) (left) and its QOP (right).

Minimizing the number of additional variables

Let n denote the number of variables involved in a POP and $\tilde{n}$ the number of variables in the corresponding QOP. The first question we are facing is, how to transform a POP into a QOP such that the number $k_0 := \tilde{n} - n$ of additional variables is as small as possible. Each additional variable x_{n+k} corresponds to

the substitution of a certain quadratic monomial $x_i x_j$ by x_{n+k}. Given an arbitrary POP, the question to find a substitution procedure minimizing $\tilde{n}$ is a difficult problem. We propose four different heuristics for transforming a POP into a QOP, which aim at reducing the number k_0 of additional variables. At the end of this section we give some motivation, why it is more important to find a strategy optimizing the quality of the SDP relaxation than one that minimizes the number k_0 of additional variables.

Our transformation algorithm iterates substitutions of pairs of quadratic monomials $x_i x_j$ in the higher degree monomials in objective function and constraints by a new variable x_{n+k}, and adding the substitution relation $x_{n+k} = x_i x_j$ as constraints to the POP. Let POP^0 denote the original POP, and POP^k the POP obtained after the k-th iteration, i.e. after substituting $x_{n+k} = x_i x_j$ and adding it as constraint to POP^{k-1}. The algorithm terminates as soon as POP^{k_0} is a QOP for some $k_0 \in \mathbb{N}$. In each iteration of the transformation algorithm we distinguish two steps. The first one is to choose which pair of variables (x_i, x_j) $(1 \le i, j \le n+k)$ is substituted by the additional variable x_{n+k+1}. The second one is to choose to which extent $x_i x_j$ is substituted by x_{n+k+1} in each higher degree monomial.

Step 1: Choosing the substitution variables

Definition 2.7 *Let POP^k be a POP of dimension $\tilde{n}$ with $\tilde{m}$ constraints $(g_1^k, \ldots, g_m^k)$. The* **higher monomial set** *$\mathcal{M}_S^k$ of POP^k is given by*

$$\mathcal{M}_S^k = \left\{ \alpha \in \mathbb{N}^{\tilde{n}} \mid \exists i \in \{0, \ldots, \tilde{m}\} \ \textit{s.t.}\ \alpha \in \textit{supp}(g_i^k) \ \textit{and}\ \mid \alpha \mid \geq 3 \right\},$$

where $g_0 := p$, $g_i^0 := g_i$, and the **higher monomial list** *$\mathcal{M}^k$ of POP^k by*

$$\mathcal{M}^k = \left\{ (\alpha, w_\alpha) \mid \alpha \in \mathcal{M}_S^k \ \textit{and}\ w_\alpha := \#\left\{ i \mid \alpha \in \textit{supp}(g_i^k) \right\} \right\}.$$

By Definition 2.7, the higher monomial list of a QOP is empty.

Definition 2.8 *Given $\alpha \in \mathbb{N}^n$ and a pair (i,j) where $1 \le i, j \le n$, we define the* **dividing coefficient** *$k_{i,j}^\alpha \in \mathbb{N}_0$ as the integer that satisfies $\frac{x^\alpha}{(x_i x_j)^{k_{i,j}^\alpha}} \in \mathbb{R}[x]$ and $\frac{x^\alpha}{(x_i x_j)^{k_{i,j}^\alpha + 1}} \notin \mathbb{R}[x]$.*

Given POP^k the k-th iterate of POP^0 and its higher monomial list $\mathcal{M}^k$, determine the symmetric matrix $C(\mathrm{POP}^k) \in \mathbb{R}^{(n+k)\times(n+k)}$ given by

$$C(\mathrm{POP}^k)_{i,j} = C(\mathrm{POP}^k)_{j,i} = \sum_{(\alpha, w_\alpha) \in \mathcal{M}^k} k_{i,j}^\alpha \, w_\alpha.$$

We consider two alternatives to choose a pair $(x_i, x_j)(1 \le i, j \le n+k)$ to be substituted by x_{n+k+1}:

A. **Naive criterion:** Choose a pair (x_i, x_j) such that there exists a $\alpha \in \mathcal{M}_S(\mathrm{POP}^k)$ which satisfies $\frac{x^\alpha}{x_i x_j} \in \mathbb{R}[x]$.

B. **Maximum criterion:** Choose a pair (x_i, x_j) such that $C(\mathrm{POP}^k)_{i,j} \geq C(\mathrm{POP}^k)_{u,v}$ $\forall 1 \le u, v \le n+k$.

Step 2: Choose the substitution strategy Next we have to decide to what extent we substitute $x_{n+k+1} = x_i x_j$ in each monomial of $\mathcal{M}_S(\mathrm{POP}^k)$. We will distinguish **full** and **partial substitution**. Let us demonstrate the importance of considering that question on the following two examples.

Example 2.5 *Consider two different substitution strategies for transforming the problem to minimize x_1^4 into a QOP:*

$$\begin{array}{ccccc} & & \min x_1^4 & & \\ (1) & \swarrow & & \searrow & (2) \\ \min & x_2^2 & & \min & x_1^2 x_2 \\ \textit{s.t.} & x_2 = x_1^2 & & \textit{s.t.} & x_2 = x_1^2 \\ & & & \downarrow & \\ & & & \min & x_1 x_3 \\ & & & \textit{s.t.} & x_2 = x_1^2 \\ & & & & x_3 = x_1 x_2 \end{array} \tag{2.68}$$

In both substitution strategies, we choose x_1^2 for substitution in the first step. In (1) we fully substituted x_1^2 by x_2, whereas in (2) we substituted x_1^2 partially. By choosing full substitution in the first iteration in (1), we need one additional variable to obtain a QOP, partial substitution requires two additional variables to yield a QOP.

Example 2.6

$$
\begin{array}{lll}
 & \min x_1^6 & \\
 & s.t.\ x_1^3 x_2 \geq 0 & \\
(1) \quad \swarrow & & \searrow \quad (2) \\
\begin{array}{ll} \min & x_3^3 \\ s.t. & x_1x_2x_3 \geq 0 \\ & x_3 = x_1^2 \end{array} & & \begin{array}{ll} \min & x_1^2x_3^2 \\ s.t. & x_1x_2x_3 \geq 0 \\ & x_3 = x_1^2 \end{array} \\
\downarrow & & \downarrow \\
\begin{array}{ll} \min & x_3x_4 \\ s.t. & x_1x_2x_3 \geq 0 \\ & x_3 = x_1^2 \\ & x_4 = x_3^2 \end{array} & & \begin{array}{ll} \min & x_4^2 \\ s.t. & x_2x_4 \geq 0 \\ & x_3 = x_1^2 \\ & x_4 = x_1x_2 \end{array} \\
\downarrow & & \\
\begin{array}{ll} \min & x_3x_4 \\ s.t. & x_2x_5 \geq 0 \\ & x_3 = x_1^2 \\ & x_4 = x_3^2 \\ & x_5 = x_1x_3 \end{array} & &
\end{array}
\tag{2.69}
$$

In this example full substitution (1) of x_1^2 requires three, and partial substitution (2) only two additional variables to yield a QOP.

The examples illustrate it depends on the structure of the higher monomial set, whether partial or full substitution requires less additional variables and results in a smaller size of the SDP relaxation. In general partial and full substitution are given as follows.

I. **Full substitution:** Let $t_{f_{i,j}}^r : \mathbb{R}[x] \to \mathbb{R}[z]$, where $x \in \mathbb{R}^r$ and $z \in \mathbb{R}^{r+1}$ for a $r \in \mathbb{N}$ and $i, j \in \{1, \ldots, r\}$, be a linear operator defined by its mappings for each monomial x^α,

$$
t_{f_{i,j}}^r(x^\alpha) = \begin{cases} z_1^{\alpha_1} \cdots z_{i-1}^{\alpha_{i-1}} z_i^{\alpha_i - \min(\alpha_i,\alpha_j)} z_{i+1}^{\alpha_{i+1}} \cdots z_{j-1}^{\alpha_{j-1}} z_j^{\alpha_j - \min(\alpha_i,\alpha_j)} z_{j+1}^{\alpha_{j+1}} \cdots z_r^{\alpha_r} z_{r+1}^{\min(\alpha_i,\alpha_j)}, & \text{if } i \neq j, \\ z_1^{\alpha_1} \cdots z_{i-1}^{\alpha_{i-1}} z_i^{\operatorname{mod}(\alpha_i,2)} z_{i+1}^{\alpha_{i+1}} \cdots z_r^{\alpha_r} z_{r+1}^{\lfloor \frac{\alpha_i}{2} \rfloor}, & \text{if } i = j. \end{cases}
$$

Thus, $t_{f_{i,j}}^r(g(x)) = \sum_{\alpha \in \operatorname{supp}(g)} c_\alpha(g) t_{f_{i,j}}^r(x^\alpha)$ for any $g \in \mathbb{R}[x]$. The operator $t_{f_{i,j}}^{n+k}$ substitutes $x_i x_j$ by x_{n+k+1} in each monomial to the maximal possible extent.

II. **Partial substitution:** Let $t_{p_{i,j}}^r : \mathbb{R}[x] \to \mathbb{R}[z]$, where $x \in \mathbb{R}^r$ and $z \in \mathbb{R}^{r+1}$ for a $r \in \mathbb{N}$ and $i, j \in \{1, \ldots, r\}$, be a linear operator defined by its mappings for each monomial x^α,

$$
t_{p_{i,j}}^r(x^\alpha) = \begin{cases} t_{f_{i,j}}^r(x^\alpha), & \text{if } i \neq j, \\ t_{f_{i,j}}^r(x^\alpha), & \text{if } i = j \text{ and } \alpha_i \text{ odd}, \\ t_{f_{i,j}}^r(x^\alpha), & \text{if } i = j \text{ and } \log_2(\alpha_i) \in \mathbb{N}_0, \\ z_1^{\alpha_1} \cdots z_{i-1}^{\alpha_{i-1}} z_i^{g_i} z_{i+1}^{\alpha_{i+1}} \cdots z_r^{\alpha_r} z_{r+1}^{\frac{1}{2}(\alpha_i - g_i)}, & \text{else}, \end{cases}
$$

where $g_i := \gcd(2^{\lfloor \log_2(\alpha_i) \rfloor}, \alpha_i)$. Thus, $t_{p_{i,j}}^r(g(x)) = \sum_{\alpha \in \operatorname{supp}(g)} c_\alpha(g) t_{p_{i,j}}^r(x^\alpha)$ for any $g \in \mathbb{R}[x]$.

We notice that full and partial substitution only differ in the case $i = j$, α_i even and $\log_2(\alpha_i) \notin \mathbb{N}_0$ holds. By pairwise combining the choice of A or B in Step 1 and the choice of I or II in Step 2, we obtain four different procedures to transform POP^{k-1} into POP^k that we denote as AI, AII, BI and BII. We do not expect AI or AII to result in a POP with a small number of substitutions, as A does not take into account

the structure of the higher degree monomial list $\mathcal{M}_S^{k-1}$, but we use AI and AII to evaluate the potential of BI and BII. The numerical performance of these four procedures is demonstrated on some example POPs in Table 2.6, where n denotes the number of variables in the original POP, *deg* the degree of the highest order polynomial in the POP, and k_0 the number of additional variables required to transform the POP into a QOP under the respective substitution strategy. The POPs *pdeBifurcation(n)* are derived from discretizing differential equations, which is the topic of Chapter 3, the other POPs are test problems from [102]. As expected, strategy B is superior to A for all but one example class of POP, when reducing the number of variables is concerned.

The entire algorithm to transform a POP into a QOP can be summarized by the scheme in Table 2.5. As mentioned before the QOP of dimension $n+k$ derived by AI, AII, BI or BII is equivalent to the original POP of dimension n. In fact it is easy to see, if $\tilde{x} \in \mathbb{R}^{n+k}$ an optimal solution of the QOP, the vector $(\tilde{x}_1, \ldots, \tilde{x}_n)$ of the first n components of $\tilde{x}$ is an optimizer of the original POP.

INPUT	POP^0 with $\mathcal{M}_S^0$
WHILE	$\mathcal{M}_S^k \neq \emptyset$
	1. Determine the pair (x_i, x_j) for substitution by A or B. 2. Apply $t_{f_{i,j}^k}$ or $t_{p_{i,j}^k}$ to each polynomial in POP^k and derive POP^{k+1}. 3. Update $k \to k+1$, $\text{POP}^k \to \text{POP}^{k+1}$, $\mathcal{M}_S^k \to \mathcal{M}_S^{k+1}$.
OUTPUT	$\text{QOP} = \text{POP}^{k_0}$

Table 2.5: Scheme for transforming a POP into a QOP.

POP	n	deg	k_0(AI)	k_0(AII)	k_0(BI)	k_0(BII)
BroydenBand(20)	20	6	229	211	60	40
BroydenBand(60)	60	6	749	691	180	120
nondquar(32)	32	4	93	93	94	94
nondquar(8)	8	4	21	21	22	22
optControl(10)	60	4	60	60	60	60
randINEQ(8,4,6,8,0)	8	8	253	307	248	238
randEQ(7,3,5,8,0)	7	8	135	146	116	115
pdeBifurcation(5)	25	3	25	25	25	25
pdeBifurcation(10)	100	3	100	100	100	100
randINEQ(3,1,3,16,0)	3	16	145	192	105	117
randUnconst(3,2,3,14,0)	3	14	86	107	63	69

Table 2.6: Number of required variables for strategies AI, AII, BI and BII.

Computational complexity

Finally, let us consider how the size of the sparse SDP relaxation of order $\omega = 1$ for a QOP depends on the number k_0 of additional variables. Let a sparse POP of dimension n be given by the polynomials $(p, g_1, \ldots, g_m)$ and the maximal cliques $(C_1, \ldots, C_p)$ of the chordal extension. With the construction in *Maintaining sparsity* above, the corresponding QOP of dimension $\tilde{n} = n + k_0$ has the maximal cliques $(\tilde{C}_1, \ldots, \tilde{C}_p)$ such that $C_i \subseteq \tilde{C}_i$ and $\tilde{n}_i \leq n_i + k_0$ for all $(i = 1, \ldots, p)$, where $n_i =\mid C_i \mid$ and $\tilde{n}_i =\mid \tilde{C}_i \mid$. All partial localizing matrices $M_0(g_k\, y, \hat{F}_k)$ are scalars in $\text{sSDP}_1(QOP)$. The size of the partial moment matrices $M_1(y, \tilde{C}_i)$ is

$$d(1, \tilde{n}_i) = \tilde{n}_i + 1 \leq n_i + k_0 + 1 = O(k_0). \tag{2.70}$$

Thus, the size of the linear matrix inequality is bounded by

$$\sum_{j=1}^{m+k_0} 1 + \sum_{i=1}^{p} d(1, \tilde{n}_i) \le m + k_0 + p\,(n_{\max} + k_0 + 1) \le m + k_0 + n\,(n_{\max} + k_0 + 1). \tag{2.71}$$

The length of the vector variable y in $\mathrm{sSDP}_1(QOP)$ is bounded by

$$\begin{array}{ll} \mid y \mid & \le \sum_{i=1}^{p} \mid y(\tilde{C}_p) \mid = \sum_{i=1}^{p} d(2, 2\tilde{n}_i) \le \frac{1}{2} p\,(2n_{\max} + 2k_0 + 2)\,(2n_{\max} + 2k_0 + 1) \\ & \le 2p\,(n_{\max} + k_0 + 1)^2 \le 2n\,(n_{\max} + k_0 + 1) = O(k_0^2). \end{array} \tag{2.72}$$

Thus, the size of the linear matrix inequalities of the sparse SDP relaxation is linear and the length of the moment vector y quadratic in the number k_0 of additional variables. For this reason the computational cost does not grow too fast, even if k_0 is not minimal. Heuristics BI and BII are sufficient in order to derive QOP with a small number k_0 of additional variables.

Moreover, the bounds (2.71) and (2.72) for the size of the primal and dual variables of the SDP relaxation for the QOP are to be compared to the respective bounds for the SDP relaxation of the POP. If we assume $\omega_{\max} = \omega_i$ for all $i \in \{1, \ldots, m\}$, the size of the linear matrix inequality in the SDP relaxation of order $\omega_{\max}$ for the original POP can be bounded by

$$\sum_{j=1}^{m} d(n_j,\, \omega_{\max} - \omega_j) + \sum_{i=1}^{p} d(n_i,\, \omega_{\max}) \le m + n \begin{pmatrix} n_{\max} + \omega_{\max} \\ \omega_{\max} \end{pmatrix}, \tag{2.73}$$

and the length of the moment vector by

$$\sum_{i=1}^{p} d(2n_i, 2\omega_{\max}) \le n \begin{pmatrix} 2n_{\max} + 2\omega_{\max} \\ 2n_{\max} \end{pmatrix}. \tag{2.74}$$

Already for $\omega_{\max} = 2$ the bounds (2.73) and (2.74) are of second and fourth degree in $n_{\max}$, whereas (2.71) and (2.72) are linear and quadratic in $n_{\max} + k_0$, respectively. Therefore we can expect a substantial reduction of the SDP relaxation under the transformation procedure. Note, we did not exploit any sparsity in the SDP relaxation or any intersection of the maximal cliques $(C_1, \ldots, C_p)$ and $(\tilde{C}_1, \ldots, \tilde{C}_p)$ when deriving these bounds. Thus, the actual size of SDP relaxations in numerical experiments may be far smaller than the one suggested by these bounds.

2.3.2 Quality of SDP relaxations for QOP

A polynomial optimization problem (POP) and the quadratic optimization problem (QOP) derived from it under one of the transformation strategies AI, AII, BI or BII are equivalent. However, the same statement does not hold for the SDP relaxations of both problems. In fact, the SDP relaxation for QOP is weaker than the SDP relaxation for the original POP. Before stating this negative result, we consider an example to illustrate it.

Example 2.7 *Let a POP and its equivalent QOP be given by*

$$\begin{array}{lllll} POP & \min & x_1^2 x_2^2 & \Leftrightarrow & QOP \\ & s.t. & x_1^2 x_2 \ge 0 & & \end{array} \qquad \begin{array}{lll} min & \tilde{x}_3^2 \\ s.t. & \tilde{x}_1 \tilde{x}_3 \ge 0 \\ & \tilde{x}_1 \tilde{x}_2 = \tilde{x}_3. \end{array}$$

The dense SDP relaxations of minimal relaxation order dSDP$_2$(POP) and dSDP$_1$(QOP) are given by

$$\begin{array}{ll}\min & y_{22}\\ s.t. & y_{21}\geq 0\end{array} \qquad \begin{array}{ll}\min & \tilde{y}_{002}\\ s.t. & \tilde{y}_{101}\geq 0\\ & \tilde{y}_{110}=\tilde{y}_{001}\end{array}$$

$$M_2(y)=\begin{pmatrix} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02}\\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12}\\ y_{01} & y_{11} & y_{02} & y_{21} & y_{13} & y_{03}\\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22}\\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13}\\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04}\end{pmatrix}\succcurlyeq 0 \qquad M_1(\tilde{y})=\begin{pmatrix} \tilde{y}_{000} & \tilde{y}_{100} & \tilde{y}_{010} & \tilde{y}_{001}\\ \tilde{y}_{100} & \tilde{y}_{200} & \tilde{y}_{110} & \tilde{y}_{101}\\ \tilde{y}_{010} & \tilde{y}_{110} & \tilde{y}_{020} & \tilde{y}_{011}\\ \tilde{y}_{001} & \tilde{y}_{101} & \tilde{y}_{011} & \tilde{y}_{002}\end{pmatrix}\succcurlyeq 0.$$

The equivalence of POP and QOP holds with the relation

$$\begin{array}{rl} & (\tilde{x}_1,\tilde{x}_2,\tilde{x}_3,\tilde{x}_1^2,\tilde{x}_1\tilde{x}_2,\tilde{x}_1\tilde{x}_3,\tilde{x}_2^2,\tilde{x}_2\tilde{x}_3,\tilde{x}_3^2)\\ = & (x_1,x_2,x_1x_2,x_1^2,x_1x_2,x_1^2x_2,x_2^2,x_1x_2^2,x_1^2x_2^2).\end{array} \tag{2.75}$$

Given a feasible solution $y\in\mathbb{R}^{d(2,4)}=\mathbb{R}^{15}$ for dSDP$_2$(POP), we exploit the relations (2.75) to define a vector $\tilde{y}=(\tilde{y}_{000},\tilde{y}_{100},\tilde{y}_{010},\tilde{y}_{001},\tilde{y}_{200},\ldots,\tilde{y}_{002})\in\mathbb{R}^{d(3,1)}=\mathbb{R}^{10}$ as

$$\tilde{y}:=(y_{00},y_{10},y_{01},y_{11},y_{20},y_{11},y_{21},y_{02},y_{12},y_{22}).$$

Then $\tilde{y}_{110}=y_{11}=\tilde{y}_{001}$ holds by definition of $\tilde{y}$, and $\tilde{y}_{101}=y_{21}\geq 0$ as y is a feasible solution of dSDP$_2$(POP). Furthermore, for the moment matrix, we have

$$M_1(\tilde{y})=\begin{pmatrix} y_{00} & y_{10} & y_{01} & y_{11}\\ y_{10} & y_{20} & y_{11} & y_{21}\\ y_{01} & y_{11} & y_{02} & y_{12}\\ y_{11} & y_{21} & y_{12} & y_{22}\end{pmatrix}.$$

Thus $M_1(\tilde{y})\succcurlyeq 0$, as $M_1(\tilde{y})$ is a principal submatrix of $M_2(y)\succcurlyeq 0$. It follows that $\tilde{y}$ is feasible for dSDP$_1$(QOP) and that $\min(dSDP_1(QOP))\leq\min(dSDP_2(POP))$ holds.

A generalization of the observation in Example 2.7 is given by the following proposition.

Proposition 2.1 *Let a POP of dimension n with $\omega_{\max}>1$ of form (2.1) be given by the set of polynomials $(p,g_1,\ldots,g_m)$ and the corresponding QOP of dimension $n+k$ derived via AI, AII, BI or BII by $(\tilde{p},\tilde{g}_1,\ldots,\tilde{g}_m)$. Then, for each feasible solution of dSDP$_{\omega_{\max}}$(POP), there exists a feasible solution of dSDP$_1$(QOP) with the same objective value. Thus, $\min$(dSDP$_1$(QOP))$\leq\min$(dSDP$_{\omega_{\max}}$(POP)).*

Proof:
Let $y\in\mathbb{R}^{d(n,2\omega_{\max})}$ be a feasible solution of SDP$_{\omega_{\max}}$(POP). Each y_α corresponds to a monomial x^α for all α with $\mid\alpha\mid\leq 2\omega_{\max}$, $x\in\mathbb{R}^n$. Moreover, with respect to the substitution relation for all monomials $\tilde{x}^\alpha\in\mathbb{R}^{n+k}$ with $\mid\alpha\mid\leq 2$ there exists a monomial $x^{\beta(\alpha)}\in\mathbb{R}^n$, such that

$$\tilde{x}^\alpha=x^{\beta(\alpha)},\mid\beta(\alpha)\mid\leq 2\omega_{\max}. \tag{2.76}$$

As $\beta(\cdot)$ in (2.76) is constructed via the substitution relations,

$$\beta(\alpha_1)=\beta(\alpha_2) \tag{2.77}$$

holds for $\alpha_1,\alpha_2\in\mathbb{N}^{n+k}$ with $\mid\alpha_1\mid=\mid\alpha_2\mid\leq 2$, whenever QOP has a substitution constraint $\tilde{x}^{\alpha_1}=\tilde{x}^{\alpha_2}$. Now, define $\tilde{y}\in\mathbb{R}^{d(n+k,2)}$ where $\tilde{y}_\alpha:=y_{\beta(\alpha)}$ for all $\mid\alpha\mid\leq 2$. Then, $\tilde{y}$ is feasible for dSDP$_1$(QOP), as all equality constraints derived from substitutions are satisfied due to (2.77), and as the principal submatrices of moment matrix $M_1(\tilde{y})$ and of the localizing matrices $M_0(\tilde{y}\tilde{g}_k)$ $(k=1,\ldots,m)$, which are obtained by simultaneously deleting rows/columns linear dependent on the remaining rows/columns, are principal

submatrices of $M_{\omega_{\max}}(y)$ and $M_{\omega_{\max}-\omega_k}(y\, g_k)$ (k=1,...,m), respectively. Finally, the objective values for y and $\tilde{y}$ coincide. □

This result for the dense SDP relaxation can be extended to the sparse SDP relaxation of minimal relaxation order in an analog manner, if the maximal cliques $(\tilde{C}_1, \ldots, \tilde{C}_m)$ of the chordal extended csp graph of the QOP are chosen appropriately with respect to the maximal cliques of the chordal extended csp graph of the POP. Therefore it seems reasonable to expect that in general sSDP_1 for the QOP provides an approximation to the global minimum of the POP which is far weaker than the one provided by $\text{sSDP}_{\omega_{\max}}$ for the original POP. One possibility to strengthen the SDP relaxation for QOPs is to increase the relaxation order to some $\omega > 1$. But, as in the case of the SDP relaxation for POP we can not guarantee to find the global minimum of a general QOP for any fixed $\omega \in \mathbb{N}$. Moreover each of the additional equality constraints results in $\frac{1}{2}(d(n,\omega-1)+1)d(n,\omega-1)$ equality constraints in $\text{sSDP}_\omega(\text{QOP})$ for $\omega > 1$. Therefore it seems more promising to consider additional techniques to improve the quality of the sSDP_1 for QOPs.

Local optimization methods

As pointed out before, the minimum of the sparse SDP relaxation converges to the minimum of the QOP for $\omega \to \infty$. Moreover, an accurate approximation can be obtained by the sparse SDP relaxation of order $\omega \in \{\omega_{\max}, \ldots, \omega_{\max}+3\}$ for many POPs [102]. However, the quality of the sparse SDP relaxation for the QOP is weaker than the one for the original POP. Therefore, the solution provided by the sparse SDP relaxation for the QOP can be understood as a first approximation to the global minimizer of the original POP, and it may serve as initial point for a locally convergent optimization technique applied to the original POP. For instance sequential quadratic programming (SQP) [8] can be applied to POP where the sparse SDP solution for the corresponding QOP is taken as starting point. In the case a POP has equality constraints only, the number of constraints coincides with the number of variables and the feasible set is finite, we may succeed in finding the global optimizer of the POP by applying Newton's method for nonlinear systems [77] to the polynomial system given by the feasible set of the POP, again starting from the solution provided by the sparse SDP relaxation for the QOP.

Higher accuracy via Branch-and-Cut bounds

The sparse SDP relaxations (2.18) incorporate lower and upper bounds for each component of the n-dimensional variable,

$$\text{lbd}_i \le x_i \le \text{ubd}_i, \qquad \forall i \in \{1,\ldots,n\}, \tag{2.78}$$

in order to establish the compactness of the feasible set of a POP. Compactness is a necessary condition to guarantee the convergence of the sequence of sparse SDP relaxations towards the global optimum of the POP. Moreover, the numerical performance for solving the sparse SDP relaxations depends heavily on the bounds (2.78). The tighter we choose these bounds, the better approximates the solution of the SDP the minimizer of the POP. Prior to solving the sparse SDP relaxations for the QOP derived from a POP, we fix the bounds (2.78) for the components of the POP and determine lower and upper bounds for the additional variables according to the substitution relation. For instance for $x_{n+1} = x_i^2$ the bounds are defined as

$$\begin{aligned} \text{lbd}_{n+1} &= \begin{cases} 0, & \text{if } \text{lbd}_i \le 0 \le \text{ubd}_i \\ \min(\text{lbd}_i^2, \text{ubd}_i^2), & \text{else,} \end{cases} \\ \text{ubd}_{n+1} &= \max(\text{lbd}_i^2, \text{ubd}_i^2). \end{aligned} \tag{2.79}$$

In 2.3.3 we will discuss the sensitivity of the choice of the lower and upper bounds on the accuracy of the SDP solution for some example POPs.

A more sophisticated technique to increase the quality of the SDP relaxation of the QOP is inspired by a Branch-and-Cut algorithm for bilinear matrix inequalities due to Fukuda and Kojima [22]. As nonconvex quadratic constraints can be reduced to bilinear ones, we are able to adapt this technique for a QOP derived from a higher degree POP. The technique is based on cutting the feasible region of the SDP, such that every feasible solution of the QOP remains feasible for the SDP. We distinguish two sets of constraints which

resemble the convex relaxations (5) proposed in [22]. Let $(p, g_1, \ldots, g_m)$ be a QOP with lower and upper bounds lbd_i and ubd_i for all components x_i $(i = 1, \ldots, n)$. The first set of constraints we consider is the following. For each constraint g_i $(i = 1, \ldots, m)$ of form $x_k = x_i x_j$ with $i \neq j$ we add the following constraints to the QOP

$$\begin{array}{rl} x_k & \leq \mathrm{ubd}_j x_i + \mathrm{lbd}_i x_j - \mathrm{lbd}_i \mathrm{ubd}_j \\ x_k & \leq \mathrm{lbd}_j x_i + \mathrm{ubd}_i x_j - \mathrm{ubd}_i \mathrm{lbd}_j. \end{array} \tag{2.80}$$

For each constraint of the form $x_k = x_i^2$ we add the following constraint to the QOP

$$x_k \leq (\mathrm{ubd}_i + \mathrm{lbd}_i)\, x_i - \mathrm{lbd}_i \mathrm{ubd}_i. \tag{2.81}$$

The second set of constraints shrinks the feasible set of the SDP relaxation even further than the constraints (2.80) and (2.81). For each monomial $x_i x_j$ of degree 2 which occurs in the objective p or one of the constraints g_i $(i = 1, \ldots, m)$ of the QOP, we add constraints as follows. If the QOP contains a constraint g_i $(i = 1, \ldots, m)$ of the form $x_k = x_i x_j$, we add the constraints (2.80) for $i \neq j$ and (2.81) for $i = j$. If the QOP does not contain a constraint $x_k = x_i x_j$, we add the quadratic constraints

$$\begin{array}{rl} x_i x_j & \leq \mathrm{ubd}_j x_i + \mathrm{lbd}_i x_j - \mathrm{lbd}_i \mathrm{ubd}_j \\ x_i x_j & \leq \mathrm{lbd}_j x_i + \mathrm{ubd}_i x_j - \mathrm{ubd}_i \mathrm{lbd}_j \end{array} \tag{2.82}$$

for $i \neq j$ and the constraint

$$x_i^2 \leq (\mathrm{ubd}_i + \mathrm{lbd}_i)\, x_i - \mathrm{lbd}_i \mathrm{ubd}_i \tag{2.83}$$

for $i = j$. When linearized, both, the linear constraints (2.80) and (2.81) and the quadratic constraints (2.82) and (2.83) result in a smaller feasible region of the SDP relaxation which still contains the feasible region of the QOP. The efficiency of these sets of additional constraints is demonstrated in Section 2.3.3 as well.

Remark 2.8 *A general QOP given by $(p, g_1, \ldots, g_m)$ of dimension n can be transformed into a QOP of dimension $n+1$ with linear objective function by adding the equality constraint $h(x, x_{n+1}) := x_{n+1} - p(x) = 0$ and choosing x_{n+1} as objective. A QOP with linear objective is a special case of a quadratic SDP (2.64), which we can apply the SDP relaxation (a) - (d) from 2.2.7 to. Thus, an arbitrary POP can be attempted by three different SDP relaxations. 1) The sparse SDP relaxations (2.18) exploiting correlative sparsity applied directly to the POP, 2) the sparse SDP relaxations (2.18) applied to an equivalent QOP, and 3) the SDP relaxations (a)-(d) from 2.2.7 exploiting d- and/or - r-space sparsity applied to an equivalent QOP.*

Remark 2.9 *The constraints (2.82) are a particular case of* **reformulation-linearization-techniques** *(RLT) [89, 90]. For the SDP relaxation (b) from 2.2.7, instead of the constraints (2.80) - (2.83) we impose RLT constraints in the following way: Add the constraints*

$$\begin{array}{rl} W_{i,j} - lbd_i W_{1,j} - lbd_j W_{1,i} & \geq -lbd_i\, lbd_j, \\ W_{i,j} - ubd_i W_{1,j} - ubd_j W_{1,i} & \geq -ubd_i\, ubd_j, \\ W_{i,j} - lbd_i W_{1,j} - ubd_j W_{1,i} & \leq -lbd_i\, ubd_j, \\ W_{i,j} - lbd_j W_{1,i} - ubd_i W_{1,j} & \leq -lbd_j\, ubd_i, \end{array} \tag{2.84}$$

to the SDP relaxation (b), if $\{i, j\}$ subset of some clique of the chordal extension of the d-space sparsity pattern graph. The constraints (2.84) strengthen the SDP relaxation and preserve the d-space sparsity structure at the same time. In the latter, whenever we apply the SDP relaxations (b) from 2.2.7 to a QOP, we impose the constraints (2.84).

2.3.3 Numerical examples

The substitution procedure and the sparse SDP relaxations are applied to a number of test problems. These test problems encompass medium and large scale POPs of higher degree. The numerical performance of the sparse SDP relaxations of these POPs under the transformation algorithm is evaluated. In the following the

Substitution	$\tilde{n}$	size($A_{\rm q}$)	nnz($A_{\rm q}$)
AI	138	[777, 6934]	7106
AII	153	[828, 6922]	7116
BI	115	[753, 5785]	5934
BII	119	[788, 6497]	6653

Table 2.7: Size of sSDP_1 for QOPs from POP *randEQ(7,3,5,8,0)* with $n = 7$.

Branch-and-Cut bounds (2.80) and (2.81) are denoted as *linear BC-bounds*, (2.82) and (2.83) as *quadratic BC-bounds*. The optional application of sequential quadratic programming starting from the solution of the SDP relaxation is abbreviated as *SQP*. Given a numerical solution x of an equality and inequality constrained POP, its *scaled feasibility error* is given by

$$\epsilon_{\rm sc} = \min\{- \mid h_i(x)/\sigma_i(x) \mid,\ \min\{g_j(x)/\hat{\sigma}_j(x), 0\}\ \forall\, i,j\},$$

where h_i $(i = 1,\ldots,k)$ denote the equality constraints, g_j $(j = 1,\ldots,l)$ the inequality constraints, and σ_i and $\hat{\sigma}_j$ are the maxima of the monomials in the corresponding polynomials h_i and g_j at x, respectively. Note, an equality and inequality constrained POP is a special case of the POP (2.1), if we define $f_i := g_i$ $(i = 1,\ldots,l)$, $g_i := h_i$ $(i = l+1,\ldots,l+k)$ and $g_i := -h_i$ $(i = k+l+1,\ldots,2k+l)$. The value of the objective function at x is given by $f_0(x)$. Let N_C denote the number of constraints of a POP. 'OOM' as entry for the scaled feasibility error denotes the size of the SDP is too large to be solved by SeDuMi [95] and results in a memory error ('Out of memory'). A two-component entry for lbd or ubd indicates that the first component is used as a bound for the first $\frac{n}{2}$ variables and the second component for the remaining $\frac{n}{2}$ variables of the POP.

All numerical experiments are conducted on a LINUX OS with CPU 2.4 GHz and 8 Gb memory. The total processing time in seconds is denoted as t_C.

Randomly generated POPs

As a first class of test problems, consider randomly generated POPs with inequality or equality constraints. We are interested in the numerical performance of the sparse SDP relaxation for the corresponding QOPs for different substitution strategies and different choices of lower, upper and Branch-and-Cut bounds. We will focus on comparing strategies BI and BII as they yield POPs with a small number of additional variables.

For the random equality constrained POP *randEQ(7,3,5,8,0)* [102] of degree 8 with 7 variables, the size of the SDP relaxation sSDP_4 is described by the matrix $A_{\rm p}$ of size [2870, 95628] with 124034 non-zero entries. This size is reduced substantially under each of the four substitution strategies, as can be seen in Table 2.7. In this table the matrix $A_{\rm q}$ in SeDuMi input format [95] and its number of nonzero entries nnz(A_q) describe the size of the sparse SDP relaxation. The reduction of the size of the SDP relaxation results in reducing the total processing time t_C by two magnitudes, as can be seen in Table 2.8.

