Solving Linear Partial Differential Equations via Semidefinite Optimization

by

Constantine Caramanis

A.B., Mathematics (1999)
Harvard University

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

Author . . . .
Department of Electrical Engineering and Computer Science
May 25, 2001

Certified by . . . .
Dimitris Bertsimas
Boeing Professor of Operations Research
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Solving Linear Partial Differential Equations via Semidefinite Optimization

by

Constantine Caramanis

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2001, in partial fulfillment of the
requirements for the degree of
Masters of Science in Computer Science and Engineering

Abstract

Using recent progress on moment problems, and their connections with semidefinite optimization, we present in this thesis a new methodology based on semidefinite optimization, to obtain a hierarchy of upper and lower bounds on both linear and nonlinear functionals defined on solutions of linear partial differential equations. We apply the proposed methods to examples of PDEs in one and two dimensions with very encouraging results. We also provide computational evidence that the semidefinite constraints are critically important in improving the quality of the bounds, that is without them the bounds are weak.

Thesis Supervisor: Dimitris Bertsimas
Title: Boeing Professor of Operations Research

Contents

List of Tables

Chapter 1

Introduction

In many real-world applications of phenomena that are described by partial differential equations (PDEs) we are primarily interested in a functional of the solution of the PDE, as opposed to the solution itself. For example, we might be primarily interested in the average temperature rather than the entire distribution of temperature in a mechanical device; or we might be interested in the lift and drag of an aircraft wing, which is computed by surface integrals over the wing; or finally we might be interested in the average inventory and its variability in a stochastic network.

Given that analytical solutions of PDEs are very scarce, there is a large body of literature on numerical methods for solving PDEs. Such methods typically involve some discretization of the domain of the solution, and thus obtain an approximate solution by solving the resulting equations, and matching boundary values and initial conditions. Such approaches scale exponentially with the dimension, i.e., if we use $O(1/\epsilon)$ points in each dimension, the size of systems we need to solve is of the order of $(1/\epsilon)^d$ for d-dimensional PDEs and results in accuracy ϵ.

Natural questions arise:

(i) *Can we obtain upper and lower bounds within ϵ of each other on functionals of the solution to a PDE in time that grows like $\left(\log \frac{1}{\epsilon}\right)^d$, thus decreasing the curse of dimensionality?*

(ii) *Can such methods be practical?*

Contributions

In this thesis, we make some progress in answering these questions for a specific class of PDEs. We consider solving for linear functionals, as well as the supremum and infimum functionals, of linear PDEs with coefficients that are polynomials of the variables. We make a connection with moment problems and use semidefinite optimization to achieve these objectives. To the best of our knowledge, this is the first connection of solving PDEs and semidefinite optimization. The proposed method finds upper and lower bounds within ϵ of each other on functionals of the solution to a PDE in this class by solving a semidefinite optimization problem. The numerical results we obtain indicate fast convergence. Moreover, we provide a result that indicates the power of a "moment–driven convergence," and suggests that under certain regularity conditions, this can be exponential. While the practicality and numerical stability of the proposed method depends on the numerical stability of semidefinite optimization codes, which are currently under intensive research, we hope that progress in semidefinite optimization codes will lead to improved performance for obtaining bounds on PDEs using the methods of the present thesis.

Moment Problems and Semidefinite Optimization

Problems involving moments of random variables arise naturally in many areas of mathematics, economics, and operations research. Recently, semidefinite optimization methods have been applied to several problems arising in probability theory, finance and stochastic optimization. Bertsimas [3] applies semidefinite optimization methods to find bounds for stochastic optimization problems arising in queueing networks. Bertsimas and Popescu [4] apply semidefinite optimization methods to find best possible bounds on the probability that a multidimensional random variable belongs in a set given a collection of its moments. In [5], they use these methods to find best possible bounds for pricing financial derivatives without assuming particular price dynamics. For a survey of this line of work, including several historical remarks on the origin of moment problems in the 20th century, see Bertsimas, Popescu and

Sethuraman [6].

Semidefinite optimization is currently in the center of much research activity in the area of mathematical programming both from the point of view of new application areas (see for example the survey paper of Vandenberghe and Boyd [25]) as well as algorithmic development.

Literature on Bounds for PDEs

Current state of the art methods for solving partial differential equations, whether they seek to solve for the actual solution of the differential equation, or just some functional of the solution, all depend on the basic idea of discretization. While there are quite a wide array of methods, each with its own focus and specialization, the majority (that we are familiar with) rest upon the idea of discretization of the domain of the equation, and then a subsequent variational formulation of the problem. Excellent references can be found in, e.g. [18], [23], [7], but we give the basic idea here as well. We deviate somewhat from the standard notation as we seek only to convey the fundamental ideas. The idea then is to find a nested sequence of finite dimensional spaces that lie inside the function space in which our solution lies. Then by solving linear systems, we essentially search in each of these finite dimensional spaces for a solution that "best" approximates the solution function, for some appropriate notion of "best."

The finite dimensional spaces, called finite element spaces, typically come from choosing a basis of functions, each supported over an element of the discretization of the domain of the PDE (called a triangulation). There are other methods based on hierarchical bases, but we do not describe these here (but see, e.g. [16]). A very common choice of basis is that given by choosing a linear polynomial over each of the domains.

The finite element space then depends on the triangulation used. Let a triangulation be indexed by the diameter of its largest element, $\delta > 0$. In a rectangular domain in $\mathbb{R}^d$ with uniform triangulation of size δ, we would have $(n_\delta)^d$ elements in our triangulation. Solving for the optimal solution in this finite dimensional space is equivalent to solving a linear system of size $(n_\delta)^d$. Variants of this method are known as Galerkin methods (see, e.g. [18]).

We drive the error to zero in these approximations by increasingly refining our approximation space, that is, by letting δ, the coarseness of the discretization, go to zero. Simultaneously, however, we drive the size of the system we need to solve to infinity, exponentially in the dimension of the problem. Indeed, especially in higher dimensions, the number of elements in the triangulation and also the size of the linear system we must solve, $(n_\delta)^d$, quickly becomes intractible. For example, for $\epsilon = 0.001$, an ϵ–discretization of the cube $[-1, 1]^3 \subseteq \mathbb{R}^3$, involves 8×10^9 elements. A good deal of work has been done to get the best of both worlds: the computational efficiency of a coarse triangulation (also called a mesh) and the computational accuracy of a fine mesh. For instance, in [13] and [14], Patera et al. describe a Lagrange multiplier method that relies on quadratic programming duality, to use coarse–mesh–multipliers to obtain sub–optimal fine–mesh–multipliers, and hence approximate the solution implied by the usual stationarity conditions. This method is extended to actually provide guidance for adaptive refinement of the mesh, in [15]. There are a number of other duality based methods as described in [7]. Regardless of the particular method's details, however, the common denominator is the search for an approximate solution in the sequence of finite dimensional subspaces.

Finally, we note that these methods, by their nature, have an inherently local quality. Indeed the error is locally minimized, and not necessarily optimized for the calculation of some integral quantity. The method we propose is, on the other hand, inherently global in nature, and thus may be better suited for computation of certain linear functionals.

Structure of the Thesis

The thesis is structured as follows. We present in Chapter 2 the proposed approach. In Section 2.2 we provide an encouraging moment driven convergence result, under certain regularity assumptions. Then we discuss the implications of this result. In Chapter 3 we present three examples that show how the method works and how it performs numerically. Finally, in Chapter 4 we discuss our perceived advantages and limitations of the proposed method.

Chapter 2

The Proposed Method

In this chapter we first provide an outline of the proposed method, and discuss each aspect in depth. In the second section, we provide a convergence result that indicates the general power of moment–driven convergence.

2.1 The Method

Suppose we are given the partial differential operators L and G operating on some distribution space $\mathcal{A}$:

$$L, G : \mathcal{A} \longrightarrow \mathcal{A},$$

and we are interested in finding

$$\int Gu(\boldsymbol{x}),$$

where $u \in \mathcal{A}$ (note also that $f \in \mathcal{A}$) satisfies the PDE,

$$Lu(\boldsymbol{x}) = f(\boldsymbol{x}), \qquad \boldsymbol{x} = (x_1, \ldots, x_d) \in \Omega \subset \mathbb{R}^d, \tag{2.1}$$

including the appropriate boundary conditions on $\partial\Omega$,

Eq. (2.1) is understood in the sense that both sides of the equation act in the same way on a given class of functions $\mathcal{D}$, i.e.,

$$Lu = f \Longleftrightarrow \int (Lu)\phi = \int f\phi, \quad \forall \phi \in \mathcal{D},$$

where $\mathcal{D}$ is taken to be some sufficiently nice class of test functions—typically a subset of the smooth functions $\mathcal{C}^\infty$.

