

Tilburg University

## A Method For Approximating Univariate Convex Functions Using Only Function Value Evaluations

Siem, A.Y.D.; den Hertog, D.; Hoffmann, A.L.

*Publication date:*  
2007

*Document Version*  
Publisher's PDF, also known as Version of record

[Link to publication in Tilburg University Research Portal](#)

*Citation for published version (APA):*  
Siem, A. Y. D., den Hertog, D., & Hoffmann, A. L. (2007). *A Method For Approximating Univariate Convex Functions Using Only Function Value Evaluations*. (CentER Discussion Paper; Vol. 2007-67). Operations research.

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Center



# Discussion Paper

No. 2007–67

## **A METHOD FOR APPROXIMATING UNIVARIATE CONVEX FUNCTIONS USING ONLY FUNCTION VALUE EVALUATIONS**

By A.Y.D. Siem, D. den Hertog, A.L. Hoffmann

August 2007

ISSN 0924-7815

# A method for approximating univariate convex functions using only function value evaluations

A.Y.D. Siem\*      D. den Hertog<sup>‡</sup>      A.L. Hoffmann<sup>§</sup>

August 31, 2007

## Abstract

In this paper, piecewise linear upper and lower bounds for univariate convex functions are derived that are only based on function value information. These upper and lower bounds can be used to approximate univariate convex functions. Furthermore, new Sandwich algorithms are proposed, that iteratively add new input data points in a systematic way, until a desired accuracy of the approximation is obtained. We show that our new algorithms that use only function-value evaluations converge quadratically under certain conditions on the derivatives. Under other conditions, linear convergence can be shown. Some numerical examples, including a Strategic investment model, that illustrate the usefulness of the algorithm, are given.

**Keywords:** approximation, convexity, meta-model, Sandwich algorithm.

**JEL Classification:** C60.

## 1 Introduction

In the field of discrete approximation, we are interested in approximating a function, given a certain discrete dataset. This is the case in black-box optimization, where we are interested in optimizing a black-box function that is time-consuming to evaluate, and of which no derivative information is available. This function could e.g. be represented by a deterministic computer simulation. Instead of using the black-box function directly, an approximation of this black-box function is used for optimization. These approximations are also called meta-models, compact models, surrogates, response surface models, emulators, or regression models.

---

\*Department of Econometrics and Operations Research/ Center for Economic Research (CentER), Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands, Phone:+31 13 4663254, Fax:+31 13 4663280, E-mail: a.y.d.siem@uvt.nl.

<sup>‡</sup>Department of Econometrics and Operations Research/ Center for Economic Research (CentER), Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands, Phone:+31 13 4662122, Fax:+31 13 4663280, E-mail: d.denhertog@uvt.nl.

<sup>§</sup>Department of Radiation Oncology, Radboud University Nijmegen Medical Centre, Geert Grooteplein 32, 6525 GA Nijmegen, The Netherlands, Phone:+31 24 3610584, Fax:+31 24 3568350, E-mail: a.hoffmann@rther.umcn.nl.

We sometimes know beforehand that the function that is to be approximated has some characteristics. It could be known e.g. that it is a nonnegative, increasing or convex function. However, the approximation does not necessarily inherit these characteristics. In Siem et al. (2007) e.g., nonnegativity preserving (trigonometric) polynomials and rational functions are studied. In that paper, also monotonicity preserving polynomials are studied. In Burkard et al. (1991), Fruhwirth et al. (1989), Rote (1992) and Yang and Goh (1997) so-called Sandwich algorithms are proposed for univariate approximation of convex functions. In these algorithms upper and lower bounds of the convex function are constructed. The methods in Burkard et al. (1991), Fruhwirth et al. (1989), and Rote (1992) make use of derivative information, which is not always available, especially in case of black-box functions. In Yang and Goh (1997) a derivative free optimization problem has to be solved in case there is no derivative information. This costs many function value evaluations, which may be time-consuming. In Section 3 we treat these Sandwich algorithms in more detail.

In this paper we present a methodology to find approximations of univariate convex functions via upper and lower bounds. An important difference with the methods studied in Burkard et al. (1991), Fruhwirth et al. (1989), and Rote (1992) is that our methodology uses only function value evaluations. Based on convexity, we construct upper and lower bounds of a convex univariate function  $y : \mathbb{R} \mapsto \mathbb{R}$ , that is only known in a finite set of points  $x^1, \dots, x^n \in \mathcal{U} \subseteq \mathbb{R}$  with values  $y(x^1), \dots, y(x^n) \in \mathbb{R}$ , and for which no derivative information is known. In Den Boef and Den Hertog (2007), these kind of bounds are used for efficient line searching of convex functions. We show that if derivative information is available, tighter lower bounds can be obtained than if this information is not available. In our paper Siem et al. (2007), it is shown that under certain conditions, these upper and lower bounds can be improved by using suitable transformations. Furthermore, we present iterative strategies, that determine in each iteration which new input data point is best to be evaluated next, until a desired accuracy is met. Different criteria can be used to select this new input data point. The iterative strategies that we use belong to the class of so-called Sandwich algorithms described in Burkard et al. (1991), Fruhwirth et al. (1989), and Rote (1992). However, these Sandwich algorithms are based on derivative information. Therefore, in Section 3, we introduce a version of the Sandwich algorithm that can be used when only function value information is available. Moreover, we introduce two other iterative strategies, based on function value information only. For these two strategies, we do not give convergence proofs. In Section 4 we give convergence proofs for our new Sandwich algorithms. Under certain conditions on the derivatives of  $y(x)$ , we can show quadratic convergence for different variants of our Sandwich algorithms. Under other conditions, linear convergence can be shown for our Sandwich algorithms. With some numerical examples, we compare different variants of our new iterative strategies, and show that our methods give better results than choosing the input data equidistantly. Also, we apply these methods to approximate the convex optimal value function of a Strategic investment model. Application of these methods in Intensity Modulated Radiation Therapy (IMRT) can be found in our paper Hoffmann et al. (2006).

