# Optimization of univariate functions on bounded intervals by interpolation and semidefinite programming

E. de Klerk G. Elabwabi D. den Hertog

April 24, 2006

**Abstract**

We consider the problem of minimizing a univariate, real-valued function $f$ on an interval $[a, b]$. When $f$ is a polynomial, we review how this problem may be reformulated as a semidefinite programming (SDP) problem, and review how to extract all global minimizers from the solution of the SDP problem.

For general $f$, we approximate the global minimum by minimizing the Lagrange or Hermite interpolant of $f$ on the Chebyshev nodes using the SDP approach. We provide numerical results for a set of test functions.

**AMS subject classification:** 65D05, 65K05, 90C22

## 1 Introduction

Global univariate optimization of a function $f$ on an interval $[a, b]$ (the line search problem) is of continuing interest in mathematical programming, since most nonlinear programming codes solve sub-problems of this type at each iteration. Algorithms for the line search problem can be roughly divided into two main groups:

1. Heuristics that return a feasible solution without any guarantee of global optimality;
2. Algorithms that use global information about $f$, *e.g.* a Lipschitz constant, and an exhaustive search to find a global minimizer with guaranteed precision.

In the first category one has classical techniques like golden section search, and in the second methods like interval analysis branch-and-bound techniques (*e.g.* [6]).

In this paper we revisit the possibilities of doing line search via interpolation of $f$ on a grid of nodes $a \leq x_0 \leq \ldots \leq x_n \leq b$, and subsequently minimizing the interpolant. In particular, we use Lagrange or Hermite interpolation on the Chebyshev nodes, since this is known to be an almost optimal choice for the purpose of good uniform approximation. Here, Lagrange interpolation corresponds to line search without using derivatives, and Hermite interpolation to line search using first derivatives.

To subsequently minimize the interpolant, we use the modern semidefinite programming (SDP) approach (see *e.g.* [15, 5, 21]). In particular, we use the 'sampling' SDP formulations suggested by Löfberg and Parrilo [12]. Thus we do not need to calculate the coefficients of the interpolant — it is implicitly defined by its values at the nodes. Another important advantage of this approach is that it leads to rank 1 data matrices in the SDP formulation. As already pointed out by Löfberg and Parrilo [12], this structure in the SDP data can be exploited by interior point algorithms. This is done in the dual scaling solver DSDP [2] by Benson, Ye and Zhang. Using this approach we can find the global minimum of polynomials of degree up to 100 in about one second on a Pentium IV PC. In order to extract a global minimizer of the interpolant from the optimal solution of the SDP problem, we use a technique developed by Henrion and Lasserre [8], as adapted by Jibetean and Laurent [10]. This only involves an eigenvalue problem of the order of the number of global minimizers of the interpolant (thus very small in practice).

It may seem more straightforward to simply find all the real roots of the derivative of the interpolant. However, this classical problem is known to be quite difficult for polynomials of high degree. In §5 of his survey of 1997, Pan [16] reviews the fact that, for degree 50 or higher, all the well-known methods exhibit numerical instability (see also Goedecker [4]). The state-of-the-art has undoubtedly improved since then, but the fundamental difficulties remain. Moreover, in order to apply methods like the classical companion matrix method or the modern algorithm described in [23], one has to calculate the coefficients of the interpolant. This must be done with care as well, since obtaining these coefficients is equivalent to solving a linear system with an ill-conditioned Vandermonde coefficient matrix (see *e.g.* §3.5 in [19]). It is possible to evaluate the derivative of the Lagrange interpolant in a numerically stable way by using the Barycentric formula (see §9.3 in [3]). This could in principle be combined with root finding algorithms that require only evaluations of the polynomial in question, like the iterative Durand-Kerner (Weierstrass) method (see *e.g.* [16]). However, in spite of its good performance in practice, the global convergence of the Durand-Kerner method is not well understood.

The SDP approach therefore seems a reasonable one for our purposes. One may also note that computing all roots of the derivative of the interpolant is a more general problem that finding its global minimum on an interval.

We demonstrate numerically that the resulting method is fast and reliable for a set of test problems from Hansen *et al.* [7], and that some obvious alternatives to our approach fare much worse.

Our line search procedure also allows some possibilities to prove (near) global optimality, by using classical uniform error bounds for Lagrange and Hermite interpolation. Thus, if we know uniform upper bounds on any derivative(s) of $f$ a priori, we can give an interval of uncertainty for the global minimum of $f$ in $[a, b]$.

In summary, our approach should be seen as a line search heuristic, but one that gives additional information if global information on $f$ is available.

## Outline

We start by reviewing uniform error bounds for Lagrange and Hermite interpolation in Section 2. We then consider the problem of minimizing a polynomial on an interval $[a, b]$ in Section 3, and state the SDP

reformulation that we use for the purpose of computation. In particular, we consider the special case where the polynomial is a Lagrange or Hermite interpolant.

We also review the procedure for extracting the global minimizers of the polynomial from the optimal solution of the SDP problem. In Section 4 we give numerical results for our approach for a set of 20 test functions from Hansen *et al.* [7]. We also compare our approach to two other line search routines, and investigate the possibility of minimizing an interpolant by computing all roots of its derivative as opposed to the SDP approach.

# 2 Uniform error bounds for polynomial interpolation

In this section we review classical uniform error bounds for polynomial interpolation on an interval. In particular, we consider some polynomial $p \in \mathbb{R}[x]$ that interpolates a continuous function $f : [a, b] \to \mathbb{R}$ at given nodes $x_k \in [a, b]$ $(k = 0, 1, ..., n)$, and review some bounds on $\|f - p\|_{\infty,[a,b]}$, where

$$\|g\|_{\infty,[a,b]} := \sup_{x\in[a,b]} |g(x)|.$$

The book by Szabados and Vértesi [22] provides a detailed survey on error bounds for univariate interpolation.

## 2.1 Lagrange interpolation

Recall that the Lagrange interpolation polynomial of degree $n$, say $L_n(f)$, is the unique polynomial given by:

$$L_n(f)(x) := \sum_{i=0}^{n} f(x_i) l_{n,i}(x), \tag{2.1}$$

where

$$l_{n,i}(x) := \prod_{j\neq i} \frac{x - x_j}{x_i - x_j}$$

is called the fundamental Lagrange polynomial with respect to $x_i$. If $f(x)$ is $n + 1$ times continuously differentiable, then the interpolation error (remainder formula) is given by the following theorem.

**Theorem 2.1** *Suppose that $f \in C^{n+1}[a, b]$ and let $L_n(x)$ be given as (2.1). Then for any $x \in [a, b]$, one has*

$$f(x) - L_n(f)(x) = \prod_{i=0}^{n} (x - x_i) \frac{f^{(n+1)}(\zeta)}{(n + 1)!} \text{ for some } \zeta \in [a, b]. \tag{2.2}$$

If $[a, b] = [-1, 1]$, then it is well-known that the uniform norm of the product $\prod_{i=0}^{n}(x - x_i)$ is minimized if we choose the $x_i$'s as the roots of the Chebyshev polynomial (of the first kind) of degree $n+1$. Recall that the Chebyshev polynomials (of the first kind) are defined as:

$$T_j(x) := \cos(j \arccos x) \qquad j = 0, 1, \ldots \tag{2.3}$$

The roots of $T_{n+1}$ (also called the Chebyshev nodes) are therefore given by

$$x_i = \cos\left(\frac{(2i+1)\pi}{2(n+1)}\right) \quad i = 0, \ldots, n, \tag{2.4}$$

and one has

$$\left\| \prod_{i=0}^{n} (x - x_i) \right\|_{\infty,[-1,1]} = 2^{-n}.$$

**Corollary 2.1** *If $f \in C^{n+1}[-1,1]$, $L_n(x)$ is given as in (2.1), and $x_0, \ldots, x_n$ are the Chebyshev nodes defined in (2.4), one has:*

$$\|f - L_n(f)\|_{\infty,[-1,1]} \leq \frac{\|f^{(n+1)}\|_{\infty,[-1,1]}}{2^n (n+1)!}. \tag{2.5}$$

If $[a,b] \neq [-1,1]$, the one simply does a linear transformation to obtain the Chebyshev nodes on $[a,b]$:

$$\frac{b-a}{2} x_k + \frac{a+b}{2} \text{ for } k = 0, 1, ..., n$$

By using this linear transformation, one easily adapts the last result to hold for any interval $[a,b]$.