We will assume that the operators L and G are linear operators with coefficients that are polynomials of the variables. In Section 2.1.5 we discuss extensions for a nonlinear operator G. In particular,

$$Lu(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}} L_{\boldsymbol{\alpha}}(\boldsymbol{x}) \frac{\partial^{\boldsymbol{\alpha}} u(\boldsymbol{x})}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}}, \qquad Gu(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}} G_{\boldsymbol{\alpha}}(\boldsymbol{x}) \frac{\partial^{\boldsymbol{\alpha}} u(\boldsymbol{x})}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}},$$

where $\boldsymbol{\alpha} = (i_1, \ldots, i_d)$ is a multi-index,

$$\frac{\partial^{\boldsymbol{\alpha}} u(\boldsymbol{x})}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} = \frac{\partial^{\sum_k i_k} u(\boldsymbol{x})}{\partial x_1^{i_1} \cdots \partial x_d^{i_d}},$$

and $L_{\boldsymbol{\alpha}}(\boldsymbol{x})$ and $G_{\boldsymbol{\alpha}}(\boldsymbol{x})$ are multivariate polynomials (we discuss extensions in Section 2.1.6). We will restrict ourselves to the case where $\mathcal{D}$ is separable, that is, it has a countable dense subset. This restriction is not as limiting as it might first appear. In particular, if the solution u has compact support, then we may also assume without loss of generality that every element of $\mathcal{D}$ has compact support as well, and thus by the Stone–Weierstrass theorem, $\mathcal{D}$ is separable. The condition that u have compact support may also be replaced by the (slightly) weaker condition that u have exponentially decaying tails.

Let $\mathcal{F} = \{\phi_1, \phi_2, \ldots\}$ generate (in the basis sense) a dense subset of $\mathcal{D}$. Then, by the linearity of integration we have

$$\begin{aligned} Lu = f \quad &\Longleftrightarrow \quad \int (Lu)\phi = \int f\phi, \quad \forall \phi \in \mathcal{D}, \\ &\Longleftrightarrow \quad \int (Lu)\phi_i = \int f\phi_i, \quad \forall \phi_i \in \mathcal{F}. \end{aligned}$$

We discuss different choices for the subset $\mathcal{F}$ in Section 2.1.6. One separable subspace around which this thesis focuses is the subspace spanned by the monomials $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{i_1} \dots x_d^{i_d}$. Polynomials have the property that they are closed under action by polynomial coefficient differential operators.

The Adjoint Operator

The adjoint operator, L^*, is defined by the equation:

$$\int (Lu)\phi = \int u(L^*\phi), \quad \forall \phi \in \mathcal{D}.$$

Therefore, if we have both L and L^*, then equality in the original PDE becomes:

$$\begin{aligned} Lu = f \quad \Longleftrightarrow \quad & \int (Lu)\phi = \int f\phi, \quad \forall \phi \in \mathcal{D}, \\ \Longleftrightarrow \quad & \int (Lu)\phi_i = \int f\phi_i, \quad \forall \phi_i \in \mathcal{F}, \\ \Longleftrightarrow \quad & \int u(L^*\phi_i) = \int f\phi_i, \quad \forall \phi_i \in \mathcal{F}. \end{aligned} \tag{2.2}$$

To illustrate the computation of the adjoint operator, we consider the one dimensional case. The general term of this operator is, up to a constant multiple:

$$x^a \frac{\partial^b}{\partial x^b}.$$

Using the notation $\tilde{\phi} = x^a\phi$, this term's contribution to the adjoint operator is as follows.

$$\begin{aligned} \int_\Omega x^a(\partial^b u)\phi &= \int_\Omega (\partial^b u)(x^a\phi)\,dx = \int_\Omega (\partial^b u)\tilde{\phi}\,dx \\ &= u^{(b-1)}\tilde{\phi}\Big|_{\partial\Omega} + \cdots + (-1)^{k+1}u^{(b-k)}\tilde{\phi}^{(k-1)}\Big|_{\partial\Omega} + \cdots \\ &\qquad + (-1)^{b+1}u\tilde{\phi}^{(b-1)}\Big|_{\partial\Omega} + (-1)^b\int_\Omega u\partial^b\tilde{\phi}\,dx. \end{aligned}$$

Thus, while perhaps notationally tedious in higher dimensions, computing the adjoint of a linear partial differential operator with polynomial coefficients is essentially only as difficult as performing the chain rule for differentiation on polynomials, and in particular, it may be easily automated.

2.1.1 Linear Constraints

Let us define variables in an optimization sense

$$M_{\boldsymbol{\alpha}} = \int_{\Omega} \boldsymbol{x}^{\boldsymbol{\alpha}} u(\boldsymbol{x}) = \int_{\Omega} {x_1}^{i_1} \cdots {x_d}^{i_d} u(\boldsymbol{x}),$$

together with variables related to the boundary $\partial\Omega$:

$$z_{\boldsymbol{\alpha}} = \int_{\partial\Omega} \boldsymbol{x}^{\boldsymbol{\alpha}} u(\boldsymbol{x}) = \int_{\partial\Omega} {x_1}^{i_1} \cdots {x_d}^{i_d} u(\boldsymbol{x}).$$

The specific form of these variables depends on the nature of the boundary conditions we are given (see Chapter 3 for specific examples). We refer to the quantities $M_{\boldsymbol{\alpha}}$ and $z_{\boldsymbol{\alpha}}$ as moments, even though $u(\cdot)$ is not a probability distribution. We select as ϕ_i's the family of monomials $\boldsymbol{x}^{\boldsymbol{\alpha}}$. Since, for the case we are considering, L, and thus L^*, are linear operators with coefficients that are polynomials in $\boldsymbol{x}$, then Eqs. (2.2) can be written as linear equations in terms of the variables $\boldsymbol{M} = (M_{\boldsymbol{\alpha}})$ and $\boldsymbol{z} = (z_{\boldsymbol{\alpha}})$.

2.1.2 Objective Function Value

Given that the operator G is also a linear operator with coefficients that are polynomials of the variables, then the functional $\int Gu$ can also be expressed as a linear function of the variables $\boldsymbol{M}$ and $\boldsymbol{z}$. So if we minimize or maximize this particular linear function, we obtain upper and lower bounds on the value of the functional.

2.1.3 Semidefinite Constraints

Let us assume that the solution we are looking for is bounded from below, that is $u(\boldsymbol{x}) \geq u_0$. The constant u_0 is in fact unknown. In certain cases, u_0 is naturally known; for example if $u(\boldsymbol{x})$ is a probability distribution, then $u(\boldsymbol{x}) \geq 0$, i.e., $u_0 = 0$; or if $u(\boldsymbol{x})$ represents temperature, then again $u(\boldsymbol{x}) \geq 0$. Most current methods do not explicitly use this constraint.

We consider the vector $\boldsymbol{F}(\boldsymbol{x}) = [\boldsymbol{x}^{\boldsymbol{\alpha}}]$ and the semidefinite matrix $\boldsymbol{F}(\boldsymbol{x})\boldsymbol{F}(\boldsymbol{x})'$. Then the matrices

$$\int_{\Omega} (u(\boldsymbol{x}) - u_0) F(\boldsymbol{x}) F(\boldsymbol{x})', \qquad \int_{\partial\Omega} (u(\boldsymbol{x}) - u_0) F(\boldsymbol{x}) F(\boldsymbol{x})'$$

are also positive semidefinite. This leads to semidefinte constraints involving the variables $(\boldsymbol{M}, u_0)$ and $(\boldsymbol{z}, u_0)$.