The remainder of this paper is organized as follows. In Section 2, we show how we can obtain upper and lower bounds to approximate univariate convex functions and show that if

derivative information is available, we can obtain even tighter bounds. In Section 3, we discuss iterative strategies for determining new data points to be evaluated. In Section 4 we consider convergence results. In Section 5, we study numerical examples. Finally, in Section 6 we give our conclusions and discuss possible directions for further research.

## 2 Approximating convex functions

### 2.1 Bounds based on function value evaluations

Suppose that  $n$  input data points  $x^1, \dots, x^n \in [x^1, x^n] \subseteq \mathbb{R}$ , are given, together with  $n$  corresponding output data points  $y(x^1), \dots, y(x^n) \in \mathbb{R}$ . It is well-known that the straight line through the points  $(x^i, y(x^i))$  and  $(x^{i+1}, y(x^{i+1}))$ , for  $1 \leq i \leq n-1$ , is an upper bound of the curve  $y(x)$ , for  $x \in [x^i, x^{i+1}]$ ; see Figure 1. Furthermore, it is known that the straight lines through the points  $(x^{i-1}, y(x^{i-1}))$  and  $(x^i, y(x^i))$ , for  $2 \leq i \leq n-1$ , and  $(x^{i+1}, y(x^{i+1}))$  and  $(x^{i+2}, y(x^{i+2}))$ , for  $1 \leq i \leq n-2$ , are lower bounds of the curve  $y(x)$ , for  $x \in [x^i, x^{i+1}]$ ; see again Figure 1. For the sake of completeness we give a proof.

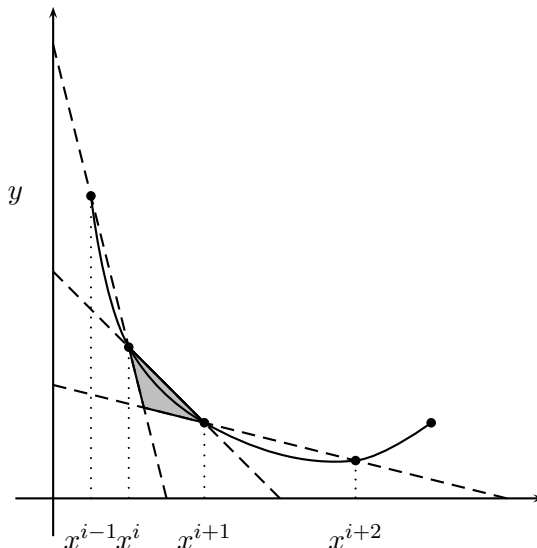


Figure 1: Upper and lower bounds for a convex function on the interval  $[x^i, x^{i+1}]$  using only function value evaluations.

**Theorem 1.** Let  $n$  input/output data points  $(x^1, y(x^1)), \dots, (x^n, y(x^n))$ , with  $x^1 < x^2 < \dots < x^n$  be given, and let  $y(x)$  be convex. Suppose furthermore that  $x^i \leq x \leq x^{i+1}$ , then

$$y(x) \leq \frac{x^{i+1} - x}{x^{i+1} - x^i} y(x^i) + \frac{x - x^i}{x^{i+1} - x^i} y(x^{i+1}), \text{ for } 1 \leq i \leq n-1, \quad (1)$$

$$y(x) \geq \frac{x - x^{i-1}}{x^i - x^{i-1}} y(x^i) + \frac{x^i - x}{x^i - x^{i-1}} y(x^{i-1}), \text{ for } 2 \leq i \leq n-1, \quad (2)$$

and

$$y(x) \geq \frac{x - x^{i+1}}{x^{i+2} - x^{i+1}}y(x^{i+2}) + \frac{x^{i+2} - x}{x^{i+2} - x^{i+1}}y(x^{i+1}), \text{ for } 1 \leq i \leq n - 2. \quad (3)$$

*Proof.* We first show (1). Since  $x^i \leq x \leq x^{i+1}$ , there exists a  $0 \leq \lambda \leq 1$  such that

$$x = \lambda x^i + (1 - \lambda)x^{i+1}. \quad (4)$$

Due to convexity we have  $y(x) \leq \lambda y(x^i) + (1 - \lambda)y(x^{i+1})$ . From (4), we may conclude that  $\lambda = \frac{x^{i+1} - x}{x^{i+1} - x^i}$ . This yields

$$y(x) \leq \frac{x^{i+1} - x}{x^{i+1} - x^i}y(x^i) + \frac{x - x^i}{x^{i+1} - x^i}y(x^{i+1}),$$

which shows (1). Next, we show inequality (2). First we consider the case that  $x^{i-1} < x^i < x$ . Then there exists a  $0 < \lambda < 1$  such that

$$x^i = \lambda x^{i-1} + (1 - \lambda)x. \quad (5)$$

Due to convexity we have  $y(x^i) \leq \lambda y(x^{i-1}) + (1 - \lambda)y(x)$ , which yields

$$y(x) \geq \frac{1}{1 - \lambda}y(x^i) - \frac{\lambda}{1 - \lambda}y(x^{i-1}). \quad (6)$$

From (5), we may conclude that  $\lambda = \frac{x - x^i}{x - x^{i-1}}$ . Substituting into (6) gives

$$y(x) \geq \frac{x - x^{i-1}}{x^i - x^{i-1}}y(x^i) + \frac{x^i - x}{x^i - x^{i-1}}y(x^{i-1}),$$

which is the second inequality. In case  $x^{i-1} < x^i = x$ , (2) holds trivially. Inequality (3) follows in a similar way as inequality (2).  $\square$

## 2.2 Bounds based on derivatives

In addition to the bounds described in Section 2.1, we can also use derivative information (if present) to obtain lower bounds. Suppose that  $y(x)$  is differentiable and that not only the  $n$  data points  $(x^1, y(x^1)), \dots, (x^n, y(x^n))$  are given, but also the derivative information  $(x^1, y'(x^1)), \dots, (x^n, y'(x^n))$ . Then we have

$$y(x) \geq y(x^i) + y'(x^i)(x - x^i), \quad \forall x \in [x^1, x^n], \forall i = 1, \dots, n. \quad (7)$$

This lower bound is schematically shown in Figure 2. In the following theorem we show that these lower bounds are tighter than the lower bounds derived in the previous subsection, which do not use derivative information.