**Corollary 2.2** *Assume that $L_n(f)(x)$ is the Lagrange polynomial that is based on the $n+1$ Chebyshev nodes on $[a,b]$. If $f \in C^{n+1}[a,b]$ then*

$$\|f - L_n(f)\|_{\infty,[a,b]} \leq \frac{2(b-a)^{n+1}}{4^{n+1}(n+1)!} \|f^{(n+1)}\|_{\infty,[a,b]}.$$

Assume now that $f$ is a function that can be analytically extended to a function that is analytic (holomorphic) in a neighborhood of $[-1,1]$ in the complex plane. Then one has the following uniform error bound.

**Theorem 2.2** *If $f$ is analytic on and inside an ellipse in the complex plane with foci $\pm 1$ and axis lengths $2L$ and $2l$, then*

$$\|f - L_n(f)\|_{\infty,[-1,1]} \leq C n^{-(l+L)}$$

*for some constant $C > 0$.*

For such functions the convergence rate of the sequence $\|f - L_n(f)\|_{\infty,[-1,1]}$ therefore therefore depends on the size of the region of analyticity (see the discussion by Berrut and Trefethen [3], §6).

If $f$ only has a fixed degree of smoothness, one may use the Jackson theorem to derive error bounds. To this end, we also require a well-known bound on the Lebesgue constant for the Chebyshev nodes.

**Definition 2.1** *(Lebesgue constant) The Lebesgue constant at a set of nodes $\{x_0, ..., x_n\} \subset [a,b]$ is defined as*

$$\Lambda_n(x_0, ..., x_n) = \max_{x \in [a,b]} \sum_{k=0}^{n} |l_{n,k}(x)|,$$

*where $l_{n,k}(x) = \Pi_{j \neq k} \frac{x - x_k}{x_k - x_j}$ as before.*

The Lebesgue constant allows us to compare the error of Lagrange interpolation to the error for the best possible approximation using degree $n$ polynomials:

$$E_n := \inf_{p \in \mathbb{R}[x], \text{degree}(p) \leq n} \|f - p\|_{\infty,[a,b]}.$$

**Theorem 2.3 (See *e.g.* Theorem 3.1 in Powell [17])** *Let $f \in C[a,b]$, and $L_n(f)$ be the Lagrange interpolating polynomial (2.1) at the set of nodes $x_0, ..., x_n$. Then*

$$\|f - L_n(f)\|_{\infty,[a,b]} \leq (1 + \Lambda_n(x_0, ..., x_n))E_n.$$

If $\{x_0, ..., x_n\}$ is the set of Chebyshev nodes, then an upper bound on $\Lambda_n$ is given by the following lemma.

**Lemma 2.1 (see *e.g.* Theorem 4.5 in Rivlin [20])** *If $\{x_0, ..., x_n\}$ is the set of Chebyshev nodes on the interval $[-1, 1]$ then*

$$\Lambda_n(x_0, ..., x_n) < \frac{2}{\pi} \ln(1 + n) + 1.$$

The famous Jackson theorem gives an error estimate for the best approximating polynomial.

**Theorem 2.4 (Jackson (see *e.g.* Theorem 1.5 in Rivlin [20]))** *If $f \in C^r[-1, 1]$ and $n > r \geq 0$, then*

$$E_n \leq 6^{r+1} e^r (1 + r)^{-1} n^{-r} \omega_r \left( \frac{1}{n - r} \right) \qquad (r = 0, \ldots, n - 1),$$

*where $\omega_r$ is the modulus of continuity of $f^{(r)}$ ($r = 0$ corresponds to $f$):*

$$\omega_r(\delta) = \sup_{x,y \in [-1,1]} (|f^{(r)}(x) - f^{(r)}(y)| : \ |x - y| \leq \delta).$$

Using Theorem 2.3 and Lemma 2.1, we have the following corollary of the Jackson theorem.

**Corollary 2.3** *If $f \in C^r[-1, 1]$ and if $n > r \geq 0$, then the interpolation error using Chebyshev nodes satisfies*

$$\|f - L_n(f)\|_{\infty,[-1,1]} \leq 6^{r+1} e^r (1 + r)^{-1} n^{-r} \left( \frac{2}{\pi} \ln(1 + n) + 1 \right) \omega_r \left( \frac{1}{n - r} \right).$$

Note, in particular, that the sequence $\|f - L_n(f)\|_{\infty,[-1,1]}$ always converges under the mild assumption

$$\ln(n)\omega_0 \left( \frac{1}{n} \right) = o(1),$$

where $\omega_0$ is the modulus of continuity of $f$, as before. This holds, for example, for Lipschitz continuous functions.

### 2.2 Hermite interpolation

The Hermite interpolating polynomial $H_n(f)$ associated with a given $f : [a, b] \to \mathbb{R}$ and a given set of nodes $a \leq x_0 < \ldots < x_n \leq b$, is defined as the unique polynomial of degree at most $2n + 1$ that satisfies

$$\begin{aligned} H_n(f)(x_i) &= f(x_i) \quad (i = 0, \ldots, n), \\ H_n(f)'(x_i) &= f'(x_i) \quad (i = 0, \ldots, n). \end{aligned}$$

The following theorem gives the interpolation error.

**Theorem 2.5** *Suppose that $f \in C^{2(n+1)}[a, b]$, then for any $x \in [a, b]$ one has*

$$f(x) - H_n(f)(x) = \prod_{i=0}^{n} (x - x_i)^2 \frac{f^{(2n+2)}(\zeta)}{(2n + 2)!} \text{ for some } \zeta \in [a, b]. \tag{2.6}$$

**Corollary 2.4** *Assume that $[a, b] = [-1, 1]$ and the nodes $x_i$ $(i = 0, \ldots, n)$ are the zeros of Chebyshev polynomial $T_{n+1}$. One has:*

$$\|f - H_n(f)\|_{\infty,[-1,1]} \leq \frac{\|f^{(2n+2)}\|_{\infty,[-1,1]}}{4^n(2n + 2)!}. \tag{2.7}$$

## 3 Optimization of polynomials on the interval

In this section we consider the optimization problem

$$\underline{p} := \min\{p(x) : \quad x \in [a, b]\}, \tag{3.8}$$

where $p$ is a univariate polynomial. We show how to reformulate this problem as an SDP problem in a way suitable for computation, and how to extract the global minimizers from the optimal solution of the SDP problem. In addition, we consider the special case where $p$ is a Lagrange or Hermite interpolant.

### 3.1 SDP formulation

Since $\underline{p}$ is the largest value for which the polynomial $p(x) - \underline{p}$ is nonnegative for all $x \in [a, b]$, problem (3.8) can be rewritten as

$$\underline{p} = \max\{\tau : \quad p(x) - \tau \geq 0, \quad \forall x \in [a, b]\}. \tag{3.9}$$

Following Nesterov [15], we obtain an SDP reformulation of problem (3.9) by using an old theorem by Lucàcs that characterizes polynomials that are nonnegative on $[a, b]$. For a discussion of this classical result, see Powers and Reznick [18].

**Theorem 3.1 (Lucàcs)** *Let $p(x)$ be a univariate polynomial of degree $2m$. Then $p(x)$ is nonnegative on the interval $[a, b]$ if and only if it has the following representation:*

$$p(x) = q_1^2(x) + (x-a)(b-x)q_2^2(x), \tag{3.10}$$

*for some polynomials $q_1(x)$ of degree $m$ and $q_2(x)$ of degree $m-1$.*
*Moreover, if the degree of $p$ is $2m+1$ then $p(x)$ is nonnegative on $[a, b]$ if and only if it has the following representation:*

$$p(x) = (x-a)q_1^2(x) + (b-x)q_2^2(x), \tag{3.11}$$

*for some polynomials $q_1(x)$ and $q_2(x)$ of degree $m$.*

Note that the Lucàcs theorem is a refinement of the more general Putinar representation theorem for the univariate case, in the sense that we have information on the degrees of the polynomials $q_1$ and $q_2$. Thus the general (multivariate) theory of Lasserre [13] applies, and we will use results from this theory without giving proofs. Where applicable, we will refer to proofs in the recent survey by Laurent [11].

To fix our ideas, we first assume that $p$ has degree $2m$, and we denote by $B_m(x)$ a basis for the space of polynomials of degree at most $m$. For the purpose of computation we will use $B_m(x) := [T_0(x), \ldots, T_m(x)]^T$, but the discussion here is independent of the choice of basis.