Note that this is an extension to multiple dimensions of the classical moment problem (see Akhiezer [2]). The problem is to determine, given some sequence of numbers, whether it is a valid moment sequence, that is to say, whether the numbers given are indeed the moments of a nonnegative function or distribution. In one dimension, if $u(x) \geq 0$, and we define $m_i = \int_{-\infty}^{+\infty} x^i u(x) dx$, then the sequence of moments $\{m_i\}$ is valid if any only if the matrix

$$M_{2n} = \begin{pmatrix} m_0 & m_1 & \cdots & m_n \\ m_1 & m_2 & \cdots & m_{n+1} \\ \vdots & & \ddots & \vdots \\ m_n & & & m_{2n} \end{pmatrix}$$

is positive semidefinite for every n. In the case where $u(x)$ has support $[0, \infty)$, we need to add the additional constraint, that the matrix

$$M_{2n+1} = \begin{pmatrix} m_1 & m_2 & \cdots & m_{n+1} \\ m_2 & m_3 & \cdots & m_{n+2} \\ \vdots & & \ddots & \vdots \\ m_{n+1} & & & m_{2n+1} \end{pmatrix}$$

also be positive semidefinite.

In multiple dimensions, it is generally unknown which are the exact necessary and sufficient conditions for $M_{\boldsymbol{\alpha}}$ and $z_{\boldsymbol{\alpha}}$ to be a valid moment sequence, when we are working over a general domain. For a wide class of domains, however, Schmüdgen [22] finds such conditions. We review his work briefly, and use it to derive the necessary and sufficient conditions for $M_{\boldsymbol{\alpha}}$ and $z_{\boldsymbol{\alpha}}$ to be a moment sequence.

An Operator Approach

Given a closed subset Ω of $\mathbb{R}^d$, a sequence of numbers $M_{\boldsymbol{\alpha}}$ defines a valid moment sequence if there exists a measure μ such that

$$M_{\boldsymbol{\alpha}} = \int_{\Omega} \boldsymbol{x}^{\boldsymbol{\alpha}} d\mu.$$

We define the linear operator

$$Hf = \int_{\Omega} f(\boldsymbol{x}) d\mu.$$

It is obviously necessary that $Hf \geq 0$, whenever $f \geq 0$ on Ω. A classical theorem says that it is also sufficient:

Theorem 1 (Haviland [11]) *If $\Omega \subseteq \mathbb{R}^n$ is closed, then $M_{\boldsymbol{\alpha}}$ defines a valid moment sequence if and only if the linear operator H is nonnegative on all polynomials that are nonnegative on Ω.*

Theorem 1 implies that the problem of finding necessary and sufficient conditions for $M_{\boldsymbol{\alpha}}$ and $z_{\boldsymbol{\alpha}}$ to be a moment sequence, reduces to checking the nonnegativity of the image of a polynomial that is nonnegative on Ω. In one dimension, we know that any polynomial that is nonnegative may be written as the sum of squares. Since the square of a polynomial may be written as a quadratic form, the nonnegativity of the operator reduces to matrix semidefiniteness conditions. The Motzkin polynomial in $\mathbb{R}^3$,

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

is an example that shows that in higher dimensions, the sum of squares decomposition of a nonnegative polynomial is not in general possible (see Reznick [19] for details).

However, Schmüdgen [22] gives a representation of all polynomials that are nonnegative over a compact finitely generated semialgebraic set Ω, as defined in the theorem below. This leads to necessary and sufficient conditions for a moment sequence to be valid on Ω.

Theorem 2 (Schmüdgen [22]) *Suppose* $\Omega := \{\boldsymbol{x} \in \mathbb{R}^n \,:\, f_i(\boldsymbol{x}) \geq 0,\ 1 \leq i \leq r\}$ *is closed and bounded, where* $f_i(\boldsymbol{x})$ *are polynomials. Then a polynomial* $g(\boldsymbol{x}) \geq 0$ *on* Ω *if and only if it is expressible as a sum of terms of the form*

$$h_I^2(\boldsymbol{x}) \prod_{k \in I} f_k(\boldsymbol{x}),$$

for $I \subset \{1, \ldots, r\}$*, and* h_I *some polynomial.*

Theorems 1 and 2 lead to the following result.

Theorem 3 *Given* $\boldsymbol{M} = [M_{\boldsymbol{\alpha}}]$*, there exists a distribution* $u(\boldsymbol{x})$ *such that*

$$M_{\boldsymbol{\alpha}} = \int_\Omega (u(\boldsymbol{x}) - u_0)\boldsymbol{x}^{\boldsymbol{\alpha}},$$

for a closed and bounded domain Ω *of the form*

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^d \,:\, f_1(\boldsymbol{x}) \geq 0, \ldots, f_r(\boldsymbol{x}) \geq 0\},$$

if and only if for all subsets $I \subseteq \{1, \ldots, r\}$ *the following matrices are positive semidefinite:*

$$\int_\Omega (u(\boldsymbol{x}) - u_0)\boldsymbol{F}(\boldsymbol{x})\boldsymbol{F}(\boldsymbol{x})' \prod_{i \in I} f_i(\boldsymbol{x}), \tag{2.3}$$

where $I \subseteq \{1, \ldots, r\}$*.*

Examples of domains for which the above result applies include the unit ball in $\mathbb{R}^d$, which can be written as

$$B = \{\boldsymbol{x} \in \mathbb{R}^d \,:\, 1 - x_1^2 - \cdots - x_d^2 \geq 0\},$$

and the unit hypercube,

$$C = \{\boldsymbol{x} \in \mathbb{R}^d \,:\, x_i \geq 0, 1 - x_i \geq 0,\ 1 \leq i \leq d\}.$$

We next make the connection to semidefinite constraints explicit. While all the results can be easily generalized to d–dimensions, for notational simplicity we consider $d = 2$, assume that $u_0 = 0$ and use Ω as the unit hypercube C in two dimensions. Note that in this case there are four functions,

$$f_1(x_1, x_2) = x_1,\ \ f_2(x_1, x_2) = 1 - x_1,\ \ f_3(x_1, x_2) = x_2,\ \ f_4(x_1, x_2) = 1 - x_2,$$

defining the set Ω. Thus, there are $2^4 = 16$ possible subsets I of $\{1, 2, 3, 4\}$. Each of these subsets gives rise to a particular semidefinite constraint as follows. Denoting the moment sequence as $\{m_{i,j}\}$, for $I = \emptyset$, we have that

$$\begin{pmatrix} m_{0,0} & m_{1,0} & m_{0,1} & m_{1,1} & \cdots \\ m_{1,0} & m_{2,0} & m_{1,1} & m_{2,1} & \cdots \\ m_{0,1} & m_{1,1} & m_{0,1} & m_{1,2} & \cdots \\ m_{1,1} & m_{2,1} & m_{1,2} & m_{2,2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \succeq \mathbf{0}.$$

For $I = \{2\}$, we obtain

$$\begin{pmatrix} m_{0,0} - m_{1,0} & m_{1,0} - m_{2,0} & m_{0,1} - m_{1,1} & m_{1,1} - m_{2,1} & \cdots \\ m_{1,0} - m_{2,0} & m_{2,0} - m_{3,0} & m_{1,1} - m_{2,1} & m_{2,1} - m_{3,1} & \cdots \\ m_{0,1} - m_{1,1} & m_{1,1} - m_{2,1} & m_{0,1} - m_{1,1} & m_{1,2} - m_{2,2} & \cdots \\ m_{1,1} - m_{2,1} & m_{2,1} - m_{3,1} & m_{1,2} - m_{2,2} & m_{2,2} - m_{3,2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \succeq \mathbf{0}.$$

Proceeding in this way, we obtain 16 semidefinite constraints. If Ω is the unit ball in d dimensions, we have exactly two semidefinite constraints.

2.1.4 The Overall Formulation

As we mentioned, the variables are the moments $M_{\boldsymbol{\alpha}} = \int_{\Omega} \boldsymbol{x}^{\boldsymbol{\alpha}} u(\boldsymbol{x})$, the boundary moments $z_{\boldsymbol{\alpha}} = \int_{\partial\Omega} \boldsymbol{x}^{\boldsymbol{\alpha}} u(\boldsymbol{x})$ and the bound u_0, which might be naturally known. The semidefinite optimization consists of linear equality constraints generated by the adjoint operator for different test functions $\boldsymbol{x}^{\boldsymbol{\alpha}}$, and of the semidefinite constraints that express the fact that the variables $M_{\boldsymbol{\alpha}}$ and $z_{\boldsymbol{\alpha}}$ are in fact moments. Subject to these constraints, we maximize and minimize a linear function of the variables that expresses the given linear functional. The overall steps of the formulation process are then summarized as follows:

(i) Compute the adjoint operator L^*.