Moreover, as reported in Table 2.8, the performance of AI, AII, BI and BII is similar - with the one of BI being slightly better than the others. In this example with few equality constraints, it is easy to obtain a feasible solution, but it requires additional techniques as SQP to obtain an optimal solution. We know, $\min(\text{sSDP}_1(QOP))$ and $\min(\text{sSDP}_4(POP))$ are lower bounds for min(POP). Moreover, $\min(\text{sSDP}_1(QOP)) \leq \min(\text{sSDP}_4(POP))$ holds with Proposition 2.1. As reported in Table 2.8 the bound provided by $\text{sSDP}_1(QOP)$ is much weaker than the one provided by $\text{sSDP}_4(POP)$. Note, the objective value $f_0(x)$ and $\min(\text{sSDP}_1(QOP))$ improve significantly if the lower and upper bounds are chosen tighter. When chosen sufficiently tight, an accurate optimal solution can be achieved without applying SQP. The main advantage of the transformation is the reduction of the total processing time by two magnitudes.

The results for the inequality constrained POP *randINEQ(8,4,6,8,0)* [102] with $\omega_{\max} = 4$ and 8 variables are given in Table 2.9. In the column for (lbd, ubd) the entries $(-0.5, 0.5)^\star$ and $(-0.5, 0.5)^{\star\star}$ denote

Substitution	SQP	BC-bounds	(lbd, ubd)	ω	n or $\tilde{n}$	N_C	$\epsilon_{\rm sc}$	$\min(\text{sSDP}_\omega)$	$f_0(x)$	t_C
-	no	none	$(-\infty,\infty)$	4	7	4	6e-11	-0.708	-0.708	333
AI	no	none	(-1,1)	1	138	135	1e-13	-247.50	-0.508	3
AII	no	none	(-1,1)	1	153	150	1e-13	-254.92	-0.517	3
BI	no	none	(-1,1)	1	115	112	1e-13	-299.11	-0.567	2
BII	no	none	(-1,1)	1	119	116	1e-13	-284.98	-0.455	3
BI	yes	none	(-1,1)	1	115	112	7e-18	-299.11	-0.708	3
BI	no	none	(-0.5,0.5)	1	115	112	9e-14	-6.55	-0.706	3
BI	no	none	(-0.3,0.3)	1	115	112	1e-13	-1.28	-0.708	2

Table 2.8: Results for SDP relaxation of *randEQ(7,3,5,8,0)*.

$\text{ubd}_2 = 0.75 \neq 0.5$ and $(\text{ubd}_2, \text{ubd}_5) = (0.75, 0) \neq (0.5, 0.5)$, respectively. By imposing linear Branch-and-Cut bounds we obtain a feasible solution, and tightening lbd and ubd improves the objective value of the approximative solution. Though we did not achieve the optimal value attained by sSDP(POP), it seems reasonable to expect that successively tightening the bounds further yields a feasible solution with optimal objective value. As in the previous example t_C could be reduced by two magnitudes.

Substitution	SQP	BC-bounds	(lbd, ubd)	ω	n or $\tilde{n}$	N_C	$\epsilon_{\rm sc}$	$f_0(x)$	t_C
-	no	none	$(-\infty,\infty)$	4	8	3	0	-1.5	1071
BI	no	none	$(-0.75, 0.75)$	1	239	234	-1.3	-0.9	14
BI	no	linear	$(-0.75, 0.75)$	1	239	680	0	-0.6	17
BI	no	linear	$(-0.5, 0.5)^\star$	1	239	680	0	-0.8	17
BI	no	linear	$(-0.5, 0.5)^{\star\star}$	1	239	680	0	-1.2	16

Table 2.9: Results for SDP relaxation of *randINEQ(8,4,6,8,0)*.

BroydenBand

Another test problem is the *BroydenBand(n)* problem [66]. It is an unconstrained POP of degree 6 and dimension n, and its global minimum is 0. Numerical results are given in Table 2.10. The performance of the sparse SDP relaxation for the QOP with initial bounds and without applying SQP is poor, the optimal value of the approximate solution and the lower bounds $\min(\text{sSDP}_1(QOP))$ are far from the global optimum. Also, SQP does not succeed in detecting the global optimum if started from an arbitrary starting point. As reported in Table 2.10, SQP detects a local minimizer with objective 3, if the initial point is an SDP solution with loose bounds for the QOP. It is interesting to observe that tight lower and upper bounds, and Branch-and-Cut bounds in combination with applying SQP are crucial to obtain the global minimum by solving the sparse SDP relaxation for the QOP. In fact, when we apply substitution strategy BI imposing quadratic Branch-and-Cut bounds yields the global minimum, whereas in the case of applying BII Branch-and-Cut bounds are not necessary to obtain the global minimum. Note, the total processing time is reduced from around 1300 seconds to less than 5 seconds.

POPs derived from partial differential equations

An important class of large scale polynomial optimization problems of higher degree is derived from discretizing systems of partial differential equations (PDE). How to derive POPs from PDEs and how to interpret their solutions is the topic of Chapter 3 and discussed in detail there. In this section we show the

Substitution	SQP	BC-bounds	(lbd, ubd)	ω	n or $\tilde{n}$	N_C	min(sSDP$_\omega$)	$f_0(x)$	t_C
-	no	none	$(-\infty, +\infty)$	3	20	0	-3e-7	5e-9	1328
BII	yes	none	(-1, 1)	1	60	40	-128	3	4
BII	yes	linear	(-1, 1)	1	60	100	-128	3	4
BII	yes	quadratic	(-1, 1)	1	60	1244	-106	3	4
BI	no	none	(-0.75,0)	1	80	60	-611	47	3
BI	no	linear	(-0.75,0)	1	80	60	-611	47	4
BI	no	quadratic	(-0.75,0)	1	80	60	-132	28	4
BI	yes	none	(-0.75, 0)	1	80	60	-1396	3	4
BI	yes	linear	(-0.75, 0)	1	80	140	-611	3	5
BI	yes	quadratic	(-0.75, 0)	1	80	1284	-611	6e-8	5
BII	no	none	(-0.75, 0)	1	60	40	-26	33	3
BII	no	linear	(-0.75, 0)	1	60	100	-24	24	3
BII	no	quadratic	(-0.75, 0)	1	60	1244	-8	9	4
BII	yes	none	(-0.75, 0)	1	60	40	-26	1e-10	5
BII	yes	linear	(-0.75, 0)	1	60	100	-24	1e-6	4
BII	yes	quadratic	(-0.75, 0)	1	60	1244	-8	2e-7	5

Table 2.10: Results for SDP relaxation for *BroydenBand(20)*.

transformation procedure from POP to QOP to be a very efficient technique for this class of POPs. Many POPs derived from PDE are of degree 3, but as the number of their constraints is in the same order as the number of variables, transformation into QOPs yields SDP relaxations of vastly reduced size. Due to the structure of the higher degree monomial set of these POPs, there is an unique way to transform them into QOPs. Therefore, we examine the impact of lower, upper and Branch-and-Cut bounds and not the choice of the substitution strategy.

POP	n	$\tilde{n}$	ω_p	size(A_p)	nnz(A_p)	ω_q	size(A_q)	nnz(A_q)
pdeBifurcation(6)	36	72	2	[2186, 17605]	23801	1	[422, 4039]	4174
pdeBifurcation(10)	100	200	2	[16592, 139245]	189737	1	[1643, 18646]	19039
pdeBifurcation(14)	196	392	2	[454497, 3822961]	5208475	1	[4126, 45189]	46000
Mimura(50)	100	150	2	[3780, 31258]	39068	1	[690, 5728]	6078
Mimura(50)	100	150	3	[19300, 280007]	354067	2	[7223, 76383]	91755
Mimura(100)	200	300	3	[39100, 565357]	713767	2	[14623, 155183]	186155
Mimura(100)	200	300	2	[7630, 63158]	78818	2	[1390, 11628]	12328
StiffDiff(6,12)	144	216	2	[18569, 163162]	219020	1	[878, 6700]	7402
ginzOrDiri(9)	162	324	2	[74628, 666987]	906558	1	[4567, 49305]	50233
ginzOrNeum(11)	242	484	2	[166092, 1451752]	2504418	1	[8063, 96367]	97776

Table 2.11: Size of the SDP relaxation for POP and QOP, respectively.

Consider the POPs in Table 2.11, where ω_p and ω_q the relaxation order of sSDP$_\omega$ for POP and QOP, respectively, to demonstrate the reduction of the size of the SDP relaxation described by the size of the matrix A in SeDuMi input format [95] and its number of nonzero entries nnz(A). Thus, the SDP relaxations for the QOPs can be solved in vastly shorter time than the one for the original POPs. The computational results of the original SDP relaxation and the SDP relaxation of the QOPs for different lower, upper and Branch-and-Cut bounds are reported in Table 2.12 for the POP *pdeBifurcation(·)*. In this example the accuracy of the sparse SDP relaxation for the QOP is improved by tightening the upper bounds for the

components of the variable $\tilde{x}$ in the QOP. Also, the additional application of SQP improves the accuracy a lot. Additional Branch-and-Cut bounds seem to have no impact on the quality of the solution. The total processing time is reduced substantially under the transformation. The original sparse SDP relaxation for *pdeBifurcation(14)* of dimension 200 cannot be solved in SeDuMi due to a memory error, but the SDP relaxation for the corresponding QOP with tight upper bounds can be solved accurately in 100 seconds.

POP	Substitution	SQP	BC-bounds	ubd	ω	n or $\tilde{n}$	$\epsilon_{\rm sc}$	$f_0(x)$	t_C
pdeBifurcation(6)	-	no	none	0.99	2	36	8e-11	-9.0	14
pdeBifurcation(6)	AI	no	none	0.99	1	72	9.6e-2	-22.1	2
pdeBifurcation(6)	AI	no	linear	0.99	1	72	9.6e-2	-22.1	2
pdeBifurcation(6)	AI	no	quadratic	0.99	1	72	9.6e-2	-22.1	2
pdeBifurcation(6)	AI	yes	none	0.99	1	72	7.3e-9	-9.0	5
pdeBifurcation(6)	AI	no	none	0.45	1	72	1.5e-2	-9.5	1
pdeBifurcation(6)	AI	yes	none	0.45	1	72	1.4e-11	-9.0	2
pdeBifurcation(10)	-	no	none	0.99	2	100	3.1e-10	-21.6	2159
pdeBifurcation(10)	AI	no	none	0.99	1	200	4.7e-2	-56.0	20
pdeBifurcation(10)	AI	yes	none	0.99	1	200	2.7e-13	-21.6	66
pdeBifurcation(10)	AI	no	none	0.45	1	200	6.4e-3	-23.2	13
pdeBifurcation(10)	AI	yes	none	0.45	1	200	1e-11	-21.6	22
pdeBifurcation(14)	-	no	none	0.99	1	196	OOM	-	-
pdeBifurcation(14)	AI	no	none	0.99	1	392	2.4e-2	-103.1	90
pdeBifurcation(14)	AI	yes	none	0.99	1	392	7.9e-14	-39.9	418
pdeBifurcation(14)	AI	no	none	0.45	1	392	3.6e-3	-43.1	85
pdeBifurcation(14)	AI	yes	none	0.45	1	392	5.2e-11	-39.9	107

Table 2.12: Results for SDP relaxation for POP *pdeBifurcation* with lbd=0.

In the case of POP *Mimura(50)*, c.f. Table 2.13, quadratic Branch-and-Cut bounds are necessary in addition to applying SQP, in order to obtain an accurate approximate solution of the global minimizer. In the POPs in Table 2.14 it is sufficient to apply SQP starting from the solution of the sparse SDP relaxation for the QOP. For these problems t_C can be reduced by up to two magnitudes. Furthermore, the original SDP relaxation for *ginzOrDiri(9)* and *ginzOrDiri(13)* is too large to be solved, whereas the SDP relaxations for the QOPs are tractable.

POP	Substitution	SQP	BC-bounds	ubd	ω	n or $\tilde{n}$	$\epsilon_{\rm sc}$	$f_0(x)$	t_C
Mimura(50)	-	no	none	$[11, 14]$	2	100	1.8e-1	-899	20
Mimura(50)	-	yes	none	$[11, 14]$	2	100	4.1e-9	-701	31
Mimura(50)	AI	no	none	$[11, 14]$	1	150	6.1e-1	-1067	2
Mimura(50)	AI	yes	none	$[11, 14]$	1	150	5.1e-3	-731	163
Mimura(50)	AI	no	quadratic	$[11, 14]$	1	150	3.3e-1	-1017	2
Mimura(50)	AI	yes	quadratic	$[11, 14]$	1	150	1.0e-13	-719	16
Mimura(100)	-	no	none	$[11, 14]$	3	200	4.5e-2	-733	532
Mimura(100)	-	yes	none	$[11, 14]$	3	200	2.0e-11	-712	557

Table 2.13: Results for SDP relaxation for POP *Mimura* with lbd = $[0, 0]$.

The POP *ginzOrNeum(·)* in Table 2.15 is another example where the global optimizer can be found in a processing time reduced by a factor 100, if the lower bounds lbd and upper bounds ubd are chosen sufficiently tight and SQP is applied. In Table 2.13 and Table 2.15 the first components of lbd and ubd correspond to the lower and upper bounds for $(x_1, \ldots, x_{\frac{n}{2}})$, respectively, whereas the second components correspond to the lower and upper bounds for $(x_{\frac{n}{2}+1}, \ldots, x_n)$.

POP	Substitution	SQP	ubd	ω	n or $\tilde{n}$	$\epsilon_{\rm sc}$	$f_0(x)$	t_C
ginzOrDiri(5)	-	no	0.6	2	50	6e-6	-25	598
ginzOrDiri(5)	-	yes	0.6	2	50	4e-15	-25	598
ginzOrDiri(5)	AI	no	0.6	1	100	3e-1	-100	7
ginzOrDiri(5)	AI	yes	0.6	1	100	4e-11	-22	10
ginzOrDiri(9)	-	no	0.6	2	162	OOM	-	-
ginzOrDiri(9)	AI	no	0.6	1	324	1e-1	-324	144
ginzOrDiri(9)	AI	yes	0.6	1	324	6e-12	-72	185
ginzOrDiri(13)	-	no	0.6	2	338	OOM	-	-
ginzOrDiri(13)	AI	yes	0.6	1	676	7e-9	-158	1992
StiffDiff(4,8)	-	yes	5	2	64	2e-11	-32	54
StiffDiff(4,8)	AI	yes	5	2	96	7e-10	-32	4
StiffDiff(6,12)	-	yes	5	1	144	4e-9	-71	1008
StiffDiff(6,12)	AI	yes	5	1	216	8e-10	-71	48

Table 2.14: Results for SDP relaxation for POP *ginzOrDiri* with lbd =0 and *StiffDiff* with lbd=0.

POP	Substitution	SQP	lbd	ubd	ω	n	$\epsilon_{\rm sc}$	$f_0(x)$	t_C
ginzOrNeum(5)	-	no	$[0, 0]$	$[4, 2]$	2	50	2.6	-47	448
ginzOrNeum(5)	-	yes	$[0, 0]$	$[4, 2]$	2	50	2e-13	-45	449
ginzOrNeum(5)	AI	no	$[0, 0]$	$[4, 2]$	1	100	24	-100	9
ginzOrNeum(5)	AI	yes	$[0, 0]$	$[4, 2]$	1	100	8e-10	-45	10
ginzOrNeum(5)	-	no	$[1, 0.5]$	$[4, 1.5]$	2	50	1e-1	-45	582
ginzOrNeum(5)	-	yes	$[1, 0.5]$	$[4, 1.5]$	2	50	2e-13	-45	583
ginzOrNeum(5)	AI	no	$[1, 0.5]$	$[4, 1.5]$	1	100	6e-2	-57	6
ginzOrNeum(5)	AI	yes	$[1, 0.5]$	$[4, 1.5]$	1	100	4e-10	-45	7
ginzOrNeum(11)	-	no	$[1, 0.5]$	$[4, 1.5]$	2	242	OOM		
ginzOrNeum(11)	AI	no	$[1, 0.5]$	$[4, 1.5]$	1	484	4e-2	-263	740
ginzOrNeum(11)	AI	yes	$[1, 0.5]$	$[4, 1.5]$	1	484	5e-11	-207	748

Table 2.15: Results for SDP relaxation for POP *ginzOrNeum*.

Chapter 3

SDP Relaxations for Solving Differential Equations

3.1 Numerical analysis of differential equations

Differential equations arise in models of many problems in engineerings, physics, chemistry, biology or economics. Only the simplest differential equations allow to find solutions given by explicit formulas. In most problems involving differential equations self-contained formulas for the solutions are not available. Therefore, one is interested in finding approximations to their solutions by applying numerical methods. In general we distinguish ordinary differential equations (ODE), which are differential equations where the unknown function is a function of a single variable, and partial differential equations (PDE), which are differential equations where the unknown function is a function of multiple independent variables and the equation involves its partial derivatives. Moreover, we distinguish linear and nonlinear differential equations. A differential equations is linear if the unknown function and its derivatives appear to the power one and nonlinear otherwise.
The beginning of numerical analysis of ODE dates back to 1850 when *Adams formulas* were proposed, which are based on polynomial interpolation in equally spaced points. The idea is, given some initial value problem with ODE $u' = f(t, u)$ for $t > t_0$ and $u(t_0) = u_0$, we choose a time step $\Delta t > 0$ and consider a finite set of time values

$$t_n = t_0 + n\Delta t,\ n \geq 0.$$

We then replace the ODE by an algebraic expression that enables us to calculate a succession of approximate values

$$v_n \approx u(t_n),\ n \geq 0,$$

where the simplest such approximate formula dates back to Euler:

$$v_{n+1} = v_n + \Delta t\, f(t_n, v_n) = v_n + \Delta t f_n,\ \ f_n := f(t_n, v_n).$$

The Adams formulas are higher order generalizations of Euler's formula that are far more efficient at generating accurate approximate solutions. For instance, the fourth-order Adams-Bashworth formula is

$$v_{n+1} = v_n + \frac{1}{24}\Delta t\left(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}\right). \tag{3.1}$$

The formula (3.1) is fourth order in the sense that it will normally converge at the rate $O\left((\Delta t)^4\right)$. The second important class of ODE algorithms are the *Runge-Kutta methods*, which were developed at the beginning of the twentieth century [50, 86]. The most commonly used member of the family of Runge-Kutta methods is the fourth-order Runge-Kutta method, which advances a numerical solution from time

step t_n to t_{n+1} with the aid of four evaluations of the function f:

$$\begin{aligned} a &= \Delta t\, f(t_n, v_n), \\ b &= \Delta t\, f(t_n + \tfrac{1}{2}\Delta t, v_n + \tfrac{1}{2}a), \\ c &= \Delta t\, f(t_n + \tfrac{1}{2}\Delta t, v_n + \tfrac{1}{2}b), \\ d &= \Delta t\, f(t_n + \Delta t, v_n + c), \\ v_{n+1} &= v_n + \tfrac{1}{6}(a + 2b + 2c + d). \end{aligned} \tag{3.2}$$

Another seminal step in the numerical analysis of ODE is the concept of *stability* due to Dahlquist [17]. He introduced what might be called the *fundamental theorem of numerical analysis*:

$$\text{consistency} + \text{stability} = \text{convergence}.$$

This theory is based on precise definitions of these three notions. Consistency is the property that the discrete formula has locally positive order of accuracy and thus models the right ODE. Stability is the property that discretization errors introduced at one time step cannot grow unboundedly at later time steps. Convergence is the property that the numerical solution converges to the correct result as $\Delta t \to 0$.
When it comes to the numerical analysis of PDEs, we distinguish three main classes of methods: *Finite Difference Methods* (FDM), *Finite Element Methods* (FEM) and *Finite Volume Methods* (FVM). The origin of the Finite Difference Method dates back to the paper [13] of Courant, Friedrichs and Lewy. A finite difference scheme is applied to formulate PDE problems as polynomial optimization problems in 3.2. We give an introduction to the FDM in 3.1.1. The FEM dates back to the 1960s and will be briefly introduced in 3.1.2. As in the case of ODEs, stability is a crucial issue in the numerical analysis of PDEs. The group around von Neumann and Lax discovered that some finite difference methods for PDEs were subject to catastrophic instabilities. The fundamental result linking convergence and stability of a finite difference scheme is the *Lax equivalence theorem* [57].
Numerical methods for finding approximate solutions to PDEs have successfully offered insights for important and difficult examples of PDE problems: The Schrödinger equation in chemistry, elasticity equations in structural mechanics, the Navier-Stokes equations in fluid dynamics, Maxwell's equations in telecommunications, Einstein's equations in cosmology, nonlinear wave equations in optical fibers, Black-Scholes equations in option pricing, reaction-diffusion equations in biological systems. Because such a variety of nonlinear PDE problems arises in many disciplines in science and engineering, which requires to solve large-scale nonlinear algebraic systems, the numerical analysis of partial differential equations remains a very challenging field.

3.1.1 The finite difference method

All discretization based methods for solving differential equations aim at algebraizing the differential equations. In the *finite difference method* (FDM) [19, 65, 96] the most important step to algebraize the equation is achieved via replacing differentials by finite differences. In a first step the domain of the differential equations needs to be discretized. Note, the FDM requires the geometry of the domain to be simple. We will restrict ourselves to intervals $[x_{\min}, x_{\max}]$ in the one-dimensional case, and to rectangular domains $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ in the two-dimensional case. We choose a discretization N_x or (N_x, N_y), respectively and define $h_x = \frac{x_{\max} - x_{\min}}{N_x - 1}$, $h_y = \frac{y_{\max} - y_{\min}}{N_y - 1}$,

$$\begin{array}{lll} x_i := x_{\min} + (i-1)\, h_x, & y_j := y_{\min} + (j-1)\, h_y, & (i = 1, \ldots, N_x;\ j = 1, \ldots, N_y), \\ u_i := u(x_i), & u_{i,j} := u(x_i, y_j), & (i = 1, \ldots, N_x;\ j = 1, \ldots, N_y), \end{array} \tag{3.3}$$

where u denotes the unknown function in a differential equation. There are three choices to approximate the first derivate u_x at some point x_i:

$$u_x(x_i) \approx \begin{cases} \frac{u_i - u_{i-1}}{h_x}, & \text{(forward difference)} \\ \frac{u_{i+1} - u_i}{h_x}, & \text{(backward difference)} \\ \frac{u_{i+1} - u_{i-1}}{2h_x}, & \text{(central difference)} \end{cases} \tag{3.4}$$

An approximation to u_{xx} is derived by successively forming $(u_x)_x$:

$$u_{xx}(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h_x^2}. \tag{3.5}$$

Choosing these approximations the question arises what errors are inherent in substituting differentials by finite differences. The following proposition provides a simple accuracy analysis.

Proposition 3.1 *Let u be a three times continuously differentiable function on $\Omega = [x_{\min}, x_{\max}]$, with $0 \in \Omega$. It holds,*

a) $\mid \frac{u_{i+1}-u_{i-1}}{2h_x} - u_x(x_i) \mid \leq \frac{1}{6}h_x^2 \max_{x\in\Omega} \mid u_{xxx}(x) \mid$,

b) $\mid \frac{u_{i+1}-u_i}{h_x} - u_x(x_i) \mid \leq \frac{1}{2}h_x \max_{x\in\Omega} \mid u_{xx}(x) \mid$,

c) $\mid \frac{u_i-u_{i-1}}{h_x} - u_x(x_i) \mid \leq \frac{1}{2}h_x \max_{x\in\Omega} \mid u_{xx}(x) \mid$,

d) $\mid \frac{u_{i+1}-2u_i+u_{i-1}}{h_x^2} - u_{xx}(x_i) \mid \leq \frac{1}{12}h_x^2 \max_{x\in\Omega} \mid y_{xxxx}(x) \mid$.

Proof:
Without loss of generality set $x_{i-1} := -h$, $x_i := 0$, $x_{i+1} := h$.
Using Taylor's theorem we expand u_{i-1} and u_{i+1} around 0 and obtain

$$\begin{aligned} u_{i-1} &= u_i - h_x\, u_x(x_i) + \tfrac{1}{2!}h_x^2 u_{xx}(x_i) - \tfrac{1}{3!}h_x^3 u_{xxx}(\xi_1), \quad \text{for some } \xi_1 \in [x_{i-1}, x_i], \\ u_{i+1} &= u_i + h_x\, u_x(x_i) + \tfrac{1}{2!}h_x^2 u_{xx}(x_i) + \tfrac{1}{3!}h_x^3 u_{xxx}(\xi_2), \quad \text{for some } \xi_2 \in [x_i, x_{i+1}]. \end{aligned}$$

Subtracting these two equations yields

$$\frac{1}{2h_x}(u_{i+1} - u_{i-1}) = u_x(x_i) + \frac{1}{2 \cdot 3!}h_x^2[u_{xxx}(\xi_1) + u_{xxx}(\xi_2)],$$

and implies a). Now, expand u_{i+1} around 0 with second order remainder:

$$u_{i+1} = u_i + h_x\, u_x(x_i) + \frac{1}{2!}h_x^2 u_{xx}(\xi_1), \text{ for some } \xi_1 \in [x_i, x_{i+1}]$$

This equation implies b):

$$\mid \frac{1}{h_x}(u_{i+1} - u_i) - u_x(x_i) \mid \leq \frac{1}{2}h_x \max_{x\in\Omega} \mid u_{xx}(x) \mid . \tag{3.6}$$

c) is shown analogously to b).
As for d), expand u_{i+1} and u_{i-1} around $x_i = 0$ with fourth order remainder:

$$\begin{aligned} u_{i-1} &= u_i - h_x\, u_x(x_i) + \tfrac{1}{2!}h_x^2 u_{xx}(x_i) - \tfrac{1}{3!}h_x^3 u_{xxx}(x_i) + \tfrac{1}{4!}h_x^4 u_{xxxx}(\xi_1), \quad \text{for some } \xi_1 \in [x_{i-1}, x_i], \\ u_{i+1} &= u_i + h_x\, u_x(x_i) + \tfrac{1}{2!}h_x^2 u_{xx}(x_i) + \tfrac{1}{3!}h_x^3 u_{xxx}(x_i) + \tfrac{1}{4!}h_x^4 u_{xxxx}(\xi_2), \quad \text{for some } \xi_2 \in [x_i, x_{i+1}]. \end{aligned}$$

Addition of these two equations yields

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h_x^2} - u_{xx}(x_i) = \frac{1}{4!}h_x^2 \left(u_{xxxx}(\xi_1) + u_{xxxx}(\xi_2)\right),$$

which implies d). □
Proposition 3.1 implies, if $u_{xxx}(x)$ is bounded between x_{i-1} and x_{i+1}, the error replacing $u_x(x_i)$ by the central difference scheme is of order h_x^2, whereas the error of using a forward or a backward difference scheme is only $O(h_x)$.
In addition to discretizing the domain of a differential equation and approximating its differentials by finite difference schemes, one needs to take into account conditions for the unknown function on the boundary

$\partial\Omega$ of the domain. The most common types of boundary conditions are *Dirichlet* and *Neumann* conditions. Given a boundary point $x \in \partial\Omega$, its boundary condition is called *Dirichlet*, if $u(x)$ is fixed on $\partial\Omega$, and it is called *Neumann* if the partial derivative $\frac{\partial u(x)}{\partial n}$ orthogonal to $\partial\Omega$ is fixed on $\partial\Omega$. We consider *periodic boundary conditions* as a third type, which is given, if values of u at the lower and upper end of its domain are identified. For example, in the one-dimensional case with $\Omega = [x_{\min}, x_{\max}]$, periodic boundary conditions are given by $u(x_{\min}) = u(x_{\max})$. A differential equation equipped with boundary conditions on the entire $\partial\Omega$ is called a *boundary value problem*. In the one-dimensional case we have N_x and in the two-dimensional case we have $N_x\,N_y$ unknown variables u_i and $u_{i,j}$, respectively. Replacing the differential equation at each interior grid point by its finite difference discretization generates $N_x - 2$ and $(N_x - 2)(N_y - 2)$ equations, respectively. By exploiting relations between the boundary and the interior variables given by Dirichlet, Neumann or periodic boundary conditions and substituting them into the equations, we can reduce the number of variables to $N_x - 2$ and $(N_x - 2)(N_y - 2)$, respectively. Thus, the number of variables coincides with the number of equations. In the case of a linear differential equation, finding the N_xN_y variables $u_{i,j}$ is therefore equivalent to solving a system of linear equations. In the case of a nonlinear differential equation, the far more challenging problem of a system of nonlinear algebraic equations needs to be solved.
As mentioned in the introduction of this section, in order to establish the convergence $u_{i,j} \to u(x_i, y_j)$ $(1 \le i \le N_x, 1 \le j \le N_y)$ for $(N_x, N_y) \to \infty$, the notions of *consistency* and *stability* are crucial. Let

$$\begin{array}{ll} D(u(x,y)) & = f(x,y) \quad \forall\,(x,y) \in \Omega, \\ B(u(x,y)) & = g(x,y) \quad \forall\,(x,y) \in \partial\Omega, \end{array} \tag{3.7}$$

a differential equation, where $D(\cdot)$ and $B(\cdot)$ are differential operators and f, g functions on Ω. Applying a finite difference discretization to (3.7) yields the system of equations

$$\begin{array}{ll} D_{i,j}((u_{k,l})_{k,l}) & = f_{i,j} \quad \forall\,(i,j) \in \{2,\ldots,N_x-1\} \times \{2,\ldots,N_y-1\}, \\ B_{i,j}((u_{k,l})_{k,l}) & = g_{i,j} \quad \forall\,(i,j) \in \{1,N_x\} \times \{1,\ldots,N_y\} \cup \{1,\ldots,N_x\} \times \{1,N_y\}, \end{array} \tag{3.8}$$

where $f_{i,j} := f(x_i, y_j), g_{i,j} := g(x_i, y_j)$, $D_{i,j}$ and $B_{i,j}$ finite difference approximations for the operators D and B, respectively.

Definition 3.1 *For the finite difference discretization (3.8) of the PDE problem (3.7) we define the* **local truncation error** $r_{i,j}$ *as*

$$r_{i,j} := D_{i,j}((u(x_k, y_l))_{k,l}) - f_{i,j},$$

where $(u(x_k, y_l))_{k,l}$ *the vector of values of the exact solution* u *of (3.7) at the grid points* (x_k, y_l)*, which is approximated by the solution* $(u_{k,l})_{k,l}$ *of (3.8). The finite difference equation (3.8) is* **consistent** *with the original equation (3.7), if* $r_{i,j} \to 0$ *as* $(h_x, h_y) \to (0, 0)$.

Consistency is a prerequisite for $u_{i,j}$ to converge to $u(x_i, y_j)$ as $(h_x, h_y) \to 0$, but it is not sufficient. We need to introduce the notion of *stability* of a difference scheme for that purpose:

Definition 3.2 *A finite difference scheme* $D_{i,j}((u_{k,l})_{k,l}) = f_{i,j}$ *for a first order PDE is* **stable** *if there is a* $J \in \mathbb{N}$ *and positive numbers* h_{x_0} *and* h_{y_0} *such that there exists a constant* C *for which*

$$h_x \sum_{l=1}^{N_y} \mid u_{k,l} \mid^2 \le C h_x \sum_{j=1}^{J} \sum_{l=1}^{N_y} \mid u_{j,l} \mid^2$$

for $k \in \{1,\ldots,N_x\}$, $0 < h_x \le h_{x_0}$, *and* $0 < h_y \le h_{y_0}$.
A finite difference scheme $D_{i,j}((u_{k,l})_{k,l}) = f_{i,j}$ *for a PDE which is second order in* x *is* **stable** *if there is a* $J \in \mathbb{N}$ *and positive numbers* h_{x_0} *and* h_{y_0} *such that there exists a constant* C *for which*

$$h_x \sum_{l=1}^{N_y} \mid u_{k,l} \mid^2 \le (1 + k^2)\, C h_x \sum_{j=1}^{J} \sum_{l=1}^{N_y} \mid u_{j,l} \mid^2$$

for $k \in \{1,\ldots,N_x\}$, $0 < h_x \le h_{x_0}$, *and* $0 < h_y \le h_{y_0}$.

For characterizing stability the following notion is useful.

Definition 3.3 *An* **explicit** *finite difference scheme is any scheme that can be written as*

$$u_{k+1,l} = \text{ a finite sum of } u_{r,s} \text{ with } r \leq k.$$

A nonexplicit scheme is called **implicit**.

In general, it is stability of a finite difference scheme which requires some effort to show, whereas consistency is straightforward. The oldest and most famous criterion for stability is the *Courant-Friedrichs-Lewy condition*:

Theorem 3.1 *For an explicit scheme for the hyperbolic PDE defined by*

$$u_x + a\, u_y = 0 \tag{3.9}$$

of the form $u_{k+1,l} = \alpha u_{k,l-1} + \beta u_{k,l} + \gamma u_{k,l+1}$*, a necessary condition for stability is the* **Courant-Friedrichs-Lewy (CFL) condition**,

$$\mid a \frac{h_x}{h_y} \mid \leq 1.$$

Proof: [13]
Moreover, Courant, Friedrichs and Lewy derived the general result:

Theorem 3.2 *There are no explicit, unconditionally stable, consitent finite difference schemes for hyperbolic systems (3.9) of partial differential equations.*

There is no general negative result like Theorem 3.2 for implicit schemes. Thus, from a stability point of view, it is advisable to choose central or backward difference approximations for the first order derivates. The result which links consistency, stability and convergence is the *Lax-Richtmyer Equivalence Theorem*:

Theorem 3.3 *A consistent finite difference scheme for a linear partial differential equation of first or second order for which the initial value problem is well-posed is convergent if and only if it is stable.*

Proof: [96]
Thus, convergence of a finite difference scheme is usually proven by showing stability. There is no general convergence result for finite difference schemes of nonlinear partial differential equations. However, for certain classes of PDEs convergence of the FDM has been shown:

Theorem 3.4 *Let a parabolic PDE problem be given by*

$$\begin{array}{lll} a(x,y)\, u_{xx} + d(x,y)\, u_x - u_y + f(y,u) & = 0 & \forall\, (x,y) \in (0,1) \times (0,T), \\ u_x(0,y) = u_x(1,y) & = 0 & \forall\, y \in (0,T), \\ u(x,0) & = u^0(x) & \forall\, x \in (0,1). \end{array}$$

If a*,* a_x*,* a_y*,* b*,* b_x *and* b_y *continuous in* $[0,1] \times [0,T]$*, there exists an* a_0 *s.t.* $a \geq a_0$ *in* $[0,1] \times [0,T]$*,* f*,* f_u *and* f_{uu} *continuous in* $[0,T] \times \mathbb{R}$*, there exists* $M_0 \in \mathbb{R}$ *s.t.* $\partial f / \partial u \leq M_0$ *in* $[0,T] \times \mathbb{R}$*, some technical conditions hold and the solution* u *of the PDE problem is smooth, then*
$(u_{i,j}(N_x, N_y))_{i,j}$ *converges uniformly to* u *in* $[0,1] \times [0,T]$ *as* $(N_x, N_y) \to \infty$.