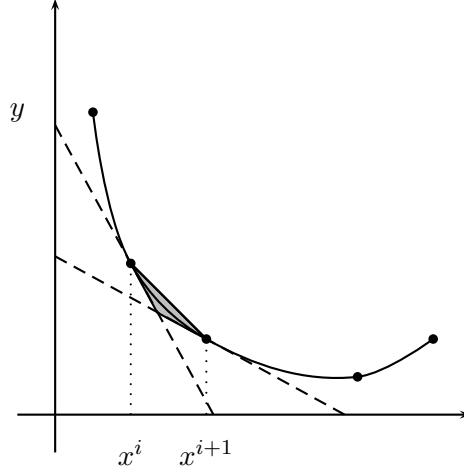


Figure 2: Upper and lower bounds for a convex function on the interval  $[x^i, x^{i+1}]$ , using derivative information.

**Theorem 2.** Let  $n$  input/output data points  $(x^1, y(x^1)), \dots, (x^n, y(x^n))$ , with  $x^1 < x^2 < \dots < x^n$  be given, and let  $y(x)$  be differentiable and convex. Suppose furthermore that  $x^i \leq x \leq x^{i+1}$ , then

$$y(x^i) + y'(x^i)(x - x^i) \geq \frac{x - x^{i-1}}{x^i - x^{i-1}}y(x^i) + \frac{x^i - x}{x^i - x^{i-1}}y(x^{i-1}), \text{ for } 2 \leq i \leq n - 1, \quad (8)$$

and

$$y(x^{i+1}) + y'(x^{i+1})(x - x^{i+1}) \geq \frac{x - x^{i+1}}{x^{i+2} - x^{i+1}}y(x^{i+2}) + \frac{x^{i+2} - x}{x^{i+2} - x^{i+1}}y(x^{i+1}), \text{ for } 1 \leq i \leq n - 2. \quad (9)$$

*Proof.* Let us denote the lefthand side of (8) by  $\ell_1(x)$  and its righthand side by  $\ell_2(x)$ . Then we have  $\ell'_1(x) = y'(x^i)$  and  $\ell'_2(x) = \frac{y(x^i) - y(x^{i-1})}{x^i - x^{i-1}}$ . Now, by the mean value theorem we know that there exists a  $\xi \in [x^{i-1}, x^i]$  such that  $y'(\xi) = \frac{y(x^i) - y(x^{i-1})}{x^i - x^{i-1}}$ . Since  $y(x)$  is convex, we have that  $\ell'_2(x) = y'(\xi) \leq y'(x^i) = \ell'_1(x)$ . Since both  $\ell_1(x)$  and  $\ell_2(x)$  are straight lines through  $(x^i, y(x^i))$ , and  $\ell'_2(x) \leq \ell'_1(x)$ , we have  $\ell_1(x) \geq \ell_2(x)$ , for all  $x \geq x^i$ , which shows (8). Inequality (9) follows in a similar way.  $\square$

### 3 Iterative strategies

In this section, we deal with iterative strategies to approximate univariate convex functions. These methods select a new input data point to evaluate in every iteration, until a desired accuracy is met. In Section 3.1 we consider so-called Sandwich algorithms that are already known from literature. These Sandwich algorithms can be used in combination with the lower bound based on derivative information (7). In Section 3.2, we introduce a version of the Sandwich algorithm that can be used in combination with the lower bounds based on only function value

evaluations, (2) and (3). Furthermore, we propose two other iterative strategies to add new input data points.

### 3.1 Sandwich algorithm with derivative information

In this section we consider Sandwich algorithms based on derivative information to construct approximations that satisfy a prescribed accuracy  $\delta$ . There is a vast literature on these Sandwich algorithms; see Burkard et al. (1991), Fruhwirth et al. (1989), Rote (1992), and Yang and Goh (1997). In these Sandwich algorithms, upper and lower bounds are generated in an iterative way. We start with evaluating the function that is to be approximated, at a 'small' number of input data points  $x^1, \dots, x^n \in [x^1, x^n] \subseteq \mathbb{R}$ , i.e., we calculate  $y(x^1), \dots, y(x^n) \in \mathbb{R}$ , and the derivative values  $y'(x^1), \dots, y'(x^n) \in \mathbb{R}$ . Then, we calculate the associated upper and lower bounds (1) and (7). The input data points  $x^1, \dots, x^n$ , with  $x^1 < \dots < x^n$  define a set of intervals  $I = \{[x^1, x^2], [x^2, x^3], \dots, [x^{n-1}, x^n]\}$ . Let  $\delta_j$  denote the error for interval  $j$ , and let  $J \subseteq I$  denote the set of intervals for which the error  $\delta_j > \delta$ . We can use different kinds of error measures, which we mention below. Next, we partition an arbitrary interval in the set  $J$  according to some of the partition rules, which we mention below, and calculate the output value  $y$  and its derivative  $y'$  at the input value  $x^0$ , where the interval is partitioned, i.e., we calculate  $y(x^0)$  and  $y'(x^0)$ . Then, we determine the new upper and lower bounds. Whenever the error of any of the two subintervals is greater than  $\delta$ , we add this interval to the set  $J$ . We repeat this procedure until all intervals in  $J$  have an error smaller than  $\delta$ , i.e., until  $J = \emptyset$ . This procedure is summarized in Algorithm 1.