Note that, if $q$ is sum-of-squares polynomial of degree $2m$, then

$$q(x) = B_m(x)^T X B_m(x)$$

for some matrix $X \succeq 0$ of order $m+1$. Such a representation of a sum-of-squares polynomial is sometimes called the Gram matrix representation. Using (3.10) and the Gram matrix representation for $q_1$ and $q_2$, the condition $p(x) - \tau \geq 0, \ \ \forall x \in [a, b]$ can now be reformulated as :

$$\begin{aligned} p(x) - \tau &= q_1(x) + (x-a)(b-x)q_2(x), \ \ q_1, q_2 \text{ are sum-of-squares polynomials} \\ &= B_m(x)^T X_1 B_m(x) + (x-a)(b-x)B_{m-1}(x)^T X_2 B_{m-1}(x), \end{aligned} \tag{3.12}$$

where $X_1, X_2$ are positive semidefinite matrices.

### 3.1.1 Standard SDP formulation

The usual way to convert this equation to a linear matrix equality is to use the standard monomial basis

$$B_m(x) = [1, x, \ldots, x^m]^T,$$

and then equate the coefficients of the polynomials on the two sides of the equation (3.12) [13, 15]. If we denote

$$p(x) = \sum_{\alpha} p_\alpha x^\alpha,$$

we obtain the standard SDP reformulation of problem (3.8):

$$\underline{p} = \max_{\tau, X_1, X_2} \tau$$

subject to

$$\begin{aligned} p_0 - \tau &= (X_1)_{00} - ab(X_2)_{00} \\ p_\alpha &= \sum_{i+j=\alpha} (X_1)_{ij} - ab \sum_{i+j=\alpha} (X_2)_{ij} + (a+b) \sum_{i+j=\alpha-1} (X_2)_{ij} - \sum_{i+j=\alpha-2} (X_2)_{ij}, \quad (\alpha = 1, \ldots, 2m), \end{aligned}$$

where $X_1$ and $X_2$ are positive semidefinite matrices of order $m+1$ and $m$ respectively. The dual SDP problem takes the form

$$\underline{p} = \min \sum_\alpha p_\alpha y_\alpha \tag{3.13}$$

subject to

$$y_0 = 1, \quad Y := \begin{bmatrix} y_0 & y_1 & y_2 & \cdots & y_m \\ y_1 & y_2 & & & \vdots \\ y_2 & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ y_m & \cdots & \cdots & \cdots & y_{2m} \end{bmatrix} \succeq 0,$$

$$-(ab) \begin{bmatrix} y_0 & y_1 & y_2 & \cdots & y_m \\ y_1 & y_2 & & & \vdots \\ y_2 & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ y_m & \cdots & \cdots & \cdots & y_{2m-2} \end{bmatrix} + (a+b) \begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_{m+1} \\ y_2 & y_3 & & & \vdots \\ y_3 & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ y_{m+1} & \cdots & \cdots & \cdots & y_{2m-1} \end{bmatrix} - \cdots$$

$$\cdots - \begin{bmatrix} y_2 & y_3 & y_4 & \cdots & y_{m+2} \\ y_3 & y_4 & & & \vdots \\ y_4 & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ y_{m+2} & \cdots & \cdots & \cdots & y_{2m} \end{bmatrix} \succeq 0.$$

From a computational point of view, the drawback of this formulation is that the dual feasible solutions are positive semidefinite Hankel matrices. Such matrices are always ill-conditioned, as the following theorem shows.

**Theorem 3.2 (Beckermann [1])** *The condition number of any positive definite Hankel matrix of order $n$ is bounded from below by $\frac{3.21^{n-1}}{16n}$.*

Any dual, or primal-dual interior point algorithm requires the computation of the inverse of the dual matrix. If the degree of the polynomial is, say, $2m = 80$, then this involves obtaining inverses of matrices with condition number greater than $10^{17}$.

### 3.1.2 'Sampling' SDP formulation

We will use an alternative SDP formulation, due to Löfberg and Parrilo [12], which uses the fact a polynomial of degree $2m$ is uniquely determined by the values it takes at $2m+1$ distinct 'sampling' points. For computation we will use the roots of the Chebyshev polynomial $T_{2m+1}$ as these 'sampling' points, but the discussion here is valid for any set $a \leq x_0 < \ldots < x_{2m} \leq b$. Thus we obtain the SDP problem

$$\underline{p} = \max_{\tau, X_1, X_2} \tau \tag{3.14}$$

subject to

$$p(x_i) - \tau = B_m(x_i)^T X_1 B_m(x_i) + (x_i - a)(b - x_i) B_{m-1}(x_i)^T X_2 B_{m-1}(x_i) \qquad (i = 0, \ldots, 2m),$$

where $X_1, X_2$ are positive semidefinite matrices. The dual problem takes the form

$$\min_{\kappa \in \mathbb{R}^{2m+1}} \sum_{i=0}^{2m} p(x_i)\kappa_i \tag{3.15}$$

subject to

$$\begin{aligned} \sum_{i=0}^{2m} \kappa_i &= 1 \\ \sum_{i=0}^{2m} \kappa_i B_m(x_i) B_m(x_i)^T &\succeq 0 \\ \sum_{i=0}^{2m} (x_i - a)(b - x_i)\kappa_i B_{m-1}(x_i) B_{m-1}(x_i)^T &\succeq 0. \end{aligned} \tag{3.16}$$

Notice that the dual variable $\kappa_i$ corresponds to the value of the fundamental Lagrange polynomial $l_{2m,i}$, since

$$p(x) = \sum_{i=0}^{2m} p(x_i) l_{2m,i}(x), \text{ and } \sum_{i=0}^{2m} l_{2m,i}(x) = 1 \quad \forall x.$$

Also note that the constraint matrices in the SDP are rank one matrices of the form $B_m(x_i)B_m(x_i)^T$. As observed by Löfberg and Parrilo [12], the 'rank one structure' can be exploited by some interior point solvers, like the implementation DSDP [2] of the dual scaling method. Note that one still has the Hankel matrix structure in (3.16) if $B_m$ is chosen as the monomial basis. For this reason we will use an orthogonal basis (the Chebyshev basis) instead for the purposes of computation.

## 3.2 If $p$ is a Lagrange interpolant

If $p$ is the Lagrange interpolant $L_{2m}(f)$ of some function $f$ at $x_i$ $(i = 0, \ldots, 2m)$, then $p(x_i) = f(x_i)$, and the following SDP problem yields the minimum of $L_{2m}(f)$.

$$\min_{x \in [a,b]} L_{2m}(f)(x) = \max_{\tau, X_1, X_2} \tau \tag{3.17}$$

subject to

$$f(x_i) - \tau = B_m(x_i)^T X_1 B_m(x_i) + (x_i - a)(b - x_i) B_{m-1}(x_i)^T X_2 B_{m-1}(x_i) \qquad (i = 0, \ldots, 2m),$$

where $X_1, X_2$ are positive semidefinite matrices.

Note that we avoid the potentially numerically unstable process of calculating the coefficients of $L_{2m}(f)$.

### 3.3 If $p$ is a Hermite interpolant

If $p$ is the Hermite interpolating polynomial of some function $f$ at $m+1$ Chebyshev nodes, then $p$ is always of odd degree $(2m+1)$. Thus, by Theorem 3.1, the minimal value of $p$ on $[a, b]$ is given by the maximal value of $\tau$ such that

$$p(x) - \tau = (x - a) B_m(x)^T X_1 B_m(x) + (b - x) B_m(x)^T X_2 B_m(x)$$

for some positive semidefinite matrices $X_1$ and $X_2$. Differentiating this equation yields

$$p'(x) = 2(x - a) B'_m(x)^T X_1 B_m(x) + B_m(x)^T X_1 B_m(x) + 2(b - x) B'_m(x)^T X_2 B_m(x) - B_m(x)^T X_2 B_m(x).$$

Thus it is easy to enforce the conditions $p(x_i) = f(x_i)$ and $p'(x_i) = f'(x_i)$ at $m+1$ Chebyshev nodes via the sampling approach. We obtain the following SDP problem that yields the minimum of the Hermite interpolant:

$$\min_{x \in [a,b]} H_{m+1}(f)(x) = \max_{\tau, X_1, X_2} \tau \tag{3.18}$$

subject to

$$\begin{aligned} f(x_i) - \tau &= (x_i - a) B_m(x_i)^T X_1 B_m(x_i) + (b - x_i) B_m(x_i)^T X_2 B_m(x_i) \\ f'(x_i) &= 2(x_i - a) B'_m(x_i)^T X_1 B_m(x_i) + B_m(x_i)^T X_1 B_m(x_i) + 2(b - x_i) B'_m(x_i)^T X_2 B_m(x_i) \\ &\quad - B_m(x_i)^T X_2 B_m(x_i), \end{aligned}$$

where $i = 0, \ldots, m$, $X_1 \succeq 0$ and $X_2 \succeq 0$.