(ii) Generate the n^{th} equality constraint by requiring that

$$\int u(L^* \phi_n) = \int f \phi_n.$$

(iii) Generate the desired semidefinite constraints; note that these only depend on the domain Ω and not on the operator L.

(iv) Compute upper and lower bounds on the given functional by solving a semidefinite optimization problem over the intersection of the positive semidefinite cone and the equality constraints.

2.1.5 The Maximum and Minimum Operator

Suppose that the given functional is

$$Gu = \min_{\boldsymbol{x} \in \Omega} u(\boldsymbol{x}).$$

Then, we will formulate the objective function

$$\min u_0.$$

This approach gives a lower bound on the minimum of $u(\boldsymbol{x})$ over Ω. However, if we maximize $\max u_0$ we do not obtain a true upper bound on the minimum of $u(\boldsymbol{x})$ over Ω, only an approximation.

Similarly, if we are interested in

$$\max_{\boldsymbol{x}\in\Omega} u(\boldsymbol{x})$$

we solve $\max v_0$ such that $u(\boldsymbol{x}) \leq v_0$, which leads to semidefinite constraints involving $\boldsymbol{M}$, $\boldsymbol{z}$ and v_0. This approach gives an upper bound on v_0, while minimizing v_0 only leads to an approximation.

Note that here the semidefinite constraints are absolutely crucial. This is because the additional variable u_0 is introduced linearly, and because of the linearity of integration, cannot possibly be calculated by the family of linear constraints. Rather, the linear constraints link it to the variables of the optimization, and then it is constrained by the semidefinite constraints.

2.1.6 Using Trigonometric Moments

Instead of choosing polynomials as test functions, we could choose other classes of test functions. Polynomials are particularly convenient as they are closed under differentiation. While this property is not a necessary condition for the proposed method to work, it significantly limits the proliferation of variables we introduce. When the linear operator has coefficients that are not polynomials, other bases might be more appropriate.

The trigonometric functions $\{\sin(nx), \cos(nx)\}$ are also closed under differentiation (again we can form products in higher dimensions, just as with monomials). Using trigonometric functions as a basis of our test functions provides a straightforward way for us to deal with linear operators with trigonometric coefficients. This is an important point, as Section 3.4 reveals, namely, that the choice of test function basis ought to depend on the coefficients of the linear operator. In Section 3.4, we present an example of the use of the method with trigonometric test functions.

2.2 The Convergence Rate

In this section we state a result (and prove it in the Appendix) that illustrates the power of moment driven convergence. In this thesis, we propose an algorithm which obtains upper and lower bounds on moments of a solution to a linear PDE. The maximization and minimization of a given moment, say the i^{th} moment, subject to the linear and semidefinite constraints generated at the n^{th} stage of the algorithm, computes the possible range of that moment, subject to the constraint that it is part of a valid moment sequence, satisfying the particular linear constraints. We can express this by the family of sets,

$$C_{\boldsymbol{\alpha},N} := \left\{ \gamma = \int u(\boldsymbol{x})\boldsymbol{x}^{\boldsymbol{\alpha}} \ : \ \exists u(\boldsymbol{x}) \geq 0, \ \int (Lu)\boldsymbol{x}^{\boldsymbol{\beta}} = \int f(\boldsymbol{x})\boldsymbol{x}^{\boldsymbol{\beta}}, \ |\boldsymbol{\beta}| \leq N \right\}.$$

If we define the family of sets

$$A_N := \left\{ u(\boldsymbol{x}) \geq 0 \ : \ \int (Lu(\boldsymbol{x}))\boldsymbol{x}^{\boldsymbol{\beta}} = \int f(\boldsymbol{x})\boldsymbol{x}^{\boldsymbol{\beta}} x^{\beta}, \ |\boldsymbol{\beta}| \leq N \right\},$$

then

$$C_{\boldsymbol{\alpha},N} = \left\{ \gamma = \int u(\boldsymbol{x})\boldsymbol{x}^{\boldsymbol{\alpha}} \ : \ u(\boldsymbol{x}) \in A_N \right\}.$$

Note that $C_{\boldsymbol{\alpha},N}$ is a subset of $\mathbb{R}$, but A_N is a subset of a function space. This shows that we are interested in the image in $\mathbb{R}$ of a linear functional of the set A_N. Note that both the sets $C_{\boldsymbol{\alpha},N}$, and also A_N, converge to a single point as $N \to \infty$. Evidently, since our linear operator is bounded, the sets $C_{\alpha,N}$ converge at least as fast as A_N (and in all likelihood, considerably faster). Finally, consider the sets

$$B_N := \left\{ v(\boldsymbol{x}) \ : \ \int v(\boldsymbol{x})\boldsymbol{x}^{\boldsymbol{\beta}} = \int f(\boldsymbol{x})\boldsymbol{x}^{\boldsymbol{\beta}}, \ |\boldsymbol{\beta}| \leq N \right\}.$$

We see that $L(A_N) \subseteq B_N$. The following theorem states that a certain regular subset of B_N converges exponentially in N.

Proposition 1 *Suppose the domain of our PDE, $K \subseteq \mathbb{R}^n$, is compact. Then in the usual topology of weak–convergence, if the sequence L_N is allowed to increase no faster* *than linearly, the diameter of the sets*

$$B_N \cap \Big\{ u \in \mathcal{C}^\infty \,:\, \Big|\Big|\nabla u\Big|\Big|_{L^1(K)} < L_N \Big\},$$

computed in the usual bounded Lipschitz metric, decreases exponentially in N.

The proof involves results of functional, and complex analysis, and is left for the Appendix.

Chapter 3

Examples

In this section, we illustrate our approach with three examples: (a) a simple homogeneous ordinary differential equation, (b) a more interesting ODE: Bessel's equation, and (c) a two-dimensional partial differential equation known as Helmholtz's equation. In all three examples, we take the solution to have support on the unit interval for the ODEs, and on the unit square for the PDE.

3.1 Example 1: $u'' + 3u' + 2u = 0$

We consider the linear ODE with constant coefficients

$$u'' + 3u' + 2u = 0 \tag{3.1}$$

with the boundary conditions $u'(0) = -2e^2$ and $u'(1) = -2$, and $\Omega = [0, 1]$. In this case, we can easily find the solution $u(x) = e^2 \cdot e^{-2x}$. Let us apply the proposed method. For simplicity of the exposition we use the fact that $u(x) \geq 0$.

We can compute the adjoint operator directly by integration by parts:

$$\int_0^1 (u'' + 3u' + 2u)\phi = u'\phi\Big|_0^1 - u\phi'\Big|_0^1 + 3u\phi\Big|_0^1 + \int_0^1 (u\phi'' - 3u\phi' + 2u\phi).$$

We use $\phi_i(x) = x^i$, $i = 0, \ldots, n$ and let

$$m_i = \int_0^1 x^i u(x) dx.$$

Together with the two unknown boundary conditions $u(0)$ and $u(1)$, we have $n+1$ variables m_i, $i = 0, \ldots, n$ for a total of $n+3$ variables. The linear equality constraints generated by the adjoint equations are:

$$\begin{aligned}
\phi = 1: &\;\Rightarrow\; 3(u(1) - u(0)) + 2m_0 = u'(0) - u'(1), \\
\phi = x: &\;\Rightarrow\; 2u(1) + u(0) - 3m_0 + 2m_1 = -u'(1), \\
\phi = x^2: &\;\Rightarrow\; u(1) + 2m_0 - 6m_1 + 2m_2 = -u'(1), \\
&\;\vdots \\
\phi = x^n: &\;\Rightarrow\; (3-n)u(1) + n(n-1)m_{n-2} - 3n \cdot m_{n-1} + 2m_n = -u'(1).
\end{aligned}$$

Since we assume that the solution has support on $[0, 1]$, we apply Theorem 3 to derive the two semidefinite constraints:

$$\begin{pmatrix} m_0 & m_1 & \cdots & m_n \\ m_1 & m_2 & \cdots & m_{n+1} \\ \vdots & & \ddots & \vdots \\ m_n & & & m_{2n} \end{pmatrix} \succeq \mathbf{0}, \qquad \begin{pmatrix} m_0 - m_1 & m_1 - m_2 & \cdots & m_n - m_{n+1} \\ m_1 - m_2 & m_2 - m_3 & \cdots & m_{n+1} - m_{n+2} \\ \vdots & & \ddots & \vdots \\ m_n - m_{n+1} & & & m_{2n} - m_{2n+1} \end{pmatrix} \succeq \mathbf{0}.$$

Subject to these constraints, we maximize and minimize each of the m_i, $0 \le i \le n$ in order to obtain values for m_i.