---

**Algorithm 1** Sandwich algorithm with derivative information

---

INPUT:

An initial set of intervals  $J$ , for which  $\delta_j > \delta$ , for all  $j \in J$ .

WHILE  $J \neq \emptyset$  DO

SELECT interval  $[a, b] \in J$ .

PARTITION  $[a, b]$  into two subintervals  $[a, c]$  and  $[c, b]$ .

Calculate  $y(c)$  and  $y'(c)$ .

Calculate the new upper and lower bounds.

IF  $\delta_{[a,c]} > \delta$

$J := J \cup \{[a, c]\}$ .

ENDIF

IF  $\delta_{[c,b]} > \delta$

$J := J \cup \{[c, b]\}$ .

ENDIF

$J := J \setminus \{[a, b]\}$ .

ENDWHILE

---

Different error measures and different partition rules have been proposed in literature. The error measures as mentioned in Rote (1992) are:

1. *Maximum error* on interval ( $\infty$ -norm):  $\delta_{[a,b]}^\infty = \max_{x \in [a,b]} \{u(x) - l(x)\}$ ;



2. *Uncertainty area* enclosed by bounds on interval (1-norm):  $\delta_{[a,b]}^1 = \int_{[a,b]} (u(x) - l(x)) dx$ ;
3. *Hausdorff distance* on interval:  $\delta_{[a,b]}^H = \max \left\{ \sup_{v \in L} \inf_{w \in U} \|w - v\|, \sup_{w \in U} \inf_{v \in L} \|w - v\| \right\}$ ,

where  $[a, b]$  is the interval of interest,  $u(x)$  is the upper bound,  $l(x)$  the lower bound,  $L = \{(x, l(x)) | x \in [a, b]\}$ , and  $U = \{(x, u(x)) | x \in [a, b]\}$ . An advantage of the last two error measures is that it does not discriminate between the two coordinate directions.

The partition rules as mentioned in Rote (1992) are:

1. *Interval bisection*: Interval is partitioned into two equal parts.
2. *Maximum error*: Interval is partitioned at the point where the maximum error is attained.
3. *Slope bisection*: Find the supporting line whose slope is the mean value of the slopes of the tangent lines at the endpoints. Partition the interval at the point where this line is tangent to the graph of the function.
4. *Chord rule*: Find the supporting tangent line whose slope is equal to the slope of the line connecting the two endpoints. Partition the interval at the point where this line is tangent to the graph of the function.

### 3.2 Iterative strategies with only function value information

We cannot use the Sandwich algorithms as described in Section 3.1 in combination with the lower bounds based on only function value evaluations (2) and (3), since we do not have derivative information. If we use the lower bounds from (2) and (3), adding a new point reduces the error not only in the interval where the point is added, but most possibly also in the neighbouring intervals. This is not the case when we use lower bounds based on derivative information. Therefore, in this section we adjust Algorithm 1, such that it can be applied in combination with the lower bounds based on only function value evaluations (2) and (3). The adjusted procedure is summarized in Algorithm 2. An important difference is that in the Algorithm 2, we have to update the set  $J$  in a different way. We have to check whether the neighbouring intervals still belong to  $J$ . Furthermore, another difference is that we select the new input data point in the interval, in which the error measure is largest, instead of selecting an arbitrary interval. This may cause the error to decrease faster. Note that for the Sandwich algorithm in Section 3.1, by selecting the interval where the error is the largest, the accuracy  $\delta$  is not reached earlier than if we select an arbitrary interval in  $J$ .

Note that we can use all three error measures as mentioned in Section 3.1. However, we cannot use the same partition rules as in Section 3.1. Since we have no derivative information, we cannot use the Slope bisection rule and the Chord rule. Also, we cannot use the Maximum error partition rule, because in the leftmost interval the Maximum error is attained at a point that is already simulated. Therefore, we can only use the Interval bisection partition rule.

---

**Algorithm 2** Sandwich algorithm with only function value information

---

INPUT:

An initial set of intervals  $J$ , for which  $\delta_j > \delta$ , for all  $j \in J$ .WHILE  $J \neq \emptyset$  DOSELECT interval  $[a, b] \in J$  for which  $\delta_{[a,b]}$  is maximal.PARTITION  $[a, b]$  into two subintervals  $[a, c]$  and  $[c, b]$ .Calculate  $y(c)$ .

Calculate the new upper and lower bounds.

IF  $\delta_{[a,c]} > \delta$  $J := J \cup \{[a, c]\}$ 

ENDIF

IF  $\delta_{[c,b]} > \delta$  $J := J \cup \{[c, b]\}$ 

ENDIF

 $J := J \setminus \{[a, b]\}$ Check if the errors of neighbouring intervals are still larger than  $\delta$ , and if not, remove them from the set  $J$ .ENDWHILE

---

Finally, we introduce two other iterative strategies. These two iterative strategies add a new input data point such that the Uncertainty area after adding that input data point is minimized until the Uncertainty area is below a certain level  $\delta$ . However, we do not know the Uncertainty area after adding a new data point, since we do not know the output value  $y$  of that input data point. We solve this problem as follows. Suppose we have the input/output data points  $(x^1, y(x^1)), \dots, (x^n, y(x^n))$ , with the corresponding upper and lower bounds; see Figure 3. Then, if we evaluate the  $(n + 1)$ -th point  $(x^0, y^0)$ , the Uncertainty area after adding this point to our data, reduces. Therefore, a first approach is that we calculate the average Uncertainty area over all possible values of  $y^0$ . A second approach is that we calculate the worst-case (i.e. the maximum) Uncertainty area of all possible values of  $y^0$ . Thus, we can evaluate the next data point  $x^0$ , according to the following rules:

- *Average area rule:* We take the value of  $x^0$ , where the average Uncertainty area after addition is minimal.
- *Worst-case area rule:* We take the value of  $x^0$ , where the maximal Uncertainty area after addition is minimal.