Note that this formulation involves rank 1 as well as rank 2 constraint matrices. Also note that we do not have to compute the coefficients of the Hermite interpolating polynomial.

### 3.4 Extracting global minimizers

Here we describe how to extract a global minimizer of problem (3.9) from an optimal solution of the dual SDP formulation (3.15). We only treat the case where $p$ has even degree, the odd degree case being similar. The methodology outlined here is a special case of a procedure due to Henrion and Lasserre [8], as adapted by Jibetean and Laurent [10], and is described in detail in the recent survey by Laurent [11].

We denote by $V = \{v_1, \ldots, v_k\}$ the set of global minimizers of problem (3.8). If we view $V$ as a finite variety, then the associated ideal is

$$I(V) := \{h \in \mathbb{R}[x] \ : \ h(v_i) = 0 \ \forall \ v_i \in V\}.$$

Now the basic procedure is as follows:

1. We obtain a basis for the quotient vector space $\mathbb{R}[x]/I(V)$ from the optimal solution of the SDP problem (3.14). Note that the quotient vector space has dimension $|V|$.

2. We use this basis together with the Theorem 3.3 below to find $V$.

**Theorem 3.3 (Stickelberger (*cf.* Theorem 21 in [11]))** *Consider the linear operator*

$$M_h : \mathbb{R}[x]/I(V) \mapsto \mathbb{R}[x]/I(V)$$

*defined by*

$$M_h(f \ mod \ I(V)) \to hf \ mod \ I(V).$$

*Then, for $h(x) = x$, the spectrum of $M_h$ is $V$.*

The next lemma shows how to obtain a basis for $\mathbb{R}[x]/I(V)$. This is needed to give a matrix representation of the linear operator in Theorem 3.3, so that we may compute its spectrum by solving an eigenvalue problem on a matrix of order $|V|$.

**Lemma 3.1 (*cf.* Lemma 24 in [11])** *Consider an optimal solution $\kappa$ of the SDP problem (3.15). Denote a maximal nonsingular principal submatrix of the matrix $\sum_{i=0}^{2m} \kappa_i B_m(x_i) B_m(x_i)^T \succeq 0$ by $\sum_{i=0}^{2m} \kappa_i \tilde{B}_m(x_i) \tilde{B}_m(x_i)^T \succ 0$, where the elements of $\tilde{B}_m(x_i)$ form the appropriate subset of the elements of $B_m(x_i)$. Then $\tilde{B}_m(x)$ is a basis for $\mathbb{R}[x]/I(V)$.*

We now describe the steps of the extraction procedure. All that remains is to give a matrix representation of the linear operator in Theorem 3.3 in terms of the basis $\tilde{B}_m(x)$ of $\mathbb{R}[x]/I(V)$ described in Lemma 3.1. This matrix representation is derived in §2.5 of Jibetean and Laurent [10] for the standard monomial basis. Since we will use the Chebyshev basis $B_m(x) = [T_0(x), \ldots, T_m(x)]^T$ for computation, we describe the matrix representation for this basis. To this end, we require the identities:

$$T_0(x) = 1, \quad T_1(x) = x, \quad xT_i(x) = \frac{1}{2}\left(T_{i-1}(x) + T_{i+1}(x)\right) \quad (i = 1, \ldots, m-1).$$

We will also use the following notation: for a matrix $M$ and index sets $\mathcal{I}$ and $\mathcal{J}$, $M_{\mathcal{I},\mathcal{J}}$ denotes the principal submatrix of $M$ with rows indexed by $\mathcal{I}$ and columns indexed by $\mathcal{J}$.

**Extraction procedure**

1. Compute an optimal solution $\kappa$ of the SDP problem (3.15).

2. Find, using a greedy algorithm, a maximal nonsingular principal submatrix of the matrix

$$M := \sum_{i=0}^{2m} \kappa_i B_m(x_i) B_m(x_i)^T \succeq 0.$$

   Let $\mathcal{A}$ denote the index set corresponding to the columns of this submatrix.

3. For each $\beta \in \mathcal{A}$, define

$$u_\beta := \begin{cases} M_{\mathcal{A},\{1\}} & \text{if } \beta = 0 \\ \frac{1}{2}\left(M_{\mathcal{A},\{\beta-1\}} + M_{\mathcal{A},\{\beta+1\}}\right) & \text{if } \beta \geq 1. \end{cases}$$

   Denote by $U$ the matrix with columns $u_\beta$ ($\beta \in \mathcal{A}$).

4. Compute the eigenvalues of the matrix $(M_{\mathcal{A},\mathcal{A}})^{-1}U$ to obtain the set of global minimizers $V$.

Note that, in step 3 of the procedure, we implicitly assume that $\beta < m$ for all $\beta \in \mathcal{A}$; otherwise the reference to column $\beta + 1$ is nonsensical. In other words, we assume that the last column of the matrix $M$ is not used in forming the maximum nonsingular principal submatrix $M_{\mathcal{A},\mathcal{A}}$. (This assumption always holds in practice.)

# 4 Numerical results

In this section we give numerical results for minimizing Lagrange and Hermite interpolants via the SDP approach. We use a set of 20 test functions from Hansen *et al.* [7]. This test set is reproduced in Table 1 (an error for entry 16 has been corrected in the table, and some values are given with more digits of accuracy). Note that these test problems are formulated as maximization problems.

We will first give results for Lagrange interpolation on the Chebyshev nodes. This corresponds to line search without using derivative information. Subsequently, we give results for Hermite interpolation, where first derivatives are used. Finally, we will compare these results to the performance of two off-the-shelf line search routines on the same set of test problems.

All computation was done on a PC with an Intel x86 processor (2128 Mhz) and with 512 MB memory. The SDP solver DSDP 5.0 [2] was used to solve all SDP problems. We used the Matlab interface for DSDP together with Matlab 7.0.4.

## 4.1 Lagrange interpolation — line search without derivatives

Table 2 gives the relative error

$$\frac{|\bar{f} - \bar{L}_n(f)|}{1 + |\bar{f}|}$$

for 20 test functions from Hansen *et al.* [7], and number of Chebyshev interpolation points from 11 to 101. Here $\bar{f} := \max_{[a,b]} f(x)$, etc. Note that, already for 81 interpolation points, the largest relative error is $1.27 \times 10^{-5}$.

The use of the Chebyshev basis $B_m(x) = [T_0(x), \ldots, T_m(x)]^T$ in (3.17) is essential for the computation — if one uses the monomial basis instead, then DSDP typically encounters numerical difficulties for more than 30 interpolation nodes.

If we use information on the derivatives of the test functions, we can give error bounds by using the results in Section 2. For example, for function 19 one trivially has $\|f^{(n)}\|_{\infty,[0,6.5]} \leq 3^{n-1}$ for all $n$. Using

| Problem | Function $f$ | $[a,b]$ | $\max_{x\in[a,b]} f(x)$ | Global maximizer(s) |
|---|---|---|---|---|
| 1 | $-\frac{1}{6}x^6+\frac{52}{25}x^5-\frac{39}{80}x^4$ $-\frac{71}{10}x^3+\frac{79}{20}x^2+x-\frac{1}{10}$ | [-1.5,11] | 29,763.233 | 10 |
| 2 | $-\sin x-\sin\frac{10}{3}x$ | [2.7,7.5] | 1.899599 | 5.145735 |
| 3 | $\sum_{k=1}^{5} k\sin((k+1)x+k)$ | [-10,10] | 12.03124 | -6.7745761<br>-0.491391<br>5.791785 |
| 4 | $(16x^2-24x+5)e^{-x}$ | [1.9,3.9] | 3.85045 | 2.868034 |
| 5 | $(-3x+1.4)\sin 18x$ | [0,1.2] | 1.48907 | 0.96609 |
| 6 | $(x+\sin x)e^{-x^2}$ | [-10,10] | 0.824239 | 0.67956 |
| 7 | $-\sin x-\sin\frac{10}{3}x$ $-\ln x+0.84x-3$ | [2.7,7.5] | 1.6013 | 5.19978 |
| 8 | $\sum_{k=1}^{5} k\cos((k+1)x+k)$ | [-10,10] | 14.508 | - 7.083506<br>-0.800321<br>5.48286 |
| 9 | $-\sin x-\sin\frac{2}{3}x$ | [3.1,20.4] | 1.90596 | 17.039 |
| 10 | $x\sin x$ | [0,10] | 7.91673 | 7.9787 |
| 11 | $2\cos x+\cos 2x$ | [-1.57,6.28] | 1.5 | 2.09439<br>4.18879 |
| 12 | $-\sin^3 x-\cos^3 x$ | [0,6.28] | 1 | $\pi$<br>4.712389 |
| 13 | $x^{2/3}+(1-x^2)^{1/3}$ | [0.001,0.99] | 1.5874 | $1/\sqrt{2}$ |
| 14 | $e^{-x}\sin 2\pi x$ | [0,4] | 0.788685 | 0.224885 |
| 15 | $(-x^2+5x-6)/(x^2+1)$ | [-5,5] | 0.03553 | 2.41422 |
| 16 | $-2(x-3)^2-e^{-x^2/2}$ | [-3,3] | -0.0111090 | 3 |
| 17 | $-x^6+15x^4-27x^2-250$ | [-4,4] | -7 | -3<br>3 |
| 18 | $\begin{cases} -(x-2)^2, & x\le 3; \\ -2\ln(x-2)-1, & \text{otherwise.} \end{cases}$ | [0,6] | 0 | 2 |
| 19 | $\sin 3x-x-1$ | [0,6.5] | 7.81567 | 5.87287 |
| 20 | $(x-\sin x)e^{-x^2}$ | [-10,10] | 0.0634905 | 1.195137 |