We applied two semidefinite optimization packages to solve the resulting SDPs: the optimization package SDPA version 5.00 by Fujisaw, Kojima and Nakata [10] and the Matlab based package SeDuMi version 1.03, by Sturm [24]. The semidefinite optimizations were run on a Sparc 5.

In Table 3.1, we report the results from SDPA using monomials up to $N = 14$. As SDPA exhibited some numerical instability, we replaced the equality constraints $\boldsymbol{a'x} = b$ with $-\varepsilon + b \le \boldsymbol{a'x} \le b + \varepsilon$ with $\varepsilon = 0.001$.

Variable	LB	UB
m_0	3.1939	3.1951
m_1	1.0969	1.1619
m_2	0.5975	0.5997
m_3	0.3957	0.3961
m_4	0.2916	0.3179
m_5	0.2580	0.8809

Table 3.1: Upper and lower bounds for the ODE (3.1) for $N = 14$, using SDPA. The total computation time was less than 15 seconds for all the twelve SDPs.

We observe that because of the perturbation we introduced the bounds are only accurate up to the second decimal point. We see, as we would expect, that the performance begins to deteriorate as we ask for higher order moments.

In Table 3.2, we report results using SeDuMi with $N = 60$. SeDuMi successfully solved for the first 45 moments, such that the upper and lower bounds agreed to 5 decimal points.

In order to test the ability of our method to find the minimum of $u(x)$, we reversed the sign of the boundary values for this linear ODE, to obtain an ODE with a solution that is no longer nonnegative:

$$u(x) = -e^2 e^{-2x}.$$

By implementing the method we outlined to compute the minimum of $u(x)$ in the previous section, and using SeDuMi, we obtain the exact value for the minimum of the function $u(x)$ to be $u_0 = -7.389$.

3.2 The Bessel Equation

In this section we consider Bessel's differential equation

$$x^2 u'' + xu' + (x^2 - p^2)u = 0.$$

Variable	LB	UB
m_0	3.1945	3.1945
m_1	1.0973	1.0973
m_2	0.5973	0.5973
m_3	0.3959	0.3959
m_4	0.2918	0.2918
m_5	0.2295	0.2295
m_6	0.1884	0.1884
m_7	0.1595	0.1595
m_8	0.1382	0.1382
m_9	0.1218	0.1218
m_{10}	0.1088	0.1088
$\vdots$	$\vdots$	$\vdots$
m_{20}	0.0524	0.0524
$\vdots$	$\vdots$	$\vdots$
m_{30}	0.0344	0.0344
$\vdots$	$\vdots$	$\vdots$
m_{40}	0.0256	0.0256

Table 3.2: Upper and lower bounds for the ODE (3.1) for $N = 60$, using SeDuMi. The total computation time was under five minutes.

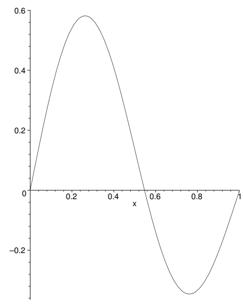

Figure 3-1: Bessel Function

The Bessel function and its variants appear in one form or another in a wide array of engineering applications, and applied mathematics. Furthermore, while there are integral and series representations, the Bessel function is not expressible in closed form. The series representation of the Bessel function, which can be found in, e.g. Watson [26], is:

$$J_p(x) := \sum_{k=0}^{\infty} \frac{(-1)^k (x/2)^{2k+p}}{k!(k+p)!}.$$

Also, over the appropriate range, the Bessel function is neither nonnegative, nor convex, as illustrated in Figure 3-1.

In order to avoid numerical difficulties from large constant factors, we solve a modified version of Bessel's equation:

$$x^2 u'' + xu' + (49x^2 - p^2)u = 0. \tag{3.2}$$

The solution is $u(x) = J_p(7x)$. Assuming we are given the value of the derivatives on the boundary, using the monomials as the test functions, we obtain the adjoint equations:

$$\begin{aligned} \phi = 1: \quad &\Rightarrow \quad -u(1) + (1 - p^2)m_0 + 49m_2 = u'(1), \\ \phi = x: \quad &\Rightarrow \quad -2u(1) + (4 - p^2)m_1 + 49m_3 = u'(1), \\ \phi = x^2: \quad &\Rightarrow \quad -3u(1) + (9 - p^2)m_2 + 49m_4 = u'(1), \\ &\vdots \end{aligned}$$

N	Minimum	Maximum
20	-0.3087	0.4986
24	-0.3101	0.5068
30	-0.3111	0.5081
40	-0.3142	0.5046

Table 3.3: Approximations for the maximum and the minimum of the solution of Eq. (3.2) using SeDuMi.

Variable	LB	UB
m_1	0.1766	0.1766
m_2	0.0903	0.0903
m_3	0.0583	0.0583
m_4	0.0438	0.0438
m_5	0.0361	0.0361

Table 3.4: Upper and lower bounds for Eq. (3.2) for $N = 24$ using SeDuMi.

$$\phi = x^n : \;\Rightarrow\; -(n+1)u(1) + ((n+1)^2 - p^2)m_n + 49m_{n+2} = u'(1).$$

In what follows, we choose $p = 1$. We used SeDuMi to compute the moments, and also to compute the max and min. Recall from the discussion in Section 2.1.5 that while we are able to obtain bounds for the moments, our method can only compute *approximations* to the max and min of the solution. In the case of the Bessel function, the approximations we obtain of the minimum are greater than the actual value, and the approximations for the maximum are less than the actual value. The true values are: min $= -0.347$ and max $= 0.583$. In Table 3.3 we report the results from using SeDuMi.

SeDuMi reported severe numerical instabilities for the computation of the maximum for the cases $N = 30$ and $N = 40$.

Next, we use these results to translate the function so that it is nonnegative, and so that we can compute the moments of the translated function. We use $u(x) - u_0 \geq 0$ with $u_0 = -0.4$. Again using SeDuMi, we obtain very accurate bounds to the moments. We give the first few in Table 3.4. We would expect by linearity, and indeed the results show, that just having a lower bound on the function is enough to find accurate results on the moments of the function.

3.3 The Helmholtz Equation

In this section we consider the two dimensional PDE

$$\triangle u + k^2 u = f \tag{3.3}$$

over $\Omega = [0,1]^2$. To compute the adjoint operator we need to use Stokes's formula:

$$\int_\Omega d\omega = \int_{\partial\Omega} \omega.$$

Recall that in two dimensions we have:

$$\omega = f\,dx + g\,dy \Longleftrightarrow d\omega = \left(\frac{\partial g}{\partial x} - \frac{\partial f}{\partial y}\right) dxdy,$$

and thus computing the adjoint operator, we have:

$$\begin{aligned}
\int_\Omega \left(\frac{\partial^2 u}{\partial x^2}\right) \phi\, dxdy &= \int_\Omega \left(\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x} \cdot \phi\right) - \frac{\partial u}{\partial x}\frac{\partial \phi}{\partial x}\right) dxdy \\
&= \int_{\partial\Omega} \frac{\partial u}{\partial x} \cdot \phi\, dy - \int_\Omega \left(\frac{\partial}{\partial x}\left(u\frac{\partial \phi}{\partial x}\right) - u\frac{\partial^2 \phi}{\partial x^2}\right) dxdy \\
&= \int_{\partial\Omega} \left(\frac{\partial u}{\partial x} \cdot \phi - u\frac{\partial \phi}{\partial x}\right) dy + \int_\Omega u\frac{\partial^2 \phi}{\partial x^2}\, dxdy,
\end{aligned}$$