Let us now describe this more mathematically. Let us denote the upper bound after adding the point  $(x^0, y^0)$  as  $u(x; (x^0, y^0))$ , and the lower bound as  $l(x; (x^0, y^0))$ . Then the area between the upper bound and the lower bound is given by:

$$A(x^0, y^0) = \int_X [u(x; (x^0, y^0)) - l(x; (x^0, y^0))] dx, \quad (10)$$

where  $X = [x^1, x^n]$  is the total interval. We are now interested in finding the value of  $x^0 \in X$ , for which  $A(x^0, y^0)$  is minimal. In general we do not know the value of  $y^0$ . Therefore, in the

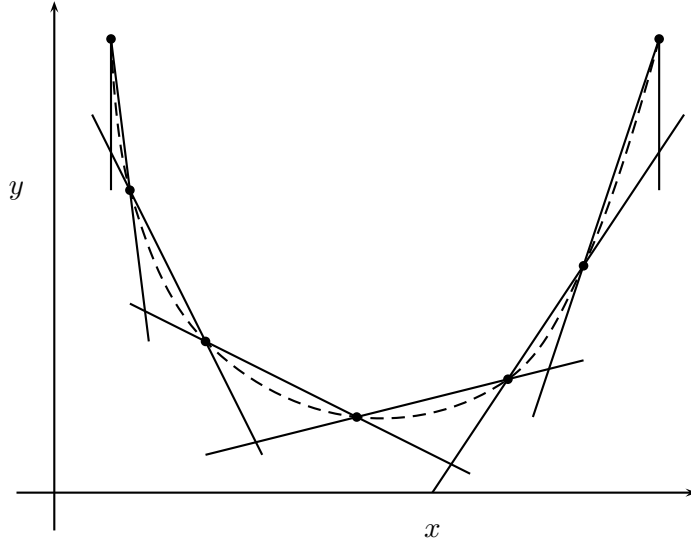


Figure 3: Upper and lower bounds for a convex function, based on function value evaluations.

first approach we take the average value over all possible values  $y^0$  as measure, and we select value  $x^0$  that solves

$$\min_{x^0 \in X} \frac{1}{u(x^0) - l(x^0)} \int_{Y(x^0)} \int_X [u(x; (x^0, y^0)) - l(x; (x^0, y^0))] dx dy^0, \quad (11)$$

where  $Y(x^0) = \{y \in \mathbb{R} | l(x^0) \leq y \leq u(x^0)\}$ , and  $u(x^0)$  and  $l(x^0)$  are the bounds, based on the original data, before adding a new point. We repeat this until the total area is below a desired accuracy level  $\delta$ .

In the second approach we take the value of  $y^0$  that yields the maximal area, which is the worst case, as measure, i.e., we select the value of  $x^0$  that solves

$$\min_{x^0 \in X} \max_{y^0 \in Y(x^0)} \int_X [u(x; (x^0, y^0)) - l(x; (x^0, y^0))] dx. \quad (12)$$

Again, we repeat this until the total Uncertainty area is below an accuracy level  $\delta$ .

Since it is rather much work to calculate the integrals explicitly, we calculate them numerically. The integral in (10) can be calculated exactly by using the fact that this integral is the total area of the triangles in Figure 3. Since the coordinates of the corners of all the triangles can be calculated easily from the expressions of the upper and lower bounds as given in Theorem 1, we can calculate the area of the triangles by using that the area of a triangle  $A_t$  with corner points  $(a_1, b_1)$ ,  $(a_2, b_2)$ , and  $(a_3, b_3)$ , is given by

$$A_t = \frac{1}{2} \det \begin{bmatrix} a_1 - a_3 & a_2 - a_3 \\ b_1 - b_3 & b_2 - b_3 \end{bmatrix}. \quad (13)$$

Moreover, instead of calculating the integral over  $Y(x^0)$ , we approximate this integral by taking

the average over a finite number of points  $\bar{N}$ , i.e. we calculate

$$\min_{x^0 \in X} \frac{1}{\bar{N}} \sum_{i=1}^{\bar{N}} \int_X [u(x; (x^0, \bar{y}_0^i)) - l(x; (x^0, \bar{y}_0^i))] dx^0 dy^0,$$

where  $\bar{y}_0^i$  are spread equidistantly over  $Y(x^0)$ , and  $\bar{N}$  is large enough.

In Section 4, we present convergence results for the (Sandwich) Algorithm 2, and in Section 5 we show some numerical examples to illustrate and compare the different iterative strategies.

## 4 Convergence

In this section, we consider the convergence of Algorithm 1 and present new convergence results of Algorithm 2.

### Sandwich algorithms

Concerning convergence proofs for Sandwich algorithms, Fruhwirth et al. (1989) proved that (Sandwich) Algorithm 1 in the case of Hausdorff distance, is of order  $O(1/n^2)$ , where  $n$  denotes the number of evaluation points. Burkard et al. (1991) obtained the same order for the Maximum error ( $\infty$ -norm). All these convergence results require that the right derivative in the left endpoint, and the left derivative in the right endpoint of the interval are finite. Gu erin et al. (2006) derived an optimal adaptive Sandwich algorithm for which they proved  $O(1/n^2)$  convergence, without assuming bounded right and left derivatives at the left and right endpoint, respectively. Note that these Sandwich algorithms use derivative information in each evaluation point. Yang and Goh (1997) proposed a Sandwich algorithm that only uses function evaluations. However, in each iteration their algorithm requires the solution of an optimization problem involving the function itself.