Table 1: Twenty test functions from P. Hansen et al. [7].

Theorem 2.2, one sees that 25 Chebyshev nodes already guarantee an absolute error of less than $10^{-8}$ in this case.

The solution times of the SDP problems (3.17) that were solved to obtain Table 2 are given in Table 3. Notice that, for a given number of nodes, the SDP solution time varies only slightly for the different test problems. Notice also, that up to an interpolant of degree 80, the solution time is less than a second.

Table 4 gives the relative error

$$\max_i \frac{|x_i^* - v_i|}{1+|x_i^*|}$$

for the 20 test functions and 81 interpolation points, where $x_i^*$ is the $i$th largest global maximizer of the test

| | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3.77e-9 | 4.18e-9 | 2.29e-10 | 5.30e-9 | 7.45e-9 | 5.79e-9 | 2.44e-9 | 1.02e-8 | 1.04e-8 | 8.88e-9 |
| 2 | 1.14e-3 | 1.01e-7 | 1.06e-7 | 1.18e-7 | 1.16e-7 | 1.18e-7 | 1.18e-7 | 1.18e-7 | 1.19e-7 | 1.12e-7 |
| 3 | 5.09e-1 | 1.29e-2 | 7.50e-2 | 4.76e-2 | 5.52e-2 | 1.53e-2 | 8.50e-5 | 3.15e-8 | 4.94e-8 | 4.79e-8 |
| 4 | 1.28e-7 | 1.40e-7 | 1.34e-7 | 1.40e-7 | 1.39e-7 | 1.43e-7 | 1.43e-7 | 1.35e-7 | 1.43e-7 | 1.44e-7 |
| 5 | 8.01e-2 | 2.80e-5 | 5.38e-8 | 1.01e-6 | 1.01e-6 | 1.01e-6 | 1.01e-6 | 1.01e-6 | 1.01e-6 | 1.01e-6 |
| 6 | 4.51e-1 | 2.98e-1 | 7.54e-2 | 5.70e-3 | 1.01e-3 | 2.15e-4 | 1.45e-5 | 6.04e-7 | 2.16e-7 | 2.16e-7 |
| 7 | 2.64e-3 | 2.89e-6 | 2.89e-6 | 2.87e-6 | 2.88e-6 | 2.89e-6 | 2.89e-6 | 2.89e-6 | 2.90e-6 | 2.89e-6 |
| 8 | 4.52e-2 | 2.12e-1 | 2.87e-1 | 1.44e-2 | 8.22e-2 | 2.19e-2 | 1.44e-4 | 6.69e-7 | 5.09e-7 | 5.10e-7 |
| 9 | 3.11e-2 | 4.64e-7 | 3.83e-7 | 3.82e-7 | 3.80e-7 | 3.83e-7 | 3.83e-7 | 3.76e-7 | 3.79e-7 | 3.83e-7 |
| 10 | 3.24e-4 | 3.01e-7 | 3.05e-7 | 2.99e-7 | 2.95e-7 | 2.98e-7 | 2.96e-7 | 3.08e-7 | 3.08e-7 | 2.98e-7 |
| 11 | 1.37e-2 | 2.17e-8 | 8.48e-10 | 1.58e-8 | 5.32e-9 | 5.15e-9 | 4.32e-9 | 1.49e-8 | 5.65e-9 | 8.79e-9 |
| 12 | 4.35e-3 | 1.39e-7 | 1.85e-9 | 1.80e-8 | 7.55e-9 | 1.38e-9 | 2.29e-9 | 5.74e-9 | 1.04e-8 | 3.96e-9 |
| 13 | 2.33e-4 | 1.11e-6 | 1.47e-8 | 1.08e-7 | 5.91e-8 | 3.00e-8 | 1.58e-8 | 2.55e-9 | 1.56e-8 | 8.87e-8 |
| 14 | 1.66e-2 | 1.91e-5 | 2.13e-7 | 2.08e-7 | 2.14e-7 | 2.12e-7 | 2.14e-7 | 2.16e-7 | 2.12e-7 | 2.15e-7 |
| 15 | 6.17e-1 | 7.15e-2 | 5.03e-3 | 1.32e-3 | 9.68e-5 | 1.66e-5 | 3.38e-6 | 6.97e-9 | 6.13e-8 | 5.78e-9 |
| 16 | 1.21e-3 | 8.92e-8 | 1.05e-8 | 6.41e-9 | 5.91e-9 | 9.48e-10 | 1.73e-9 | 4.70e-10 | 3.04e-9 | 4.40e-9 |
| 17 | 8.69e-10 | 1.58e-8 | 1.06e-8 | 1.08e-8 | 9.87e-9 | 2.16e-8 | 1.12e-9 | 1.68e-9 | 3.87e-9 | 6.31e-9 |
| 18 | 1.59e-2 | 2.29e-3 | 6.80e-4 | 2.59e-4 | 1.14e-4 | 5.44e-5 | 2.67e-5 | 1.27e-5 | 5.63e-6 | 1.81e-6 |
| 19 | 1.01e-2 | 2.63e-7 | 5.01e-7 | 5.14e-7 | 5.05e-7 | 4.96e-7 | 5.09e-7 | 5.08e-7 | 1.00e-5 | 5.11e-7 |
| 20 | 5.88e-2 | 7.44e-3 | 4.24e-3 | 6.48e-3 | 1.40e-4 | 1.57e-4 | 1.04e-5 | 1.38e-7 | 6.35e-9 | 2.36e-8 |

Table 2: Relative errors in approximating the maximum of the 20 test functions with the maximum of the Lagrange interpolant on $n+1$ Chebyshev nodes. The columns are indexed by the degree of the Lagrange interpolant ($n$).

function, and $v_i$ is the $i$th largest global maximizer of the Lagrange polynomial. The $v_i$ values were obtained via the extraction procedure described in Section 3.4.

The errors for functions 6 and 9 seem relatively large, but this is only because the global maximizers are given only to 5 significant digits in Table 1. For function 18 the error is indeed relatively large: the global maximizer is 2 and the global maximizer of the degree 80 interpolant was 1.99979 (to 6 digits). The relatively large error for function 18 is due to the relatively poor uniform approximation by Lagrange interpolation (see Table 2). The poor convergence of the Lagrange interpolantion is in turn due to the fact that function 18 is only continuously differentiable (*i.e.* the higher order derivatives are not continuous at $x = 3$).