Proof: Theorem 2.1 in [97].
Note, Theorem 3.4 can be extended to the case of arbitrary rectangular domains and arbitrary Dirichlet and Neumann conditions at $x_{\min}$ and $x_{\max}$. To proof convergence of classes of elliptic or hyperbolic PDE problems is far more difficult and no result for a broad class of problems like Theorem 3.4 has been found in those cases.
Some simple accuracy analysis of finite difference approximations was provided in Proposition 3.1, where we have seen that the accuracy of central differences is better than the one of forward and backward

differences. On the other hand, it is a well known phenomenon that central difference approximations for the first derivatives may cause oscillations in the numerical solution of a boundary value problem [65]. These oscillations do not occur under forward or backward difference schemes. However, we have also discussed that a forward finite difference scheme may not be stable, whereas implicit finite difference schemes are unconditionally stable for many classes of PDE problems. Thus, when choosing a difference scheme we have to consider accuracy, stability and how to avoid oscillations. It depends on the PDE problem which finite difference scheme is "the best" one. Therefore, numerous difference schemes have been proposed, which may use different difference approximations on certain segments of the domain, may use different difference approximations in the different dimensions, may depend of the type of the PDE or may depend of the type of boundary conditions. Detailed analysis of these issues for a variety of finite difference schemes for different classes of PDE problems is discussed in detail in [96].
To summarize, when solving a linear or nonlinear ODE or PDE problem with the FDM, the three main tasks are 1) to choose a finite difference scheme whose solutions are accurate approximations for solutions of the PDE problem, 2) to show convergence of the difference scheme, and 3) to solve the (nonlinear) algebraic system of equations (3.8). To solve a system of nonlinear equations is a very hard problem in general. As mentioned in 2.1 solving a system of nonconvex polynomial equations is NP-hard. A standard method to solve the system of algebraic equations resulting from a finite difference approximation of a nonlinear PDE is to solve the system of equations corresponding to the linear part of the PDE, and to take the solution of the linear system as a starting point for gradient type methods, Newton's method or other iterative methods applied to the nonlinear system or to a system where the nonlinear part is successively increased. The eventual success of such a continuation type method is base on the assumption that the solution of nonlinear system does not change much if the nonlinear part is increased by a small factor. In this thesis we attempt problem 3) by reformulating (3.8) as a polynomial optimization problem (POP) and solving this POP by sparse semidefinite programming relaxation techniques introduced in Chapter 2. One of the main advantages of this approach is the fact, that we do not require any initial guess and the nonlinearity of the scheme is taken into account directly. This is presented in detail in 3.2.

3.1.2 The finite element method and other numerical solvers

The finite element method

In addition to the finite difference method which our technique to be proposed in 3.2 is based on, we briefly introduce alternative methods for solving a PDE problem numerically. The most important one is the finite element method (FEM). The FEM is a very active field of research and there is exhaustive literature on it. For example, see [11, 59, 106] for detailed introductions and more advanced studies. The FEM is based on the idea to approximate a solution u of a PDE problem by a function $\tilde{u}$ which is an element of a finite-dimensional subspace of the function space u belongs to. I.e., the FEM can be understood as a method which discretizes the space we are searching for solutions of a PDE problem, whereas in the FDM the PDE is discretized. The origins of the FEM date back to works of Rayleigh [83], Ritz [84] and Galerkin [23] at the beginning of the 20th century. The FEM in its modern formulation is due to Courant [16] and Turner, Clough, Martin and Topp [99], among others. Typically, a FEM approach to solve a PDE consists of the following steps. First of all one is looking for a solution u for a PDE problem defined on a domain Ω in a certain function space. The most common function space to this end is the Sobolov space $H_0^s(\Omega) \subset L^s(\Omega)$. Given that function space, one replaces the PDE problem by a weak, variational formulation where the test functions are elements of the same space $H_0^s(\Omega)$. In a second step known as *meshing* the domain Ω is partitioned into a finite number of subdomains of simple geometry, which are called *elements*. We denote such a partition by $\mathcal{T}$. In the one-dimensional case intervals, in the two-dimensional case triangles, and in the three-dimensional case tetrahedra are a common choice for the elements. These elements define a mesh for Ω with n_d nodes. Then, in a third step $H_0^s(\Omega)$ is approximated by the n_d-dimensional subspace which is spanned by functions $f_1, \ldots, f_{n_d}$. A common choice for this basis are for instance piece-wise linear functions f_i which equal one at the node i and are 0 at all other nodes. The larger n_d, i.e., the finer we choose the mesh, the better approximates $\text{span}(f_1, \ldots, f_{n_d})$ the space $H_0^s(\Omega)$. When replacing $H_0^s(\Omega)$ by $\text{span}(f_1, \ldots, f_{n_d})$ in the weak formulation of the PDE problem and approximating u by $\tilde{u} = \sum_{i=1}^{n_d} d_i f_i$,

ones obtains a finite number of equations in the unknowns $d_1, \dots, d_{n_d}$. Solving this system of equations yields the numerical approximation $\tilde{u}$ for a solution of the original PDE problem. Finally, as for the Finite Difference Method, convergence of a finite element discretization needs to be shown, i.e., ones has to show that $\text{span}(f_1, \dots, f_{n_d})$ converges to a subspace dense in $H_0^s(\Omega)$ if the number n_d of nodes in the mesh and the corresponding number of basis functions goes to infinity. One of the biggest advantages of the FEM is its sound mathematical basis. As the PDE is formulated as a variational problem one has lots of powerful tools from functional analysis at hand to proof convergence of a finite element discretization. Let us demonstrate the outlined procedure on the following, simple ODE problem. Find $u \in H_0^2(\Omega)$ such that

$$\begin{array}{ll} u''(x) & = g(x) \qquad \text{on } x \in \Omega := [0,1], \\ u(0) & = u(1) = 0. \end{array} \tag{3.10}$$

Its weak formulation is given by

$$\int_0^1 u''(x)\,\phi(x)dx = \int_0^1 g(x)\,\phi(x)dx \quad \forall\, \phi \in H_0^2(\Omega), \tag{3.11}$$

a partition is given by $\mathcal{T}_h := \{x_i := i\,h \mid i \in \{0, \dots, \frac{1}{h}\}\}$ for any $h > 0$ with $\frac{1}{h} \in \mathbb{N}$. With this partition the nodes of the mesh are given by $x_1, \dots, x_{\frac{1}{h}-1}$, i.e., $n_d = \frac{1}{h} - 1$. We define the finite dimensional subspace $V_h = \text{span}(f_1, \dots, f_{n_d})$ of $H_0^2(\Omega)$ via the basis functions $f_i : \Omega \to \mathbb{R}$ with $f_i(x_j) := \delta_{i,j}$ and f_i linear on each interval (x_j, x_{j+1}). Then, let $\tilde{u} := \sum_{i=1}^{n_d} d_i\, f_i$ satisfy the finite-dimensional relaxation of the weak formulation (3.11):

$$\begin{array}{lll} & \int_0^1 \left(\sum_{i=1}^{n_d} d_i\, {f''}_i(x)\right) \phi(x)\,dx = \int_0^1 g(x)\phi(x)\,dx & \forall\, \phi \in V_h \\ \Leftrightarrow & \sum_{i=1}^{n_d} d_i \int_0^1 {f''}_i(x)\, f_j(x)\,dx = \int_0^1 g(x) f_j(x)\,dx & \forall\, j \in \{1, \dots, n_d\} \\ \Leftrightarrow & \frac{d_{j-1} - 2d_j + d_{j+1}}{h} = \int_0^1 g(x) f_j(x)\,dx & \forall\, j \in \{1, \dots, n_d\} \end{array} \tag{3.12}$$

which is a system of linear equations in $d_1, \dots, d_{n_d}$. Solving it provides a numerical approximation $\tilde{u}$ to a weak solution u of ODE (3.10). If we moreover assume $g(x) = g$ constant on $[0,1]$, we obtain the system of equations

$$\frac{d_{j-1} - 2d_j + d_{j+1}}{h^2} = g \quad \forall\, j \in \{1, \dots, n_d\}. \tag{3.13}$$

With the definition of the basis functions f_i it is clear, in this example holds $d_i = f(x_i)$ for all $i \in \{1, \dots, n_d\}$. Thus, (3.13) is actually identical to the system of equations we obtain when approximating (3.10) by a finite difference scheme. However, this connection with finite difference methods does not hold in general. The Finite Element Method provides the user with a great deal of freedom, such as how to choose the finite dimensional subspace to approximate $H_0^s(\Omega)$ or the mesh for Ω, and for most other finite element discretizations there is no equivalent finite difference scheme.
When comparing the FEM to the FDM, we already mentioned its sound basis from functional analysis as one main advantage. Another one is, it is highly flexible for different domains of PDE problems: complicated geometries can be dealt with easily, whereas the FDM is restricted to relatively simple geometries. On the other hand, the FDM is far easier to implement than the FEM for many PDE problems arising in applications. However, in both methods one of the greatest challenges is to solve a system of nonlinear algebraic equations, in the case they are applied to a system of nonlinear differential equations. In general it is highly dependent on the PDE problem to be solved numerically which method, FDM or FEM, provides a better approximation to the continuous world when choosing a similar discretization.

Other numerical solvers

Beside the finite difference method and the finite element method another important class of methods to solve differential equations numerically is the *finite volume method* (FVM). Similar to the FDM, values are calculated at discrete points on a mesh. In the FVM these values are derived from calculating volume integrals over small volumes around the node points of the mesh. The FVM applies the divergence theorem

to convert volume into surface integrals, and it exploits, that a flux entering a given, small volume is equal to that leaving the volume. Each integration over a volume results in an algebraic equation. Thus, as in the FDM and FEM a system of algebraic equations needs to be solved to obtain a numerical solution. The FVM is popular for solving hyperbolic PDE problem, in particular in computational fluid dynamics. For a detailed introduction, see [58].
The *Spectral method* [27] is a class of techniques involving the use of the Fast Fourier Transform. It is suitable for PDE problems with very smooth solutions and provides highly accurate approximations for these problems. It is based on replacing the unknown function in the PDE by its Fourier series to get a system of ODEs in the time-dependent coefficients of the Fourier series. The spectral method and the FEM share the idea to approximate the solution of a PDE by a linear combination of basic functions. The difference being these basic functions in the Spectral method are nonzero over the entire domain, whereas the basic functions of the FEM are nonzero on small subdomains only. For this reason, the spectral method can be understood as a global approximation approach, whereas the FEM constitutes a local approximation approach.
There are numerous other methods, such as multigrid methods, domain decomposition methods, level-set methods or meshfree methods. As PDE problems arise from very different settings and applications, it is highly dependent on a particular PDE which numerical method is most suitable for providing an accurate approximation. In all numerical methods a potentially hard system of algebraic equations need to be solved. In the next section we will attempt this problem for the FDM by sparse semidefinite programming relaxation and polynomial optimization techniques.

Remark 3.1 *We have seen, in each discretization-based method for nonlinear PDEs a system of algebraic equations needs to be solved. The classical tool for a system of nonlinear equations is Newton's method, which converges locally quadratic. However, Newton's method requires a starting point close to a solution of the system, in order to converge. For difficult nonlinear problems it may be a very challenging problem to find a good initial guess. There are various techniques to find a good initial guess for Newton's method or other locally fast convergent techniques. One is to apply gradient methods or other first order techniques to find a rough approximation of a solution. Another one is to apply some homotopy-like continuation method, where the nonlinear problem is linearized and a solution of the linear problem is taken as initial point for a problem where the weight of the nonlinear part is increased incrementally. Finally, in many problems partial information of the solution, which may be obtained by numerical simulation, is utilized to get a sufficiently close initial guess. We will show in the following that SDP relaxation for polynomial problems are very useful to find a good initial guess for locally fast convergent methods.*

3.2 Differential equations and the SDPR method

3.2.1 Transforming a differential equation into a sparse POP

In the previous section we gave an introduction to the numerical analysis of differential equations. In Chapter 2 we introduced polynomial optimization problems and their semidefinite programming (SDP) relaxations. In this section we will see how to transform a problem involving differential equations into a polynomial optimization problem (POP), in order to apply SDP relaxations to solve these differential equations numerically. For discretizing a differential equation we choose the Finite Difference Method (FDM). The FDM has the advantage of being easy to implement. Moreover, applying the FDM to a differential equation yields a sparse POP, as we will show in this section. In the following we will restrict ourselves to discussing the two-dimensional case. However, the procedures for differential equations with domains of different dimension are derived analogously. Recall, a general differential equation is given by

$$\begin{array}{lll} D(u(x,y)) & = f(x,y) & \forall\ (x,y) \in \Omega, \\ B(u(x,y)) & = g(x,y) & \forall\ (x,y) \in \partial\Omega, \end{array} \tag{3.14}$$

where $D(\cdot)$ and $B(\cdot)$ are differential operators and f, g functions on $\Omega := [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$. Applying the FDM yields the system of equations

$$\begin{array}{ll} D_{i,j}((u_{k,l})_{k,l}) & = f_{i,j} \quad \forall\, (i,j) \in \{2, \ldots, N_x - 1\} \times \{2, \ldots, N_y - 1\}, \\ B_{i,j}((u_{k,l})_{k,l}) & = g_{i,j} \quad \forall\, (i,j) \in \{1, N_x\} \times \{1, \ldots, N_y\} \cup \{1, \ldots, N_x\} \times \{1, N_y\}, \end{array} \tag{3.15}$$

where $f_{i,j} := f(x_i, y_j)$, $g_{i,j} := g(x_i, y_j)$, $D_{i,j}$ and $B_{i,j}$ finite difference approximations for the operators D and B, respectively. We assume the operators D and B are both polynomial in $u(\cdot, \cdot)$ and its derivates, which implies (3.15) is a system of polynomial equations. Note, we can reduce the dimension of this system of equations by exploiting the boundary conditions. In their most basic form Dirichlet, Neumann and periodic boundary conditons at $x_{\min}$ are given by

$$u_{1,j} = g_{1,j}, \qquad u_{1,j} = g_{1,j} h_x + u_{2,j} \quad \text{and} \quad u_{1,j} = u_{N_x,j}, \tag{3.16}$$

respectively. If we substitute the $u_{i,j}$ corresponding to the boundary grid points in (3.15) by the terms given in (3.16), the number of variables in each direction is reduced by 2 in the case of Dirichlet and Neumann boundary conditions, or by 1 in the case of periodic boundary conditions. Thus, (3.15) is reduced to a system of n equations in n variables,

$$\hat{D}_{i,j}((u_{k,l})_{k,l}) - \hat{f}_{i,j} = 0 \quad \forall\, (i,j) \in \{1, \ldots, \hat{N}_x\} \times \{1, \ldots, \hat{N}_y\}, \tag{3.17}$$

where $n := \hat{N}_x \hat{N}_y$. For instance under Neumann condition in x direction and periodic condition in y direction, n is given by $n = \hat{N}_x \hat{N}_y = (N_x - 2)(N_y - 1)$. For simplicity of notation we will denote a solution of the original PDE problem (3.14) as $u(\cdot, \cdot)$ and a solution $(u_{k,l})_{k,l}$ of (3.17) as the vector $u \in \mathbb{R}^{\hat{N}_x \hat{N}_y}$, with

$$u = (u_{1,1}, \ldots, u_{1,\hat{N}_y}, u_{2,1}, \ldots, u_{\hat{N}_x,\hat{N}_y}).$$

As (3.15) is polynomial in the variable u, so is (3.17). We attempt to solve this system of equations by transforming it into a POP of type,

$$\begin{array}{lll} \min & p(u) & \\ \text{s.t.} & g_i(u) \geq 0 & \forall i = 1, \ldots, m, \\ & h_j(u) = 0 & \forall j = 1, \ldots, k. \end{array} \tag{3.18}$$

Note, (3.18) is a special case of (2.1), as an equality constraint $h(x) = 0$ is equivalent to the pair of inequality constraints $h(x) \geq 0$ and $-h(x) \geq 0$. Given a PDE problem (3.14), we take (3.17) derived from it as system of equality constraints for an optimization problem. Moreover, we choose lower and upper bounds of type (2.25),

$$\text{lbd}_{i,j} \leq u_{i,j} \leq \text{ubd}_{i,j} \qquad \forall\, (i,j) \in \{1, \ldots, \hat{N}_x\} \times \{1, \ldots, \hat{N}_y\}. \tag{3.19}$$

Choosing an objective function

To derive a POP, it remains to choose an objective function F that is polynomial in u as well. The choice of an appropriate objective function is dependent on the PDE problem we are aiming to solve. In case there is at most one solution of the PDE problem, we are interested in the feasibility of the POP we construct. Thus any objective is a priori acceptable for that purpose. However, the accuracy of obtained solutions may depend on the particular objective function. In the case the solution of the PDE problem is not unique, the choice of the objective function determines the particular solution to be found. The objective function may correspond to a physical quantity a solution needs to optimize. For instance, in problems in fluid dynamics one is often interested in finding a solution of minimal kinetic energy. For such a problem we obtain the objective function by discretizing the kinetic energy function. A large class of PDEs which occur in many applications can be written as Euler-Lagrange equations. A typical case is a stable state equation of reaction-diffusion type. In this case, a canonical choice is a discretization of the corresponding energy integral as in 3.3.1. Another case is optimal control: A finite difference discretization of state and control constraints yields the feasible set and a discretization of the optimal value function yields the objective. We discuss numerical examples for optimal control problem in 3.3.6.

Additional polynomial constraints

We mentioned above, it is crucial to add lower and upper bounds (3.19) for each $u_{i,j}$ when constructing a POP from a differential equation. When choosing these bounds care has to be taken. A choice of lbd and ubd which is too tight may exclude solutions of (3.17) from the feasible set of the POP, while a choice which is too loose may cause inaccurate results. In addition to those constraints we may impose further inequality or equality constraints

$$g_l(u) \geq 0 \quad \text{and} \quad h_j(u) = 0, \tag{3.20}$$

respectively, with $g_l, h_j \in \mathbb{R}[u]$. Constraints (3.20) can be understood as restrictions for the admissible space of functions we search for solutions of a differential equation. One possibility to obtain such bounds is derived by constraining the partial derivatives. We call bounds of this type **variation bounds**. For the derivative in $x-$direction they are given by

$$|\, \frac{\partial u(x_i, y_j)}{\partial x} \,| \leq M \qquad \forall i \in \{2, \ldots, N_x - 1\},\ \forall j \in \{2, \ldots, N_y - 1\}. \tag{3.21}$$

Expression (3.21) can be transformed into polynomial constraints easily. Another possibility is to impose bounds in the spirit of [22] like (2.80), (2.81), (2.82) and (2.83) introduced in 2.3. Add

$$\begin{array}{ll} u_{s,t} & \leq \mathrm{ubd}_{k,l} u_{i.j} + \mathrm{lbd}_{i,j} u_{k,l} - \mathrm{lbd}_{i,j}\mathrm{ubd}_{k,l} \\ u_{s,t} & \leq \mathrm{lbd}_{k,l} u_{i,j} + \mathrm{ubd}_{i,j} u_{k,l} - \mathrm{ubd}_{i,j}\mathrm{lbd}_{k,l}. \end{array} \tag{3.22}$$

for each constraint $u_{s,t} = u_{i,j}u_{k,l}$ in the POP, and for each constraint $u_{s,t} = u_{i,j}^2$ we add

$$u_{s,t} \quad \leq (\mathrm{ubd}_{i,j} + \mathrm{lbd}_{i,j})\, u_{i,j} - \mathrm{lbd}_{i,j}\mathrm{ubd}_{i,j}. \tag{3.23}$$

If there is no quadratic constraint of this type for (i, j, k, l) or (i, j), respectively, we may add the quadratic constraints

$$\begin{array}{ll} u_{i,j}u_{k,l} & \leq \mathrm{ubd}_{k,l} u_{i,j} + \mathrm{lbd}_{i,j} u_{k,l} - \mathrm{lbd}_{i,j}\mathrm{ubd}_{k,l} \\ u_{i,j}u_{k,l} & \leq \mathrm{lbd}_{k,l} u_{i,j} + \mathrm{ubd}_{i,j} u_{k,l} - \mathrm{ubd}_{i,j}\mathrm{lbd}_{k,l} \end{array} \tag{3.24}$$

for $(i, j) \neq (k, l)$ and the constraint

$$u_{i,j}^2 \quad \leq (\mathrm{ubd}_{i,j} + \mathrm{lbd}_{i,j})\, u_{i,j} - \mathrm{lbd}_{i,j}\mathrm{ubd}_{i,j} \tag{3.25}$$

for $(i, j) = (k, l)$. Note, by the construction of the constraints (3.22) - (3.25), they shrink the feasible set of an SDP relaxation for a POP, but they do not change the feasible set of the POP. I.e., they may be added to improve the numerical accuracy of SDP relaxations for solving the POP, but they have no impact on the space of functions we are searching for discrete approximations to a solution for a differential equation.

A POP derived from a differential equation

If we take together all constraints and the chosen objective function, we obtain the following POP:

$$\begin{array}{lllll} \min & F(u) & & & \\ \text{s.t.} & D_{i,j}(u) & = f_{i,j} & \forall\, (i,j) \in \{1, \ldots, \hat{N}_x\} \times \{1, \ldots, \hat{N}_y\}, \\ & g_l(u) & \geq 0 & \forall\, l \in \{1, \ldots, s\}, \\ & h_k(u) & = 0 & \forall\, k \in \{1, \ldots, m\}, \\ & \mathrm{lbd}_{i,j} & \leq u_{i,j} \leq \mathrm{ubd}_{i,j} & \forall\, (i,j) \in \{1, \ldots, \hat{N}_x\} \times \{1, \ldots, \hat{N}_y\}. \end{array} \tag{3.26}$$

Every feasible solution u of (3.26) is a solution of the finite difference scheme for the PDE problem (3.14). Let demonstrate how to derive (3.26) for an example.

Example 3.1 *Consider the nonlinear elliptic PDE*

$$\begin{array}{rll} u_{xx}(x,y) + u_{yy}(x,y) + \lambda u(x,y)\ \left(1 - u(x,y)^2\right) & = 0 & \forall\, (x,y) \in [0,1]^2, \\ u(x,y) & = 0 & \forall\, (x,y) \in \partial[0,1]^2, \\ 0 \leq u(x,y) & \leq 1 & \forall\, (x,y) \in [0,1]^2, \end{array} \tag{3.27}$$

where the parameter λ is set to $\lambda = 22$. We apply the standard finite difference discretization for $\hat{N} = \hat{N}_x = \hat{N}_y$, choose $F(u) = -\sum_{1\le i,j\le \hat{N}} u_{i,j}$ as objective function, and obtain the POP

$$\begin{array}{lll} \min & -\sum_{1\le i,j\le \hat{N}} u_{i,j} & \\ s.t. & \frac{1}{h_x^2}\left(u_{i+1,j}+u_{i,j+1}+u_{i,j-1}+u_{i-1,j}-4u_{i,j}\right)+22u_{i,j}\left(1-u_{i,j}^2\right) & = 0 \quad \forall\,(i,j)\in\{1,\dots,\hat{N}\}^2, \\ & 0 \le u_{i,j} & \le 1 \quad \forall\,(i,j)\in\{1,\dots,\hat{N}\}^2, \end{array} \tag{3.28}$$

where $u_{0,k} = u_{k,0} = u_{\hat{N}+1,k} = u_{k,\hat{N}+1} = 0$ for all $k \in \{1,\dots,\hat{N}\}$. The choice for F is motivated by the fact, that (3.27) is known to have one strictly positive and the trivial solution. The optimal solution of (3.28) is a discrete approximation to the strictly positive solution of (3.27).

Correlative sparsity

In 3.2.2 we will introduce a method to solve (3.26) by sparse SDP relaxations. In order to apply this method efficiently, we need to show (3.26) satisfies a structured sparsity pattern. To show *correlative sparsity* is straight-forward:

Proposition 3.2 *Let two differential operators D_1 and D_2 be given by*

$$D_1(u) := a(u)u_{xx} + c(u)u_{yy} + d(u)u_x + e(u)u_z + \tilde{f}(u)$$

and

$$D_2(u) := a(u)u_{xx} + b(u)u_{xy} + c(u)u_{yy} + d(u)u_x + e(u)u_z + \tilde{f}(u),$$

where $a(\cdot),\dots,\tilde{f}(\cdot)$ polynomial in the function $u(\cdot)$. Let $\hat{N} = \hat{N}_x = \hat{N}_y$ and $n := \hat{N}^2$ Let F be linear function in u. Let $n_z(R)$ denote the number of nonzero entries in the CSP matrix R of the POP (3.26). Then,

$$n_z(R) \le 13n,$$

if (3.26) derived from (3.15) with $D := D_1$, and

$$n_z(R) \le 25n,$$

if (3.26) derived from (3.15) with $D := D_2$, for any choice of B and f. This implies, (3.26) is correlatively sparse in both cases.

Proof:
As F linear, the objective function does not cause any nonzero entries in R by Definition 2.5. Due to the finite difference discretization at most 12 unknown $u_{k,l}$ can occur in some equality constraint with a particular unknown $u_{i,j}$ for $D = D_1$, as pictured in Figure 3.1. Hence the maximum number of nonzero elements in the row of R corresponding to $u_{i,j}$ is 13, which implies $n_z(R) \le 13n$. With the same argument holds $n_z(R) \le 25n$ for $D = D_2$; see Figure 3.1. These bounds are tight; they are attained in the case of periodic conditions for x and y. □

Let R' denote the $n \times n$ matrix corresponding to the graph $G(N, E')$, which is a chordal extension of CSP graph $G(N, E)$. For the computational efficiency it is also useful to know whether R' is sparse or not. $n_z(R')$ depends on the employed ordering method P for R, which is used to avoid fill-ins in the symbolic sparse Cholesky factorization LL^T of the ordered matrix PRP^T. R' is constructed as $R' = L + L^T$. We examine two different methods of ordering R, the **symmetric minimum degree (SMD) ordering** and **reverse Cuthill-McKee (RCM) ordering**. See [24] for details about these orderings.

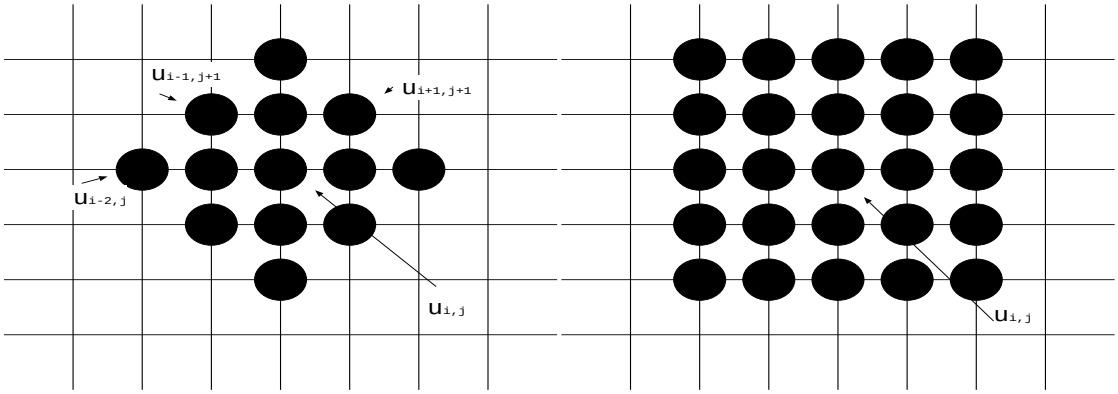


Figure 3.1: $u_{k,l}$ involved in some constraint with $u_{i,j}$ for $D = D_1$ (left) and $D = D_2$ (right).

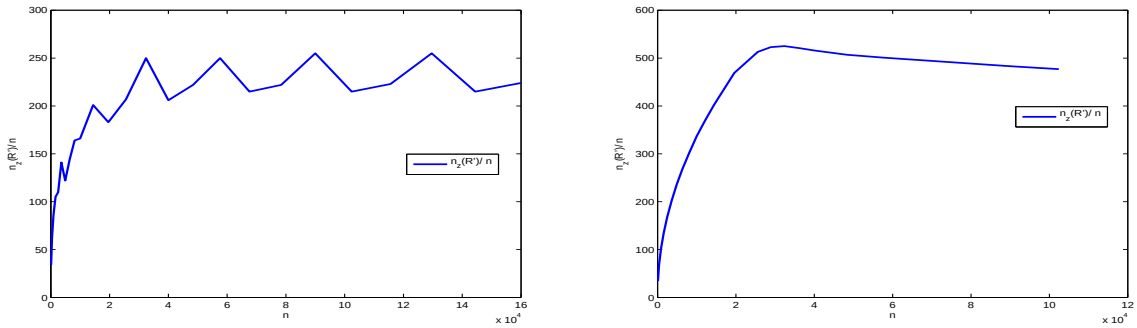


Figure 3.2: $\frac{n_z(R')}{n}$ for SMD (left) and RCM (right) ordering if $D = D_1$.

We conduct some numerical experiments, in order to estimate the behavior of $n_z(R')$. Figure 3.3 shows examples of R' after SMD and RCM ordering, and Figure 3.2 shows $\frac{n_z(R')}{n}$ obtained by the SMD and RCM orderings for the $n \times n$-matrix R, respectively, for $D = D_1$ and Dirichlet or Neumann condition in x and periodic condition in y. For $n \in [100, 160000]$ it holds $\frac{n_z(R')}{n} \leq 300$ for SMD ordering and $\frac{n_z(R')}{n} \leq 600$ for RCM ordering, respectively. The behavior of $\frac{n_z(R')}{n}$ may suggest $n_z(R') = O(n)$ for both ordering methods. Hence we expect the sparse SDP relaxations to be efficient for solving (3.26) in numerical experiments. However, since the constants 300 and 600 are large, we can not always expect a quick solution of the sparse SDP relaxation.

Domain-space sparsity

In 2.2 we introduced the concept of domain-space and range-space sparsity of an optimization problem with matrix variables. Moreover, in 2.2.7 we constructed some linear SDP relaxations for quadratic SDP exploiting this sparsity. A QOP is a special case of a quadratic SDP. As all constraints in a QOP are scalar, it does not satisfy a range-space sparsity pattern. Thus, if we transform (3.26) into a QOP, we can apply the relaxations (a) or (b) from 2.2.7 to find approximate solutions for (3.26). In order to apply these relaxations efficiently, the question arises, whether QOPs derived from (3.26) satisfy domain-space sparsity. For certain classes of PDE problems, we obtain the following sparsity results.

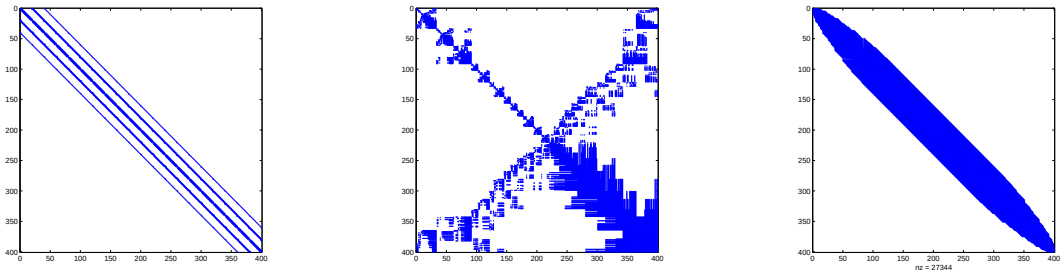


Figure 3.3: R (left), R' obtained by SMD (center) and RCM (right) orderings for $D = D_1$ with $n = 400$.

Example 3.2 *Given a rectangular domain* Ω, $f : \Omega \to \mathbb{R}$ *and differential operator* $B(\cdot)$. *Define the operators*

$$\begin{aligned} D_1(u) := &\quad au_{xx} + cu_{yy} + du_x + eu_y + gu + hu^2, \\ D_2(u) := &\quad au_{xx} + cu_{yy} + du_x + eu_y + gu + hu^3, \end{aligned}$$

with $a, c, d, e, g, h : \Omega \to \mathbb{R}$. *Moreover, choose* F *linear in* u *when constructing (3.26) for a discretization* $n = \hat{N}_x \hat{N}_y$. *In the case* $D = D_1$ *(3.26) is a quadratic SDP, in the case* $D = D_2$ *we need to apply the method from 2.3 to (3.26) to transform it into a quadratic SDP. In fact, for the example with* $D = D_2$ *there is an unique way to transform the POP into an QOP by defining* n *variables* $v_{i,j} := u_{i,j}^2$. *Then, it is easy to see that the domain-space sparsity patterns of the quadratic SDP corresponding to (3.26) for* $D = D_1$ *and* $D = D_2$, *respectively, is given by Figure 3.4.*

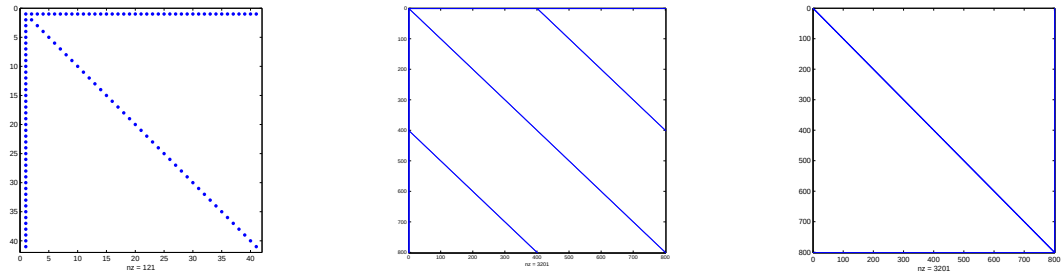


Figure 3.4: d-space sparsity pattern of (3.26) for $D = D_1(left)$, and $D = D_2$ before (center) and after (right) reordering of rows and columns.

Note, for the two cases in Example 3.2 the number of nonzero entries in every row but the first and last one of the domain-space sparsity pattern matrix is less or equal to two and three, respectively. This is considerably smaller than the upper bound 13 for the number of nonzero entries in each row of the correlative sparsity pattern matrix provided by Proposition 3.2. Likewise the size of an average maximal clique of the chordal extension of the domain-space sparsity pattern graph is far smaller than the one of the chordal extension of the correlative sparsity pattern graph. Thus, the primal SDP relaxation (b) from 2.2.7 for an QOP derived from a PDE of one of the two classes in Example 3.2 is far smaller than the sparse SDP relaxation (2.18) of relaxation order $\omega = 1$ for the same QOP, and can be solved for much finer discretizations. However, in the primal SDP relaxation (b) there is no relaxation order we can increase,

and there is no general result how well the primal SDP relaxation approximates a QOP. The sparse SDP relaxations (2.18) provide a sequence of SDPs whose minima converge to the optimum of a QOP. In fact, their approximation accuracy is improving monotonously for increasing order ω. Therefore, as we will see in numerical examples in 3.3, the primal SDP relaxation (b) is only useful for QOPs, where the solution of the primal SDP relaxation is a good approximation of an optimal solution of the QOP.

3.2.2 The SDPR method

In this section we introduce the method to solve the POP (3.26) derived from a PDE problem (3.14) to obtain discrete approximations to solutions of (3.14). In order to derive (3.26) from a PDE problem, we need to choose a discretization (N_x, N_y), the bounds lbd and ubd, the objective F, if not given by the PDE problem, and possibly additional polynomial constraints g_l and h_k. If (3.26) is a POP of degree three or larger, we can either apply the sparse SDP relaxations (2.18) for some relaxation order ω, or we apply one of the heuristics AI, AII, BI, BII from 2.3.1 to transform (3.26) into a QOP. To the QOP we apply either the sparse SDP relaxations (2.18) with relaxation order $\omega = 1$ or the primal SDP relaxation (b) exploiting domain-space sparsity from 2.2.7. Solving the SDP relaxations for the POP or the QOP, we obtain a first approximation $\hat{u}$ to an optimal solution of (3.26). The solution $\hat{u}$ can be used as an initial guess for locally fast convergent methods. One possibility is to apply sequential quadratic programming (SQP) [8] to (3.26), another one is to apply Newton's method for nonlinear systems [77] to (3.17), both starting from $\hat{u}$, in order to obtain a more accurate discrete approximation u to a solution of the PDE problem (3.14). This procedure is called the **semidefinite programming relaxation (SDPR) method** for solving a PDE problem of type (3.14), it is summarized in the following chart:

Method 3.1 ***(SDPR method)***

I. Given a PDE problem (3.14), choose (N_x, N_y), lbd, ubd, F, g_l and h_k to derive (3.26).

II. If POP (3.26) of degree three or larger, we may apply AI, AII, BI or BII to transform it into a QOP. Then, apply $sSDP_1$ (2.18) or relaxation (b) from 2.2.7 to this QOP. Denote the first n components of the solution vector of the applied SDP relaxation as $\hat{u}$.