In this section, we prove that our upper and lower bounds which do not use derivative information, for equidistant input data points, are of order  $O(1/n^2)$ , for the Maximum error ( $\infty$ -norm), for the Uncertainty area (1-norm), and for the Hausdorff distance. These results also require bounded right and left derivatives at the left and right endpoint, respectively. Notice that especially in the case of approximating a Pareto frontier, this assumption may be violated; see e.g. Example 5.2. When this assumption does not hold, we prove an  $O(1/n)$  convergence for our upper and lower bounds for equidistant input data points in the case of Hausdorff distance and the Uncertainty area (1-norm). Note that such a convergence result certainly does not hold for the Maximum error ( $\infty$ -norm). From these results it will follow in this section that (Sandwich) Algorithm 2, using the Interval bisection partitioning rule, converges at least at the same rate, as in the equidistant case, for all error measures.

### Approximation theory

In approximation theory, error bounds are usually given for the  $\infty$ -norm, and involves some global property of the function. For convex approximation, to the best of our knowledge, the

best error bound (in the  $\infty$ -norm) known is  $O(1/\sqrt{n})$  for Lipschitz  $f$  for  $n$  function evaluations, obtained by Bernstein approximation. This improves to  $O(1/n)$  if  $f'$  is Lipschitz.

If the approximation is allowed to be nonconvex, then an  $O((\log n)/n)$  error bound is obtained for Lipschitz  $f$  by Lagrange interpolation at the Chebyshev nodes, and  $O(1/n^2)$  if  $f'$  is continuous. Note that our convergence results improve Bernstein's convergence results for convex approximation on one hand, but that on the other hand our Sandwich method does not yield one function as an approximation.

### Convergence rates

In Theorems 3, 4, and 5, we give convergence results for equidistant input data points for three different error measures. For simplicity, we write  $y^i = y(x^i)$

**Theorem 3.** *Suppose that  $y : [x^1, x^n] \mapsto \mathbb{R}$  is convex, and is known in the equidistant input data  $x^1, \dots, x^n$ . Furthermore, suppose that the right derivative  $y'_+$  in  $x^1$  exists, the left derivative  $y'_-$  in  $x^n$  exists, and that  $y'_-(x^n) - y'_+(x^1) < \infty$ . Then we have for the Maximum error,  $\delta_{[x^1, x^n]}^\infty$ , between the upper and lower bounds  $u(x)$  and  $l(x)$  of Theorem 1 that*

$$\delta_{[x^1, x^n]}^\infty \leq \frac{y'_-(x^n) - y'_+(x^1)}{n-1}.$$

Furthermore, suppose that  $y''$  exists on  $[x^1, x^n]$  and that  $\|y''\|_\infty < \infty$ . Then, we have for the Maximum error,  $\delta_{[x^1, x^n]}^\infty$ , between the upper and lower bounds  $u(x)$  and  $l(x)$  of Theorem 1 that

$$\delta_{[x^1, x^n]}^\infty \leq \frac{1}{(n-1)^2} \|y''\|_\infty.$$

*Proof.* Let  $\lambda^i(x) = \frac{x^{i+1}-x}{h}$ , where  $h$  is the length of the interval  $[x^i, x^{i+1}]$ . For the intervals  $[x^i, x^{i+1}]$ , with  $i = 1, \dots, n-2$ , we subtract the 'right' lower bound (3) from the upper bound (1):

$$\begin{aligned} \Delta y(x) &= \lambda^i(x)y^i + (1 - \lambda^i(x))y^{i+1} - (1 + \lambda^i(x))y^{i+1} + \lambda^i(x)y^{i+2} \\ &= \lambda^i(x)(y^i - 2y^{i+1} + y^{i+2}). \end{aligned} \tag{14}$$

If we assume that the right derivative  $y'_+$  in  $x^1$  exists, the left derivative  $y'_-$  in  $x^n$  exists, and that  $y'_-(x^n) - y'_+(x^1) < \infty$ , using Taylor's remainder formula, we have that

$$y^{i+2} = y^{i+1} + hy'(\xi_1), \tag{15}$$

where  $\xi_1 \in [x^{i+1}, x^{i+2}]$  and

$$y^i = y^{i+1} - hy'(\xi_2), \tag{16}$$

where  $\xi_2 \in [x^i, x^{i+1}]$ . Substituting (15) and (16) in (14) gives

$$\Delta y(x) \leq h(y'(\xi_1) - y'(\xi_2)) \leq h(y'_-(x^n) - y'_+(x^1)).$$

Note that  $h = \frac{1}{n-1}$ , so

$$\Delta y(x) \leq \delta_{[x^1, x^n]}^\infty \leq \frac{y'_-(x^n) - y'_+(x^1)}{n-1}. \quad (17)$$

For the interval  $[x^{n-1}, x^n]$ , we can also obtain (17) by subtracting the 'left' lower bound (3) from the upper bound (1).