**Minimizing $L_n(f)$ using root-finding routines**

As an alternative to minimizing the Lagrange interpolant $L_n(f)$ by solving the SDP problem (3.17), we also tried to find all real roots of $(L_n(f))'$ using two root finding packages:

- The 'roots' function of Matlab 7.0.4, that is an implementation of the companion matrix method;
- The Matlab package 'MultRoot' by Zeng [23], that is an implementation of the algorithm described in

| | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 6.25e-2 | 7.81e-2 | 1.25e-1 | 1.71e-1 | 2.65e-1 | 3.43e-1 | 5.00e-1 | 6.71e-1 | 8.90e-1 | 1.25 |
| 2 | 3.12e-2 | 7.81e-2 | 1.09e-1 | 2.03e-1 | 2.50e-1 | 3.12e-1 | 4.68e-1 | 6.09e-1 | 8.90e-1 | 1.28 |
| 3 | 4.68e-2 | 6.25e-2 | 1.25e-1 | 1.87e-1 | 2.18e-1 | 3.12e-1 | 4.53e-1 | 6.56e-1 | 8.90e-1 | 1.26 |
| 4 | 4.68e-2 | 7.81e-2 | 1.25e-1 | 2.03e-1 | 2.65e-1 | 3.28e-1 | 5.15e-1 | 6.56e-1 | 9.68e-1 | 1.34 |
| 5 | 4.68e-2 | 6.25e-2 | 1.09e-1 | 1.71e-1 | 2.50e-1 | 3.28e-1 | 5.00e-1 | 6.40e-1 | 9.37e-1 | 1.26 |
| 6 | 3.12e-2 | 6.25e-2 | 1.09e-1 | 1.56e-1 | 2.50e-1 | 3.12e-1 | 4.53e-1 | 6.71e-1 | 8.90e-1 | 1.21 |
| 7 | 3.12e-2 | 7.81e-2 | 1.25e-1 | 1.71e-1 | 2.65e-1 | 3.75e-1 | 4.53e-1 | 6.56e-1 | 8.90e-1 | 1.31 |
| 8 | 4.68e-2 | 7.81e-2 | 1.09e-1 | 1.40e-1 | 2.50e-1 | 3.12e-1 | 4.84e-1 | 5.93e-1 | 8.43e-1 | 1.25 |
| 9 | 4.68e-2 | 9.37e-2 | 9.37e-2 | 1.56e-1 | 2.50e-1 | 3.28e-1 | 4.84e-1 | 6.40e-1 | 8.75e-1 | 1.28 |
| 10 | 4.68e-2 | 7.81e-2 | 9.37e-2 | 1.71e-1 | 2.34e-1 | 3.28e-1 | 4.53e-1 | 6.71e-1 | 9.68e-1 | 1.32 |
| 11 | 4.68e-2 | 6.25e-2 | 1.09e-1 | 2.03e-1 | 2.81e-1 | 3.59e-1 | 5.00e-1 | 7.03e-1 | 9.68e-1 | 1.32 |
| 12 | 4.68e-2 | 7.81e-2 | 1.09e-1 | 1.87e-1 | 2.65e-1 | 3.12e-1 | 4.68e-1 | 6.40e-1 | 1.00e+0 | 1.29 |
| 13 | 4.68e-2 | 7.81e-2 | 1.09e-1 | 1.87e-1 | 2.65e-1 | 3.43e-1 | 4.84e-1 | 7.03e-1 | 1.03e+0 | 1.35 |
| 14 | 4.68e-2 | 7.81e-2 | 1.09e-1 | 1.56e-1 | 2.18e-1 | 3.43e-1 | 4.53e-1 | 6.56e-1 | 9.21e-1 | 1.26 |
| 15 | 6.25e-2 | 7.81e-2 | 9.37e-2 | 1.87e-1 | 2.65e-1 | 3.90e-1 | 5.15e-1 | 6.71e-1 | 1.00e+0 | 1.28 |
| 16 | 4.68e-2 | 7.81e-2 | 1.25e-1 | 1.71e-1 | 2.81e-1 | 3.59e-1 | 4.84e-1 | 7.34e-1 | 9.84e-1 | 1.37 |
| 17 | 4.68e-2 | 7.81e-2 | 1.25e-1 | 2.03e-1 | 2.81e-1 | 3.43e-1 | 5.46e-1 | 6.87e-1 | 9.37e-1 | 1.35 |
| 18 | 4.68e-2 | 7.81e-2 | 1.25e-1 | 1.87e-1 | 2.96e-1 | 3.43e-1 | 4.68e-1 | 6.71e-1 | 9.37e-1 | 1.40 |
| 19 | 4.68e-2 | 6.25e-2 | 1.09e-1 | 1.71e-1 | 2.50e-1 | 3.59e-1 | 4.84e-1 | 6.71e-1 | 9.68e-1 | 1.25 |
| 20 | 4.68e-2 | 7.81e-2 | 9.37e-2 | 1.56e-1 | 2.34e-1 | 3.43e-1 | 4.53e-1 | 6.87e-1 | 8.90e-1 | 1.21 |

Table 3: Solution times (in seconds) of the SDP problems (3.17) corresponding to the entries in Table 2 using DSDP.

[24].

In order to compute the coefficients of $L_n(f)$ we used the Matlab algorithm by Matthews and Fink [14], and subsequently obtained the coefficients of $(L_n(f))'$ by using the Matlab 7.0.4 routine 'polyder'.

We found this approach to be very numerically unstable for more than 50 Chebychev nodes, for both root finding packages. The main difficulty seems to lie in computing the coefficients of $L_n(f)$ in a stable way; see *e.g.* the discussion in §3.5 of [19].

As mentioned in the introduction, we do not rule out other 'root finding' approaches, like using the Durand-Kerner iterative root finding algorithm (see *e.g.* Pan [16]) in conjunction with the Barycentric representation [3] of $(L_n(f))'$. However, in view of the robustness and speed of the SDP approach, we have not further explored these possibilities.

## 4.2 Hermite interpolation — line search with derivatives

The following tables give the computational results for Hermite interpolation on the Chebyshev nodes. Table 5 give the relative errors in approximating the optimal values using DSDP. Note that the relative errors are not significantly smaller than when using Lagrange interpolation. For a few functions, like 11, 12, 13, higher

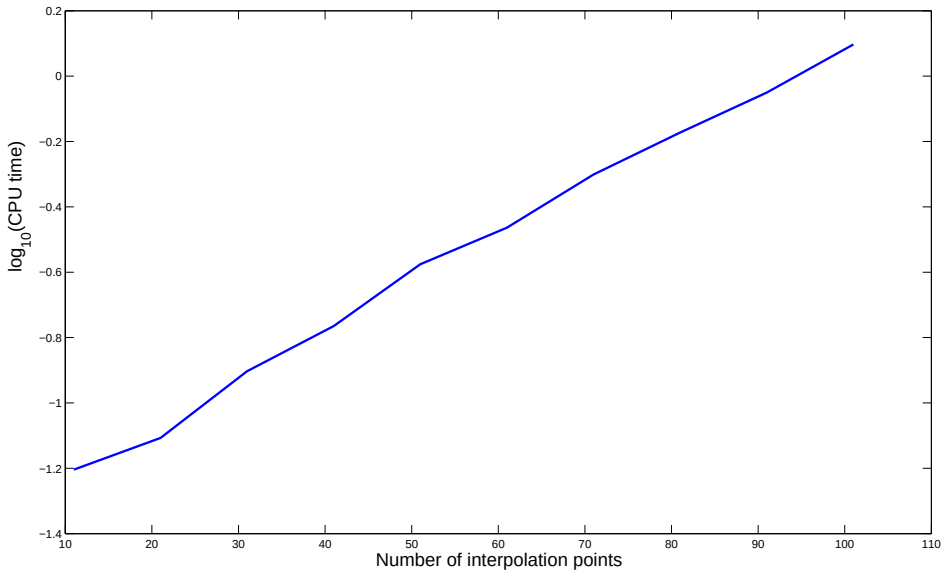


Figure 1: Logarithm of the typical CPU time (in seconds) to solve the SDP (3.17) using DSDP, as a function of the number of interpolation nodes.

accuracy is obtained than with Lagrange interpolation. Table 6 gives the SDP solution times corresponding to the entries in Table 5. Note that these solutions times are comparable (only slightly higher) than for the Lagrange interpolants of similar degree.

In conclusion, there does not seem to be a significant gain in using Hermitian interpolation as opposed to Lagrange interpolation on the Chebyshev nodes for the set of test functions.

### 4.3 Comparison with other line search routines

In order to show that off-the-shelf global optimization routines do not routinely find the global maxima of the test problems in Table 1, we include results for two line search routines:

- The routine FMINBND of the Matlab version 7.0.4 Optimization Toolbox;
- The routine GLS by Neumaier, available at `http://www.mat.univie.ac.at/~neum/software/ls/` This line search routine is used as a subroutine in the global optimization algorithm described in [9].