III. If POP (3.26) of degree three or larger which has not been transformed into a QOP, choose relaxation order $\omega \geq \omega_{\max}$, apply $sSDP_\omega$ (2.18) to that POP and obtain its solution $\hat{u}$.

IV. Apply Newton's method to (3.17) or SQP to (3.26), both with initial guess $\hat{u}$, and obtain u as an approximate to an optimal solution of (3.26) and as a discrete approximation for a solution of the PDE problem (3.14).

Recall, when applying the SDP relaxation (b) from 2.2.7 to a QOP, we impose the additional constraints (2.84) as explained in Remark 2.9. For locally fast convergent methods we consider SQP and Newton's method, which we describe briefly in the following. However, we are by no means restricted to these two for choosing an iterative method which is fast convergent towards an highly accurate discrete approximation of a solution to (3.17) when starting from a guess close to the accurate approximation.

Newton's method

The discretized PDE (3.17) is a special case of the problem

$$r(x) = 0,$$

where $r : \mathbb{R}^n \to \mathbb{R}^n$, $r(x) = [r_1(x), \ldots, r_n(x)]^T$ and $r_i : \mathbb{R}^n \to \mathbb{R}$ are smooth functions for all $i \in \{1, \ldots, n\}$. The functions r_i may be nonlinear in x. The basic form of Newton's method for solving nonlinear equations

is given by

NEWTON Choose x_0;
for $k = 0, 1, 2, \ldots$
Calculate a solution p_k to the Newton equations
$$J(x_k)p_k = -r(x_k);$$
$x_{k+1} = x_k + p_k$;
end (for)

This algorithm is motivated by the multidimensional Taylor's theorem.

Theorem 3.5 *Suppose that $r : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable in some convex open set D and that x and $x + p$ are vectors in D. We then have that*

$$r(x + p) = r(x) + \int_0^1 J(x + tp)pdt.$$

$J(x + tp)$ is the Jacobian of r at $x + tp$, it is defined as

$$J(x) = \left[\frac{\partial r_j}{\partial x_i}\right]_{i,j=1,\ldots,n} = \begin{bmatrix} \nabla r_1(x)^T \\ \vdots \\ \nabla r_n(x)^T \end{bmatrix}.$$

We define a linear model $M_k(p)$ of $r(x_k + p)$ given in Theorem 3.5, i.e., we approximate the second term on the right-hand-side by $J(x)p$, and write

$$M_k(p) = r(x_k) + J(x_k)p.$$

The vector $p_k = -J(x_k)^{-1}r(x_k)$ satisfies $M_k(p_k) = 0$. It is equivalent to the solution p_k of the Newton equations in the NEWTON algorithm. As shown in [77], if x_0 close to a nondegenerate root $x^\star$ and r continuously differentiable, then the sequence $(x_k)_k$ of Newton's method converges superlinearly to $x^\star$. If r furthermore Lipschitz continuously differentiable, the convergence is quadratic.

Sequential Quadratic Programming

The POP (3.26) is a special case of the nonlinear programming problem

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & h(x) = 0, \\ & g(x) \leq 0, \end{array} \tag{3.29}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ three times continuously differentiable. The basic idea of SQP is to model (3.29) at a given approximate solution x_k by a quadratic program, and to use the solution to this subproblem to construct a better approximation x_{k+1} to the solution of (3.29). This method can be viewed as the natural extension of Newton's method to the constrained optimization setting. It shares with Newton's method the property of rapid convergence, when the iterates are close to the solution, and possible erratic behavior, when the iterates are far from the solution. SQP has two key features: First, it is not a feasible point method, its iterates x_{k+1} do not need to be feasible for (3.29). Second, in each iteration of an SQP approach a quadratic program is to be solved, which is not too demanding since highly efficient procedures for quadratic programs, i.e., programs with quadratic objective and linear constraints, exist. Let the Lagrange function of (3.29) be given by $L(x, u, v)$ and a quadratic subproblem by

$$\begin{array}{ll} \min & \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T B_k(x - x_k) \\ \text{s.t.} & \nabla h(x_k)^T(x - x_k) + h(x_k) = 0, \\ & \nabla g(x_k)^T(x - x_k) + g(x_k) \leq 0, \end{array} \tag{3.30}$$

where B_k an approximation of the Hessian of the Lagrangian function at the iterate (x_k, u_k, v_k). Then, an SQP approach in its most basic form is given by the algorithm

SQP Choose $(x_0, u_0, v_0), B_0,$ and a merit function ϕ
for $k = 0, 1, 2, \ldots$
Form and solve (3.30) to obtain its optimal solution $(x^\star, u^\star, v^\star)$.
Choose step length α so that $\phi(x_k + \alpha(x^\star - x_k)) < \phi(x^k)$.
Set $x_{k+1} = x_k + \alpha(x^\star - x_k)$, $u_{k+1} = u_k + \alpha(u^\star - u_k)$, $v_{k+1} = v_k + \alpha(v^\star - v_k)$.
Stop if (x_k, u_k, v_k) satisfies some convergence criterion.
Compute B_{k+1} from (x_k, u_k, v_k).
end (for)

A merit function is a function whose reduction implies progress towards the global optimum of problem (3.29). For more details about SQP see [8].

Grid-refining method

In order to guarantee, that a discretized PDE problem (3.15) is a good approximation of (3.14), i.e., its solutions are good discrete approximations of continuous functions $u(\cdot, \cdot)$, it is necessary to choose fine grid discretizations (N_x, N_y). However, a fine grid discretization results in a large scale POP (3.26). Even when exploiting correlative or domain-space sparsity, transforming it into a QOP and imposing tight lower and upper bounds, the SDP relaxation, which needs to be solved, is often computationally demanding - in particular in the cases where we need to choose a high relaxation order to obtain an accurate approximation to an optimal solution of (3.26). Thus, for many difficult PDE problems the SDP relaxations resulting from fine grid discretizations are intractable for current SDP solvers. To overcome this problem, we consider a **grid-refining method**. In our grid-refining method a solution obtained by applying the SDPR method to (3.14) for a coarse grid discretization in a first step is extended stepwise to finer and finer grids by subsequently interpolating coarse grid solutions and applying the SDPR method or locally convergent methods. This method is described by the following scheme:

Step 1 - Initialize	Apply SDPR method with $N_x(1), N_y(1), F_1, \mathrm{lbd}^1, \mathrm{ubd}^1, g(1), h(1), \omega_1$.	obtain u^1
Step 2 - Extend	$N_x(k) = 2\ N_x(k-1) - 1$ or $N_y(k) = 2\ N_y(k-1) - 1$ Interpolation of u^{k-1}	obtain $u^{k-1^\star}$.
Step 3a Step 3b	Apply SDPR method with , $N_x(k), N_y(k), F_k, \mathrm{lbd}^k, \mathrm{ubd}^k, g(k), h(k), \omega_k$. Apply Newton's method or SQP.	obtain u^k
Iterate	Step 2 and Step 3	

Step 1 - SDPR method: Choose an objective function $F_1(u)$, a discretization grid size ($N_x(1)$, $N_y(1)$), lower bounds lbd^1, upper bounds ubd^1 and an initial relaxation order ω_1. Apply SDPR with these parameters to (3.14) and obtain a solution u^1.
Step 2 - Extension: Extend the *(k-1)*th iteration's solution u^{k-1} to a finer grid. Choose either x- or y-direction as the direction of refinement, i.e. choose either $N_x(k) = 2\ N_x(k-1)-1$ and $N_y(k) = N_y(k-1)$, or $N_x(k) = N_x(k-1)$ and $N_y(k) = 2\ N_y(k-1)-1$. In order to extend u^{k-1} to the new grid with the doubled number of grid points, assume without loss of generality the direction of extension is x. The interpolation of the solution u^{k-1} to $u^{k-1^\star}$ is given by the scheme

$$\begin{array}{lcll} u^{k-1\ \star}_{2i-1,j} & = & u^{k-1}_{i,j} & \forall i \in \{1,\dots,N_x(k-1)\},\ \forall j \in \{1,\dots,N_y(k)\}, \\ u^{k-1^\star}_{2i,j} & = & \frac{1}{2}\left(u^{k-1}_{i+1,j} + u^{k-1}_{i,j}\right) & \forall i \in \{1,\dots,N_x(k-1)-1\},\ \forall j \in \{1,\dots,N_y(k)\}. \end{array}$$

The interpolated solution $u^{k-1^\star}$ is a first approximation to a solution of POP (3.26) for the $N_x(k) \times N_y(k)$-grid.

Step 3a - Apply SDPR method: We choose new parameters F_k, lbd^k, ubd^k, ω_k, $g(k)$ and $h(k)$. It is based on the idea, that we may be able to choose $\omega_k < \omega_{k-1}$, if we exploit information given by the interpolated solution $u^{k-1^\star}$. One possibility to do this, is to choose the new objective function $F_k = F_M$, where F^k_M is defined by

$$F^k_M(u) = \sum_{i,j}(u_{i,j} - u^{k-1^\star}_{i,j})^2. \tag{3.31}$$

We may choose this objective function as we are interested in finding a feasible solution of (3.26) with minimal Euclidean distance to the interpolated solution $u^{k-1^\star}$. Another possibility to utilize the information given by $u^{k-1^\star}$ is to tighten the lower and upper bounds by

$$\begin{array}{ll} \mathrm{lbd}^k_{i,j} & = \max\left\{\mathrm{lbd}^{k-1}_{i,j},\ u^{k-1^\star}_{i,j} - \delta\right\} \quad \forall\, i,j, \\ \mathrm{ubd}^k_{i,j} & = \min\left\{\mathrm{ubd}^{k-1}_{i,j},\ u^{k-1^\star}_{i,j} + \delta\right\} \quad \forall\, i,j, \end{array}$$

for some $\delta > 0$. Apply the SDPR method to obtain u^k.

Step 3b - Apply Newton's method or SQP: It may occur the SDP relaxations of (3.26) for the finer grid become intractable, even if $\omega_k < \omega_{k-1}$. Therefore, we may just apply Newton's method to (3.17) or SQP to (3.26) for the finer discretization $(N_x(k), N_y(k))$, both starting with the interpolated solution $u^{k-1^\star}$ as initial guess, to obtain a better approximate solution u.

The steps 2 and 3 are repeated until an accurate solution for a high resolution grid is obtained.

The SDPR method with all its options and the grid-refining method are demonstrated on a variety of numerical examples in 3.3.

3.2.3 Enumeration algorithm

The SDPR method aims at finding a discrete approximation to a solution of a PDE problem. The freedom to choose an objective function for detecting particular solutions of a PDE problem is a feature of the SDPR method which is particularly interesting for a PDE problem with many solutions. Beside finding a particular solution of a PDE problem, another challenging problem is to find all solutions of a PDE problem. The problem of finding discrete approximations for all solutions of a PDE problem (3.14) is the problem of finding all real solutions of the system of polynomial equations (3.17). Classical methods to solve this problem are Gröbner basis or polyhedral homotopy method, which we describe briefly at the end of this section. They compute all complex solutions to (3.17) and it remains to choose the real solutions among them. A recent method which avoids computing all complex solutions and directly computes all real solutions is given by [55] for the case the solution set of (3.17) is finite. Another method is the extraction algorithm presented in [36] for finding all optimal solutions of (3.26). However, both methods, [55] and [36] do not exploit sparsity in (3.17) and (3.26), respectively, which restricts their applicability to small- and medium-scale systems.

In the following we present an algorithm to enumerate all solutions of a system of polynomial equations first proposed in [63] for the cavity flow problem, but which can be extended to a more general case of (3.17). For this method we need to assume the number of solutions of (3.17) is finite, i.e. the feasible set of (3.26) is finite. We also assume that all feasible solutions of (3.26) are distinct with respect to the objective function F, i.e., there is no pair of feasible solutions with identical objective value. The SDPR method enables us to approximate the global minimal solution $u^\star =: u^{(1)\star}$ of (3.26). Beside the minimal solution, we are also interested in finding the solution $u^{(2)\star}$ with the second smallest objective value, the solution $u^{(3)\star}$ with the third smallest objective value or in general the solution $u^{(k)\star}$ with the kth smallest objective value. Based

on the SDPR method we propose an algorithm that enumerates the solutions of (3.17) with the k smallest objective values. Our algorithm shares the idea of separating the feasible set by additional constraints with Branch-and-Bound and cutting plane methods that are used for solving mixed integer linear programs and general concave optimization problems [38]. In contrast to the linear constraints of those methods we impose quadratic constraints to separate the feasible set.

Algorithm 3.1 *Find the approximations to the solutions of (3.17) with the k smallest objective values: Given $u^{(k-1)}$, the approximation to the solution with the $(k-1)$th smallest objective value obtained by applying the SDPR method with relaxation order ω to POP^{k-1} from the $(k-1)$th iteration of the algorithm.*

1. *Choose $\epsilon^k > 0$.*
2. *Choose a vector $b^k \in \{0,1\}^n$.*
3. *Add the following quadratic constraints to POP^{k-1} and denote the resulting POP with smaller feasible set as POP^k.*
$$(u_j - u_j^{(k-1)})^2 \geq \epsilon^k \quad \text{for all } j \text{ with } b_j = 1. \tag{3.32}$$
4. *Apply the SDPR method with relaxation order ω to POP^k. Obtain an approximation $u^{(k)}$ for $u^{(k)\star}$.*
5. *Iterate steps 1–4.*

The idea of Algorithm 3.1 is to impose an additional polynomial inequality constraint (3.32) to the POP (3.26) in iteration k, that excludes the previous iteration's solution $u^{(k-1)}$ from the feasible set of (3.26). In the case the feasible set of (3.26) is finite and $u^{(k-1)}$ is sufficiently close to $u^{(k-1)\star}$, the new constraint excludes $u^{(k-1)\star}$ from the feasible set of (3.26) and $u^{(k)\star}$ is the new global minimizer of (3.26). Of course, there are various alternatives to step 3 in Algorithm 3.1, in order to exclude $u^{(k-1)\star}$ from the feasible set of the POP. One alternative constraint is

$$\left(u_i - u_i^{(k-1)\star}\right) u_{n+i} - \epsilon_i = 0 \quad \text{for all } i \text{ with } b_i = 1, \tag{3.33}$$

where $b \in \{0,1\}^n$, $\epsilon_i > 0$ and u_{n+i} an additional slack variable bounded by -1 and 1. It is easy to see that (3.33) is violated, if $u = u^{(k-1)\star}$. However, it turned out that the numerical performance of (3.33) is inferior to the one of (3.32) for problems of type (3.26) as the tuning parameters ϵ_i and b is far more difficult for (3.33) compared to (3.32). A second alternative to exclude $u^{(k-1)\star}$ are l_p-norm constraints such as

$$\| u - u^{(k-1)\star} \|_p = \left(\sum_{i=1}^{n} \left(u_i - u_i^{(k-1)\star}\right)^p \right)^{\frac{1}{p}} \geq \epsilon, \tag{3.34}$$

for $p \geq 1$. The disadvantage of the constraints (3.34) is, they destroy the correlative sparsity of (3.26), as all u_i $(i = 1, \ldots, n)$ occur in the same constraint. Therefore the advantage of the sparse SDP relaxations is lost and the POP can not be solved efficiently anymore. These observations justify to impose (3.32) as additional constraints in Algorithm 3.1. We obtain the following results for Algorithm 3.1.

Proposition 3.3 *Let $(u^{(1)}, \ldots, u^{(k-1)})$ be the output of the first $(k-1)$ iterations of Algorithm (3.1). If this output is a sufficiently close approximation of the vector $(u^{(1)\star}, \ldots, u^{(k-1)\star})$ of the $(k-1)$ solutions with smallest objective value, and if the feasible set of POP (3.26) is finite and distinct in terms of the objective, i.e. $F(u^{(1)\star}) < F(u^{(2)\star}) < \ldots$, then there exist $b \in \{0,1\}^n$ and $\epsilon \in \mathbb{R}^n$ such that the output $u^{(k)}$ of Algorithm 3.1 (for kth iteration) satisfies*

$$u^{(k)}(\omega) \to u^{(k)\star} \quad \text{when } \omega \to \infty. \tag{3.35}$$

Proof: As each $u^{(j)}$ is in a neighborhood of $u^{(j)\star}$ for all $j \in \{1,\ldots,k-1\}$, we can choose $b \in \{0,1\}^n$ and a vector $\epsilon \in \mathbb{R}^n$, s.t.

$$\forall j \in \{1,\ldots,k-1\}\ \exists i \text{ with } b_i = 1 \text{ s.t. } \left(u_i^{(j)} - u_i^{(j)\star}\right)^2 < \epsilon_i,$$

and for each $j \in \{1,\ldots,k-1\}$ holds

$$\left(u_i^{(j)} - u_i^{(l)\star}\right)^2 \geq \epsilon_i\ \forall l \geq k\ \forall i \text{ with } b_i = 1.$$

Let $\text{POP}^{(k)}$ denote (3.26) with the k systems of additional constraints given by step 3 in Algorithm 3.1, where the kth constraints are given by (3.32) for the constructed b and ϵ. Then it holds

$$\text{feas}\left(\text{POP}^{(k)}\right) = \text{feas}\,(3.26) \setminus \left\{u^{(1)\star},\ldots,u^{(k-1)\star}\right\}.$$

Thus, $u^{(k)\star}$ is the global minimizer of $\text{POP}^{(k)}$ and the global minimum is $F(u^{(k)\star})$. As the bounds (3.19) guarantee the compactness of the feasible set, it holds with the convergence theorem for the sparse SDP relaxations [52], if $\omega \to \infty$,

$$u^{(k)}(\omega) \quad \to \quad u^{(k)\star}. \qquad \square \tag{3.36}$$

Although we have proven convergence, the capacity of current SDP solvers restricts the choice of the relaxation order ω to small integers, typically $\omega = \omega_{\max} + 1$ or $\omega = \omega_{\max} + 2$. Moreover, we need to choose the parameters ϵ and b appropriately, to obtain good approximations of the k feasible solutions with the smallest objective value. In the numerical experiments in 3.3.5, we see the Gröbner basis method is an useful tool to tune the two parameters ϵ and b, as it allows to confirm whether we derive the k solutions of smallest objective value successfully in case (N_x, N_y) is small. In the following we briefly describe Gröbner basis method and Polyhedral Homotopy Continuation as methods to test the SDPR method and to tune the parameters in Algorithm 3.1.

Gröbner basis method

The *Gröbner basis method* to find all complex solutions of a given system of zero dimensional polynomial equations is an useful tool for tuning the parameters of the SDPR method and Algorithm 3.1, and for validating its numerical results. In order to do this, we study (3.17) by the rational univariate representation [85], [78], which is a variation of the Gröbner basis method, for coarse discretizations (N_x, N_y). For a mesh with $N := N_x = N_y$ small (for instance $N = 5$ in the example in 3.3.5), (3.17) is solvable with this method (Groebner(Fgb) in Maple 11, nd_gr_trace and tolex_gsl in Risa/Asir). Applying Gröbner basis method to solve (3.17) for a problem satisfying the assumptions of Proposition 3.3, and enumerating all solutions by their objective value allows us to confirm whether the solutions of the SDPR method are indeed the minimal solutions of (3.17) and to determine which relaxation order ω is sufficient to derive this global minimizer. The result is also used to tune parameters ϵ_i^k in Algorithm 3.1. We have no theorem which states that the tuning based on the coarse mesh case is good for the fine mesh case. However, we believe this tuning provides a better approximation for the fine mesh case, too. Note, whereas the Gröbner basis method finds all complex solutions of (3.17), the SDPR method finds the real solution of (3.17) that minimizes F.

Polyhedral Homotopy Continuation Method

Another recent approach for solving (3.17) is the *Polyhedral Homotopy Continuation Method* for polynomial systems [39]. Consider the problem to find all isolated zeros of a system of n polynomials

$$f(x) = (f_1(x),\ldots,f_n(x)) = 0$$

in a $n-$dimensional complex vector variable $x = (x_1,\ldots,x_n) \in \mathbb{C}^n$. The idea of homotopy continuation methods is to define a smooth **homotopy system** with a continuation parameter $t \in [0,1]$,

$$h(x,t) = (h_1(x),\ldots,h_n(x)) = 0,$$

using the algebraic structure of the polynomial system. The homotopy system is constructed, such that the solutions of the starting polynomial system $h(x,0) = 0$ can be computed easily and that the target polynomial system $h(x,1) = 0$ coincides with the system $f(x) = 0$ to be solved. Furthermore, every connected component of the solutions $(x,t) \in \mathbb{C}^n \times [0,1)$ of $h(x,t) = 0$ forms a smooth curve. The number of homotopy curves that are necessary to connect the isolated zeros of the target system to isolated zeros of the starting system determines the computational work involved in tracing homotopy curves.
A recent software package to determine all complex isolated solutions of $f(x) = 0$ is PHoM by Gunji et al. [31]. Thus, we may apply PHoM to find the complex solutions of a discretized PDE problem $p_{i,j}(u) = 0$ for all (i,j). Then, we select the real among all complex solutions of (3.17) and compare those solutions to the solutions obtained by the SDPR method. Beside its property of finding all complex solutions, PHoM has the drawback, that the computation time grows exponentially in the dimension n of the system. This is due to the fact, the number of isolated solutions increases exponentially in n. Therefore we are restricted to very coarse meshes with $n \leq 10$ when applying PHoM to (3.17). As the Gröbner base method, the polyhedral homotopy method can be used for tuning the parameters in Algorithm 3.1, too.

3.2.4 Discrete approximations to solutions of differential equations

In 3.1.1 we discussed, if a finite difference scheme is convergent and the discretization (N_x, N_y) is chosen sufficiently fine, then a solution of (3.17) is a discrete approximation to a solution of the differential equation (3.14). However, there is no theorem proving the convergence of finite difference schemes for general classes of nonlinear differential equations. In the case of many nonlinear PDE we can not guarantee that a solution of (3.17) is indeed a discrete approximation of a solution of (3.14).

Definition 3.4 *A solution of (3.17) that is not a discrete approximation of a solution of the PDE problem (3.14) is called a fake solution.*

In numerical experiments our main indicator for a solution u of (3.17) not being a fake solution is if we succeed to extend u from a coarse grid to finer and finer grid via the grid refining method.
Another property of solution to (3.17) is the notion of stability.

Definition 3.5 *Let $J(\cdot)$ denote the Jabobian of (3.17) and $m_e(\cdot)$ its maximal eigenvalue. A solution u to (3.17) is called* **stable***, if all eigenvalues of $J(u)$ are non-positive, i.e., if $m_e(u) \leq 0$. If not, it is called* **unstable***.*

Distinguishing stable and unstable solutions is of interest for certain classes of nonlinear PDE problems. In 3.3.3 we will discuss Reaction-Diffusion equations as an example of such a class.

3.3 Numerical experiments

In 3.2.2 we introduced the **SDPR method** for computing discrete approximations to solutions of differential equations. In this section we demonstrate the broad scope of this method by applying it to problems involving nonlinear differential equations which arise from a wide range of fields. It is indeed possible to find highly accurate approximations to many nonlinear differential equations by techniques based on sparse SDP relaxations. For our numerical experiments we apply the software SparsePOP [103] as an implementation of the sparse SDP relaxations (2.18). For applying domain-space sparsity conversion methods we use SparseCoLO [20], and as an SQP based solver we utilize the MATLAB Optimization toolbox routine *fmincon*. As SDP solver for SparsePOP or the primal SDP relaxation from (2.2.7) we apply SeDuMi [95]. For an eventual transformation from POP into a QOP we use one of the heuristics AI, AII, BI or BII from 2.3. When applying the SDPR method to obtain an approximate solution for (3.26), the most important measure of accuracy of its solution u is the **scaled feasibility error** $\epsilon_{\rm sc}$ defined by

$$\epsilon_{\rm sc} = \min \left\{ - \mid \frac{D_{i,j}(u) - f_{i,j}}{\sigma_{i,j}(u)} \mid, \;\; - \mid \frac{h_k(u)}{\sigma_k(u)} \mid, \;\; \min \left\{ \frac{g_l(u)}{\hat{\sigma}_l(u)}, 0 \right\} \quad \forall\, i,j,k,l \right\},$$

where $\sigma_{i,j}(u), \sigma_k(u)$ and $\hat{\sigma}_l(u)$ the maxima over all monomials in the corresponding enumerator polynomials. Note, $\epsilon_{\text{sc}}(u)$ does measure how accurate u approximates a feasible solution of (3.26), but it does not measure how accurate the finite difference scheme approximates the continuous problem (3.14). Another question is, how well $F(u)$ approximates the minimum of (3.26). In the case we choose the primal relaxations (a) and (b) from 2.2.7 or the dual relaxations (2.18) for some linear objective F, $F(u)$ approximates the minimum of (3.26) very accurately, if the feasibility error of u is small. But in case F is nonlinear and we choose the dual relaxations, we have to consider the **optimality error** ϵ_{obj} defined by

$$\epsilon_{\text{obj}} = \frac{|\min(\text{sSDP}_\omega) - F(u)|}{\max\{1, |F(u)|\}}$$

as a measure for the optimality of u. Recall, when applying the SDPR method to solve a differential equation, the most important choices are: The objective function F, bounds lbd and ubd, relaxation order ω in the case of dual relaxations (2.18), Newton's method or SQP as locally fast convergent method, possibly additional constraintss h_k and g_l. Finally, ones needs to decide whether to apply the grid-refining method from 3.2.2 with extension strategy 3a or 3b.
To evaluate the results of the SDPR method for approximating the solutions of differential equations, we consider (a) to apply the SDPR method to PDE problems where an analytical solution is known, and (b) to compare the performance of the SDPR method with the performance of MATLAB Optimization Toolbox, a general purpose solver based on the Finite Element Method. Finally, we may apply Gröbner basis computation or the Polyhedral Homotopy Method to verify that the SDPR method provides accurate approximations to feasible solutions of (3.26), as mentioned in 3.2.3. Moreover, $J(u)$ denotes the Jacobian of (3.17) at u and $m_e(u)$ its largest eigenvalue.
All numerical experiments are conducted on a LINUX OS with CPU 2.4 GHz and 8 Gb memory. The total processing time in seconds is denoted as t_C.

3.3.1 A nonlinear elliptic equation with bifurcation

A well known yet interesting nonlinear elliptic PDE problem, which we have already seen in Example 3.27, is given by

$$\begin{aligned} u_{xx}(x,y) + u_{yy}(x,y) + \lambda u(x,y)\ \left(1 - u(x,y)^2\right) &= 0 \quad \forall\, (x,y) \in [0,1]^2, \\ u(x,y) &= 0 \quad \forall\, (x,y) \in \partial[0,1]^2, \\ 0 \le u(x,y) &\le 1 \quad \forall\, (x,y) \in [0,1]^2, \end{aligned} \tag{3.37}$$

where $\lambda \ge 0$. In fact, this PDE is known as the **Allen-Cahn Equation**. It was shown in [93], there exists a unique nontrivial solution for this problem if $\lambda > \lambda_0 = 2\pi^2 \approx 19.7392$, and there exists only the trivial zero solution if $\lambda \le \lambda_0$. Due to the bifurcation at λ_0, homotopy-like continuation methods, which start from a solution of a system with weak non-linearity to attempt the system with strong non-linearity, cannot be applied to solve (3.37). We fix $\lambda = 22$ and apply the SDPR method with $\omega = 2$ and $F(u) = -\sum_{i,j} u_{i,j}$. In order to study the efficiency of the various options of the SDPR method, we consider different settings: Dual SDP relaxations with and without additional local solver for the POP derived from (3.37), dual and primal SDP relaxations for a QOP equivalent to the POP, the grid-refining method starting from a coarse grid solution, tight and loose upper bounds. The numerical results are given in Table 3.1 and pictured in Figure 3.5. When applying dual SDP relaxations to the original POP, we observe that the solution provided by the SDPR method is very accurate even in the case no additional local method is used. Moreover, the size of the SDP relaxations and thus the computational cost increases rapidly for increasing N_x and N_y. One way to address this problem is to apply the transformation from POP into an equivalent QOP. Both, the primal and dual SDP relaxations are substantially smaller than the dual SDP relaxations for the original POP. But, ubd needs to be tightened when applying SDP relaxations to the QOP, in order to preserve numerical accuracy. Note, for this problem d-space sparsity is richer than correlative sparsity, as the size of the corresponding SDP relaxations and the resulting t_C are smaller. The most efficient mean to obtain high resolution approximations to a solution of (3.37) is the grid-refining method. However, the grid-refining method relies on the assumption that the behavior of the discretized system does not change much for increasing N_x and N_y, which is not the case when starting from a very coarse discretization in general.

Therefore, SDP relaxations for the equivalent QOP are a promising tool to attempt a PDE problem on a higher resolution grid directly. Finally, we notice that the solution of the sparse SDP relaxation remains a sufficiently good initial guess for SQP, even if its feasibility error $\epsilon_{\rm sc}$ is not that small.

SDP relaxation	Local solver	POP to QOP	ubd	n	N_x	N_y	$\epsilon_{\rm sc}$	t_C
Dual	SQP	no	0.99	16	6	6	-1e-14	3
Dual + Grid-refining 3b	SQP	no	0.99	1521	41	41	-1e-13	426
Dual	none	no	0.99	81	11	11	-1e-10	418
Dual	SQP	no	0.99	81	11	11	-4e-15	420
Dual + Grid-refining 3b	SQP	no	0.99	1521	41	41	-1e-13	1016
Dual	SQP	no	0.99	49	9	9	-9e-15	39
Dual + Grid-refining 3b	SQP	no	0.99	225	17	17	-1e-9	49
Dual	SQP	no	0.99	196	16	16	-	OOM
Dual	SQP	yes	0.45	196	16	16	-5e-11	107
Primal	SQP	yes	0.45	196	16	16	-8e-15	35
Primal	none	yes	0.6	841	31	31	-5e-4	1133
Primal	SQP	yes	0.6	841	31	31	-5e-14	3763

Table 3.1: SDPR method results for (3.37), where OOM stands for 'Out of Memory'.

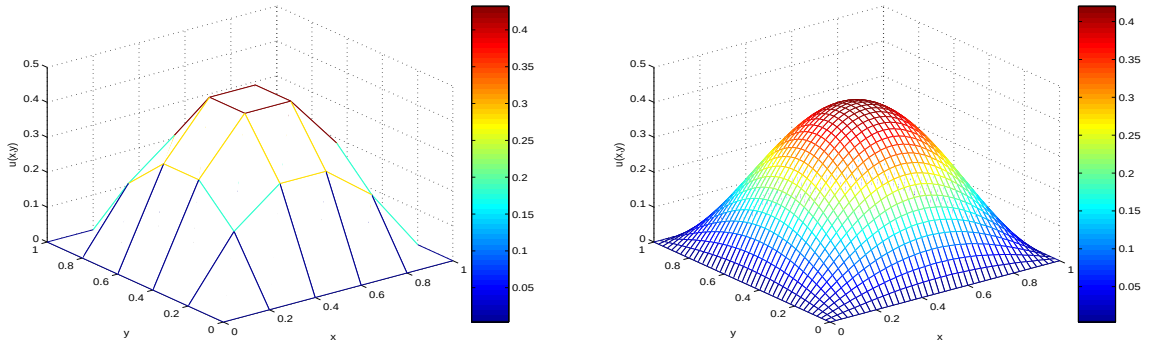


Figure 3.5: Solution for (3.37) in case $\lambda = 22$ for $(N_x, N_y) = (6, 6)$ and $(N_x, N_y) = (41, 41)$.

To examine whether the obtained solution is the only strictly positive one of the discretized PDE, we impose additional constraints

$$\begin{array}{ll} \mid u_x(x,y) \mid & \leq M \\ \mid u_y(x,y) \mid & \leq M \end{array} \qquad \forall\ (x,y) \in [0,1]^2. \tag{3.38}$$

We apply the SDPR method with $\omega = 2$ and $N_x = N_y = 6$ to (3.37) for $\lambda = 22$ under the additional constraints (3.38). For $M > 0.8$ we detect the positive solution obtained before. If we decrease M sufficiently, we obtain the zero solution. Hence, it seems there exists exactly one positive non-trivial solution to the discretization of (3.37), and this solution converges to the strictly positive solution of (3.37) for $N_x, N_y \to \infty$.

As another way to confirm the accuracy of the SDPR method, we take advantage of a further property of (3.37). It was shown in [15], a function $u : [0,1]^2 \to \mathbb{R}$ that is a minimizer of the optimization problem

$$\begin{array}{lll} \min_{u:\,[0,1]^2 \to \mathbb{R}} & \int_{[0,1]^2} u_x^2 + u_y^2 - 2\lambda \left(\frac{u^2}{2} - \frac{u^4}{4} \right) dx\, dy & \\ \text{s.t.} & u = 0 & \text{on } \partial[0,1]^2, \\ & 0 \leq u \leq 1 & \text{on } [0,1]^2, \end{array} \tag{3.39}$$

is a solution to (3.37). The integral to be minimized in this problem is called the **energy integral**. By discretizing (3.39) via a finite difference scheme it can be transformed into a POP analogously to a PDE of form (3.14). In opposite to (3.26) that we derive from (3.14), the objective function F is not of free choice but canonically given by the discretization of the objective function in (3.39). We apply the SDPR method with relaxation order $\omega = 2$ to (3.37) and (3.39) on a 6×6- and a 11×11- grid and obtain an identical solution for both problems. These results are reported in Table 3.2, where Δu, given by

$$\Delta u = \max_{i,j} \mid u_{i,j} - \hat{u}_{i,j} \mid,$$

evaluates the deviation of the SDPR method solutions for both problems; $u_{i,j}$ denotes the SDPR solution to (3.37) and $\hat{u}_{i,j}$ the SDPR solution to (3.39).

Problem	N_x	N_y	t_C	ϵ_{obj}	ϵ_{sc}	Δu
(3.37)	6	6	3	2e-14	-1e-14	2e-6
(3.39)	6	6	2	1e-10	-	2e-6
(3.37)	11	11	418	4e-15	-9e-15	9e-7
(3.39)	11	11	98	2e-10	-	9e-7

Table 3.2: SDPR results for (3.37) and (3.39).

The solutions to both problems are highly accurate and we note that the total computation time to minimize the energy integral is less than the time required to solve the polynomial optimization problem corresponding to (3.37).
Finally, we compare the numerical performance of the SDPR method for (3.37) to existing solvers for nonlinear PDE problems. We apply the nonlinear solver from Matlab PDE toolbox to (3.37). The Matlab solver is FEM based and requires an initial guess to search for a solution of the PDE problem. When starting from the zero-solution or a number of random positive functions, this solvers detects the trivial solution. Even when choosing u_0 with $u_0(x, y) := 0.43 \sin(\pi x) \sin(\pi y)$ on $[0, 1]^2$ as initial guess, the FEM solver detect the trivial solution not the non-trivial one, although u_0 is close to the non-trivial solution (On 41×41-grid: $\max \mid u - u_0 \mid = 0.006$, $\frac{1}{41^2} \sum_{i,j} \mid u_{i,j} - u_{0i,j} \mid = 0.003$). Although the FEM solver finds the trivial solution in less than 60 seconds on a mesh of much higher resolution (67356 nodes, 134204 triangles) than those solved by the SDPR method, it needs a very good initial guess to find the more interesting, non-trivial solution. It is the advantage of the SDPR method, that no initial guess is required to find a accurate approximation of the strictly positive solution of (3.37).