By a similar process for the $\frac{\partial^2 u}{\partial y^2}$ term, we obtain

$$\begin{aligned}
\int_\Omega \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f\right) \phi dxdy = \int_\Omega \left(f\phi + \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2}\right) dxdy \\
+ \int_{\partial\Omega} \left(\frac{\partial u}{\partial x} \cdot \phi - u\frac{\partial \phi}{\partial x}\right) dy - \int_{\partial\Omega} \left(\frac{\partial u}{\partial y} \cdot \phi - u\frac{\partial \phi}{\partial y}\right) dx.
\end{aligned}$$

Again we consider the family of monomials,

$$\mathcal{F} = \{x^i \cdot y^j\}, \quad \text{for } i, j \in \mathbb{N} \cup \{0\}.$$

In addition to the variables

$$m_{i,j} = \int_0^1 \int_0^1 x^i y^j u(x, y) dx dy,$$

we also introduce the boundary moment variables:

$$\begin{aligned}
b_i^{x=1} &:= \int_0^1 u(x = 1, y) y^i \, dy, \quad & d_i^{x=1} &:= \int_0^1 \partial_x u(x = 1, y) y^i \, dy \\
b_i^{x=0} &:= \int_0^1 u(x = 0, y) y^i \, dy, \quad & d_i^{x=0} &:= \int_0^1 \partial_x u(x = 0, y) y^i \, dy \\
b_i^{y=1} &:= \int_0^1 u(x, y = 1) x^i \, dx, \quad & d_i^{y=1} &:= \int_0^1 \partial_y u(x, y = 1) x^i \, dx \\
b_i^{y=0} &:= \int_0^1 u(x, y = 0) x^i \, dx, \quad & d_i^{y=0} &:= \int_0^1 \partial_y u(x, y = 0) x^i \, dx.
\end{aligned}$$

Then the adjoint relationship above yields:

$$\begin{aligned}
\phi = 1 : \quad &\Rightarrow \quad d_0^{x=1} - d_0^{x=0} + d_0^{y=1} - d_0^{y=0} + m_{0,0} = \int_\Omega f \, dx dy \\
\phi = x^i : \quad &\Rightarrow \quad d_0^{x=1} + i(i-1) m_{i-2,0} + d_i^{y=1} - d_i^{y=0} + m_{i,0} = \int_\Omega f \cdot x^i \, dx dy \\
\phi = y^j : \quad &\Rightarrow \quad d_j^{x=1} - d_j^{x=0} + d_0^{y=1} + j(j-1) m_{0,j-2} + m_{0,j} = \int_\Omega f \cdot y^i \, dx dy \\
\phi = x^i y^j : \quad &\Rightarrow \quad d_j^{x=1} + i(i-1) m_{i-2,j} + d_i^{y=1} + j(j-1) m_{i,j-2} + m_{i,j} = \int_\Omega f \cdot x^i y^j \, dx dy.
\end{aligned}$$

Note that either the $\{d_i^x, d_j^y\}$, or the $\{b_i^x, b_j^y\}$, are given as boundary values. In order to compare with the exact solution, we selected the boundary conditions such that $f(x, y) = 3e^{x+y}$.

In order for the $m_{i,j}$ to be a valid moment sequence, we need to impose 16 semidefinite matrix constraints. Similarly, we need to impose two semidefinite con-

Variable	Value
$m_{0,0}$	2.9525
$m_{1,0}$	1.7183
$m_{1,1}$	1.0000
$m_{2,0}$	1.2342
$m_{2,1}$	0.7183
$m_{2,2}$	0.5159
$m_{3,0}$	0.9681
$m_{3,1}$	0.5634
$m_{3,2}$	0.4047
$m_{3,3}$	0.3175

Table 3.5: Exact Results for the solution of the PDE (3.3).

Variable	LB, $N = 5$	UB, $N = 5$	LB, $N = 10$	UB, $N = 10$	LB, $N = 20$	UB, $N = 20$
$m_{0,0}$	0.0000	$+\infty$	0.0000	$+\infty$	0.0000	$+\infty$
$m_{1,0}$	0.0000	4.3142	0.0000	4.0822	0.0000	3.8694
$m_{1,1}$	0.7559	1.0881	0.8557	1.0419	0.9400	1.0120
$m_{2,0}$	0.0000	4.6790	0.0000	4.6790	0.0000	4.6790
$m_{2,1}$	0.0545	1.0563	0.1417	1.0059	0.1749	0.9753
$m_{2,2}$	0.0000	0.9447	0.0000	0.9087	0.0000	0.9087
$m_{3,0}$	0.0000	4.8743	0.0000	4.4932	0.0000	4.4414
$m_{3,1}$	0.1692	0.6806	0.4015	0.6063	0.5025	0.5758
$m_{3,2}$	0.0000	0.6383	0.0000	0.6105	0.0000	0.5989
$m_{3,3}$	0.0000	0.5291	0.1684	0.3640	0.2573	0.3296

Table 3.6: Upper and lower bounds from linear optimization for $N = 5$, $N = 10$, $N = 20$.

straints for all boundary variables. In order to illustrate the power of the semidefinite constraints, we run our optimization problem in two different stages. First, we provide the results of solving the linear optimization problem generated by the adjoint equation, using the commercial software AMPLE. Next, we enforce the semidefinite constraints. The true results, as computed by Maple 6.0, are reported in Table 3.5. Solving a linear optimization problem, ignoring the semidefinite constraints and only imposing nonnegativity constraints on the variables, we obtain the bounds in Table 3.6 for $N = 5, 10, 20$.

We then add the semidefinite constraints and use SDPA to solve the correspond-

Variable	LB, $N=5$	UB, $N=5$	LB, $N=10$	UB, $N=10$	LB, $N=20$	UB, $N=20$
$m_{0,0}$	2.7687	5.5689	2.9818	2.9809	2.9598	2.9609
$m_{1,0}$	1.8377	2.2006	1.7271	1.7276	1.7189	1.7185
$m_{1,1}$	0.9679	0.9841	0.9971	0.9968	0.9988	0.9988
$m_{2,0}$	1.0860	1.8173	1.2433	1.2437	1.2346	1.2353
$m_{2,1}$	0.6089	0.6749	0.7146	0.7146	0.7177	0.7172
$m_{2,2}$	0.3192	0.4040	0.5126	0.5126	0.5148	0.5148
$m_{3,0}$	0.9789	1.7799	0.9790	0.9779	0.9687	0.9683
$m_{3,1}$	0.5296	0.6076	0.5605	0.5606	0.5623	0.5624
$m_{3,2}$	0.2021	0.3339	0.4009	0.4016	0.4036	0.4036
$m_{3,3}$	0.1025	1.1311	0.3139	0.3141	0.3163	0.3161

Table 3.7: Upper and lower bounds for Eq. (3.3) for $N = 5,\ 10$ using SDPA. The computation of each bound took less than 0.5 seconds for $N = 5$, 3-5 seconds for $N = 10$, and 1-3 minutes for $N = 20$. Note that because of numerical errors, some times the lower bounds are slightly higher than the upper bounds.

ing semidefinite optimization problems. SDPA gave very tight bounds; however, as the numerical results demonstrate, there were nevertheless some numerical instabilities. In Table 3.7 we report upper and lower bounds for $N = 5,\ 10,\ N = 20$. Note that when we use monomials up to degree N, there are in fact N^2 such monomials.

We notice that the tightness of the bounds is nearly as dramatic as in the one dimensional case. Moreover, the bounds using semidefinite optimization are significantly tighter than the ones obtained using linear optimization. This observation is significant and emphasizes the importance of the semidefinite constraints. For example, without the semidefinite constraints, the upper bound on $m_{0,0}$ is $+\infty$, whereas we obtain very tight bounds for $N = 10$ using the semidefinite constraints. There are, however, numerical difficulties with the software. For example, the lower bound is sometimes slightly higher than the upper bound. Even more disturbingly, the interval between the lower and upper bound does not contain the exact answer, even though it is close. We attribute these problems to the numerical stability of the particular software we use, and not to the method itself.