For the routine FMINBND, the maximum number of function evaluations is limited to 80. The routine GLS chooses the number of functions evaluation dynamically, and the actual number of function evaluations are shown in Table 7. We used the option 'smax = 80' to allow a list size of approximately 80 and the option 'nloc=10' to allow saturation of 10 local minima. The idea was to obtain results that are comparable to Lagrange interpolation on 80 nodes.

| Problem | Relative error |
|---|---|
| 1 | 2.97e-7 |
| 2 | 4.46e-8 |
| 3 | 1.52e-6 |
| 4 | 3.69e-7 |
| 5 | 2.14e-6 |
| 6 | 1.01e-5 |
| 7 | 2.49e-7 |
| 8 | 7.16e-7 |
| 9 | 1.10e-5 |
| 10 | 3.83e-6 |
| 11 | 1.66e-7 |
| 12 | 3.68e-8 |
| 13 | 8.46e-6 |
| 14 | 3.72e-6 |
| 15 | 7.35e-6 |
| 16 | 1.80e-9 |
| 17 | 1.31e-7 |
| 18 | 6.99e-5 |
| 19 | 6.17e-7 |
| 20 | 8.53e-6 |

Table 4: Relative errors between the test function maximizers and the extracted maximizers of the Lagrange interpolant for 81 nodes.

The results for these routines are shown in Table 7. Note that the routines FMINBND and GLS find the global maxima with high accuracy in most cases, but fail in some cases, if we define failure as a relative error greater than $10^{-4}$. Thus, FMINBND fails for the functions 5, 9, 14 and 19, and GLS fails for the functions 3, 9, 10, 13, 15 and 19. Note, however, that the routine GLS uses very few function evaluations in some cases. For example, for function 19, GLS terminates after 5 function evaluations even thought the relative error is 0.104. It is therefore difficult to compare our approach to such line search routines, and the main purpose of this section was to show that the set of test problems is quite challenging for existing line search routines.

# 5 Conclusion

We have revisited the classical 'line search' problem of minimizing a univariate function $f$ on an interval $[a, b]$. When $f$ is a polynomial, we have shown how to reformulate this problem as a semidefinite programming (SDP) problem in such a way that it is possible to solve these SDP reformulations for degrees up to 100 in about one second on a Pentium IV PC. Moreover, the global minimizers may be extracted in a reliable way from the solution of the SDP problem. The important choices in formulating the SDP problem are:

| | 9 | 19 | 29 | 39 | 49 | 59 | 69 | 79 |
|---|---|---|---|---|---|---|---|---|
| 1 | 5.46e-9 | 2.44e-9 | 1.25e-9 | 1.34e-8 | 9.59e-9 | 1.02e-8 | 2.35e-9 | 9.93e-9 |
| 2 | 1.28e-3 | 1.84e-7 | 4.07e-6 | 1.19e-7 | 1.12e-7 | 1.18e-7 | 1.18e-7 | 1.18e-7 |
| 3 | 3.03e+0 | 2.86e+0 | 7.23e-1 | 1.03e+0 | 1.48e-1 | 1.53e-1 | 2.15e-4 | 5.06e-7 |
| 4 | 1.31e-7 | 1.36e-7 | 1.42e-7 | 1.44e-7 | 1.32e-7 | 1.30e-7 | 1.43e-7 | 1.36e-7 |
| 5 | 3.77e-2 | 6.17e-5 | 1.01e-6 | 1.00e-6 | 1.01e-6 | 9.99e-7 | 1.01e-6 | 1.01e-6 |
| 6 | 1.15e+0 | 2.04e-1 | 9.55e-2 | 7.69e-4 | 2.08e-3 | 3.27e-5 | 4.17e-6 | 3.29e-8 |
| 7 | 6.72e-3 | 2.61e-6 | 2.89e-6 | 2.89e-6 | 2.89e-6 | 2.89e-6 | 2.89e-6 | 2.90e-6 |
| 8 | 4.71e+0 | 7.85e-1 | 1.35e+0 | 5.01e-1 | 1.20e-1 | 6.48e-4 | 2.13e-3 | 5.39e-7 |
| 9 | 7.28e-4 | 2.37e-7 | 3.75e-7 | 3.79e-7 | 3.78e-7 | 3.82e-7 | 3.82e-7 | 3.83e-7 |
| 10 | 1.59e-5 | 2.95e-7 | 2.96e-7 | 3.05e-7 | 2.96e-7 | 2.98e-7 | 2.96e-7 | 3.01e-7 |
| 11 | 8.05e-4 | 6.89e-9 | 6.63e-9 | 1.04e-8 | 3.57e-8 | 6.91e-9 | 1.07e-9 | 6.94e-9 |
| 12 | 1.09e-2 | 1.11e-6 | 6.48e-9 | 9.54e-9 | 2.09e-9 | 7.39e-9 | 1.35e-8 | 1.34e-9 |
| 13 | 5.93e-4 | 3.77e-6 | 5.22e-7 | 5.44e-7 | 1.98e-7 | 3.76e-8 | 5.95e-9 | 3.87e-9 |
| 14 | 9.61e-2 | 1.12e-6 | 2.14e-7 | 2.09e-7 | 2.14e-7 | 2.15e-7 | 2.13e-7 | 2.14e-7 |
| 15 | 5.43e-1 | 2.28e-3 | 9.87e-3 | 8.20e-4 | 2.00e-7 | 1.82e-5 | 3.20e-6 | 3.09e-8 |
| 16 | 5.00e-3 | 3.30e-7 | 2.16e-8 | 5.47e-9 | 4.51e-9 | 1.05e-8 | 1.80e-9 | 1.02e-8 |
| 17 | 1.00e-8 | 3.58e-8 | 7.24e-9 | 3.96e-9 | 7.64e-9 | 3.86e-9 | 1.04e-8 | 1.64e-8 |
| 18 | 1.43e-1 | 9.36e-3 | 6.78e-3 | 1.18e-3 | 1.17e-3 | 2.49e-4 | 2.66e-4 | 5.56e-5 |
| 19 | 9.73e-2 | 1.79e-6 | 5.12e-7 | 5.08e-7 | 5.05e-7 | 5.14e-7 | 5.06e-7 | 5.09e-7 |
| 20 | 5.97e-2 | 4.15e-3 | 3.21e-2 | 8.24e-3 | 4.14e-5 | 1.76e-4 | 1.04e-5 | 2.88e-8 |

Table 5: Relative errors in approximating the maximum of the test functions with the maximum of the Hermite interpolant on $n$ Chebyshev nodes using DSDP. The columns are indexed by the degree of the Hermite interpolant.

- Using the 'sampling approach' due to Parrilo and Löfberg [12];
- Using the Chebyshev nodes as sampling points;
- Using an orthogonal basis (the Chebyshev basis) to represent polynomials;
- Using the solution extraction procedure of Henrion and Lasserre [8] as adapted by Jibetean and Laurent [10].

For general $f$, we have applied the SDP approach to Lagrange or Hermite interpolants of $f$ at the Chebyshev nodes. The key ingredients for success of this approach are:

- We do not have to compute the coefficients of the interpolants, since 'sampling' at the Chebyshev nodes is used;
- The constraints of the SDP problems involve only rank one matrices, and this is exploited by the SDP solver DSDP;
- The solution extraction procedure only involves an eigenvalue problem of order the number of global minimizers of the interpolant (*i.e.* very small in practice).