3.3.2 Illustrative nonlinear PDE problems

A problem in Yokota's text book

Simple ODE problems can be solved by the SDPR method with ease. To demonstrate this, consider the easy solvable nonlinear boundary value problem

$$\begin{aligned} \ddot{u}(x) + \tfrac{1}{8} u(x) \dot{u}(x) - 4 - \tfrac{1}{4} x^3 &= 0 \qquad \forall\, x \in [1, 3], \\ u(1) &= 17, \\ u(3) &= \tfrac{43}{3}, \\ 10 \leq u(x) &\leq 20 \qquad \forall\, x \in [1, 3]. \end{aligned} \tag{3.40}$$

For details about problem (3.40) see [105]. Applying the SDPR method with $\omega = 2$, objective function F, defined by

$$F(u) = \sum_{i=1}^{N_x} u_i,$$

and without using a locally fast convergent method, yields the highly accurate, stable solution, that is documented in Table 3.3 and pictured in Figure 3.6.

N_x	$\epsilon_{\rm sc}$	$\epsilon_{\rm obj}$	m_e	t_C
200	2e-9	-4e-11	-3	104

Table 3.3: Numerical results for problem (3.40).

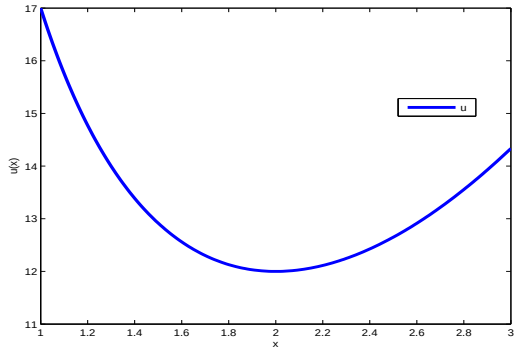


Figure 3.6: SDPR method solution u for problem (3.40).

Nonlinear wave equation

As an example of a hyperbolic PDE problem we study time periodic solutions of the **nonlinear wave equation**

$$\begin{aligned} -u_{xx} + u_{yy} + u\,(1-u) + 0.2\sin(2x) &= 0 \\ &\forall\ (x,y) \in [0,\pi] \times [0,2\pi], \\ u(0,y) = u(\pi,y) &= 0 \qquad \forall\ y \in [0,2\pi], \\ u(x,0) &= u(x,2\pi) \qquad \forall\ x \in [0,\pi], \\ -3 \le u(x,y) &\le 3 \qquad \forall\ (x,y) \in [0,\pi] \times [0,2\pi]. \end{aligned} \tag{3.41}$$

As far as we have checked on the mathsci data base, there is no mathematical proof of the existence of periodic solution of this system. However, the SDPR method finds some periodic solutions. We observed the POP corresponding to problem (3.41) has various solutions. Therefore, the choice of the objective determines the solution found by the sparse SDP relaxation. We consider the functions

$$F_1(u) = \sum_{i,j} \sigma_{i,j} u_{i,j}, \qquad F_2(u) = \sum_{i,j} u_{i,j},$$

as objective for (3.26), where $\sigma_{i,j}$ $(i = 1,\ldots,N_x,\ j = 1,\ldots,N_y)$ are random variables that are uniformly distributed on $[-0.5, 0.5]$. We apply the SDPR method with $\omega = 2$ and Newton's method as a local solver.

The results are enlisted in Table 3.4 and pictured in Figures 3.7 and 3.8.

Imposing Variation bounds: Beside choosing different objective functions in the SDPR method, a second possibility to detect other solutions of a PDE is to impose additional constraints polynomial in the unknown functions. In 3.2.1 we introduced *variation bounds* (3.21) to restrict the space of functions we are searching for solutions. For (3.41) we impose the variation bounds

$$|\ u_y(x,y)\ | \le 0.5 \quad \forall\ (x,y) \in (0,\pi) \times (0,2\pi). \tag{3.42}$$

SDP relaxation	objective	N_x	N_y	ϵ_{sc}	t_C
Dual	F_1	5	6	-4e-10	151
Dual + Grid-refining 3b	F_1	33	40	-3e-8	427
Dual	F_2	5	5	-3e-11	19
Dual + Grid-refining 3b	F_2	33	33	-5e-10	86

Table 3.4: SDPR method results for (3.41).

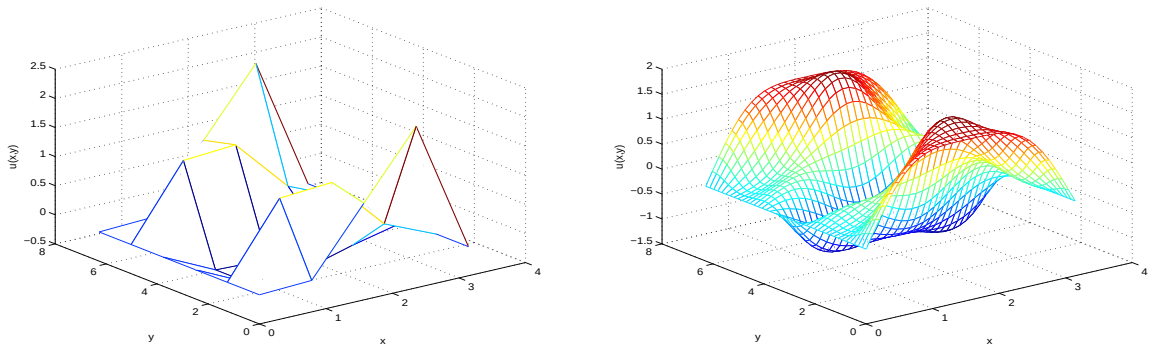


Figure 3.7: Solutions of (3.41) by SDPR method with objective F_1.

If the SDPR method is applied to (3.41) with additional condition (3.42) and F_1 as objective, another solution to the PDE problem is obtained, which is documented in Table 3.5 and pictured in Figure 3.9. Thus several solutions of (3.41) are detected by choosing different objective functions (3.26) and by imposing additional polynomial inequality constraints.

SDP relaxation	objective	N_x	N_y	ϵ_{sc}	t_C
Dual	F_1	5	6	-3e-15	437
Dual + Grid-refining 3b	F_1	17	21	-3e-14	1710

Table 3.5: SDPR method results for (3.41) under additional constraint (3.42).

A system of elliptic nonlinear PDEs

As a PDE problem of two unknown functions in two variables, consider the following problem, where we distinguish two types of boundary conditions, Case I (Dirichlet condition) and Case II (Neumann condition).

$$\begin{array}{lll} u_{xx} + u_{yy} + u\ \left(1 - u^2 - v^2\right) & = 0, & \\ v_{xx} + v_{yy} + v\ \left(1 - u^2 - v^2\right) & = 0, & \forall\ (x, y) \in [0, 1]^2. \\ 0 \leq u, v & \leq 5. & \end{array} \tag{3.43}$$

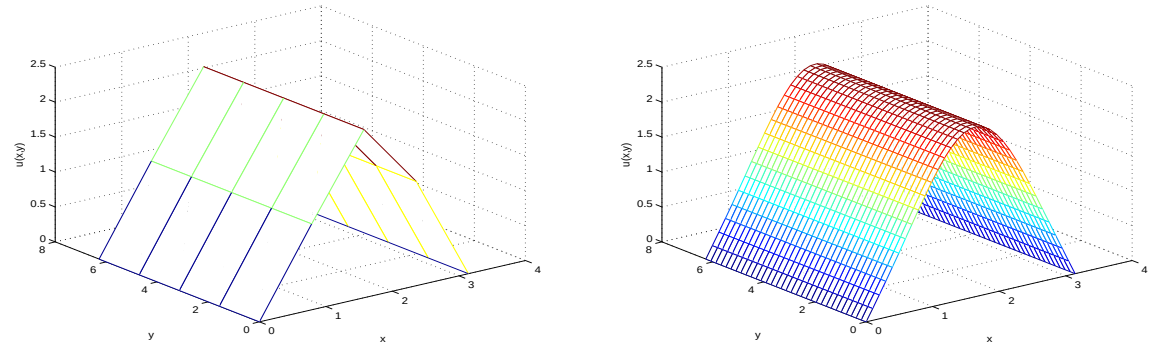


Figure 3.8: SDPR solutions of (3.41) with objective F_2.

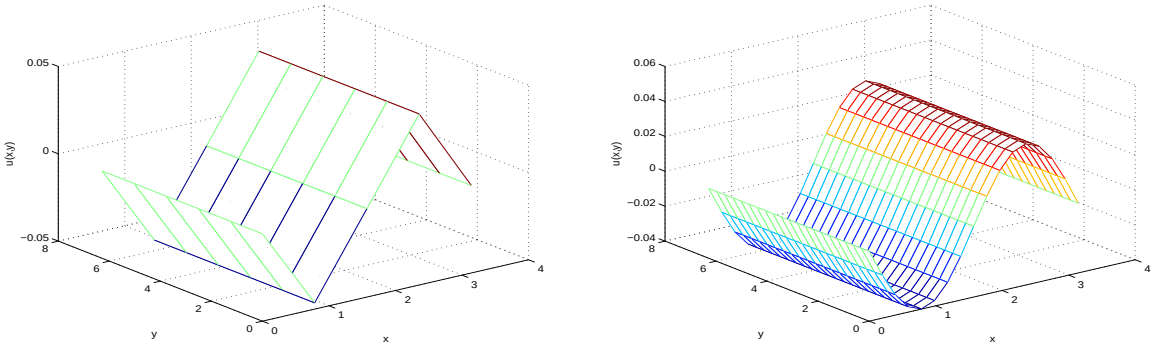


Figure 3.9: SDPR solutions of (3.41) under additional constraint (3.42) with objective F_1.

Case I:

$$\begin{array}{llll} u(0,y) = 0.5y + 0.3\sin(2\pi y), & u(1,y) = 0.4 - 0.4y & & \forall\, y \in [0,1], \\ u(x,0) = 0.4x + 0.2\sin(2\pi x), & u(x,1) = 0.5 - 0.5x & & \forall\, x \in [0,1], \\ v(x,0) = v(x,1) & v(0,y) = v(1,y) = 0 & & \forall\, x \in [0,1], \\ & & & \forall\, y \in [0,1]. \end{array}$$

or

Case II:

$$\begin{array}{llll} u_x(0,y) = -1, & u_x(1,y) = 1 & & \forall\, y \in [0,1], \\ u_y(x,0) = 2x, & u_y(x,1) = x + 5\sin(\frac{\pi x}{2}) & & \forall\, x \in [0,1], \\ v_x(0,y) = 0, & v_x(1,y) = 0 & & \forall\, y \in [0,1], \\ v_y(x,0) = -1, & v_y(x,1) = 1 & & \forall\, x \in [0,1]. \end{array}$$

In both cases, we choose $F(u,v) = \sum_{i,j} u_{i,j}$ for the SDPR method.

Case I. We apply the SDPR method with both primal and dual SDP relaxations exploiting sparsity. Also, as the degree of this PDE problem is three, we can apply the POP to QOP transformation to reduce the size of the SDP relaxations. Finally we apply the grid refining method to extend the coarse grid solutions to finer grids. When applying the dual SDP relaxations to the POP, lbd $= 0$ and ubd $= 5$ are given by (3.43). The bound ubd is tightened to ubd $= 0.6$ for the primal and dual SDP relaxations of the QOP in order to obtain accurate solutions. The numerical results of the SDPR method with $\omega = 2$, and SQP as

a local solver are reported in Table 3.6. We observe, exploiting d-space sparsity and applying the primal SDP relaxations is very efficient for this problem. In fact, the d-space sparsity is richer than the correlative sparsity, as a comparison of the total computation time of primal and dual SDP relaxations for the QOP derived from (3.43) for $N_x = N_y = 11$ reveals. When applying the dual SDP relaxations, the grid-refining method is useful to extend coarse grid solutions to high resolution grids. The approximate solution for $u(\cdot,\cdot)$ is pictured in Figure 3.10, the corresponding v equals zero on the entire domain.

SDP relaxation	Transform POP to QOP	n	N_x	N_y	$\epsilon_{\rm sc}$	t_C
Dual	no	18	5	5	-4-e13	2
Dual + Grid-refining 3b	no	7938	65	65	-2e-14	12280
Dual	no	32	6	6	-4e-13	150
Dual + Grid-refining 3b	no	162	11	11	-2e-13	156
Dual + Grid-refining 3b	no	722	21	21	-4e-16	238
Dual	yes	162	11	11	-9e-9	185
Primal	yes	162	11	11	-3e-8	19
Primal	yes	338	15	15	-3e-8	107

Table 3.6: Results of SDPR method for (3.43) in Case I.

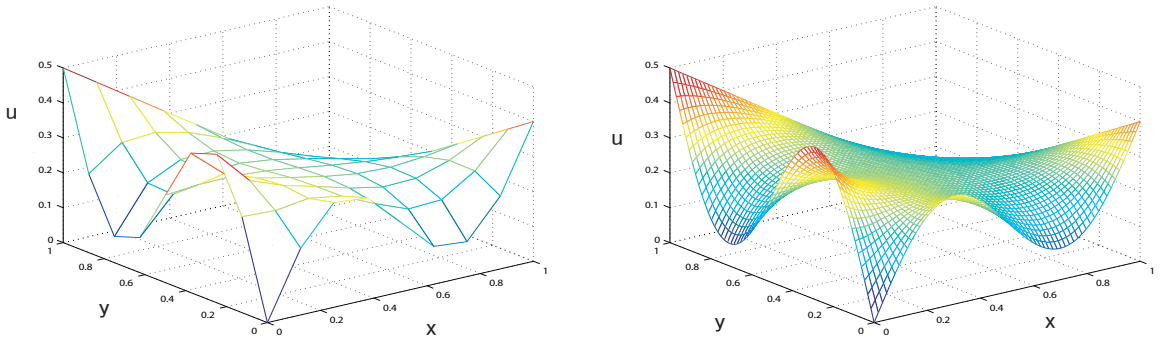


Figure 3.10: SDPR method solution u of (3.43) for Case I and two different discretizations.

We compare the performance of the SDPR method for Case I of (3.43) to Matlab PDE toolbox. Starting from an arbitrary initial guess the Matlab solver detects the same solution as the SDPR method in 2 seconds on a mesh with 2667 nodes and 5168 triangles, and in 15 seconds on a mesh with 10501 nodes and 20672 triangles, c.f. Figure 3.11. Thus, the FEM solver from Matlab is much more efficient in finding the solution. However, the discretization of (3.43) under Dirichlet condition has exactly one real solution. In a more difficult PDE problem with many solutions a good initial guess is required for the Matlab solver to find a solution of interest.

Case II. We apply the SDPR method with the same settings as in Case I, and compare the efficiency of primal and dual SDP relaxations and the grid-refining method. For the primal and dual SDP relaxations of the QOP the upper bounds are tightened to $\text{ubd}_u = 4$ and $\text{ubd}_v = 1.5$. The numerical performance of the SDPR method is reported in Table 3.7. The single solution (u, v) of the discretized differential equation is illustrated in Figure 3.12. As in Case I, we observe that the transformation from POP to QOP is efficient to reduce the size of the SDP relaxations while the accuracy of the approximation is preserved. Moreover, d-space sparsity is richer than correlative sparsity, as the primal SDP relaxations for $n = 242$ can be solved in 58s whereas solving the dual SDP relaxations for the same QOP requires 748s.

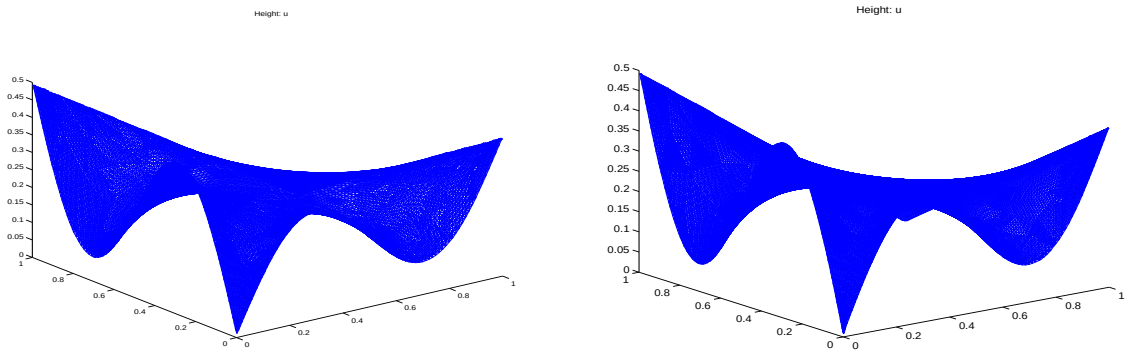


Figure 3.11: Solution u of (3.43) by Matlab PDE Toolbox for mesh with 2667 nodes/ 5168 triangles (left) and 10501 nodes/ 20672 triangles (right).

SDP relaxation	Transform POP to QOP	n	N_x	N_y	$\epsilon_{\rm sc}$	t_C
Dual	no	32	6	6	-7e-13	128
Dual + Grid-refining 3b	no	3042	41	41	-3e-15	2160
Dual	no	50	7	7	-1e-10	713
Dual + Grid-refining 3b	no	242	13	13	-6e-13	728
Dual	yes	242	13	13	-5e-11	748
Primal	yes	242	13	13	-6e-13	58

Table 3.7: Results of the SDPR method for (3.43) in Case I.

Nonlinear parabolic equation

Consider a nonlinear parabolic PDE problem of two dependent scalar functions

$$\begin{aligned} \tfrac{1}{50}u_{xx} - u_y + 1 + u^2\,v - 4u &= 0 && \forall x \in [0,1],\ y \ge 0,\\ \tfrac{1}{50}v_{xx} - v_y + 3u - u^2\,v &= 0 && \forall x \in [0,1],\ y \ge 0,\\ u(0,y) = u(1,y) &= 1 && \forall\ y \ge 0,\\ v(0,y) = v(1,y) &= 3 && \forall\ y \ge 0,\\ u(x,0) &= 1+\sin(2\pi x) && \forall\ x \in [0,1],\\ v(x,0) &= 3 && \forall\ x \in [0,1]. \end{aligned} \tag{3.44}$$

In order to bring (3.44) into form (3.14), we need to cut y at $y = T$. Since problem (3.44) is parabolic, the solutions $((u,v)(N_x,N_y))$ of the discretized problems converge to solutions $(u(\cdot,\cdot),v(\cdot,\cdot))$ of (3.44) with Theorem 3.4. We apply the grid-refining method with strategy 3b, where F is given by $F(u,v) = \sum_{i,j} u_{i,j}$ and $\omega = 3$. Furthermore, lbd $\equiv 0$ and ubd $\equiv 5$ are chosen as bounds for u and v. The grid-refinig method yields a highly accurate, stable solutions on a 33×65-grid; see Table 3.8 and Figure 3.13.

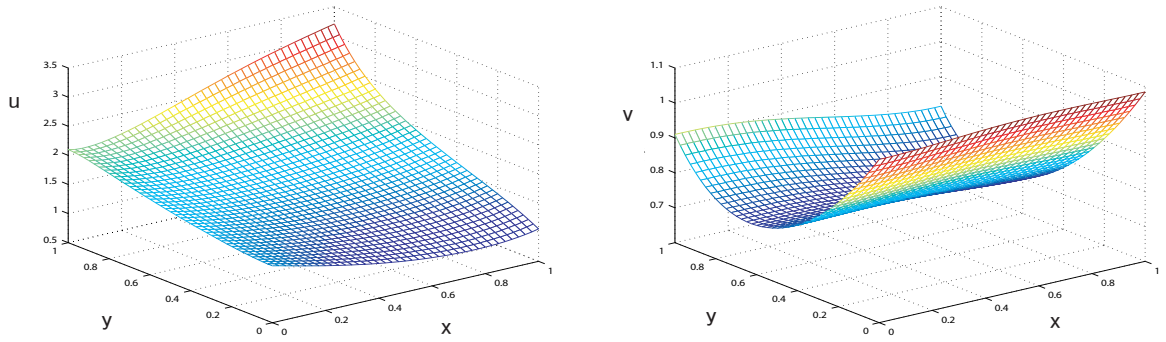


Figure 3.12: SDPR method solutions u (left) and v (right) of (3.43) for Case II.

strategy	N_x	N_y	m_e	ϵ_{sc}
initial SDPR method	5	9	-4.12	-2e-10
Grid-refining 3b	33	65	-2.88	-5e-11

Table 3.8: Results for diffusion problem (3.44).

First order PDEs

An optimization based approach to attempt first order PDE was proposed by Guermond and Popov [29, 30]. In [29] the following example of a first order PDE with a discontinuous solution is solved on a $40 \times 40-$grid:

$$\begin{array}{lll} u_x(x,y) & = 0 & \forall\,(x,y) \in [0,\,2] \times [0.2,\,0.8], \\ u(0,y) & = 1 & \text{if } y \in [0.5,\,0.8], \\ u(0,y) & = 0 & \text{if } y \in [0.2,\,0.5[. \end{array} \tag{3.45}$$

Applying the SDPR method with an forward or central difference approximation for the first derivative in (3.45) we detect the discontinuous solution

$$u(x,y) = \begin{cases} 1 \text{ if } y \geq 0.5 \\ 0 \text{ otherwise} \end{cases}$$

on a 40×40-grid.

A more difficult first order PDE problem is given by

$$\begin{array}{lll} u_x(x,y) + u(x,y) - 1 & = 0 & \forall\,(x,y) \in [0,\,1]^2, \\ u(0,y) = u(1,y) & = 0 & \forall\, y \in [0,\,1], \\ 0 \leq u(x,y) & \leq 1 & \forall\,(x,y) \in [0,\,1]^2. \end{array} \tag{3.46}$$

As can be seen easily and was pointed out in [30], problem (3.46) is not well-posed since the outflow boundary condition is over-specified. Problem (3.46) is discussed in detail in [30] and the authors obtained an accurate approximation to the exact solution by L^1 approximation on a 10×10-grid. Applying the SDPR method with $\omega = 1$ and objective function $F(u) = \sum_{i,j} u_{i,j}$ on a 10×10 grid, we obtain a highly accurate

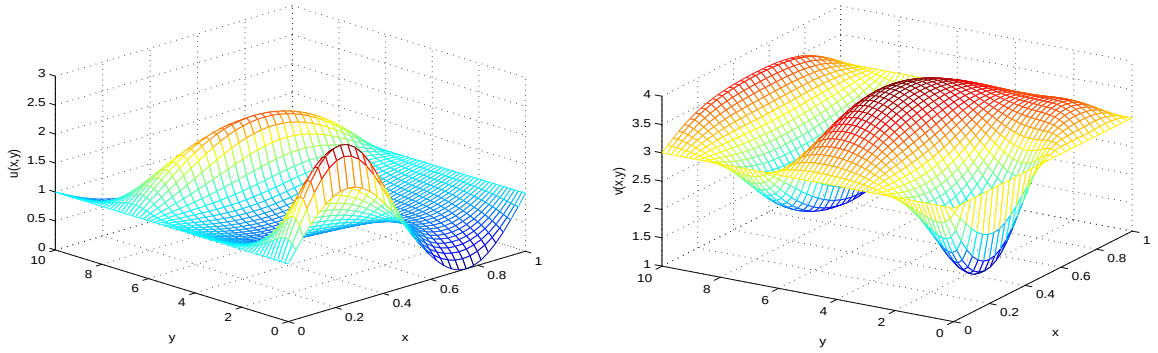


Figure 3.13: Solutions u (left) and v (right) for diffusion problem (3.44).

approximation ($\mid \epsilon_{\text{sc}} \mid < 1e-14$) to this solution in less than 10 seconds in the case we choose a forward difference approximation for the first derivative. In the case of choosing a central or a backwards difference scheme the dual problem in the resulting SDP relaxation becomes infeasibe. Moreover, by applying the SDPR method on a 50×50-grid we are able to obtain a highly accurate approximation to the solution of (3.46) in less than 250 seconds, as pictured in Figure 3.14.

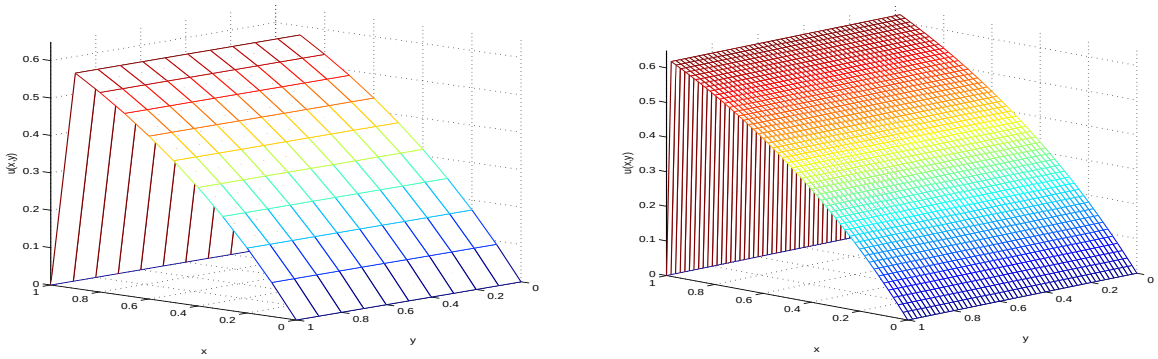


Figure 3.14: Solution u by SDPR method for (3.46).

3.3.3 Reaction-diffusion equations

An interesting class of PDE problems to analyze by the SDPR method is the class of **reaction-diffusion equations**. Many reaction-diffusion equations are systems of nonlinear PDE with multiple solutions. For some of them special unstable solutions exist that are difficult to detect by standard homotopy-like continuation methods. We demonstrate how the features of the SDPR method can be used to find special solutions of a reaction-diffusion equation, in particular interesting unstable ones.

A reaction-diffusion equations due to Mimura

An exciting and difficult reaction-diffusion problem of two dependent functions is a problem by M. Mimura [64] which arises from the context of planktonic prey and predator models in biology. This problem is given below and we briefly call it **Mimura's problem**.

$$\begin{aligned} \tfrac{1}{20}\, u''(t) + \tfrac{1}{9}\left(35 + 16u(t) - u(t)^2\right)\, u(t) - u(t)\, v(t) &= 0, \\ 4\, v''(t) - \left(1 + \tfrac{2}{5}v(t)\right)\, v(t) + u(t)\, v(t) &= 0, \\ \dot{u}(0) = \dot{u}(5) = \dot{v}(0) = \dot{v}(5) &= 0, \\ 0 \le u(t) &\le 14, \\ 0 \le v(t) &\le 14, \\ \forall t \in [0,5]\,. & \end{aligned} \tag{3.47}$$

In [64] the problem is analyzed, and the existence of continuous solutions is shown in [82]. In order to construct a POP of the type (3.26), we consider different **objective functions**:

$$\begin{aligned} &F_1(u,v) = -u_{\lceil \frac{N}{2} \rceil}, \quad F_2(u,v) = -\textstyle\sum_{i=1}^{N} u_i, \quad F_3(u,v) = -u_2, \\ &F_4(u,v) = -u_{N-1}, \quad F_5(u,v) = -u_2 - u_{N-1}, \quad F_6(u,v) = \textstyle\sum_{i=1}^{N}(u_i + v_i). \end{aligned} \tag{3.48}$$

First, we apply the SDPR method with $\omega = 3$ and $N = 5$. In order to confirm the numerical results obtained for this very coarse grid, we apply PHoM [31], which is a C++ implementation of the polyhedral homotopy continuation method for computing all isolated complex solutions of a polynomial system of equations, to the system of discretized PDEs. In that case the dimension n of the polynomial system equals 6, as there are 2 unknown functions with 3 interior grid points each. PHoM finds 182 complex, 64 real and 11 nonnegative real solutions. Varying the upper and lower bounds for u_2 and u_4 and choosing one of the functions $F_1, \ldots, F_5$ as an objective function, all 11 solutions are detected accurately ($\mid \epsilon_{\rm sc} \mid < 1e-7$) by the SDPR method, as enlisted in Table 3.9.

u_2	u_3	u_4	v_2	v_3	v_4	objective	ubd_{u2}	ubd_{u4}
4.623	6.787	0.939	9.748	10.799	5.659	F_3	5	1.5
4.607	6.930	0.259	9.737	10.831	5.166	F_3	5	0.5
0.259	6.930	4.607	5.166	10.831	9.737	F_2	0.5	6
5.683	2.971	5.683	10.388	8.248	10.388	F_3	6	6
6.274	0.177	6.274	10.638	6.404	10.638	F_3	7	7
0.970	7.812	0.970	5.735	10.94	5.735	F_3	2	2
0.297	7.932	0.966	5.230	10.94	5.729	F_4	0.5	2
0.962	7.932	0.297	5.729	10.94	5.230	F_3	2	0.5
0.304	8.045	0.304	5.234	10.94	5.234	F_1	14	14
0.939	6.787	4.623	5.659	10.80	9.748	F_4	2	14
5.000	5.000	5.000	10.000	10.000	10.000	F_2	14	14

Table 3.9: SDPR method solutions of (3.47) for $N = 5$.

The confirmation of our SparsePOP results by PHoM encourages us to solve Mimura's problem for a higher discretization. Relaxation order $\omega = 3$ is necessary to obtain an accurate solution in case $N = 7$ (Table 3.10, row 1). The upper bounds for u_2 and u_{N-1} are chosen to be 1. When extending the grid size from 7 to 13, the accuracy of the SDPR solution deteriorates. Also, if we choose $\omega = 2$ for the initial application of the SDPR method, or if Newton's method is applied with another arbitrary starting point, or if we start for instance with $N = 5$ or $N = 9$, it is not possible to get an accurate solution. One possibility to overcome these difficulties is to start the grid-refining method with strategy 3b on a finer grid. We obtain a highly accurate stable solution *2teeth* when we start with $N = 7$ and F_2 as objective function, and a highly accurate stable solution *2,3peak* when we start with $N = 25$ and F_5 as objective function. See Table 3.10 and Figure 3.15. It seems reasonable to state that the SDPR method provides an appropriate initial guess for Newton' s method, which leads to accurate solutions for sufficiently high discretizations.

SDP relaxation	QOP	Local solver	obj	ubd_{u1}	N	solution	ϵ_{sc}	m_e
Dual	no	Newton	F_2	14	7		-5e-9	-2.2
Grid-ref 3b	no	Newton		14	13		-2e+0	2.7
Grid-ref 3b	no	Newton		14	25		-2e-6	-0.9
Grid-ref 3b	no	Newton		14	49		-1e-12	0.1
Grid-ref 3b	no	Newton		14	97		-5e-12	0.2
Grid-ref 3b	no	Newton		14	193		-2e-4	0.3
Grid-ref 3b	no	Newton		14	385	2teeth	-5e-5	0.2
Dual	no	Newton	F_5	14	26		-1e-1	2.09
Grid-ref 3b	no	Newton		14	51		-5e-2	-0.18
Grid-ref 3b	no	Newton		14	101		-2e-15	-0.07
Grid-ref 3b	no	Newton		14	401	2,3peak	-6e-16	-0.07

Table 3.10: Results of grid-refining strategy 3b.

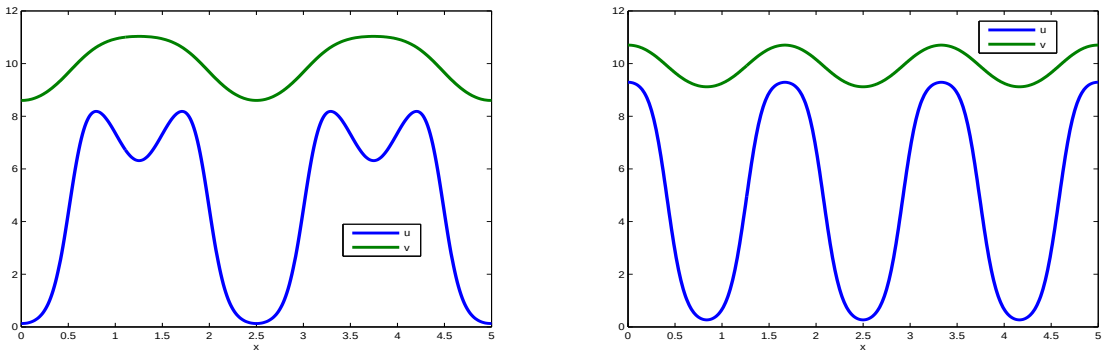


Figure 3.15: Unstable solution *2teeth* (left) and stable solution *2,3peak* (right).

As the most powerful approach we apply the grid-refining method with strategy 3b, $\omega_1 = 3$ and $\omega_k = 2$ for $k \geq 2$. We obtain the highly accurate stable solutions *3peak* and *4peak*, that are documented in Table 3.11 and pictured in Figure 3.16. As objective function for the POPs to be solved in each iteration we choose the function F_M from (3.31).
Another way to attempt finer grids directly is to transform the POP derived from 3.47 into a QOP, and to apply both primal and dual SDP relaxations to that QOP. As reported in Table 3.11, the total computation time in the case $N = 51$ is reduced by two magnitudes under this method. A highly accurate solution is obtained when applying SQP as local solver in the SDPR method. Finally, we yield various stable and unstable solutions to Mimura's problem, when choosing different functions as objective F and when tightening or loosening ubd_{u1}. By the SDPR method we obtain the stable solutions *3peak, 4peak, 2,3peak* and the unstable solutions *2teeth, peak3unstable, peak4unstable, 2valley.*

Reaction-diffusion equations from collision processes

Another interesting class of reaction-diffusion equations arises from collision processes of particle-like patterns in dissipative systems. Various different input-output relations such as annihilation, repulsion, fusion, and even chaotic dynamics are observed after collision of these patterns. The reaction-diffusion equations

SDP Relaxation	QOP	Local solver	obj	ubd_{u1}	N	solution	ϵ_{sc}	m_e	t_C
Dual	no	Newton	F_5	0.5	26		-2e-1	2.09	203
Grid-ref 3a	no	Newton	F_M	0.5	51		-4e-2	-0.05	224
Grid-ref 3a	no	Newton	F_M	0.5	101		-4e-4	-0.02	383
Grid-ref 3a	no	Newton	F_M	0.5	201	4peak	-3e-11	-0.02	1082
Dual	no	Newton	F_1	0.5	26		-1e-3	-0.12	270
Grid-ref 3a	no	Newton	F_M	0.5	51		-4e-3	-0.08	348
Grid-ref 3a	no	Newton	F_M	0.5	101		-3e-16	-0.08	511
Grid-ref 3a	no	Newton	F_M	0.5	201	3peak	-2e-11	-0.07	1192
Dual	no	SQP	F_1	0.5	51		-5e-13	-0.07	470
Grid-ref 3b	no	SQP	F_1	0.5	201	3peak	-1e-12	-0.07	501
Dual	no	SQP	F_1	14	51		-1e-10	0.70	393
Grid-ref 3b	no	SQP	F_1	14	201	peak3unstable	-4e-12	0.70	427
Dual	no	SQP	F_2	0.5	51		-1e-2	1.62	806
Grid-ref 3b	no	SQP	F_2	0.5	201	peak4unstable	-3e-16	1.43	868
Dual	yes	SQP	F_6	0.5	51	2valley	-1e-13	2.10	16
Primal	yes	SQP	F_6	0.5	51	2valley	-1e-9	2.10	8

Table 3.11: Results of grid-refining strategy 3a.

describing the collision processes have special unstable solutions, so-called **scattors**, which are difficult to detect and attract lots of interest [73, 74, 75, 76]. We show how scattors are detected by the SDPR method.