3.4 Trigonometric Test Functions

In this section we illustrate the use of trigonometric test functions. We consider the differential equation

$$u'' + 2u' + \sin(2\pi x)u = 10\sin x - 20\cos x + (10 - 10\sin x)\sin(2\pi x). \tag{3.4}$$

Note that if we attempted to use a polynomial basis we would encounter a proliferation of variables, since the polynomials are not closed by action of the adjoint (which has a $\sin(2\pi x)$ term). We use the family of functions

$$\phi_{2n}(x) := \sin(2\pi nx), \quad \phi_{2n+1}(x) := \cos(2\pi nx).$$

We define the variables:

$$\begin{aligned} m_{2n} &:= \int_\Omega u(x)\phi_{2n}(x)\,dx \\ m_{2n+1} &:= \int_\Omega u(x)\phi_{2n+1}(x)\,dx. \end{aligned}$$

The adjoint equations become:

$$\begin{aligned} \phi_1 = 1: \quad &\Rightarrow \quad 2u(1) - 2u(0) + m_2 = \int_\Omega f\,dx, \\ \phi_{2n} = \sin(2\pi nx): \quad &\Rightarrow \quad 2\pi n(u(0) - u(1)) + \frac{1}{2}m_{2(n-1)+1} - \frac{1}{2}m_{2(n+1)+1} \\ &\qquad -4\pi^2n^2m_{2n} - 4\pi n m_{2n+1} = \int_\Omega f\phi_{2n}\,dx \\ \phi_{2n+1} = \cos(2\pi nx): \quad &\Rightarrow \quad 2u(1) - 2u(0) + \frac{1}{2}m_{2(n+1)} - \frac{1}{2}m_{2(n-1)} \\ &\qquad -4\pi^2n^2m_{2n+1} - 4\pi n m_{2n} = \int_\Omega f\phi_{2n+1}\,dx. \end{aligned}$$

Note that for the semidefinite constraints products $\cos(2\pi nx)\cdot\sin(2\pi mx)$ appear, which can be rewritten as follows:

Variable	LB	UB
m_0	0.1128	3.1239
m_1	0.6730	1.0954
m_2	0.0000	0.0294
m_3	0.4471	0.7192
m_4	0.0127	0.0130
m_5	0.3349	0.5390
m_6	0.0072	0.0073
m_7	0.2678	0.4310
m_8	0.0046	0.0047
m_9	0.2231	0.3591
m_{10}	0.0032	0.0032

Table 3.8: Upper and lower bounds for the ODE (3.4) for $N = 20$ using SeDuMi.

$$\begin{aligned}
\sin(2\pi nx)\cdot\cos(2\pi mx) &= \frac{1}{2}(\sin(2\pi(n+m)x) + \operatorname{sgn}(n-m)\sin(2\pi|n-m|x)) \\
\sin(2\pi nx)\cdot\sin(2\pi mx) &= \frac{1}{2}(\cos(2\pi(n-m)x) - \cos(2\pi(n+m)x)) \\
\cos(2\pi nx)\cdot\cos(2\pi mx) &= \frac{1}{2}(\cos(2\pi(n-m)x) + \cos(2\pi(n+m)x)).
\end{aligned}$$

Using SeDuMi we report in Table 3.8 upper and lower bounds for this ODE using trigonometric test functions. We see that the bounds are much tighter for the even moments. While the bounds are not as tight as in the earlier cases, nevertheless they do give an indication that the proposed method may have further applications than polynomial moments.

3.5 Insights From The Computations

In this section we summarize the major insights from the computations we performed.

(i) In both one and two dimensions, the proposed method gave strong bounds in reasonable times.

(ii) Perhaps the most encouraging finding is that the semidefinite constraints sig-

nificantly improve over the bounds from the linear constraints.

(iii) The software packages we used exhibited some numerical difficulties.

(iv) Our experiments with trigonemetric moments indicate that the proposed method is not restrictive to PDEs with polynomial coefficients, but can accomodate more general coefficients by appropriately changing the underlying basis.

Chapter 4

Concluding Remarks

We have presented a method for providing bounds on functionals defined on solutions of PDEs using semidefinite optimization methods.

The algorithm proposed in this thesis uses N elements of our chosen function family (for example polynomials), and uses $O(N^d)$ variables. Compared to traditional discretization methods, the proposed method provides bounds, as opposed to approximate solutions by solving a semidefinite optimization problem on $O(N^d)$ variables. The computational results at least for one or two dimensions indicate that we obtain relatively tight bounds even with small to moderate N, which is encouraging.

Despite a lot of progress in recent years, the current state of the art of semidefinite optimization codes, especially with respect to stability of the numerical calculations is not yet at the level of linear optimization codes. This is one of the major limitations of the proposed method, as it relies on the semidefinite optimization codes. Moreover, we use general purpose semidefinite codes even though we have a very particular formulation with a lot of structure. The hope is that progress in the area of numerical methods for semidefinite optimization codes will improve the ability of the proposed method as well.

Appendix A

Moment Driven Convergence Proof

In this section we prove the result stated previously,

Theorem 4 *Suppose the domain of our PDE, $K \subseteq \mathbb{R}^d$, is compact. Then in the usual topology of weak–convergence, if the sequence L_n is allowed to increase no faster than linearly, the diameter of the sets*

$$B_{n,L_n} := \Big\{ u \in B_n \cap \mathcal{C}^\infty \, : \, \Big|\Big| \frac{du}{dx} \Big|\Big|_{L^1(K)} < L_n \Big\},$$

computed in the usual bounded Lipschitz metric, decreases exponentially in n.

In the proof below, we consider only the unit interval in $\mathbb{R}^1$. However this is only for notational convenience. All the results stated have precise analogs in higher dimensions, and the results proved carry over exactly as stated and proved.

Proof. The bounded Lipschitz metric is well–known to be a metrization of weak convergence (see [8]). It is given by

$$\beta(f,g) := \sup_{G \in BL_1} \langle f - g \, , \, G \rangle = \sup_{G \in BL_1} \int_0^1 (f-g) G \, d\mu,$$

where we have

$$BL_1 := \{ f \, : \, ||f||_\infty + ||f||_L \leq 1 \},$$

where $||\cdot||_\infty$ is the usual supremum norm, and $||\cdot||_L$ is the Lipschitz norm given by

$$||f||_L := \sup_{x\neq y} \frac{f(x)-f(y)}{|x-y|},$$

and is thus a smoothness condition, uniformly bounding the pointwise magnitude of the first derivative. We have:

$$\begin{aligned}
\beta(f,g) &= \sup_{G\in BL_1} \langle f-g\,,\,G\rangle \\
&= \sup_{G\in BL_1} \langle \hat{f}-\hat{g}\,,\,\hat{G}\rangle \\
&= \sup_{G\in BL_1} \langle \frac{\hat{f}-\hat{g}}{1+|\xi|}\,,\,\hat{G}\cdot(1+|\xi|)\rangle \\
&\leq \left\|\frac{\hat{f}-\hat{g}}{1+|\xi|}\right\|_{L^2(\mathbb{R})} \cdot \sup_{G\in BL_1} ||\hat{G}\cdot(1+|\xi|)||_{L^2(\mathbb{R})} \\
&= C_{BL_1}\left\|\frac{\hat{f}-\hat{g}}{1+|\xi|}\right\|_{L^2(\mathbb{R})}.
\end{aligned}$$

The first equality is by definition. In the second, $\hat{f}$ denotes the usual Fourier transform of f, and the equality follows by the fact that the Fourier transform is an isometry ([20]). The inequality in the fourth step is a consequence of the Cauchy–Schwartz inequality. The final equality comes from the fact that G belongs to a family of uniformly Lipshitz functions, and hence the transform has integrability properties as claimed (see any reference on Sobolev spaces such as, e.g. Folland (1995), or Adams (1975)).

We have the following facts about f, g and their transforms ([20]): by the compact support and smoothness of f, g we know that $\hat{f}, \hat{g}$ are analytic, and such that, for any $N \in \mathbb{N}$,

$$|\hat{f}(z)| \leq \gamma_N(1+|z|)^{-N}\cdot e^{|\mathrm{Im}\,z|},$$

where $\gamma_N \leq ||D_N f||_{L^1}$. The analyticity follows from Morera's theorem, and the growth condition from standard Fourier transform techniques (differentiate under the integral...). Thanks to our restrictions to B_{n,L_n}, we can choose a γ_1 *independent* of

$f \in B_{n,L_n}$. This is important, as we intend to take the supremum over all $f, g \in B_{n,L_n}$.