If we assume that  $y''$  exists and that  $\|y''\|_\infty < \infty$ , using Taylor's remainder formula, we have that

$$y^{i+2} = y^{i+1} + hy'(x^{i+1}) + \frac{1}{2}h^2y''(\xi_1), \quad (18)$$

where  $\xi_1 \in [x^{i+1}, x^{i+2}]$  and

$$y^i = y^{i+1} - hy'(x^{i+1}) + \frac{1}{2}h^2y''(\xi_2), \quad (19)$$

where  $\xi_2 \in [x^i, x^{i+1}]$ . Substituting (18) and (19) in (14) gives

$$\Delta y(x) \leq \frac{1}{2}h^2(y''(\xi_1) + y''(\xi_2)) \leq h^2\|y''\|_\infty.$$

Note that  $h = \frac{1}{n-1}$ , so

$$\Delta y(x) \leq \delta_{[x^1, x^n]}^\infty \leq \frac{1}{(n-1)^2}\|y''\|_\infty. \quad (20)$$

For the interval  $[x^{n-1}, x^n]$ , we can also obtain (20) by subtracting the 'left' lower bound (3) from the upper bound (1).  $\square$

**Theorem 4.** *Suppose that  $y : [x^1, x^n] \mapsto \mathbb{R}$  is convex, and is known in the equidistant input data  $x^1, \dots, x^n$ . Then, we have for the total Uncertainty area  $\delta_{[x^1, x^n]}^1$ , between the upper and lower bounds  $u(x)$  and  $l(x)$  of Theorem 1 that*

$$\delta_{[x^1, x^n]}^1 \leq \frac{y^{\max} - y^{\min}}{n-1}. \quad (21)$$

*Furthermore, suppose that the right derivative  $y'_+$  in  $x^1$  exists, the left derivative  $y'_-$  in  $x^n$  exists, and that  $y'_-(x^n) - y'_+(x^1) < \infty$ . Then, we have for the total area  $\delta_{[x^1, x^n]}^1$ , between the upper and lower bounds  $u(x)$  and  $l(x)$  of Theorem 1 that*

$$\delta_{[x^1, x^n]}^1 \leq \frac{y'_-(x^n) - y'_+(x^1)}{2(n-1)^2}. \quad (22)$$

*Proof.* As in the proof of Theorem 3, let  $\lambda^i(x) = \frac{x^{i+1}-x}{h}$ , where  $h$  is the length of the interval  $[x^i, x^{i+1}]$ . For the intervals  $[x^i, x^{i+1}]$ , with  $i = 1, \dots, n-2$ , we subtract the 'right' lower bound (3) from the upper bound (1). Then, we again obtain (14). Integrating this gives:

$$\begin{aligned} A_i &\leq \int_{x^i}^{x^{i+1}} \lambda^i(x)(y^i - 2y^{i+1} + y^{i+2})dx \\ &= h(y^i - 2y^{i+1} + y^{i+2}) \int_0^1 \lambda^i d\lambda^i \\ &= \frac{1}{2}h(y^i - 2y^{i+1} + y^{i+2}), \end{aligned} \tag{23}$$

where  $A_i$  denotes the Uncertainty area on  $[x^i, x^{i+1}]$ . The inequality in (23) comes from the fact that we only used the 'right' lower bound. For the interval  $[x^{n-1}, x^n]$ , we do the same but then with the 'left' lower bound. We then obtain:

$$\begin{aligned} A_{n-1} &= \int_{x^{n-1}}^{x^n} \lambda^{n-1}(x)(y^{n-2} - 2y^{n-1} + y^n)dx \\ &= h(y^{n-2} - 2y^{n-1} + y^n) \int_0^1 \lambda^{n-1} d\lambda^{n-1} \\ &= \frac{1}{2}h(y^{n-2} - 2y^{n-1} + y^n). \end{aligned}$$

Then the total Uncertainty area is given by:

$$\delta_{[x^1, x^n]}^1 = \sum_{i=1}^{n-1} A_i \leq \frac{1}{2}h(y^1 - y^2 - y^{n-1} + y^n) \leq h(y^{\max} - y^{\min}), \tag{24}$$

which shows (21).

Now we assume that the right derivative  $y'_+$  in  $x^1$  exists, the left derivative  $y'_-$  in  $x^n$  exists, and that  $y'_-(x^n) - y'_+(x^1) < \infty$ . Using the Taylor expansion we have that

$$y^1 = y^2 - hy'(\xi_1), \tag{25}$$

where  $\xi_1 \in [x^1, x^2]$  and

$$y^n = y^{n-1} + hy'(\xi_2), \tag{26}$$

where  $\xi_2 \in [x^{n-1}, x^n]$ . Then, with (25) and (26), instead of (24), we obtain

$$\delta_{[x^1, x^n]}^1 \leq \frac{1}{2}h^2(y'(\xi_2) - y'(\xi_1)).$$

Note that  $h = \frac{1}{n-1}$ , so

$$\delta_{[x^1, x^n]}^1 \leq \frac{y'(\xi_2) - y'(\xi_1)}{2(n-1)^2} \leq \frac{y'_-(x^n) - y'_+(x^1)}{2(n-1)^2},$$

which shows (22). □

**Theorem 5.** *Suppose that  $y : [x^1, x^n] \mapsto \mathbb{R}$  is convex, and is known in the equidistant input data  $x^1, \dots, x^n$ . Furthermore, suppose that the right derivative  $y'_+$  in  $x^1$  exists, the left derivative  $y'_-$  in  $x^n$  exists, and that  $y'_-(x^n) - y'_+(x^1) < \infty$ . Then, we have for the Hausdorff distance,  $\delta_{[x^i, x^{i+1}]}^H$ , between the upper and lower bounds  $u(x)$  and  $l(x)$  of Theorem 1 on the interval  $[x^i, x^{i+1}]$  that*

$$\delta_{[x^i, x^{i+1}]}^H \leq \frac{y'_-(x^n) - y'_+(x^1)}{n-1}. \quad (27)$$

Furthermore, suppose that  $y''$  exists and that  $\|y''\|_\infty < \infty$ . Then, we have for the Hausdorff distance,  $\delta_{[x^i, x^{i+1}]}^H$ , between the upper and lower bounds  $u(x)$  and  $l(x)$  of Theorem 1 on the interval  $[x^i, x^{i+1}]$  that