Gray-Scott model in 1D: As an first example we consider the stationary equation of a Gray-Scott model for the dynamics of two traveling pulses in one dimension from [74]:

$$\begin{aligned} D_u u_{xx}(x) - u(x)v(x)^2 + f(1-u(x)) &= 0 \quad \forall\, x \in [0,4]\,, \\ D_v v_{xx}(x) + u(x)v(x)^2 - (f+k)v(x) &= 0 \quad \forall\, x \in [0,4]\,, \end{aligned} \tag{3.49}$$

where $D_u > 0$, $D_v > 0$, and $f > 0$ and $k > 0$ are two parameters related to inflow and removal rate of chemical species. Moreover, we impose Neumann boundary conditions. The existence and the shape of stable solutions and scattors *doublePeak* depends heavily on the choice for f and k. We set the parameters to $D_u = 5e-5$, $D_v = 2.5e-5$, $f = 0.0198$ and $k = 0.0497859$. For this setting a scattor with positive eigenvalues $(\lambda_1, \lambda_2, \lambda_3) = (0.0639, 0.0638, 0.0023)$ was reported in [74]. We choose $F(u,v) = -\sum_{i=1}^{N} u_i$, $\text{lbd}_u = \text{lbd}_v = 0$, $\text{ubd}_u = 1.0$, $\text{ubd}_v = 0.8$, $\text{lbd}_{u1} = \text{lbd}_{uN} = 0.4$, $\text{ubd}_{u1} = \text{ubd}_{uN} = 0.6$ and $\text{ubd}_{u\frac{N}{2}} = 0.9$. Applying the SDPR method with these settings yields the scattor *doublePeak* as reported in Table 3.12 and pictured in Figure 3.18.

SDP Relaxation	QOP	Local solver	ω	N	solution	λ_1	λ_2	λ_3	t_C
Dual	yes	SQP	2	20		0.0441	0.0434	-	32
Grid-ref 3b	yes	SQP		640	doublePeak	0.0639	0.0638	0.0023	469
Grid-ref 3b	yes	SQP		1280	doublePeak	0.0638	0.0637	0.0023	1247

Table 3.12: Results of the SDPR method for (3.49).

In (3.49) the shape and number of scattors depends on the choice of the two parameters F and k. It is well known, that if we fix f there is a bifurcation point k_0, to be precise, there are no scattors for (3.49) if $k < k_0$ and several scattors if $k > k_0$. For a bifurcation analysis by the SDPR method we fix $f = 0.0270$.

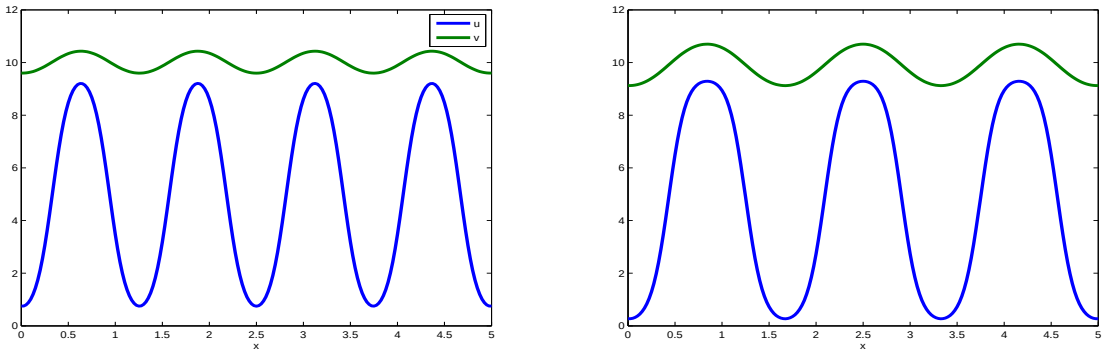


Figure 3.16: Stable solutions *4peak* and *3peak*.

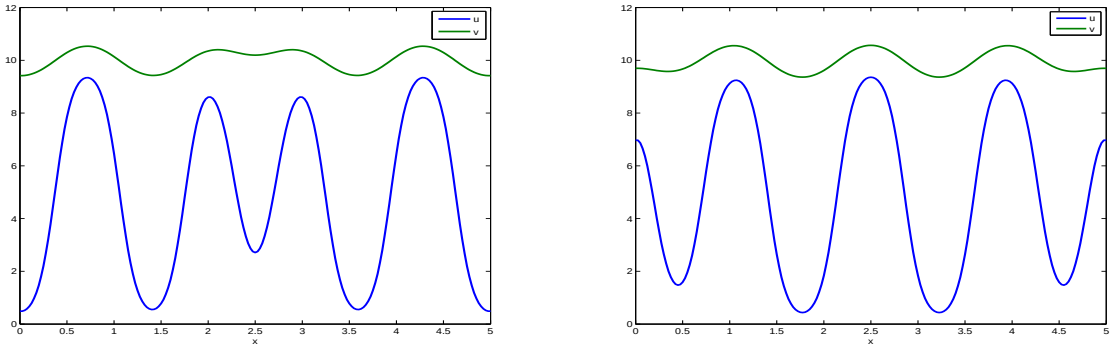


Figure 3.17: Unstable solutions *peak4unstable* and *peak3unstable*.

Moreover, we define the norm of a numerical solution u by

$$\| u \| = \left(\sum_{i=1}^{N} u_i^2 \right)^{\frac{1}{2}}.$$

We define $F(u,v) = -\sum_{i=1}^{N} u_i^2$ as objective function for the SDPR method. Bounds lbd and ubd are chosen as follows:

$$\text{lbd}_{ui} = \begin{cases} 0.5, & \text{if } i \in \{1, \ldots, \lceil \frac{N}{10} \rceil\} \cup \{\lceil \frac{9N}{10} \rceil, \ldots, N\} \\ 0, & \text{else} \end{cases}, \tag{3.50}$$

$\text{ubd}_u = 0.8$, $\text{lbd}_v = 0$ and $\text{ubd}_v = 0.8$. If we apply the SDPR method with $\omega = 2$ and $N = 256$ to (3.49) for $k \leq 0.05281$, we do not obtain any nontrivial solution. If we apply the SDPR method for the same settings to (3.49) with k increasing from 0.05282 to 0.0535 we obtain a scattor $(u,v)^1$ with $\| u^1 \|$ increasing from

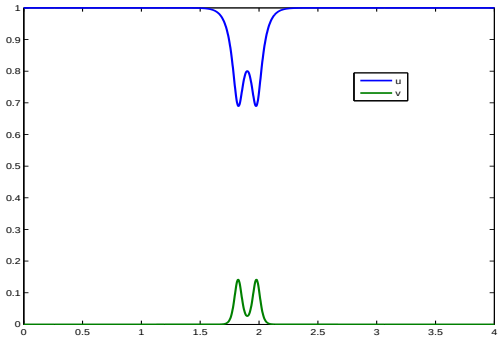


Figure 3.18: SDPR method detects double peak scattor for (3.49).

0.1682 to 0.1843. To obtain a second scattor requires to tighten the bounds lbd_u and ubd_u:

$$\begin{aligned} \text{lbd}_{ui} &= \begin{cases} 0.4, & \text{if } i \in \{1,\dots,\lceil \frac{N}{10} \rceil\} \cup \{\lceil \frac{9N}{10} \rceil,\dots,N\} \\ 0, & \text{else} \end{cases}, \\ \text{ubd}_{ui} &= \begin{cases} 0.6, & \text{if } i \in \{1,\dots,\lceil \frac{N}{10} \rceil\} \cup \{\lceil \frac{9N}{10} \rceil,\dots,N\} \\ 0.8, & \text{else} \end{cases}. \end{aligned} \tag{3.51}$$

The bounds for v remain unchanged. Applying the SDPR method with the tighter bounds (3.51) for $k = 0.0535$ yields a second solution $(u,v)^2$ with $\| u^2 \| = 0.1566 \neq 0.1843 = \| u^1 \|$. However, it is not possible to obtain accurate approximations for the second solution when applying the SDPR method for smaller choices of k. Therefore we apply a continuation technique based on the SDPR method to obtain the second solution for smaller k: Set $\tilde{k} = 0.0535$, choose some stepsize Δk and apply the SDPR method to (3.49) for $k = \tilde{k} - \Delta k$ with objective function $G_{\tilde{k}}$ defined by

$$G_{\tilde{k}}(u,v) = \sum_{i=1}^{N} \left(u_i - u_{\tilde{k}i}\right)^2,$$

where $u_{\tilde{k}}$ solution of (3.49) for $k = \tilde{k}$. Update $\tilde{k} = \tilde{k} - \Delta k$ and iterate. Following this procedure we obtain $(u,v)^2$ for k decreasing from 0.0535 to 0.05281 and $\| u^2 \|$ increasing from 0.1566 to 0.1655. Thus, the bifurcation point of (3.49) is $k_0 \approx 0.05281$ for $f = 0.0270$. The results of the SDPR method are illustrated in the bifurcation diagram in Figure 3.19.

Three-component reaction-diffusion equation Consider another one-dimensional steady state equation from [74]:

$$\begin{aligned} D_u u_{xx}(x) + 2u(x) - u(x)^3 - \kappa_3 v(x) - \kappa_4 w(x) + \kappa_1 &= 0 \quad \forall\, x \in [0,0.5]\,, \\ \tfrac{1}{\tau}\left(D_v v_{xx}(x) + u(x) - v(x)\right) &= 0 \quad \forall\, x \in [0,0.5]\,, \\ \tfrac{1}{\theta}\left(D_w w_{xx}(x) + u(x) - w(x)\right) &= 0 \quad \forall\, x \in [0,0.5]\,, \\ -2 \leq u(x), v(x), w(x) &\leq 2 \quad \forall\, x \in [0,0.5]\,, \end{aligned} \tag{3.52}$$

under Neumann boundary conditions. We set the parameters to $D_u = 5e-6$, $D_v = 5e-5$, $D_w = 1e-2$, $\kappa_1 = -7$, $\kappa_3 = 1$, $\kappa_4 = 8.5$, $\tau = 16.1328$ and $\theta = 1$. In [74] the existence of two scattors, *twin-horn* and *fusion*, is shown for this setting. We choose $F(u,v,w) = -u_{\lceil \frac{N}{2} \rceil}$. *fusion* has one positive eigenvalue and

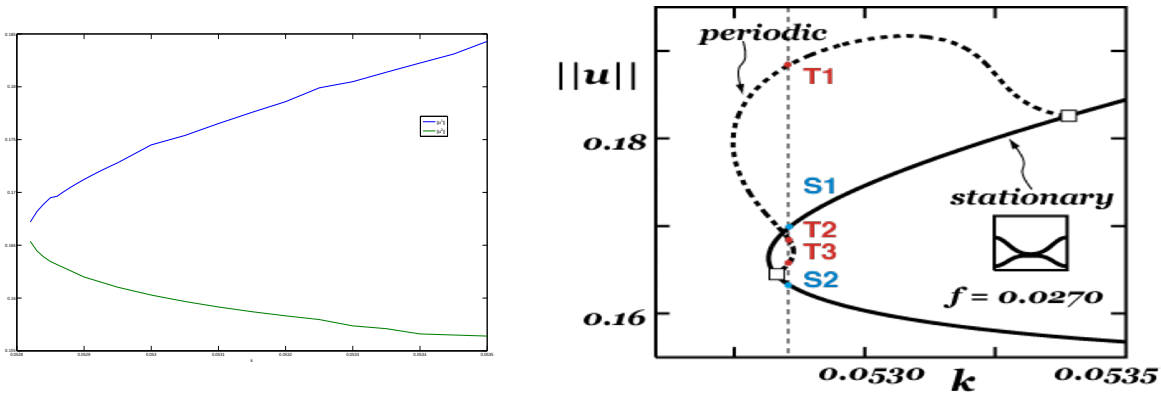


Figure 3.19: Bifurcation diagram by the SDPR method (left) compared to the one (right) from [100].

twin-horn three positive eigenvalues $(\lambda_1, \lambda_2, \lambda_3) = (0.9069, 0.1297, 0.0138)$. Applying the SDPR method with SQP as local solver starting from $N = 40$ and subsequent application of the grid-refining method with strategy 3b, we obtain a highly accurate approximation of the scattor *fusion* pictured in Figure 3.20.

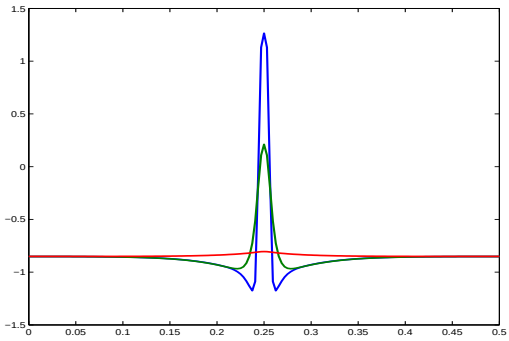


Figure 3.20: SDPR method detects *fusion* for (3.52), with u (blue), v (green) and w (red).

Swift-Hohenberg equation

Another type of reaction-diffusion equations arising from modeling pattern formations is given by **Swift-Hohenberg equations** [80]. Swift-Hohenberg equations are interesting from the point of view of pattern formation because they have many qualitatively different stationary solutions. Differential equations with many solutions are of particular interest for applying the SDPR method. We examine a stationary Swift-Hohenberg equation from [80]:

$$\begin{array}{lll} -u_{xxxx}(x) - 2u_x(x) - (1-\alpha)u(x) - u(x)^3 & = 0 & \forall\, x \in [0, L], \\ u(0) = u(L) = u_{xx}(0) = u_{xx}(L) & = 0, & \end{array} \tag{3.53}$$

where $\alpha \in \mathbb{R}$. It is known that the shape and number of solutions depends on the choice for L. For our numerical analysis we set the parameters to $L = 9$ and $\alpha = 0.3$. The forth order derivative is approximated

by

$$u_{xxxx}(x_i) \approx \frac{1}{h_x^4}\left(u_{i+2} - 4u_{i+1} + 6u_i - 4u_{i-1} + u_{i-2}\right).$$

We consider the following functions as objective functions:

$$\begin{array}{lll} F_1(u) = -\sum_{i=1}^N u_i, & F_2(u) = \sum_{i=1}^N u_i, & F_3(u) = -u_2, \\ F_4(u) = -u_{\lceil \frac{N}{3} \rceil} & F_5(u) = -u_{\lceil \frac{2N}{3} \rceil}, & F_6(u) = -u_{\lceil \frac{N}{4} \rceil}, \end{array} \tag{3.54}$$

We apply the SDPR method with $\omega = 3$ for $N = 40$ and varying objective functions, and obtain five different solutions for (3.53) as reported in Table 3.13 and pictured in Figure 3.21.

SDP relaxation	POP to QOP	Local Solver	Objective	ϵ_{sc}	t_C
Dual	no	SQP	F_1	-1e-9	29
Dual	no	SQP	F_2	-3e-15	32
Dual	no	SQP	F_3	-3e-10	31
Dual	no	SQP	F_4	-3e-10	69
Dual	no	SQP	F_5	-1e-13	58

Table 3.13: Results of the SDPR method with $\omega = 3$, $N = 40$ and varying objective function.

In a next step we investigate a more systematic approach to enumerate solutions of (3.53). Applying Algorithm 3.1 does not yield an accurate enumeration of solutions for (3.53). The numerical accuracy of the solutions deteriorates from the second iteration. This may be explained by the fact, that the relaxation of the quadratic constraints added in step 3 of Algorithm 3.1 is too weak to provide a good starting point for the initial solver. We therefore consider a variant of Algorithm 3.1. Instead of adding constraint

$$(u_i - u_i^{(k-1)})^2 \geq \epsilon^k \text{ for all } i \text{ with } b_i = 1$$

to the POP, we consider $2^{\sum_{i=1}^n b_i}$ POPs where we add the constraints

$$u_i \leq u_i^{(k-1)} - \epsilon_i^k$$

or

$$u_i \geq u_i^{(k-1)} + \epsilon_i^k$$

for every i with $b_i = 1$. It is clear, that the solution with kth smallest objective value is feasible for exactly one of the $2^{\sum_{i=1}^n b_i}$ POPs. This procedure has the advantage that the added linear constraint remains hard under the SDP relaxation, it has the disadvantage that many SDPs need to be solved in course of the algorithm. Therefore, we consider this modified enumeration algorithm for the SDPR method with small relaxation order. We apply this modified enumeration algorithm and SDPR method with $N = 50$, $b^k := b := (0,0,0,0,0,0,0,0,0,1,1,0,\ldots,0)$, $\epsilon^k := \epsilon := 0.1$, POP to QOP transformation and $\omega = 2$, F_6 as objective and 6 iterations of the enumeration algorithm. In its first five iterations we obtain the same five solutions as those obtained by the SDPR method when choosing different objective functions, the output of the sixth iteration is the same as for the fifth one, although it is not a feasible solution for the POP from the sixth iteration. Our numerical results are reported in Table 3.14 and pictured in Figure 3.21. Note, applying the modified enumeration algorithm requires a longer time to obtain the five solutions to (3.53). However, enumerating the five solutions one by one with respect to the same objective function is a more systematic approach than choosing five arbitrary objective functions.

3.3.4 Differential algebraic equations

The class of PDE problems (3.14) contains so called **differential algebraic equations (DAE)**. These are differential equations, where the derivatives of several unknown functions do not occur explicitly. We

k	SDP relaxation	POP to QOP	Local Solver	ϵ_{sc}	t_C	$F_6(u^{(k)})$
0	Dual	yes	SQP	-4e-15	30	-0.43
1	Dual	yes	SQP	-9e-11	153	-0.17
2	Dual	yes	SQP	-5e-10	293	0.00
3	Dual	yes	SQP	-9e-13	400	0.17
4	Dual	yes	SQP	-1e-11	500	0.43
5	Dual	yes	SQP	-1e-11	593	0.43

Table 3.14: Results of the modified enumeration algorithm and SDPR method with $\omega = 2$, $N = 50$ and F_6 as objective.

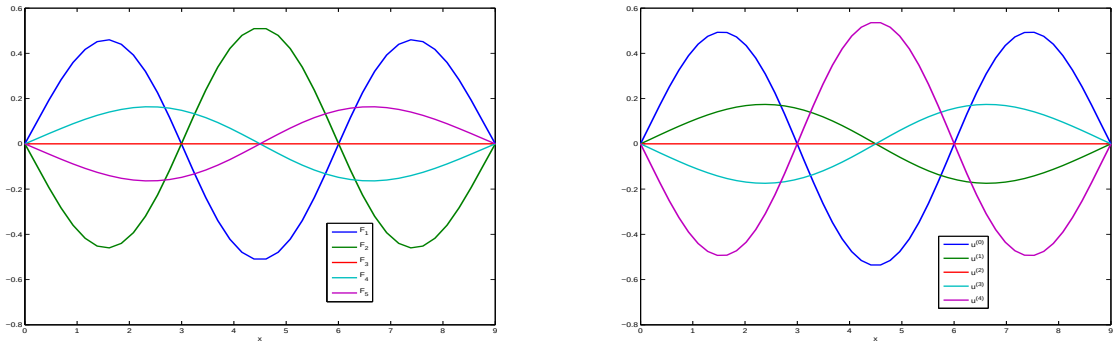


Figure 3.21: SDPR method with varying objective for $N = 40$ (left) and modified enumeration algorithm with SDPR method for $N = 50$ (right).

demonstrate for the following example that the SDPR method can be applied to solve DAE as well. Consider the DAE problem

$$\begin{aligned} \dot{u}_1(x) &= u_3(x), \\ 0 &= u_2(x)\,(1-u_2(x))\,, \\ 0 &= u_1(x)\,u_2(x) + u_3(x)\,(1-u_2(x)) - x, \\ u_1(0) &= u_0. \end{aligned} \qquad \forall\, x \in [0,T] \tag{3.55}$$

It is easy to see that two closed-form solutions u^1 and u^2 are given by

$$\begin{aligned} u^1(x) &= \left(u_0 + \tfrac{x^2}{2},\, 0,\, x\right)^T \quad x \in [0,T], \\ u^2(x) &= (x,\, 1,\, 1)^T \quad x \in [0,T]. \end{aligned}$$

We choose lbd $\equiv 0$ and ubd $\equiv 10$ for each function u_1, u_2 and u_3 and define two objective functions F_1 and F_2,

$$F_1(u) = \sum_{i=1}^{N_x} u_{1i}, \qquad F_2(u) = \sum_{i=1}^{N_x} u_{2i}.$$

First we choose $u_0 = 0$ and apply the SDPR method with F_2 as an objective, and we obtain an highly accurate approximation for u^2, which is documented in Table 3.15 and Figure 3.22.

objective	ω	N_x	u_0	ϵ_{obj}	ϵ_{sc}	t_C
F_2	2	100	0	4e-10	-4e-10	29
F_2	2	200	0	3e-9	-3e-9	122
F_1	2	200	1	8e-9	-2e-6	98
F_1	2	200	2	3e-10	-4e-8	107
F_1	2	10	0.5	3e-10	-3e-7	4
F_1	2	20	0.5	3e-10	-9e-6	9
F_1	2	30	0.5	8e-10	-3e-3	15
F_1	2	40	0.5	7e-8	-1e-1	24
F_1	3	30	0.5	9e-9	-2e-3	51
F_1	4	30	0.5	8e-9	-6e-4	210

Table 3.15: Results of SDPR method for (3.55) with $T = 2$.

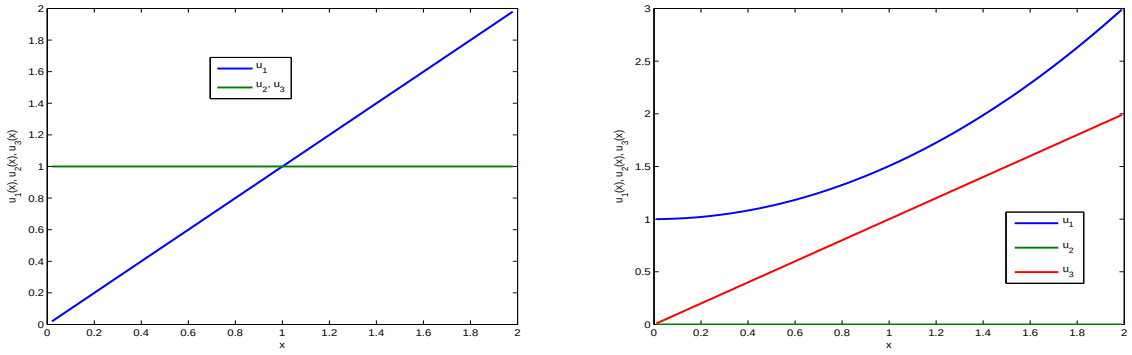


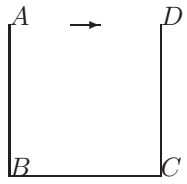
Figure 3.22: Solutions u^2 (left) and u^1 (right) of DAE problem (3.55).

Next we apply the SDPR method with F_1 as objective. For $u_0 \in \{1,\ 2\}$ highly accurate approximations of u^1 are obtained. An interesting phenomenon is observed in case u_0 is small. For instance, if we choose $u_0 = 0.5$ and $\omega = 2$, we get a highly accurate solution for $N_x = 10$. But, as we increase N_x stepwise to 40, the accuracy decreases, although the relaxation order remains constant. For numerical details see Table 3.15. This effect can be slightly compensated by increasing ω, as demonstrated for the case $N_x = 30$. But due to the limited capacity of current SDP solvers it is not possible to increase ω as much as needed to obtain a high accuracy solution. However, we obtain highly accurate approximations to both solutions of (3.55) by the SDPR method even without applying a locally fast convergent method.

3.3.5 The steady cavity flow problem

One of the most challenging present PDE problems is the numerical analysis of the Navier-Stokes equations. As a first step to attempt this class of PDE problems by the SDPR method we consider the **steady cavity flow problem**, which contains a steady state version of the Navier-Stokes equations. The steady cavity flow problem is a simple model of a flow with closed streamlines and is used for examining and validating numerical solution techniques in fluid dynamics. Although it has been discussed in the literature of numerical analysis of fluid mechanics (see, e.g., [40], [12], [32], [14], [98]), it is still an interesting problem to a number of researchers for a range of Reynolds numbers. The setting of the steady cavity flow problem is the following.

Let $(v_1(x,y,t), v_2(x,y,t))$ be the velocity of the two dimensional cavity flow of an incompressible fluid on the cavity region $ABCD$ with the coordinates $A = (0,1)$, $B = (0,0)$, $C = (1,0)$, $D = (1,1)$.



It follows from the continuity equation of the incompressible fluid (preservation of the mass) $\frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y} = 0$ that there exists a function $\psi(x,y,t)$ such that

$$\frac{\partial \psi}{\partial x} = -v_2, \ \frac{\partial \psi}{\partial y} = v_1. \tag{3.56}$$

Put $\mathbf{v} = (v_1, v_2, 0)$. $\vec{\phi} = \mathrm{rot}\,\mathbf{v}$ is called the vorticity. Since the last coordinate of $\mathbf{v}$ is 0, $\vec{\phi}$ can be written as $(0,0,\phi(x,y,t))$. The continuity equation and the Navier-Stokes equation (preservation of the momentum) can be written as follows in terms of ψ and ϕ.

$$\Delta\psi = -\phi \tag{3.57}$$

$$\frac{\partial \phi}{\partial t} = \frac{\partial \psi}{\partial y}\frac{\partial \phi}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \phi}{\partial y} + \frac{1}{R}\Delta\phi, \tag{3.58}$$

where the parameter R is called the **Reynolds number**.

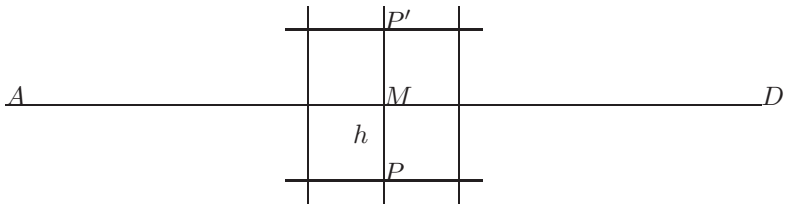
The steady cavity flow problem is (3.57) and (3.58) with the steady condition $\frac{\partial \phi}{\partial t} = 0$ and the boundary condition

$$v_1(0,y) = v_1(x,0) = v_1(1,y) = 0 \quad \text{on } AB, BC, CD \tag{3.59}$$

$$v_2(0,y) = v_2(x,0) = v_2(1,y) = 0 \quad \text{on } AB, BC, CD \tag{3.60}$$

$$v_1(x,1) = s, v_2(x,1) = 0 \quad \text{on } AD \tag{3.61}$$

Here s is the velocity of the stream out of the cavity $ABCD$. We set the boundary velocity at AD to $s := 1$. Due to its boundary conditions PDE problem (3.56), (3.57), (3.58), (3.59) - (3.61) is not of form (3.14). Therefore, we need to follow a specialized strategy to discretize this problem via a finite different scheme. We devide the square $ABCD$ into a $N \times N$ mesh and define $h := \frac{1}{N-1}$. We translate the boundary conditions for v_1 and v_2 into boundary conditions for ψ and ϕ. It follows from (3.59), (3.60), (3.61) that the function ψ is constant on the boundaries AB, BC, CD, DA. Since ψ is continuous, we suppose that $\psi = 0$ on the boundaries. The boundary condition for ϕ is a little complicated. We derive it from the discussion in [98], p 162: Let us consider the case of the boundary AD first. We take a mesh point M on AD. Let P be the mesh point inside the cavity adjacent to M and P' the mirror image of P with respect to AD. We supposed that the size of the mesh is h.



We denote the value of ψ at the point P by $\psi(P)$ or ψ_P. Moreover, we have $-\phi(M) = \Delta\psi(M) = \psi_{yy} \approx \frac{\psi_P - 2\psi_M + \psi_{P'}}{h^2}$. We need to determine the value of $\psi_{P'}$ to get an approximate value of ϕ at M. By using the central difference approximation, $s = 1 = v_2 = \frac{\partial\psi}{\partial y}(M) \approx \frac{\psi_{P'} - \psi_P}{2h}$ holds. Then, $\psi_{P'} \approx 2h + \psi_P$. Therefore, we have

$$\phi_M \approx -\frac{2\psi_P + 2h}{h^2}. \tag{3.62}$$

Analogously, we obtain

$$\phi_M \approx -\frac{2\psi_P}{h^2} \tag{3.63}$$

when M is a grid point on AB or BC or CD and P is the adjacent internal grid point of M. From this discussion follows, we obtain the following finite difference scheme for the steady cavity flow problem.

$$g^1_{i,j}(\psi, \phi) = 0 \quad \forall\, 2 \le i, j \le N-1, \tag{3.64}$$

$$g^2_{i,j}(\psi, \phi) = 0 \quad \forall\, 2 \le i, j \le N-1, \tag{3.65}$$

$$\begin{array}{ll} \psi_{1,j} = \psi_{N,j} = 0 & \forall\, j \in \{1, \ldots, N\}, \\ \psi_{i,1} = \psi_{i,N} = 0 & \forall\, i \in \{1, \ldots, N\}, \\ \phi_{1,j} = -2\frac{\psi_{2,j}}{h^2} & \forall\, j \in \{1, \ldots, N\}, \\ \phi_{N,j} = -2\frac{\psi_{N-1,j}}{h^2} & \forall\, j \in \{1, \ldots, N\}, \\ \phi_{i,1} = -2\frac{\psi_{i,2}}{h^2} & \forall\, i \in \{1, \ldots, N\}, \\ \phi_{i,N} = -2\frac{\psi_{i,N-1}+h}{h^2} & \forall\, i \in \{1, \ldots, N\}, \end{array} \tag{3.66}$$

where

$$\begin{array}{rl} g^1_{i,j}(\psi, \phi) := & -4\phi_{i,j} + \phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} \\ & +\frac{R}{4}(\psi_{i+1,j} - \psi_{i-1,j})(\phi_{i,j+1} - \phi_{i,j-1}) \\ & -\frac{R}{4}(\psi_{i,j+1} - \psi_{i,j-1})(\phi_{i+1,j} - \phi_{i-1,j}), \\ g^2_{i,j}(\psi, \phi) := & -4\psi_{i,j} + \psi_{i+1,j} + \psi_{i-1,j} + \psi_{i,j+1} \\ & +\psi_{i,j-1} + h^2\phi_{i,j}. \end{array}$$

We call the polynomial system (3.64), (3.65), (3.66) the **discrete steady cavity flow problem** denoted as DSCF(R, N). It depends on two parameters, the **Reynolds number** R and the discretization N of the cavity region $ABCD = \Omega = [0, 1]^2$. Its dimension is given by $n = 2(N-2)^2$.

Remark 3.2 *We conjecture that the discrete cavity flow problem DSCF(R, N) has finite complex solutions. In other words, it defines a zero-dimensional ideal. We checked this conjecture up to $N = 5$ by Gröbner basis computation.*

DSCF(R, N) is a sparse, polynomial system which we apply the SDPR method and Algorithm 3.1 to. The choice for F in 3.26 is motivated by the fact, that one is interested in the solution to the PDE problem which minimizes the kinetic energy given by

$$\int\int_{ABCD} \left(\frac{\partial\psi}{\partial y}\right)^2 + \left(\frac{-\partial\psi}{\partial x}\right)^2 dxdy. \tag{3.67}$$

Thus, by discretizing (3.67) we yield the following function as a canonical choice for F:

$$\begin{array}{rl} F(\psi, \omega) = \frac{1}{4} \displaystyle\sum_{2 \le i,j \le N-1} & \psi^2_{i+1,j} + \psi^2_{i-1,j} + \psi^2_{i,j+1} + \psi^2_{i,j-1} \\ & -2\psi_{i+1,j}\psi_{i-1,j} - 2\psi_{i,j+1}\psi_{i,j-1}. \end{array} \tag{3.68}$$

We denote the optimization problem to minimize F subjected to DSCF(R, N) as he **steady cavity flow optimization problem CF(R, N)**. It is characterized by a simple proposition.

Proposition 3.4 *a) CF$(0, N)$ is a **convex quadratic program** for any N.*

b) CF(R, N) is ***non-convex*** *for any N, if* $R \neq 0$.

Proof:

a) In case $R = 0$ all constraints are linear. Furthermore, the objective function can be written as $F = \sum_{i,j} F^1_{i,j} + F^2_{i,j}$, where

$$F^1_{i,j}(\psi, \omega) = \begin{pmatrix} \psi_{i-1,j} \\ \psi_{i+1,j} \end{pmatrix}^T \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix} \begin{pmatrix} \psi_{i-1,j} \\ \psi_{i+1,j} \end{pmatrix}.$$

It follows that $F^1_{i,j}$ is convex as $\begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}$ positive semidefinite with eigenvalues 0 and 4. The convexity of $F^2_{i,j}$ follows analoguously. Thus, F can be written as a sum of convex function and is therefore convex as well. The proposition follows.

b) In case $R \neq 0$, the equality constraint function $g^1_{i,j}$ is indefinite quadratic. Thus, CF(R, N) is a non-convex quadratic program. □

Solving CF(R, N) by the SDPR method

First, we apply the SDPR method with $\omega = 1$ and Newton's method as local solver to CF$(100, N)$. Highly accurate solutions with $\epsilon_{\text{sc}} < 1e-10$ are obtained for $N \in \{10, 15, 20\}$. By applying the grid-refining method we succeed in extending the solutions to grids of size 30×30 and 40×40, as pictured for $N = 40$ in Figure 3.23 and reported in Table 3.16. Thus, it seems reasonable to conclude, that the minimal energy solution of CF$(100, N)$ converges to a continuous solution of the steady cavity flow problem for $N \to \infty$. The discrete steady cavity flow problem has multiple solutions. It is an advantage of the SDPR method to show that the minimal kinetic energy solution $u^\star(N)$ of CF(R, N) converges to an analytic solution for $N \to \infty$.

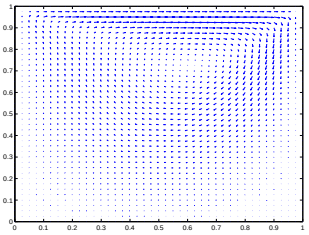


Figure 3.23: (v_1, v_2) for solution u of CF$(100, 40)$.

N	ω	ϵ_{sc}	t_C	$F(u^\star)$
10	1	4e-11	14	0.0169
15	1	6e-16	255	0.0313
20	1	6e-16	948	0.0409
30	1	4e-11	1759	0.0503
40	1	4e-11	4156	0.0554

Table 3.16: Results for $CF(100, N)$ for increasing N.

Second, we apply the SDPR method to CF(R, N) for a much larger R. For example we examine $CF(10000, N)$ for $N \in \{8, \ldots, 18\}$. For all tested discretizations we were able to find accurate solutions by the SDPR method with $\omega = 1$ and SQP as local solver, c.f. Table 3.17 and Figure 3.24.

N	ω	ϵ_{sc}	t_C	$F(u^{(k)})$
8	1	2e-7	7	1.5e-6
10	1	3e-10	21	3.2e-6
12	1	1e-7	49	6.0e-6
14	1	5e-9	99	1.1e-5
16	1	4e-12	199	1.9e-5
18	1	2e-8	501	3.9e-5

Table 3.17: Results for $CF(10000, N)$ for increasing N.

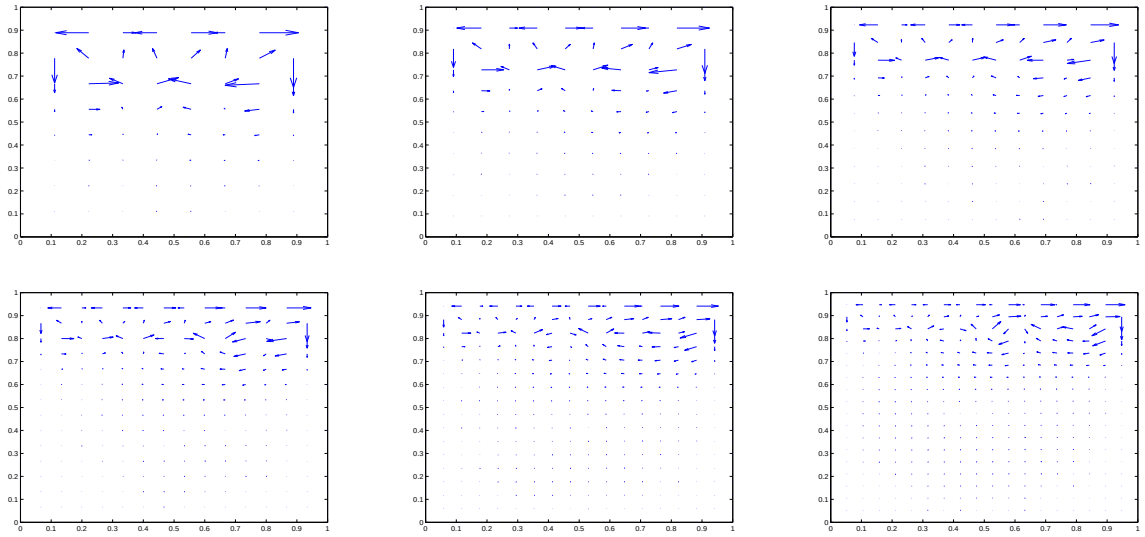


Figure 3.24: Solutions of $CF(10000, N)$ of the SDPR method for $N = 8$ (left, top), $N = 10$ (center, top), $N = 12$ (right, top), $N = 14$ (left, bottom), $N = 16$ (center, bottom) and $N = 18$ (right, bottom).