But then, given any threshold $\varepsilon > 0$, we can choose some $K_\varepsilon \in \mathbb{R}_+$ such that

$$\left\| \frac{\hat{f} - \hat{g}}{1 + |z|} \right\|_{L^2(\mathbb{R})} \leq \left\| \frac{\hat{f} - \hat{g}}{1 + |z|} \right\|_{L^2[-K,K]} + \varepsilon.$$

Let $\hat{h} := \hat{f} - \hat{g}$. Then by our assumption on the moments of f, g, we know that $\hat{h}$ has a zero of order n at 0. In other words, the first n terms of the Taylor series for $\hat{h}$ are zero. Therefore we have

$$\hat{h}(z) = \hat{h}_n(z) \cdot z^n, \quad \forall z \in \mathbb{C},$$

where $\hat{h}_n$ is the remainder term in the usual finite form of Taylor's theorem. There is a convenient representation of this remainder in terms of a contour integral:

$$\hat{h}_n(z) = \frac{1}{2\pi i} \int_{|z|=R} \frac{\hat{h}(\zeta)\, d\zeta}{\zeta^n \cdot (\zeta - z)}.$$

Taking $R > K_\varepsilon$, we obtain a bound on the remainder (see Ahlfors [1] for one dimension, and Hörmander [12] for the straightforward extension to several variables):

$$\begin{aligned} |\hat{h}_n(z)| &\leq \frac{|z|^n}{R^n} \cdot \frac{M_R \cdot R}{R - |z|} \\ \Rightarrow \quad \frac{|\hat{h}_n(z)|}{1 + |z|} &\leq \frac{|z|^n}{R^n} \cdot \frac{M_R \cdot R}{(R - |z|) \cdot (1 + |z|)}, \end{aligned}$$

where $M_R := sup_{|z|<R} |\hat{h}(z)|$. This is valid $\forall\, |z| < R$, and in particular, for any $x \in [-K, K]$. Now, again by our choice of γ_1 which recall, is independent of f, g, we have the bound, for any $f \in B_{n,L_n}$:

$$\begin{aligned} |\hat{f}(z)| &< ||D_1 f||_{L^1} \cdot (1 + |z|)^{-1} \cdot e^{|\mathrm{Im}\, z|} \\ &\leq L_n \cdot (1 + R)^{-1} \cdot e^R \quad \forall\, |z| < R, \end{aligned}$$

and therefore we can take

$$M_R := 2L_n \cdot (1+R)^{-1} \cdot e^R.$$

Taking R sufficiently large, say $R = 2K$, we have $(R - |x|) > |x|, \ \forall\, x \in [-K, K]$, and thus:

$$\begin{aligned}
\Big\| \frac{\hat{f} - \hat{g}}{1+|x|} \Big\|_{L^2[-K,K]} &= \Big\| \frac{\hat{h}}{1+|x|} \Big\|_{L^2[-K,K]} \\
&\leq \int_{-K}^{K} \frac{|x|^{2n}}{R^{2n}} \cdot \frac{M_R^2 \cdot R^2}{(R-|x|)^2(1+|x|)^2}\, dx \\
&\leq 2 \cdot \frac{M_R^2}{R^{2n-2}} \int_0^K x^{2n-4}\, dx \\
&= 2M_R^2 K^{-1} \cdot \left(\frac{K}{R}\right)^{2n-2} \cdot \frac{1}{2n-3} \\
&\leq 2M_R^2 K^{-1} \cdot \left(\frac{1}{2}\right)^{2n-2} \cdot \frac{1}{2n-3}.
\end{aligned}$$

Since the RHS is independent of $f, g \in B_{n,L_n}$, we have:

$$\begin{aligned}
\sup_{f,g \in B_{n,L_n}} ||f-g||_{L^2[0,1]} &\leq \sup_{f,g \in B_{n,L_n}} \Big\| \frac{\hat{f} - \hat{g}}{1+|x|} \Big\|_{L^2[-K,K]} \cdot C_{BL_1} + \varepsilon \\
&\leq 2C_{BL_1} \cdot M_R^2 K^{-1} \cdot \left(\frac{1}{2}\right)^{2n-2} \cdot \frac{1}{2n-3} + \varepsilon.
\end{aligned}$$

From here it is clear that we can allow L_n to increase linearly in n, while still maintaining an exponential convergence rate. This concludes the proof. □

Bibliography

[1] Ahlfors, L., Complex Analysis, 3rd Ed., McGraw–Hill, New York, 1966.

[2] Akhiezer, N., The Classical Moment Problem Hafner, New York 1965.

[3] Bertsimas, D., "The achievable region method in the optimal control of queueing systems; formulations, bounds and policies," *Queueing Systems and Applications*, 21, 3-4, 337-389, 1995.

[4] Bertsimas, D. and Popescu, I., "Optimal Inequalities in Probability Theory: A Convex Optimization Approach," submitted for publication, 2000.

[5] Bertsimas, D. and Popescu, I., "On the relation between option and stock prices: a convex optimization approach", to appear in *Operations Research*, 2001.

[6] Bertsimas, D., Popescu, I. and Sethuraman, J., "Moment problems and semidefinite programming", in Semidefinite programming H. Wolkovitz, ed., 469–509, 2000.

[7] Brezzi, F. and Fortin, M. Mixed and Hybrid Finite Element Methods, Springer–Verlag, New York, 1991.

[8] Dudley, R. M., Real Analysis and Probability, Pacific Grove, Brooks/Cole Pub. Co., San Franscisco, 1989.

[9] Folland, G., Introduction to Partial Differential Equations, Princeton University Press, Princeton, New Jersey, 1995.

[10] Fujisawa, K., Kojima, M. and Nakata, K., SDPA (Semidefinite Programming Algorithm) User's Manual, Version 4.10, Research Report on Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, Japan, 1998.

[11] Haviland, E. K., "On the momentum problem for distribution functions in more than one dimension II," *Amer. J. Math.*, 58, 164-168, 1936.

[12] Hörmander, L., An Introduction to Complex Analysis in Several Variables, American Elsevier Pub. Co., New York, 1973.

[13] Paraschivoiu, M. and Patera, A., "A Hierarchical Duality Approach to Bounds for the Outputs of Partial Differential Equations," *Comput. Methods Appl. Engrg.*, **158** 389-407, 1998.

[14] Paraschivoiu, M.; Peraire, J.; Patera, A.; " A Posteriori Finite Element Bounds for Linear–Functional Outputs of Elliptic Partial Differential Equations," *Comput. Meth. Appl. Mech. Engrg.*, **150**, 289-312, 1997.

[15] Peraire, J.; Patera, A. (1997) "Bounds for linear-functional outputs of coercive partial differential equations: Local indicators and adaptive refinement," *New Advances in Adaptive Computational Methods in Mechanics*, available at http://raphael.mit.edu/peraire/.

[16] Patera, A. "Spring 2001 Class Notes," Cambridge, MIT.

[17] Powers, V., Scheiderer, C.; "The Moment Problem for Non-Compact Semialgebraic Sets," Advances in Geometry **1** (2001), 71-88.

[18] Quarteroni, A.; Valli, A., Numerical Approximation of Partial Differential Equations, Springer–Verlag, Berlin, 1997.

[19] Reznick, B., "Some concrete aspects of Hilbert's 17th problem." Preprint, available at: http://www.math.uiuc.edu/Reports/reznick/98-002.html, 1999.

[20] Rudin, W., Functional Analysis, McGraw Hill, Boston, 1991.

[21] Schwerer, E., "A Linear Programming Approach to the Steady–State Analysis of Markov Processes," Palo Alto, CA: Ph.D. Thesis, Stanford University, 1996.

[22] Schmüdgen, K. "The K–moment problem for compact semi–algebraic sets," *Math. Ann.*, 289, 203–206, 1991.

[23] Strang, G. and Fix, G., An Analysis of the Finite Element Method, Prentice–Hall, Englewood Cliffs, NJ, 1973.

[24] Sturm, J., SeDuMi Semidefinite Software, can be downloaded from: http://www/unimaas.nl/ sturm.

[25] Vandenberghe, L.; Boyd, S., "Semidefinite Programming," *SIAM Review*, 38, 49–95, 1996.

[26] Watson, G. N., A Treatise on the Theory of Bessel Functions, 2nd Edition, The Macmillan Company, Cambridge, 1994.