| | 9 | 19 | 29 | 39 | 49 | 59 | 69 | 79 |
|---|---|---|---|---|---|---|---|---|
| 1 | 4.68e-2 | 6.25e-2 | 1.09e-1 | 1.71e-1 | 2.50e-1 | 3.59e-1 | 6.25e-1 | 7.18e-1 |
| 2 | 4.68e-2 | 6.25e-2 | 9.37e-2 | 1.56e-1 | 2.65e-1 | 3.43e-1 | 6.40e-1 | 7.03e-1 |
| 3 | 4.68e-2 | 7.81e-2 | 1.09e-1 | 1.56e-1 | 2.81e-1 | 3.59e-1 | 7.03e-1 | 7.81e-1 |
| 4 | 4.68e-2 | 9.37e-2 | 1.09e-1 | 1.56e-1 | 2.65e-1 | 4.06e-1 | 7.65e-1 | 8.12e-1 |
| 5 | 4.68e-2 | 6.25e-2 | 1.25e-1 | 1.71e-1 | 2.65e-1 | 4.06e-1 | 6.71e-1 | 7.81e-1 |
| 6 | 4.68e-2 | 6.25e-2 | 1.09e-1 | 1.56e-1 | 2.34e-1 | 3.75e-1 | 6.56e-1 | 7.81e-1 |
| 7 | 3.12e-2 | 6.25e-2 | 9.37e-2 | 1.56e-1 | 2.96e-1 | 3.59e-1 | 7.50e-1 | 7.50e-1 |
| 8 | 3.12e-2 | 6.25e-2 | 9.37e-2 | 1.40e-1 | 2.65e-1 | 4.21e-1 | 6.87e-1 | 7.03e-1 |
| 9 | 3.12e-2 | 7.81e-2 | 1.09e-1 | 1.40e-1 | 2.81e-1 | 4.06e-1 | 7.34e-1 | 7.50e-1 |
| 10 | 1.56e-2 | 6.25e-2 | 1.09e-1 | 1.56e-1 | 2.34e-1 | 4.21e-1 | 6.09e-1 | 7.18e-1 |
| 11 | 4.68e-2 | 6.25e-2 | 1.09e-1 | 1.56e-1 | 2.81e-1 | 4.06e-1 | 6.71e-1 | 7.65e-1 |
| 12 | 4.68e-2 | 6.25e-2 | 1.09e-1 | 1.71e-1 | 2.65e-1 | 3.90e-1 | 7.34e-1 | 7.50e-1 |
| 13 | 4.68e-2 | 6.25e-2 | 1.09e-1 | 1.56e-1 | 2.96e-1 | 4.53e-1 | 7.18e-1 | 7.96e-1 |
| 14 | 4.68e-2 | 6.25e-2 | 1.09e-1 | 1.56e-1 | 2.34e-1 | 3.75e-1 | 6.56e-1 | 7.18e-1 |
| 15 | 4.68e-2 | 7.81e-2 | 1.25e-1 | 1.71e-1 | 3.12e-1 | 4.68e-1 | 7.65e-1 | 8.43e-1 |
| 16 | 4.68e-2 | 7.81e-2 | 1.09e-1 | 1.71e-1 | 2.81e-1 | 3.90e-1 | 7.18e-1 | 8.28e-1 |
| 17 | 4.68e-2 | 7.81e-2 | 1.25e-1 | 1.71e-1 | 3.12e-1 | 4.37e-1 | 7.81e-1 | 8.28e-1 |
| 18 | 3.12e-2 | 7.81e-2 | 1.09e-1 | 1.56e-1 | 2.96e-1 | 3.90e-1 | 7.03e-1 | 7.81e-1 |
| 19 | 4.68e-2 | 6.25e-2 | 1.09e-1 | 1.56e-1 | 2.81e-1 | 3.59e-1 | 7.34e-1 | 7.50e-1 |
| 20 | 4.68e-2 | 6.25e-2 | 1.09e-1 | 1.40e-1 | 2.65e-1 | 3.75e-1 | 7.03e-1 | 7.81e-1 |

Table 6: SDP solution times (seconds) for approximating the global maxima of the test functions with the maxima of their Hermite interpolants on $n$ Chebyshev nodes using DSDP. The columns are indexed by the degree of the Hermite interpolant.

We have been able to approximate all the global maximizers for a set of twenty test functions by Hansen *et al.* [7] to 5 digits of accuracy within one second of CPU time on a Pentium IV PC. Perhaps surprisingly, Hermite interpolation did not give superior results to Lagrange interpolation for the test set.

In view of the encouraging numerical results, we feel that the time has come to reconsider the idea of line search via interpolation, since interpolants of high degree can be minimized efficiently and in a numerically stable way with current SDP technology.

## Acknowledgements

Etienne de Klerk would like to thank Monique Laurent for valuable discussions on the 'sampling SDP formulation' of a polynomial minimization problem.

| Test function | Rel. err. FMINBND | Rel. err. GLS | Function evals. GLS |
|---|---|---|---|
| 1 | 1.04e-8 | 4.10e-8 | 8 |
| 2 | 1.20e-7 | 1.17e-7 | 29 |
| 3 | 4.74e-8 | 1.79e-4 | 90 |
| 4 | 1.46e-7 | 5.07e-7 | 70 |
| 5 | 1.14 | 9.95e-7 | 38 |
| 6 | 2.18e-7 | 4.92e-5 | 30 |
| 7 | 2.90e-6 | 2.88e-6 | 32 |
| 8 | 5.04e-7 | 2.88e-5 | 87 |
| 9 | 3.11e-1 | 5.29e-3 | 12 |
| 10 | 2.94e-7 | 1.25e-4 | 15 |
| 11 | 3.96e-10 | 1.13e-9 | 29 |
| 12 | 1.83e-9 | 1.90e-6 | 12 |
| 13 | 6.16e-12 | 2.73e-4 | 6 |
| 14 | 7.21e-1 | 7.54e-6 | 30 |
| 15 | 5.39e-9 | 1.49e-3 | 12 |
| 16 | 4.82e-6 | 3.43e-18 | 5 |
| 17 | 2.95e-8 | 1.17e-5 | 35 |
| 18 | 0 | 0 | 11 |
| 19 | 3.11e-1 | 1.04e-1 | 5 |
| 20 | 2.60e-8 | 6.79e-5 | 29 |

Table 7: Relative errors in approximating the global maxima of the 20 test functions for the line search routines FMINBND and GLS. The number of function evaluations of FMINBND was limited to 80. The numbers of function evaluations for GLS are shown.

# References


[1] B. Beckermann. The Condition Number of real Vandermonde, Krylov and positive definite Hankel matrices. *Numer. Mathematik*, **85**, 553–577, 2000.

[2] S.J. Benson, Y. Ye and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2), 443–461, 2000.

[3] J.-P. Berrut and L.N. Trefethen. Barycentric Lagrange interpolation. *SIAM Review*, **46**, 501–517, 2004.

[4] S. Goedecker. Remark on algorithms to find roots of polynomials. *SIAM J. Sci. Comput.* **15**(5), 1059-1063, 1994.

[5] Y. Hachez. *Convex optimization over non-negative polynomials: structured algorithms and applications*, Ph.D. thesis, Université Catholique de Louvain, Belgium, 2003.

[6] E. Hansen. Global optimization using interval analysis. *Journal of Optimization Theory and Applications*, **29**, 331–343, 1979.

[7] P. Hansen, B. Jaumard, and S-H. Lu. Global optimization of univariate Lipschitz functions : II. New algorithms and computational comparisons. *Mathematical Programming*, **55**, 273-292, 1992.

[8] D. Henrion and J.B. Lasserre. Detecting global optimality and extracting solutions in GloptiPoly. In D. Henrion, A. Garulli (eds). *Positive Polynomials in Control*. Lecture Notes on Control and Information Sciences, 312, Springer Verlag, Berlin, January 2005.

[9] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search, *J. Global Optimization*, **14**, 331-355, 1999.

[10] D. Jibetean and M. Laurent. Semidefinite approximations for global unconstrained polynomial optimization. SIAM Journal on Optimization, **16**(2), 490–514, 2005.

[11] M. Laurent. Moment matrices and optimization over polynomials — A survey on selected topics. Preprint, CWI, 2005. Available at: `http://homepages.cwi.nl/~monique/`

[12] J. Löfberg and P. Parrilo, From coefficients to samples: a new approach to SOS optimization, *IEEE Conference on Decision and Control*, December 2004.

[13] J.B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optimization*, **11**, 796–817, 2001.

[14] J.H. Mathews and K.D. Fink. Numerical Methods: Matlab Programs. Complementary Software to accompany the book: *NUMERICAL METHODS: Using Matlab*, Fourth Edition. Prentice-Hall Pub. Inc., 2004.

[15] Yu. Nesterov, Squared functional systems and optimization problems. In H. Frenk, K. Roos, T. Terlaky and S. Zhang (eds.), *High Performance Optimization*. Dordrecht, Kluwer Academic Press, 405–440, 2000.

[16] V.Y. Pan, Solving a Polynomial Equation: Some History and Recent Progress, *SIAM Review*, **39**(2), 187-220, 1997.

[17] M.J.D. Powell. *Approximation Theory And Methods*, Cambridge University Press, 1981.

[18] V. Powers and B. Reznick. Polynomials that are positive on an interval. *Trans. Amer. Math. Soc.*, **352**, 4677–4692, 2000.

[19] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P,. Flannery. *Numerical recipes in FORTRAN 77: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992. Available online at: `http://library.lanl.gov/numerical/bookfpdf.html`

[20] T.J. Rivlin. *An Introduction to the Approximation of Functions*, Dover, New York, 1981.

[21] T. Roh and L. Vandenberghe. Discrete transforms, semidefinite programming, and sum-of-squares representations of nonnegative polynomials. *SIAM J. Optimization*, **16**(4), 939–964, 2006.

[22] J. Szabados and P. Vértesi. *Interpolation of Functions*, World Scientific, Singapore, 1990.

[23] Z. Zeng. Algorithm 835: MultRoot – A Matlab package for computing polynomial roots and multiplicities. *ACM Transaction on Mathematical Software*, **30**, 218–315, 2004.

[24] Z. Zeng. Computing multiple roots of inexact polynomials. *Mathematics of Computation*, **74**, 869–903, 2005.