If we compare the pictures in Figure 3.24, it seems the SDPR(1) solution of $CF(10000, 1, N)$ evolves into some **stream-like** solution for increasing N. However, unlike the solutions of $CF(100, 1, N)$, we have not been able to expand this solution to a grid of higher resolution by the grid-refinement method. Therefore, it is possible the solution pictured in Figure 3.24 is a fake solution, which confirms that the Steady Cavity Flow problem becomes a hard problem for increasing Reynolds number.

Enumerating the solutions of DSCF(R, N)

A further interesting question is to find all solutions of the cavity flow problem, in particular for large Reynolds number. Therefore, we examine the efficiency of Algorithm 3.1 for enumerating the solutions of DSCF(R, N) with respect to their discretized kinetic energy. For the parameter $b^k \in \{0, 1\}^n$ to be chosen in each iteration of Algorithm 3.1 we restrict ourselves to the case where b^k is given by

$$b_i^k = \begin{cases} 1, & \text{if } i \in \{1, \ldots, b_1^k\} \cup \{\frac{n}{2} + 1, \ldots, \frac{n}{2} + b_2^k\}, \\ 0, & \text{else.} \end{cases}$$

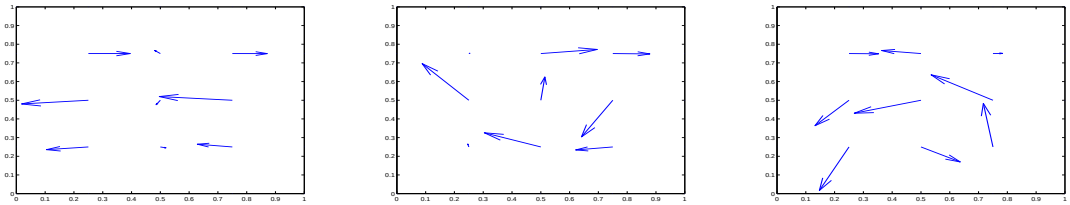


Figure 3.25: (v_1, v_2) for solutions $u^{(0)}$ (left), $u^{(1)}$ (center) and $u^{(2)}$ (right) of CF$(4000, 5)$ on the interior of $[0, 1]^2$.

Thus b^k is defined by the two parameters $b_1^k, b_2^k \in \{1, \ldots, \frac{n}{2}\}$. The parameters ϵ_1^k and ϵ_2^k are corresponding to the constraints imposed by b_1^k and b_2^k, respectively.

CF(4000,5): In a first setting we choose the discretization $N = 5$, i.e. the dimension is $n = 2 \cdot 3^2 = 18$. This dimension is small enough to apply the Gröbner basis method to determine all complex solutions of DSCF(R, N). Therefore, we are able to verify whether the solutions provided by Algorithm 3.1 are optimal. The computational results are given in Table 3.18. Comparing the solutions of the SDPR method to all

k	ω	ϵ_1^k	b_1^k	b_2^k	t_C	ϵ_{sc}	$F(u^{(k)})$	solution
0	1	-	-	-	2	2e-10	4.6e-4	$u^{(0)}$
1	1	1e-3	3	0	5	5e-4	6.3e-4	$u^{(1)}$
2	1	1e-3	3	0	8	5e-4	1.0e-3	$u^{(2)}$

Table 3.18: Results of Algorithm 3.1 for CF$(4000, 5)$.

solutions of the polynomial system obtained by polyhedral homotopy method or Gröbner basis method, it turns out that the solutions $u^{(0)}$, $u^{(1)}$ and $u^{(2)}$ indeed coincide with the three smallest energy solutions $u^{(0)\star}$, $u^{(1)\star}$ and $u^{(2)\star}$. The velocities (v_1, v_2) derived from these three solutions via (3.56) are displayed in Figure 3.25. Note, that the third smallest energy solution $u^{(2)}$ shows a vortex in counter-clockwise direction, which may indicate that this solution is a fake solution.

CF(20000,7): We apply Algorithm 3.1 with $\omega = 1$ to CF$(20000, 7)$ and obtain the results in Table 3.19. The two parameter settings $(\epsilon_1^1, b_1^1) = (1e-3,\ 1)$ and $(\epsilon_1^1, b_1^1) = (1e-6,\ 5)$ are not sufficient to obtain an other solution than $u^{(0)}$, whereas $(\epsilon_1^1, b_1^1) = (1e-5,\ 5)$ yields $u^{(1)}$, a solution of larger energy. After another iteration with $(\epsilon_1^2, b_1^2) = (1e-5,\ 5)$ we obtain a third solution $u^{(3)}$ of even larger energy.

k	ω	ϵ_1^k	b_1^k	b_2^k	t_C	ϵ_{sc}	$F(u^{(k)})$	solution
0	1	-	-	-	2	3e-7	3.4e-4	$u^{(0)}$
1	1	1e-3	1	0	5	5e-4	3.4e-4	$u^{(0)}$
1	1	1e-6	5	0	5	6e-6	3.4e-4	$u^{(0)}$
1	1	1e-5	5	0	9	5e-6	5.9e-4	$u^{(1)}$
2	1	1e-5	5	0	14	5e-6	5.2e-3	$u^{(2)}$

Table 3.19: Results of Algorithm 3.1 for CF$(20000, 7)$.

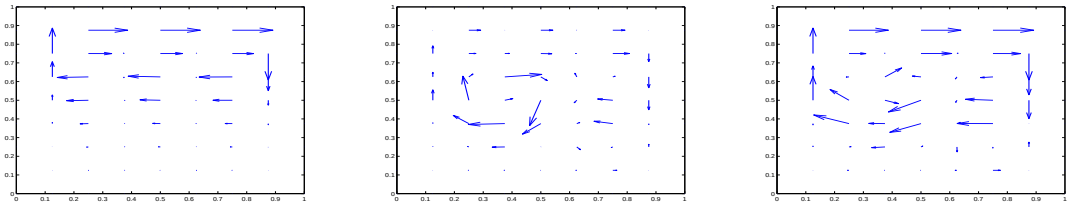


Figure 3.26: (v_1, v_2) for solutions $u^{(0)}$ (left), $u^{(1)}$ (center) and $u^{(2)}$ (right) of CF$(20000, 7)$ on $[0, 1]^2$.

It is interesting to observe in Figure 3.26 that $u^{(1)}$ and $u^{(2)}$ are one-vortex solutions, whereas there seems to be no vortex in the smallest energy solution $u^{(0)}$.

CF(40000,7): Next, we examine $CF(40000, 7)$, which is a good example to demonstrate that solving DSCF(R, N) and CF(R, N) are becoming more difficult for larger Reynolds numbers. As for the previous problem, the dimension of the POP is $n = 50$, which is too large to be solved by Gröbner basis. Our computational results are reported in Table 3.20.

k	ω	ϵ_1^k	b_1^k	b_2^k	t_C	$\epsilon_{\rm sc}$	$F(u^{(k)})$	solution
0	1	-	-	-	3	2e-7	3.4e-4	$u^{(0)}(1)$
1	1	5e-6	5	0	7	6e-9	7.3e-4	$u^{(1)}(1)$
2	1	5e-6	5	0	11	3e-6	5.9e-4	$u^{(2)}(1)$
3	1	8e-6	5	0	16	5e-6	2.3e-4	$u^{(3)}(1)$
0	2	-	-	-	5872	8e-10	2.6e-4	$u^{(0)}(2)$

Table 3.20: Results of Algorithm 3.1 for $CF(40000, 7)$.

Solution $u^{(2)}(1)$ is of smaller energy than $u^{(1)}(1)$, and $u^{(3)}(1)$ is even of smaller energy than $u^{(0)}(1)$. Thus, unlike the solutions for CF(20000,7) reported in Table 3.19, the solutions of CF(40000,7) are not enumerated in the correct order. This phenomenon can be explained by the fact, that the SDP relaxation with $\omega = 1$ is not tight enough to yield a solution that converges to $u^\star$ under the local optimization procedure. The energy of $u^{(0)}(2)$ obtained by SDPR(2) is smaller than the one of $u^{(0)}(1)$, but it is not the global minimizer as well. In fact, Algorithm 3.1 with $\omega = 1$ generates a better solution $u^{(3)}(1)$ (with smaller energy) in 3 iterations requiring 16 seconds computation time, compared to solution $u^{(0)}(2)$ obtained by applying the SDPR method with $\omega = 2$ to $CF(40000, 7)$ requiring 5872 seconds. Thus, despite failing to enumerate the smallest energy solutions in the right order with $\omega = 1$, applying the enumeration algorithm with relaxation order $\omega = 1$ is far more efficient than the original sparse SDP relaxation (2.18) with $\omega = 2$ for approximating the global minimizer of POP (3.26). It is a future problem to make this construction systematic.

Alternative finite difference scheme

To derive DSCF(R, N) we discretize the Jacobian $\frac{\partial \psi}{\partial y}\frac{\partial \phi}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \phi}{\partial y}$ by the standard central difference scheme. Arakawa [3] showed that the standard central difference scheme does not keep important physical invariants. Therefore, Arakawa proposed an alternative finite difference discretization for the Jacobian, that is shown to preserve those invariants. We use this alternative scheme to derive an *alternative discrete steady cavity flow problem* ADSCF(R, N) and solve it via the SDPR method. In ADSCF(R, N), the finite difference

approximation for $\frac{\partial\psi}{\partial y}\frac{\partial\phi}{\partial x} - \frac{\partial\psi}{\partial x}\frac{\partial\phi}{\partial y}$ is replaced by

$$\begin{aligned}
&\tfrac{\partial\psi}{\partial y}\tfrac{\partial\phi}{\partial x} - \tfrac{\partial\psi}{\partial x}\tfrac{\partial\phi}{\partial y}(x_i,\, y_j) \approx \\
&-\tfrac{1}{12h^2}[(\phi_{i,j-1} + \phi_{i+1,j-1} - \phi_{i,j+1} - \phi_{i+1,j+1})\,(\psi_{i+1,j} + \psi_{i,j}) \\
&-(\phi_{i-1,j-1} + \phi_{i,j-1} - \phi_{i-1,j+1} - \phi_{i,j+1})\,(\psi_{i,j} + \psi_{i-1,j}) \\
&+(\phi_{i+1,j} + \phi_{i+1,j+1} - \phi_{i-1,j} - \phi_{i-1,j+1})\,(\psi_{i,j+1} + \psi_{i,j}) \\
&-(\phi_{i+1,j-1} + \phi_{i+1,j} - \phi_{i-1,j-1} - \phi_{i-1,j})\,(\psi_{i,j} + \psi_{i,j-1}) \\
&+(\phi_{i+1,j} - \phi_{i,j+1})\,(\psi_{i+1,j+1} + \psi_{i,j}) - (\phi_{i,j-1} - \phi_{i-1,j})\,(\psi_{i,j} + \psi_{i-1,j-1}) \\
&+(\phi_{i,j+1} - \phi_{i-1,j})\,(\psi_{i-1,j+1} + \psi_{i,j}) - (\phi_{i+1,j} - \phi_{i,j-1})\,(\psi_{i,j} + \psi_{i+1,j-1})].
\end{aligned} \tag{3.69}$$

Note, ADSCF(R, N) is less sparse than DSCF(R, N) and it is more difficult to derive accurate solutions by the SDPR method with relaxation order $\omega = 1$. However, we succeed in solving ADSCF(R, N) in some instances. For example, in Table 3.21 and Figure 3.27 we compare the minimum kinetic energy solutions obtained for DSCF$(5000, N)$ and ADSCF$(5000, N)$. It is interesting that the vortex in the minimum kinetic energy solution for ADSCF$(5000, N)$ is preserved for increasing N, whereas the vortex in solution for DSCF$(5000, N)$ seems to deteriorate.

Problem	ϵ_{sc}	t_C	$F(u^\star)$
ADSCF(5000,14)	7e-12	1304	1.8e-4
ADSCF(5000,16)	5e-10	2802	3.1e-4
DSCF(5000,14)	1e-11	419	5.6e-4
DSCF(5000,16)	3e-10	768	1.1e-4

Table 3.21: Results for solving ADSCF(5000,N) compared to DSCF(5000,N).

Solutions of CF(R, N) for increasing R

In order to understand why convergence of discrete approximations to the analytic solution is a lot more difficult to obtain for large R, we examine the behavior of the minimal energy solution of DSCF(R, N) and CF(R, N), respectively, for increasing Reynolds number R. The SDPR method is one possible approach to solve DSCF(R, N). If ω is chosen sufficiently large, the output u of the SDPR method is guaranteed to accurately approximate the minimal energy solution $u^\star$ of CF(R', N) and DSCF(R', N), respectively. In order to show the advantage of the SDPR method we compare our results to solutions of DSCF(R', N) obtained by the following standard procedure:

Method 3.2 *Naive homotopy-like* **continuation method**

1. *Choose the parameters R', N and a step size ΔR.*
2. *Solve DSCF$(0, N)$, i.e. a linear system, and obtain its unique solution u^0.*
3. *Increase R^{k-1} by ΔR: $R^k = R^{k-1} + \Delta R$*
4. *Apply Newton's method to DSCF(R^k, N) starting from u^{k-1}. Obtain solution u^k as an approximation to a solution of the discrete cavity flow problem.*
5. *Iterate 3. and 4. until the desired Reynold's number R' is reached.*

Note, the continuation method does not necessarily yield the minimal kinetic energy solution of DSCF(R, N). Let $u^\star(R, N)$ denote the global minimizer of CF(R, N), the minimal energy is given by $E_{\min}(R, N) = F(u^\star(R, N))$. Obviously, $E_{\min}(0, N) = F(u_0(N))$ holds. In a next step, the solution of DSCF(R, N) obtained by the continuation method starting from u_0 is denoted as $\tilde{u}(R)$, and its energy as $E_C(R, N) := F(\tilde{u}(R, N))$. As illustrated for $N = 5$ in Figure 3.28, it is possible to find a continuation $\tilde{u}$ of u_0 for all R.

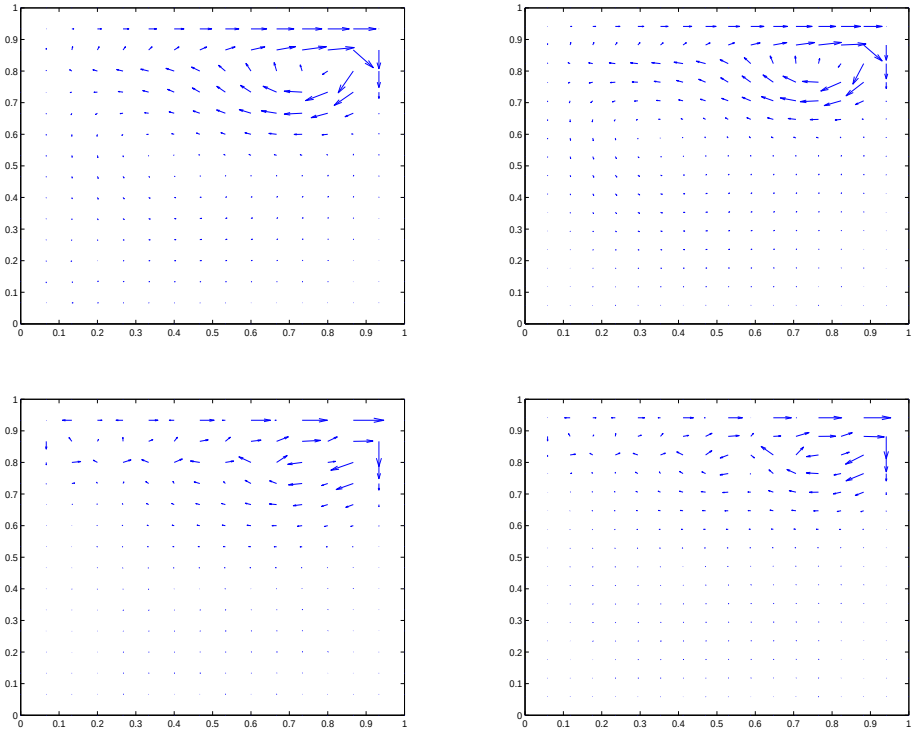


Figure 3.27: Solutions for ADSCF(5000,14) (top left), ADSCF(5000,16) (top right), DCSF(5000,14) (bottom left) and DCSF(5000,16) (bottom right).

For $N = 5$ the dimension of DSCF(R, N) is $n = 18$. This dimension is small enough to solve a polynomial system by Gröbner basis method and to determine all complex solutions of the system. Therefore, we can verify whether SDPR method detects the global minimizer of CF(R, N) or not. It is worth pointing out, that we are able to find the minimal energy solution of CF(R, N) by applying the SDP relaxation method, whereas this solution cannot be obtained by the continuation method. We observe applying the SDPR method with $\omega = 1$ is sufficient to detect the global optimizer for $R \leq 10000$, and for $R \geq 20000$ the global optimizer is obtained by the SDPR method with $\omega = 2$.

In the case of $N = 6$ and $N = 7$ the dimension of the polynomial system is too large to be solved by Gröbner basis method for $R > 0$. For $N = 6$ the continuation method, and the SDPR method with $\omega = 1$ and $\omega = 2$ yield the same solution for all tested R. And in the case of $N = 7$ the continuation solution $\tilde{u}(R)$ is detected by the SDPR method with $\omega = 1$ as well, except the case $R = 6000$, where a solution with slightly smaller energy is detected, as documented in Table 3.23.

Summarizing these results, $F(u_0(N)) \geq F(\tilde{u}(R, N))$ for any of the tested $R > 0$. It is an advantage of the SDPR method to show, $\tilde{u}(R, N)$ is in general not the optimizer of $CF(R, N)$ for increasing R. In fact, for some settings we obtain far better approximations to the minimal energy solution than $\tilde{u}(R, N)$. Furthermore, $E_{\min}(R)$ and $E_C(R)$ are both decreasing in R. The behavior of E_C, E_{SDPR} and $E_{\min}$ coincides for all chosen discretizations N and motivates the following conjecture.

Conjecture 3.1 *Let discretization N be fixed.*

a) $F(u_0(N)) = E_{\min}(0, N) \geq E_{\min}(R, N) \geq 0 \quad \forall R \geq 0.$

R	$N_{\mathcal{C}}$	$N_{\mathbb{R}}$	E_C	$E_{SDPR(1)}$	$E_{SDPR(2)}$
0	1	1	0.0096	0.0096	0.0096
100	37	13	0.0030	0.0030	0.0030
500	37	13	6.2e-4	6.2e-4	6.2e-4
1000	37	13	5.4e-4	5e-4	5e-4
2000	37	13	6.2e-4	6.2e-4	6.2e-4
4000	37	17	6.3e-4	4.6e-4	4.6e-4
6000	36	16	5.7e-4	4.5e-4	4.5e-4
8000	36	16	5.2e-4	4.5e-4	4.5e-4
10000	35	17	4.7e-4	4.5e-4	4.5e-4
30000	35	17	4.5e-4	4.5e-4	2.5e-4
100000	34	16	4.5e-4	4.5e-4	8.8e-5

Table 3.22: Numerical results for $CF(R,5)$, where $E_{SDPR(\omega)}$ the discretized kinetic energy of the solution of SDPR method with relaxation order ω.

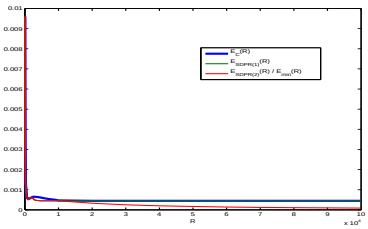


Figure 3.28: $E_C(R)$, $E_{\mathrm{SDPR}(1)}(R)$, $E_{\mathrm{SDPR}(2)}(R)$ and $E_{\min}(R)$ for $N = 5$.

b) $E_{\min}(R,N) \to 0$ *for* $R \to \infty$.

As an application, Conjecture 3.1 can be used as a certificate for the non-optimality of a feasible solution u' of $\mathrm{CF}(R,N)$ in the case $F(u'(R,N)) > E_{\min}(0,N)$. If it is possible to extend u_0 to R via continuation method, $\tilde{u}(R,N)$ can serve as a non-optimality certificate in the case $F(u'(R,N)) > F(\tilde{u}(R,N))$.

3.3.6 Optimal control problems

A class of challenging problems involving differential equations that goes beyond the class of PDE problems (3.14) is optimal control, in particular nonlinear optimal control. To solve nonlinear optimal control problems (OCP) analytically is a challenging problem, even though powerful techniques such as the maximum principle and Hamilton-Jacobi-Bellman optimality equations exist. For numerical methods to solve OCP, we distinguish direct and indirect methods [18, 25, 94, 101]. But, in particular for OCPs with state constraints many numerical methods are difficult to use. A recent approach by Lasserre et al. [54] takes advantage of semidefinite programming (SDP) relaxations to generate a convergent sequence of lower bounds for the optimal value of an OCP. We demonstrate on the following examples that the SDPR method can be applied as well to solve OCPs numerically. An OCP can be discretized via finite difference approximations to obtain a POP satisfying a structured sparsity pattern. The POP we derive from an OCP is essentially of the form (3.26); the main difference being that we do not choose the objective function F, but that F is given as the discretization of the objective of the OCP. By applying the SDPR method to an OCP we (a) obtain

R	0	100	4000	6000	10000
E_C	2.0e-2	7.7e-3	4.1e-4	3.7e-4	3.4e-4
$E_{\mathrm{SDPR(1)}}$	2.0e-2	7.7e-3	4.1e-4	3.6e-4	3.4e-4

Table 3.23: Numerical results for $CF(R,7)$, where $E_{SDPR(\omega)}$ the discretized kinetic energy of the solution of SDPR method with relaxation order ω.

a lower bound for its optimal value, and (b) unlike the approach in [54] we obtain approximations for the optimal value, the optimal control and trajectory. As in the PDE case, it is a feature of the SDPR method that state and/or control contraints can be incorporated by defining additional polynomial equality and inequality constraints.

Control of production and consumption

The following problem arises from the context of control of production and consumption of a factory. Let $x(t)$ be the **amount of output** produced at time $t \geq 0$, $\alpha(t)$ the **control variable** which denotes the **fraction of output reinvested at time** $t \geq 0$, with $0 \leq \alpha(t) \leq 1$. The dynamics of the system are provided by the ODE problem

$$\begin{aligned} \dot{x}(t) &= k\alpha(t)x(t) \quad \forall\, t \in [0,T], \\ x(0) &= x^0, \end{aligned}$$

where $k > 0$ a constant modeling the growth rate of a reinvestment. It is the aim to maximize the functional

$$P\left(\alpha(\cdot), x(\cdot)\right) = \int_0^T \left(1-\alpha(t)\right) x(t) dt,$$

i.e., the total consumption of the output, our consumption at a given time t being $(1-\alpha(t))x(t)$. Thus, the control problem can be written as

$$\begin{array}{lllll} \max & \int_0^T \left(1-\alpha(t)\right) x(t) dt & & & \\ \text{s.t.} & \dot{x}(t) & = k\,\alpha(t)x(t) & \forall\, t \in [0,T], & \\ & x(0) & = x^0, & & \\ & 0 \leq \alpha(t) & \leq 1 & \forall\, t \in [0,T]. & \end{array} \tag{3.70}$$

The constraining ODE problem can be discretized by a finite difference scheme in the same way as (3.14). In contrast to the previous examples, we are not free to choose the objective function of (3.26). It is given by the objective function $P\left(\alpha(\cdot), x(\cdot)\right)$ of the optimal control problem (3.70). We obtain the POP's objective function F by discretizing $P\left(\alpha(\cdot), x(\cdot)\right)$ as

$$F(\alpha, x) = \sum_{i=1}^{N} (1-\alpha_i) x_i.$$

It is easy to show with the **Pontryagin Maximum Principle**, see for example [60], the optimal control law $\alpha^\star(\cdot)$ is given by

$$\alpha^\star(t) = \begin{cases} 1 & \text{if } 0 \leq t \leq t^\star \\ 0 & \text{if } t^\star < t \leq T \end{cases}$$

for an appropriate **switching time** $t^\star$, $0 \leq t^\star \leq T$. In the case $k = 1$ the switching time is given by $t^\star = T - 1$. We apply the SDPR method to (3.70) with objective function F and $\omega = 2$, and can confirm numerically $t^\star = T - 1$ holds for $k = 1$. Our results are reported in Table 3.24 and illustrated in Figure 3.29.

Thus, the solution of the control problem and in particular the optimal control law $\alpha^\star$ are approximated accurately. Moreover, we observe that the switching time fits the predicted value.

T	N_t	ϵ_{sc}	ϵ_{obj}	t_C	$t^\star$
3	200	-3.9e-6	4.4e-9	102	2.00
3	300	-1.3e-4	3.9e-9	354	2.00
4	200	-4.5e-4	1.9e-8	126	3.00
4	300	-2.7e-4	3.9e-8	367	3.00

Table 3.24: Results of the SDPR method for (3.70) with $k = 1, x^0 = 0.25$.

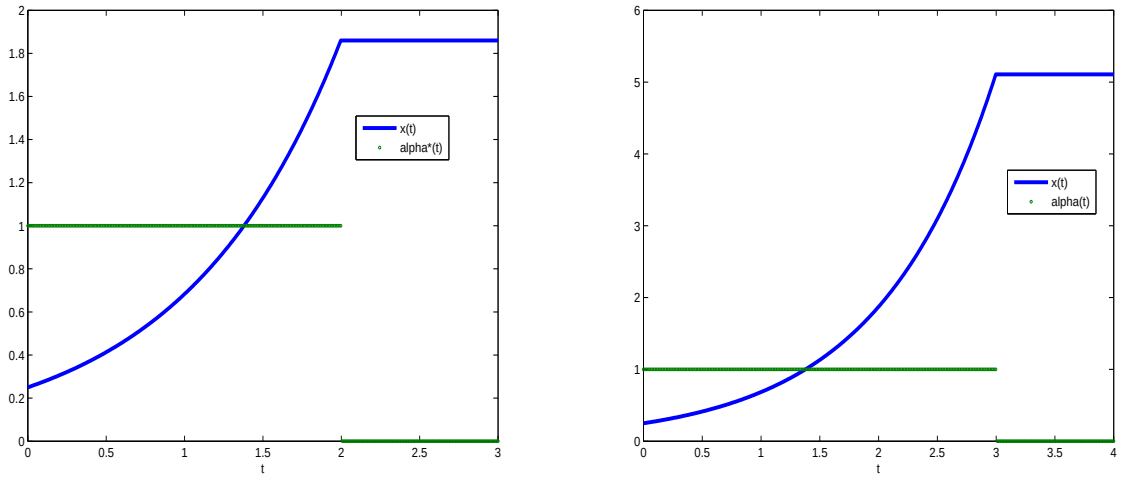


Figure 3.29: SDPR method solutions for (3.70), $x(t)$ (blue) and $\alpha(t)$ (green) for $N_t = 300$, $T = 3$ (left) and $T = 4$ (right), respectively.

Control of reproductive strategies of social insects

As another example, consider a problem arising from reproductive strategies of social insects.

$$\begin{array}{lrll} \max & P\left(w(\cdot), q(\cdot), \alpha(\cdot)\right) & = q(T) & \\ \text{s.t.} & \dot{w}(t) & = -\mu w(t) + b\, s(t)\alpha(t)\, w(t) & \forall\, t \in [0,T], \\ & w(0) & = w^0, & \\ & \dot{q}(t) & = -\nu q(t) + c(1-\alpha(t))s(t)w(t) & \forall\, t \in [0,T], \\ & q(0) & = q^0, & \\ & 0 \le \alpha(t) & \le 1 & \forall\, t \in [0,T], \end{array} \tag{3.71}$$

where $w(t)$ the number of workers at time t, $q(t)$ the number of queens, $\alpha(t)$ the **control variable**, which denotes the fraction of the colony effort devoted to increasing work force, μ the workers death rate, ν the queens death rate, $s(t)$ a known rate at which each worker contributes to the bee-economy, b and c constants. It follows from Pontryagin Maximum Principle [60], the optimal control law α of problem (3.71) is a bang-bang control law for any rate $s(t)$, i.e., $\alpha(t) \in \{0,\, 1\}$ for all $t \in [0, 1]$.
For the SDPR method we choose as objective the function F given by

$$F(w, q, \alpha) = q_{N_t},$$

which is a discretization of the objective function $P(w(\cdot), q(\cdot), \alpha(\cdot))$. Table 3.25 shows the numerical results, which are illustrated in Figure 3.30.

$s(t)$	N_t	ϵ_{obj}	ϵ_{sc}
1	300	2e-7	-2e-6
$\frac{1}{2}(\sin t + 1)$	300	1e-4	-4e-5

Table 3.25: Results of the SDPR method for (3.71) with $T = 3$, $\mu = 0.8$, $b = 1$, $w^0 = 10$, $\nu = 0.3$, $c = 1$, $q^0 = 1$ and $\omega = 2$.

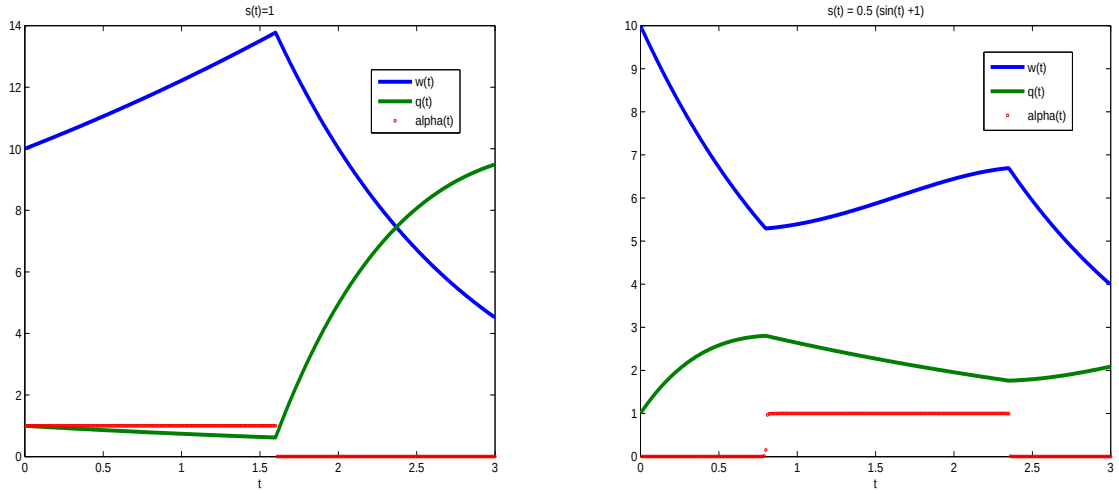


Figure 3.30: SDPR method solutions for (3.71) for $s(t) = 1$ (left) and $s(t) = 0.5(\sin(t) + 1)$ (right).

In the case of $s(t) = 1$, it is sufficient to choose $w(t)$, $q(t) \leq 20$ as upper bounds to get accurate results. For the more difficult problem with $s(t) = 0.5(\sin(t) + 1)$ it is necessary to tighten the upper bounds to $w(t) \leq 10$ and $q(t) \leq 3$, in order to obtain fairly accurate results. In both cases, the bang-bang control law is approximated with high precision.

The double integrator

Consider the optimal control problem given by

$$\begin{array}{lll} \min & T & \\ \text{s.t.} & \dot{x}_1(t) = x_2(t) & \forall t \in [0, T], \\ & \dot{x}_2(t) = u(t) & \forall t \in [0, T], \\ & x_1(0) = x_{1,0}, & \\ & x_1(T) = 0, & \\ & x_2(0) = x_{2,0}, & \\ & x_2(T) = 0, & \\ & x_2(t) \geq -1, & \\ & u(t) \in [-1, 1]. & \end{array} \tag{3.72}$$

Note, we can not apply the SDPR method directly to (3.72), since the length T of the domain is not specified. Furthermore as a system of first order PDE with terminal condition is overspecified, we replace

the constraints $x_1(T) = x_2(T) = 0$ by $\mid x_1(T) \mid + \mid x_2(T) \mid \leq \epsilon$ for a small $\epsilon > 0$. We apply a standard coordinate transformation to (3.72) and obtain the equivalent problem

$$\begin{array}{lll} \min & T & \\ \text{s.t.} & \dot{x}_1(t) = T\, x_2(t) & \forall t \in [0,1], \\ & \dot{x}_2(t) = T\, u(t) & \forall t \in [0,1], \\ & x_1(0) = x_{1,0}, & \\ & x_2(0) = x_{2,0}, & \\ & \mid x_1(1) \mid + \mid x_2(1) \mid \leq \epsilon, & \\ & x_2(t) \geq -1, & \\ & u(t) \in [-1,1]. & \end{array} \tag{3.73}$$

Optimal control problem (3.73) is of a form we can apply the SDPR method to. Lower bounds $\text{lbd}_u = \text{lbd}_x = -1$ and upper bound $\text{ubd}_u = 1$ are given, and we choose $\text{ubd}_x = 10$. Given some starting point $x_0 \in \mathbb{R}^2$ it is the aim to find the minimal time $T^\star$ to steer $x(t)$ into the origin. For this simple problem it is possible to determine the minimal time $T^\star(x_0)$ analytically, c.f. [54]. Thus, for each choice x_0 we can calculate the ratio $\frac{\min(\text{sSDP}_\omega(x_0))}{T^\star(x_0)}$ and evaluate the performance of our approach. We apply the SDPR method with $\omega = 3$ for discretization $N = 50$. We choose the same set of x_0 as in [54], $x_{0,1} \in \{0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$ and $x_{0,2} \in \{-1.0, -0.8, -0.6, -0.4, -0.2, 0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. In Table 3.26 we report the fraction $\frac{\min(\text{sSDP}_3(x_0))}{T^\star(x_0)}$ for the 11×11 different values of x_0. Some entries are larger than 1, which can be explained by the fact that there is a discretization error for the medium scale discretization $N = 50$. Compared to the corresponding table in [54], we achieve better lower bounds for $T^\star$ in most cases. Moreover, the SDPR method approximates optimal control and trajectory in addition to generating lower bounds for the optimal value. See Figure 3.31 for approximations of $u^\star$ and $x^\star$ before and after applying sequential quadratic programming with the SDP solution as initial guess. We observe, that the approximation $(\tilde{u}, \tilde{x})$ provided by the sparse SDP relaxation is already close to the highly accurate approximation (u, x) to the optimal solution of the discretized OCP obtained by additionally applying SQP.

$x_{0,1} \backslash x_{0,2}$	-1	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4	0.6	0.8	1.0
0	0.8783	0.6162	0.5543	0.5665	0.8472	1.0000	0.8420	0.5447	0.8191	0.8858	0.9128
0.2	0.8158	0.4756	0.9060	0.8756	0.8281	0.7869	0.7362	0.7420	0.9068	0.9339	0.9537
0.4	0.7228	0.9440	0.9237	0.9258	0.9023	0.8708	0.8539	0.8495	0.9423	0.9692	0.9811
0.6	1.0139	0.9975	0.9971	0.9886	0.9991	0.9876	0.9382	0.9588	0.9507	0.9875	0.9848
0.8	1.0079	1.0071	1.0090	0.9972	0.9983	1.0035	1.0005	0.9962	0.9772	1.0025	0.9901
1.0	1.0141	1.0124	1.0119	1.0050	1.0009	0.9926	1.0086	1.0016	1.0020	1.0026	0.9961
1.2	1.0162	1.0131	1.0109	1.0064	1.0044	1.0018	1.0086	1.0076	1.0018	1.0043	0.9974
1.4	1.0185	1.0156	1.0135	1.0114	1.0086	1.0067	1.0042	1.0017	1.0065	0.9991	0.9967
1.6	1.0189	1.0195	1.0148	1.0136	1.0114	1.0069	1.0070	1.0062	1.0021	1.0009	0.9997
1.8	1.0196	1.0182	1.0187	1.0150	1.0133	1.0082	1.0085	1.0076	1.0065	1.0027	1.0010
2.0	1.0234	1.0205	1.0177	1.0168	1.0140	1.0109	1.0095	1.0082	1.0045	1.0028	1.0024

Table 3.26: $\min(\text{sSDP}_3(x_0))/T^\star(x_0)$ for different choices of x_0.

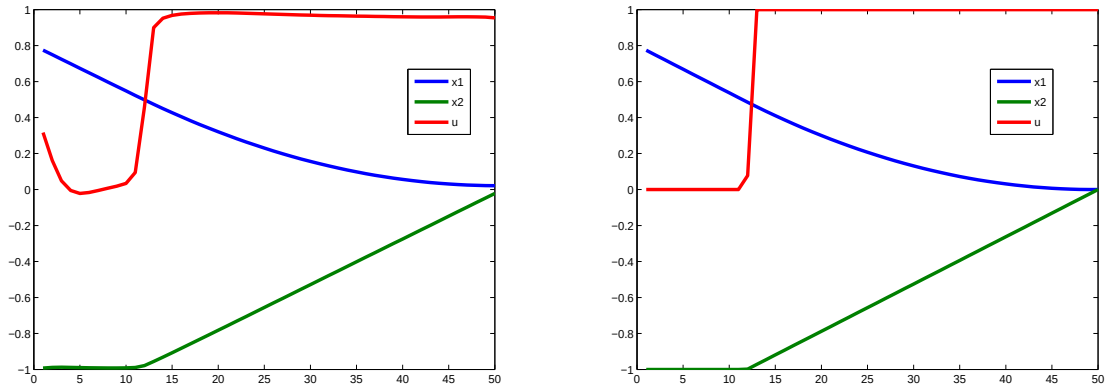


Figure 3.31: Optimal control and trajectories for $x_0 = (0.8, -1)$ for Double Integrator OCP before (left) and after (right) applying SQP.

The Brockett Integrator

Consider the nonlinear optimal control problem given by

$$\begin{aligned}
\min \quad & T \\
\text{s.t.} \quad & \dot{x}_1(t) = u_1(t) && \forall t \in [0,T], \\
& \dot{x}_2(t) = u_2(t) && \forall t \in [0,T], \\
& \dot{x}_3(t) = u_1(t)x_2(t) - u_2(t)x_1(t) && \forall t \in [0,T], \\
& x_1(0) = x_{1,0}, \\
& x_2(0) = x_{2,0}, \\
& x_3(0) = x_{3,0}, \\
& x_1(T) = 0, \\
& x_2(T) = 0, \\
& x_3(T) = 0, \\
& u_1(t)^2 + u_2(t)^2 \leq 1.
\end{aligned} \tag{3.74}$$

Applying the same transformation as for (3.72), we bring (3.74) into a form we can apply the SDPR method to:

$$\begin{aligned}
\min \quad & T \\
\text{s.t.} \quad & \dot{x}_1(t) = T\,u_1(t) && \forall t \in [0,1], \\
& \dot{x}_2(t) = T\,u_2(t) && \forall t \in [0,1], \\
& \dot{x}_3(t) = T\,u_1(t)x_2(t) - T\,u_2(t)x_1(t) && \forall t \in [0,1], \\
& x_1(0) = x_{1,0}, \\
& x_2(0) = x_{2,0}, \\
& x_3(0) = x_{3,0}, \\
& \mid x_1(1) \mid + \mid x_2(1) \mid + \mid x_3(1) \mid \leq \epsilon, \\
& u_1(t)^2 + u_2(t)^2 \leq 1,
\end{aligned} \tag{3.75}$$

for some small $\epsilon > 0$. As in example (3.72), it is the aim to find the minimal time $T^\star(x_0)$ to steer some point $x(t)$ with $x(0) = x_0 \in \mathbb{R}^3$ into the origin. For the lower and upper bounds of u and x, we choose

$$\text{lbd}_u = -1, \;\; \text{ubd}_u = 1, \;\; \text{lbd}_x = -5, \;\; \text{ubd}_x = 5.$$

For this optimal control problem it is possible to calculate the minimal time $T^\star$ exactly [54]. Thus, we can compare the performance of the SDPR method to the approach in [54]. We apply the SDPR method

$x_{0,2}\backslash x_{0,3}$	0	1	2	3
0	0.0000	1.0081	2.0276	3.0145
1	1.6049	1.2827	2.0337	3.0125
2	2.7816	2.0269	2.1959	3.0170
3	3.9705	2.8498	2.6177	3.1906

Table 3.27: $\min(\text{sSDP}_3(x_0))$ for $x_{0,1} = 0$ and $(x_{0,2}, x_{0,3}) \in \{0,1,2,3\}^2$.

$x_{0,2}\backslash x_{0,3}$	0	1	2	3
0	0.0000	1.0000	2.0000	3.0000
1	2.5066	1.7841	2.1735	3.0547
2	3.5449	2.6831	2.5819	3.2088
3	4.3416	3.4328	3.0708	3.4392

Table 3.28: $T^\star(x_0)$ for $x_{0,1} = 0$ and $(x_{0,2}, x_{0,3}) \in \{0,1,2,3\}^2$.

with $\omega = 3$ to (3.75). The numerical results for $N = 50$ and $x_{0,1} = 0$, $(x_{0,2}, x_{0,3}) \in \{0,1,2,3\}^2$ are reported in Table 3.27, the results for $N = 30$ and $x_{0,1} = 1$, $(x_{0,2}, x_{0,3}) \in \{1,2,3\}^2$ in Table 3.29. Again, $\min(\text{sSDP}_w(x_0))$ is larger than $T^\star(x_0)$ for some x_0, which is explained by the discretization error due to the medium scale choice $N \in \{30, 50\}$. This gap closes for $N \to \infty$.

In particular for choices x_0 to be found in the lower left corner of Table 3.27 and 3.29, we obtain better lower bounds than [54]. Again, unlike the method in [54] we also obtain an accurate approximation of optimal control and trajectory, as pictured in Figure 3.32.

$x_{0,2}\backslash x_{0,3}$	1	2	3
1	1.8862	2.4412	3.3145
2	2.6077	2.7737	3.4516
3	3.2969	3.2033	3.6618

Table 3.29: $\min(\text{sSDP}_3(x_0))$ for $x_{0,1} = 1$ and $(x_{0,2}, x_{0,3}) \in \{1,2,3\}^2$.

$x_{0,2} \backslash x_{0,3}$	1	2	3
1	1.8257	2.3636	3.2091
2	2.5231	2.6856	3.3426
3	3.1895	3.1008	3.5456

Table 3.30: $\min(T^\star(x_0))$ for $x_{0,1} = 1$ and $(x_{0,2}, x_{0,3}) \in \{1, 2, 3\}^2$.

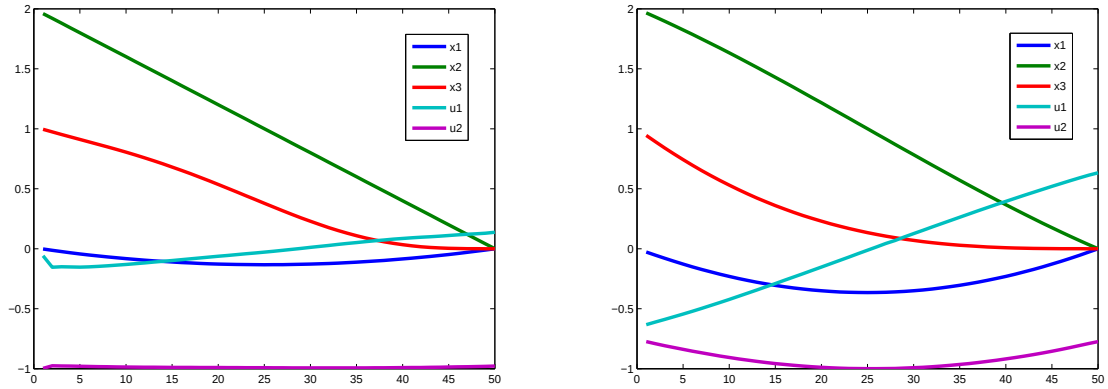


Figure 3.32: Optimal control and trajectories for $x_0 = (0, 2, 1)$ for Brockett Integrator OCP before (left) and after (right) applying SQP.

Chapter 4

Concluding Remarks and Future Research

4.1 Conclusion

Hierarchies of SDP relaxations are a powerful tool to solve general, severely nonconvex POPs. However, to solve large scale POPs remains a very challenging task due to the limited capacity of contemporary SDP solvers. In this thesis we discussed two major approaches to attempt large scale POPs by reducing the size of the SDP relaxations.

In the first one presented in 2.2, our focus has been on developing a theoretical framework consisting of the d- and r-space conversion methods to exploit structured sparsity, characterized by a chordal graph structure, via the positive semidefinite matrix completion for an optimization problem involving linear and nonlinear matrix inequalities. The two d-space conversion methods are provided for a matrix variable X in objective and/or constraint functions of the problem, which is required to be positive semidefinite. The methods decompose X into multiple smaller matrix variables. The two r-space conversion methods are aimed at a matrix inequality in the constraint of the problem. In these methods, the matrix inequality is converted into multiple smaller matrix inequalities. As mentioned in Remarks 2.3, 2.5 and 2.7, the d-space conversion method using clique trees and the r-space conversion method using clique trees have plenty of flexibilities in implementation. This should be explored further for increasing the computational efficiency. In 2.2.7 we constructed linear SDP relaxations for general quadratic SDP that exploit d- and r-space sparsity. When applying these relaxations to quadratic SDP arising from different applications, we observed that the computational performance is greatly improved compared to the classical SDP relaxations, which do not apply d- and r-space conversion methods. Of particular interest for the numerical analysis of differential equation is the linear SDP relaxation exploiting d-space sparsity. It reduces the size of SDP relaxations for POPs derived from certain differential equations a lot. It will be interesting topic to study the efficiency of d- and r-space conversion methods for further classes of nonlinear SDPs.

In 2.3, we proposed four different heuristics to transform a general POP into a QOP. The advantage of this transformation is that the sparse SDP relaxation of order one can be applied to the QOP. The sparse SDP relaxation of order one is of vastly smaller size than the sparse SDP relaxation of minimal order $\omega_{\max}$ for the original POP. By solving the sparse SDP relaxation of the QOP, approximates to the global minimizer of a large scale POP of higher degree can be derived. The reduction of the SDP relaxation and the gain in numerical tractability come at the cost of deteriorating feasibility and optimality errors of the approximate solution obtained by solving the SDP relaxation. In general the SDP relaxation of order one for the QOP is weaker than the SDP relaxation of order $\omega_{\max}$ for the original POP. We discussed how to overcome this difficulty by imposing tighter lower and upper bounds for the components of the n-dimensional variable of a POP, by adding linear or quadratic Branch-and-Cut bounds, and by applying local convergent optimization methods such as sequential quadratic programming to the POP starting from the solution provided by the SDP relaxation for the QOP. The proposed heuristics have been demonstrated with success

on various medium and large-scale POPs. We have seen that imposing additional Branch-and-Cut bounds was necessary to yield accurate approximations to the global optimizer for some problems. However, for most problems it was crucial to choose the lower and upper bounds for the variable x sufficiently tight and to apply SQP to obtain a highly accurate approximation of the global optimizer. The total processing time could be reduced by up to three magnitudes for the problems we tested. For these reasons we think the proposed technique is promising to find first approximate solutions for POPs, whose size is too large to be solved by the more precise, original SDP relaxations due to Waki et al.

Our most important application for both approaches to reduce the size of SDP relaxations is the numerical analysis of nonlinear differential equations. We were able to transform nonlinear PDE problems with polynomial structure into POPs. The description is based on the discretization of a PDE problem, the approximation of its partial derivatives by finite differences and the choice of an appropriate objective function. Due to the finite difference discretization, the POPs derived from PDEs satisfy both, correlative and domain-space sparsity patterns under fairly general assumptions. Therefore, we can apply dual standard form SDP relaxations exploiting correlative sparsity and primal standard form SDP relaxations exploiting domain-space sparsity efficiently. For many PDE problems the solution of the SDP relaxation is an appropriate initial guess for locally fast convergent methods like Newton's method and SQP. However, we have seen it is often necessary to impose tighter or additional bounds to the POPs to derive highly accurate solutions. Moreover, we demonstrated how to choose an objective function and bounds for the unknown function, in order to detect particular solutions of a discretized PDE problem. In other words, one of the features of using the SDPR method instead of several existing methods is that a function space to find solutions may be translated into natural constraints for a sparse POP. In the case we have partial information about a particular solution we want to find, this information can be exploited by the SDPR method to provide an appropriate initial guess for a local method. The reduction techniques from Chapter 2 are highly efficient to solve POPs derived from PDEs for higher resolution grids and to obtain accurate discrete approximations for solutions of the continuous PDE problem. Another technique to extend solutions to finer and finer grid is the grid-refining method, which is efficient even when starting from solutions on very coarse grids.

We have shown that the SDPR method is very promising for nonlinear differential equations with several solutions. One feature of the SDPR method is the ability to detect a particular solution. Another challenging problem is to enumerate all solutions of a discretized PDE problem and ultimately to enumerate accurate approximations for all solutions of the underlying continuous PDE problem. We proposed an algorithm based on the SDPR method to approximately enumerate all real solutions of a zero dimensional radical polynomial system with respect to a cost function. If the order of the SDP relaxations tends to infinity, we can guarantee the convergence of the algorithm's output to the smallest kinetic energy solutions of the polynomial system. The algorithm can be applied successfully to enumerate the solutions of the discrete cavity flow problem with the kinetic energy of the flow as cost function. A variant of the enumeration algorithm has been applied to detect all solutions of an interesting reaction-diffusion equation. Since both, the enumeration algorithm and its variant, are based on the SDPR method that exploits sparsity, it is possible to attempt POPs of much larger scale than by the approaches in [35] and [55].

To conclude, the SDPR method constitutes a general purpose method for solving problems involving differential equations polynomial in the unknown functions. We demonstrated the potential of the SDPR method on differential equations arising from a range of areas: Elliptic, parabolic and hyperbolic PDE, reaction-diffusion equations, fluid dynamics, nonlinear optimal control, differential algebraic equations and first order PDEs. This list of differential equations we may analyze by the SDPR method is by no means complete. But it illustrates that the SDPR method provides a powerful tool to get new insights in the numerical analysis of differential equations.

4.2 Outlook on future research directions

Efficient software for large scale POPs and their SDP relaxations remains a challenging field with many open problems. The research presented in this thesis motivates to look for answers to a number of questions:

We discussed four heuristics to transform an arbitrary POP into an equivalent QOP. Moreover, we

encountered that correlative sparsity and domain-space sparsity of a QOP can differ significantly. Therefore, the size of the resulting dual and primal form SDP relaxations may be of vastly different size. The question remains whether (a) there is a way to transform a POP into a QOP that enhances these types of sparsity, and (b) whether we may find a more general concept of sparsity for a QOP that combines these two types of sparsity. We have also seen that the approximation accuracy of the SDP relaxation for the QOP is weaker than for the original POP. It remains a future problem to strengthen the sparse SDP relaxation of order one for a QOP further. In that respect, it is desirable to find a systematic approach to tighten lower and upper bounds successively, without shrinking the optimal set of the POP. Furthermore, the additional quadratic constraints derived under the transformation algorithm allow to express some moments as linear combination of other moments. As proposed by Henrion and Lasserre in [35] and Laurent in [56] these linear combinations can be substituted in the moment and localizing matrices of the SDP relaxation to reduce the size of the moment vector y. Exploiting this technique will shrink the size of the sparse SDP relaxations for QOPs further and may enable us to solve POPs of even larger scale.

Compared to the methods [35, 55] for finding all real solutions of a zero-dimensional radical polynomial system, our enumeration algorithm can be applied to problems of much larger scale. However, the numerical stability depends heavily on the choice of the parameters in the algorithm. The variant of the enumeration algorithm for the Swift-Hohenberg equation constitutes a promising first step to improve the numerical stability, since the additional linear constraints remain unchanged under the SDP relaxation. The idea of this variant may be exploited more systemically in future.

Although we are able to solve some PDE problems with minimal relaxation order $\omega = \omega_{\max}$, in many cases it is a priori not possible to predict the relaxation order ω which is necessary to attain an accurate solution. As the size of the sparse SDP relaxation increases polynomially in ω, the tractability of the SDP is limited by the capacity of current SDP solvers. It is a further challenging question whether we can characterize a class of differential equation problems that is guaranteed to be approximated accurately for a certain fixed relaxation order. At the moment there are only very few results for error bounds of SDP relaxations for general, nonconvex POPs [72].

Not every solution of a discretized differential equation is a discrete approximation for an actual solution of this differential equation, as we encountered in the analysis of the steady cavity flow problem. It is therefore interesting to close the gap between the discrete and the continuous world. An approach based on the SDPR method for narrowing this gap takes advantage of *maximum entropy estimation* [10, 53]. In this approach the solution of the SDPR method is used to compute discrete approximations to moments of a measure corresponding to the differential equation, and when applying maximum entropy estimation to these discretized moments we obtain a smooth approximation for a solution of the differential equation. This is the topic of some ongoing joint work with Jean Lasserre and Didier Henrion.

Finally, we applied the SDPR method to a wide variety of problems involving differential equations. However, the classes of problems we may attempt by this approach are by no means exhausted. It will be an interesting topic of future research to apply the SDPR methods to challenging nonlinear differential equations satisfying a polynomial structure. Also, nonlinear optimal control seems an challenging area to apply this methodology to, as the numerical experiments for the simple optimal control problems presented in this thesis suggest.

Bibliography

[1] J. Agler, J. W. Helton, S. McCullough, L. Rodman, *Positive semidefinite matrices with a given sparsity pattern*, Linear Algebra Appl. (1988), Vol. 107, pp. 101-149.

[2] E.L. Allgower, D.J. Bates, A.J. Sommese, C.W. Wampler, *Solution of polynomial systems derived from differential equations*, Computing, 76 (2006), No. 1, pp. 1-10.

[3] A. Arakawa, *Computational design for long-term numerical integration of the equation of fluid motion: two dimensional incompressible flow, part I.*, Journal of Computational Physics 135 (1997), pp. 103-114.

[4] J.R.S. Blair, B. Peyton, *An introduction to chordal graphs and clique trees, Graph Theory and Sparse Matrix Computation*, Springer Verlag (1993), pp. 1-29.

[5] D. Bertsimas, C. Caramanis, *Bounds on linear PDEs via semidefinite optimization*, Math. Programming, Series A 108 (2006), pp. 135-158.

[6] P. Biswas, Y. Ye, *A distributed method for solving semidefinite programs arising from Ad Hoc Wireless Sensor Network Localization*, Multiscale Optimization Methods and Applications, 69-84, Springer-Verlag

[7] J. Bochnak, M. Coste, M.-F. Roy, *Real Algebraic Geometry*, Springer-Verlag (1998).

[8] P.T. Boggs, J.W. Tolle, *Sequential Quadratic Programming*, Acta Numerica 4 (1995), pp. 1-50.

[9] B. Borchers, *SDPLIB 1.2, A library of semidefinite programming test problems*, Optim. Methods Softw. (1999), 11-12, pp. 683-689.

[10] J. Borwein, A.S. Lewis, *On the convergence of moment problems*, Trans. Am. Math. Soc., 325 (1991), pp. 249-271.

[11] D. Braess, *Finite Elements, Theory, fast solvers, and applications in solid mechanics*, Cambridge University Press (2001).

[12] O.R. Burggraf, *Analytical and numerical studies of the structure of steady separated flows*, J. Fluid Mech 24 (1966), pp. 113-151.

[13] R. Courant, K. Friedrichs, H. Lewy, *Über die partiellen Differenzengleichungen der mathematischen Physik*, Math. Ann. 100 (1928), No. 1, pp. 32-74.

[14] M. Cheng, K.C. Hung, *Vortex structure of steady flow in a rectangular cavity*, Computers & Fluids, Volume 35, Issue 10 (2006), pp. 1046-1062.

[15] R. Courant, D. Hilbert, *Methoden der Mathematischen Physik*, Vol 1 (1931), Chapter 4, *The method of variation.*

[16] R. Courant, *Variational methods for the solution of problems of equilibrium and vibrations*, Bull. Amer. Soc. 49 (1943), pp. 1-23.

[17] G. Dahlquist, *Convergence and stability in the numerical integration of ordinary differential equations*, Math. Scand. 4 (1956), pp. 33-53.

[18] R. Fletcher, *Practical Methods of Optimization. Vol. 1 Unconstrained Optimization*, John Wiley, Chichester (1980).

[19] I. Fried, *Numerical Solutions of Differential Equations*, Academic Press (1979).

[20] K. Fujisawa, S. Kim, M. Kojima, Y. Okamoto, M. Yamashita, *User's Manual for SparseCoLO: Conversion Methods for SPARSE COnic-form Linear Optimization Problems*, Research reports on Mathematical and Computing Sciences B-453, Tokyo Institute of Technology.

[21] M. Fukuda, M. Kojima, K. Murota, K. Nakata *Exploiting sparsity in semidefinite programming via matrix completion I: General framework*, SIAM J. Optim., 11 (2000), pp. 647-674.

[22] M. Fukuda, M. Kojima, *Branch-and-Cut Algorithms for the Bilinear Matrix Inequality Eigenvalue Problem, Computational Optimization and Applications*, 19 (2001), pp. 79-105.

[23] B.G. Galerkin, *Series solution of some problems in elastic equilibrium of rods and plates*, Vestn. Inzh. Tech. 19 (1915), pp. 897-908.

[24] A. George, J.W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall (1981).

[25] P.E. Gill, W. Murray, M.H. Wright, *Practical Optimization*, Academic Press, London, New York (1981).

[26] M. Goemans, D.P. Williamson, *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming*, Journal of the ACM, 42 (1995), No. 6, pp. 1115-1145.

[27] D. Gottlieb, S. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM, Philadelphia (1977).

[28] R. Grone, C. R. Johnson, E. M. Sá, H. Wolkowitz, *Positive definite completions of a partial hermitian matrices*, Linear Algebra Appl. 58 (1984), pp. 109-124.

[29] J.L. Guermond, *A finite element technique for solving first-order PDEs in* L^P, SIAM Journal Numerical Analysis 42 (2004), No. 2, pp. 714-737.

[30] J.L. Guermond, B. Popov, *Linear advection with ill-posed boundary conditions via* L^1*-minimization*, International Journal of Numerical Analysis and Modeling 4 (2007), No. 1, pp. 39-47.

[31] T. Gunji, S. Kim, M. Kojima, A. Takeda, K. Fujisawa, and T. Mizutani, *PHoM - a Polyhedral Homotopy Continuation Method for Polynomial Systems*, Research Reports on Mathematical and Computing Sciences, Dept. of Math. and Comp. Sciences, Tokyo Inst. of Tech., B-386 (2003)

[32] K. Gustafson, K. Halasi, *Cavity flow dynamics at higher Reynolds number and higher aspect ratio*, Journal of Computational Physics 70 (1987), pp. 271-283.

[33] W. Hao, J.D. Hauenstein, B. Hu, Y. Liu, A.J. Sommese, Y.-T. Zhang, *Multiple stable steady states of a reaction-diffusion model on zebrafish dorsal-ventral patterning*, Discrete and Continuous Dynamical Systems, Series S, To appear.

[34] J.D. Hauenstein, A.J. Sommese, C.W. Wampler, *Regeneration Homotopies for Solving Systems of Polynomials*, Mathematics of Computation, To appear.

[35] D. Henrion, J.B. Lasserre, *Detecting global optimality and extracting solutions in GloptiPoly*, Chapter in D. Henrion, A. Garulli, editors, *Positive polynomials in control. Lecture Notes in Control and Information science*, Springer Verlag (2005), Berlin.

[36] D. Henrion, J. B. Lasserre, *Convergent relaxations of polynomial matrix inequalities and static output feedback*, IEEE Trans. Automatic Control (2006), 51, pp. 192-202.

[37] C. W. J. Hol, C. W. Scherer, *Sum of squares relaxations for polynomial semidefinite programming*, Proc. Symp. on Mathematical Theory of Networks and Systems (MTNS), Leuven, Belgium, 2004.

[38] R. Horst, P.M. Pardalos, N.V. Thoai, *Introduction to Global Optimization*, Kluwer Academic Publishers (2000).

[39] B. Huber, B. Sturmfels, *A polyhedral method for solving sparse polynomial systems*, Math. of Comp. 64 (1995), pp. 1541-1555.

[40] M. Kawaguti, *Numerical solution of the Navier-Stokes equations for the flow in a two dimensional cavity*, J. Phys. Soc. Jpn. 16 (1961), pp. 2307-2315.

[41] S. Kim, M. Kojima, *Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations*, Computational Optimization and Applications, 26 (2003), pp. 143-154.

[42] S. Kim, M. Kojima, M. Mevissen, M. Yamashita, *Exploiting Sparsity in Linear and Nonlinear Matrix Inequalities via Positive Semidefinite Matrix Completion*, Mathematical Programming, To Appear.

[43] S. Kim, M. Kojima, H. Waki, *Exploiting Sparsity in SDP Relaxation for Sensor Network Localization*, SIAM Journal of Optimization 20 (2009), No. 1, pp. 192-215.

[44] S. Kim, M. Kojima H. Waki, M. Yamashita, *SFSDP: a Sparse Version of Full Semidefinite Programming Relaxation for Sensor Network Localization Problems*, Research Report B-457, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology (2009).

[45] K. Kobayashi, S. Kim, M. Kojima, *Correlative sparsity in primal-dual interior-point methods for LP, SDP and SOCP*, Appl. Math. Optim. (2008), 58, pp. 69-88.

[46] M. Kojima, *Sums of Squares Relaxations of Polynomial Semidefinite Programs*, Research Report B-397, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology (2003).

[47] M. Kojima, S. Kim, H. Waki, *Sparsity in sums of squares of polynomials*, Mathematical Programming, 103 (2005), pp. 45-62.

[48] M. Kojima, M. Muramatsu, *An Extension of Sums of Squares Relaxations to Polynomial Optimization Problems over Symmetric Cones*, Math. Programming (2007), 110, pp. 315-336.

[49] M. Kojima, M. Muramatsu, *A note on sparse SOS and SDP relaxations for polynomial optimization problems over symmetric cones*, Comput. Optim. Appl. (2009), 42, pp. 31-41.

[50] W. Kutta, *Beitrag zur näherungsweisen Integration totaler Differentialgleichungen*, Zeitschrift Math. Physik 46 (1901), pp. 435-453.

[51] J.B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM Journal on Optimization, 11 (2001), pp. 796-817.

[52] J.B. Lasserre, *Convergent SDP-Relaxations in Polynomial Optimization with Sparsity*, SIAM Journal on Optimization, 17 (2006), No. 3, pp. 822-843.

[53] J.B. Lasserre, *Semidefinite programming for gradient and Hessian computation in maximum entropy estimation*, Proc. IEEE Conf. Dec Control, 2007.

[54] J.B. Lasserre, D. Henrion, C. Prieur, E. Trelat, *Nonlinear optimal control via occupation measures and LMI-relaxations*, SIAM Journal on Control and Optimization, 47 (2008), pp. 1649-1666.

[55] J.B. Lasserre, M. Laurent, P. Rostalski, *Semidefinite characterization and computation of real radical ideals*, Foundations of Computational Mathematics, Vol. 8 (2008), No. 5, pp. 607-647.

[56] M. Laurent, *Sums of squares, moment matrices and optimization over polynomials*, Emerging Applications of Algebraic Geometry, Vol. 149 of IMA Volumes in Mathematics and its Applications (2009), M. Putinar and S. Sullivant (eds.), Springer, pp. 157-270.

[57] P.D. Lax, R.D. Richtmyer, *Survey of the stability of linear finite difference equations*, Comm. Pure Appl. Math. 9 (1956), pp. 267-293.

[58] R.J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press (2002).

[59] G.R. Liu, S.S. Quek, *The Finite Element Method, A practical course*, Elsevier (2003).

[60] J. Macki, A. Strauss, *Introduction to Optimal Control Theory*, Springer-Verlag (1982), pp. 108.

[61] M. Mevissen, M. Kojima, J. Nie and N. Takayama, *Solving partial differential equations via sparse SDP relaxations*, Pacific Journal of Optimization, 4 (2008), No. 2, pp. 213-241.

[62] M. Mevissen, M. Kojima, *SDP Relaxations for Quadratic Optimization Problems Derived from Polynomial Optimization Problems*, Asia-Pacific Journal for Operations Research 27 (2010), No. 1, pp. 1-24.

[63] M. Mevissen, K. Yokoyama and N. Takayama, *Solutions of Polynomial Systems Derived from the Cavity Flow Problem*, Proceedings of the 2009 International Symposium on Symbolic Computation, 2009, pp. 255 - 262.

[64] M. Mimura, *Asymptotic Behaviors of a Parabolic System Related to a Planktonic Prey and Predator Model*, SIAM Journal on Applied Mathematics, 37 (1979), no. 3, pp. 499-512.

[65] A.R. Mitchell, D.F. Griffiths, *The Finite Difference Method in Partial Differential Equations*, John Wiley and Sons (1980).

[66] J.J. More, B.S. Garbow and K.E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7 (1981), pp. 17-41.

[67] K.G. Murty, S.N. Kabadi, *Some NP-complete problems in quadratic and nonlinear programming*, Mathematical Programming, 39 (1987), pp. 117-129.

[68] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, K. Murota *Exploiting sparsity in semidefinite programming via matrix completion II: Implementation and numerical results*, Math. Programming, 95 (2003), pp. 303-327.

[69] Ju. E. Nesterov, A. S. Nemirovski, *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, SIAM, Philadelphia, PA, 1994.

[70] Y. Nesterov, *Squared functional systems and optimization problems*, in J.B.G. Frenk, C. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization*, pp. 405-440. Kluwer Academic Publishers (2000).

[71] J. Nie, *Sum of squares method for sensor network localization*, Computational Optimization and Applications 43 (2009), No. 2, pp. 151-179.

[72] J. Nie, *An Approximation Bound Analysis for Lasserre's Relaxation in Multivariate Polynomial Optimization*, preprint (2009).

[73] Y. Nishiura, D. Ueyama, *Spatio-temporal chaos for the Gray-Scott model*, Physica D, 150 (2001), pp. 137 - 162.

[74] Y. Nishiura, T. Teramoto, K. Ueda, *Dynamic transitions through scattors in dissipative systems*, Chaos, 13 (2003), No. 3, pp. 962 - 972.

[75] Y. Nishiura, T. Teramoto, K. Ueda, *Scattering of traveling spots in dissipative systems*, Chaos, 15 (2005), 047509.

[76] Y. Nishiura, T. Teramoto, X. Yuan, K. Ueda, *Dynamics of traveling pulses in heterogeneous media*, Chaos, 17 (2007), 037104.

[77] J. Nocedal, S.J. Wright, *Numerical Optimization*, Series in Operations Research, Springer, New York 2006.

[78] M. Noro, K. Yokoyama, *A modular method to compute the rational univariate representation of zero-dimensional ideals*, Journal of Symbolic Computation 28 (1999), pp. 243–263.

[79] P.A. Parrilo, *Semidefinite programming relaxations for semialgebraic problems*, Math. Programming, 96 (2003), pp. 293 - 320.

[80] L.A. Peletier, V. Rottschäfer, *Pattern selection of solutions of the Swift-Hohenberg equation*, Physica D, 194 (2004), pp. 95 - 126.

[81] M. Putinar, *Positive Polynomials on Compact Semi-algebraic Sets*, Indiana Univ. Math. Journal 42 (1993), No. 3, pp. 969-984

[82] J. Rauch, J. Smoller, *Qualitative theory of the FitzHugh-Nagumo equations*, Advances in Mathematics, 27 (1978), pp. 12-44.

[83] L. Rayleigh, *On the theory of resonance*, Trans. Roy. Soc. A 161 (1870), pp. 77 - 118.

[84] W. Ritz, *Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik*, Journal für die reine und angewandte Mathematik 135 (1908), pp. 1-61.

[85] F. Rouillier, *Solving zero-dimensional systems through the rational univariate representation*, Applicable Algebra in Engineering, Communication and Computing 9 (1999), pp. 433–461.

[86] C. Runge, *Über die numerische Auflösung von Differentialgleichungen*, Math. Ann. 46 (1895), pp. 167 -178.

[87] K. Schmüdgen, *The K-moment problem for compact semi-algebraic sets*, Math. Ann. 289 (1991), pp. 203-206.

[88] M. Schweighofer, *Optimization of polynomials on compact semialgebraic sets*, SIAM J. Optimization 15 (2005), pp. 805-825.

[89] H.D. Sherali, C.H. Tuncbilek, *A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique*, Journal of Global Optimization, 2 (1992), pp. 101-112.

[90] H.D. Sherali, C.H. Tuncbilek, *New reformulation-linearization technique based relaxations for univariate and multivariate polynomial programming problems*, Operations Research Letters, 21 (1997), 1, pp. 1-10.

[91] N.Z. Shor, *Class of global minimum bounds of polynomial functions*, Cybernetics, 23 (1987), 6, pp. 731-734.

[92] N.Z. Shor, *Nondifferentiable Optimization and Polynomial Problems*, Kluwer (1998).

[93] J. Smoller, *Shock Waves and Reaction-Diffusion Equations*, Springer-Verlag (1983), pp. 106.

[94] J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, 3rd edition, Springer-Verlag, New York (2002).

[95] J.F. Sturm, *SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11 and 12 (1999), pp. 625-653.

[96] J.C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth and Brooks (1989).

[97] M. Tabata, *A finite difference approach to the number of peaks of solutions for semilinear parabolic problems*, J. Math. Soc. Japan, 32 (1980), pp. 171-192.

[98] T. Takami, T. Kawamura, *Solving Paritial Differential Equations with Difference schemes*, Tokyo University Press (1994).

[99] M.J. Turner, R.M. Clough, H.C. Martin, L.J. Topp, *Stiffness and deflection analysis of complex structures*, J. Aeron. Sci. 23 (1956), pp. 805-823, pp. 854.

[100] T. Teramoto, *Personal communication.*

[101] O. Von Stryk, R. Bulirsch, *Direct and indirect methods for trajectory optimization*, Ann. Oper. Res. 37 (1992), pp. 357-373.

[102] H. Waki, S. Kim, M. Kojima, M. Muramatsu, *Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity*, SIAM Journal of Optimization 17 (2006) 218-242.

[103] H. Waki, S. Kim, M. Kojima, M. Muramatsu, *SparsePOP: a Sparse Semidefinite Programming Relaxation of Polynomial Optimization Problems*, Research Reports on Mathematical and Computing Sciences, Dept. of Math. and Comp. Sciences, Tokyo Inst. of Tech., B-414 (2005).

[104] M. Yamashita, K. Fujisawa, M. Kojima, *Implementation and evaluation of SDPA 6.0 (SemiDefinite Programming Algorithm 6.0)*, Optimization Methods and Software 18 (2003), pp. 491-505.

[105] Yokota, http://next1.cc.it-hiroshima.ac.jp/MULTIMEDIA/numeanal2/node24.html.

[106] O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, *The Finite Element Method, Its Basis and Fundamentals*, Elsevier (2005).