$$\delta_{[x^i, x^{i+1}]}^H \leq \frac{1}{(n-1)^2} \|y''\|_\infty. \quad (28)$$

*Proof.* In Fruhwirth et al. (1989) it is stated that the Hausdorff distance is always less than or equal to the Maximum error. Therefore (27) and (28) follow immediately from Theorem 3. If we assume that the right derivative  $y'_+$  in  $x^1$  exists, the left derivative  $y'_-$  in  $x^n$  exists, and that  $y'_-(x^n) - y'_+(x^1) < \infty$ , then we can write

$$\delta_{[x^i, x^{i+1}]}^H \leq \delta_{[x^i, x^{i+1}]}^\infty \leq \frac{y'_-(x^n) - y'_+(x^1)}{n-1},$$

which shows (27). If we assume that  $y''$  exists and that  $\|y''\|_\infty < \infty$ , we can write

$$\delta_{[x^i, x^{i+1}]}^H \leq \delta_{[x^i, x^{i+1}]}^\infty \leq \frac{1}{(n-1)^2} \|y''\|_\infty,$$

which shows (28). □

In the following Corollary, we show that the results of Theorems 3, 4, and 5, imply that (Sandwich) Algorithm 2 with the Interval bisection rule converges at least at the same rate.

**Corollary 1.** *The results of Theorems 3, 4, and 5 also hold if we apply Algorithm 2 in combination with the Interval bisection rule, instead of equidistant input data points.*

*Proof.* Suppose that we want to obtain a certain level of uncertainty  $\delta$ . Then, Theorems 3, 4, and 5 give us the number (say  $N$ ) of equidistant points that are required to achieve that uncertainty. Let  $\tilde{N}$  be the smallest number such that  $\tilde{N} = 2^k$ , with  $k \in \mathbb{N}$ , and  $\tilde{N} \geq N$ . The points that are generated by (Sandwich) Algorithm 2 after each iteration, are a subset of the  $\tilde{N}$



equidistant points. Also, the Sandwich algorithm needs at most  $\tilde{N}$  iterations (then, the desired uncertainty  $\delta$  is reached for sure). Note that  $\tilde{N} \leq 2N$ . Therefore (Sandwich) Algorithm 2 with the Interval bisection rule converges at least with the same rate as in the equidistant case, for all error measures.  $\square$

### Area reduction per iteration

Next, we consider Algorithm 2 using the Uncertainty area as error measure and the Interval bisection partitioning rule. We give a more precise result on the area reduction per iteration. By adding a point in Algorithm 2, the triangle in which the data point is added is divided into two triangles. In the following lemma we show that the total area of the two 'new' triangles is at most half the area of the 'original' triangle. We denote the area of the 'original' triangle by  $A_t$  and we denote the area of the 'new' triangles by  $A_1$  and  $A_2$ .

**Theorem 6.** *Let  $y(x)$  be convex and decreasing. Suppose we use Algorithm 2 to approximate  $y(x)$ , and that we use the Interval bisection partitioning rule and the Uncertainty area as error measure. Then we have that*

$$\frac{A_1 + A_2}{A_t} \leq \frac{1}{2}.$$

*Proof.* First, we construct a parametrization for a general triangle, which captures all possible triangles that can occur in the algorithm. The chosen parametrization is shown in Figure 4. In

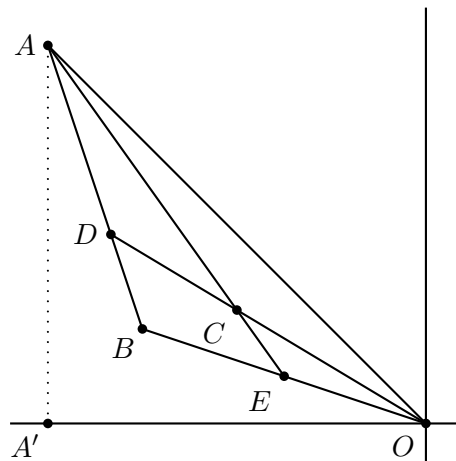


Figure 4: Parametrization for a general triangle occurring in Algorithm 2, in the approximation of a decreasing function.

this figure, the triangle  $\triangle OAB$  represents the 'original' triangle. Suppose that Algorithm 2 is applied for the approximation of a univariate convex and decreasing function  $y(x)$ . Then, the line  $OA$  is an upperbound of the function  $y(x)$  on the interval  $[x_A, 0]$ . Suppose that there is a data point  $P$  on the left hand side of data point  $A$ , and that there is a data point  $Q$  on the right

hand side of the data point  $O$ . Then, both  $PA$  and  $OQ$  are lower bounds for the function  $y(x)$  on the interval  $[x_A, 0]$ . We denote the point where both lines intersect by  $B$ .

Since the  $y(x)$  is convex and decreasing, we have for the coordinates of point  $P$  that  $x_P < x_A$  and  $y_P \geq \frac{y_A}{x_A}x_P$ . Similarly, for data point  $Q$  we have that  $x_Q > 0$  en  $0 \geq y_Q \geq \frac{y_A}{x_A}x_Q$ . From this it follows directly that  $B$  lies inside the triangle  $\Delta OAA'$ , where  $A'$  is the projection of  $A$  onto the  $x$ -axis, i.e.,  $x_{A'} = x_A$ , and  $y_{A'} = 0$ .

We parameterize the  $x$ -coordinate of point  $B$  as  $x_B = \alpha x_A$ , where  $0 \leq \alpha \leq 1$ . The  $y$ -coordinate of point  $B$  is determined by the upper bound  $OA$ :  $y_B = \beta y_A$ , where  $0 \leq \beta \leq \alpha$ .

In Figure 4, the point  $C$  denotes a new data point. Since we use the Interval bisection partitioning rule, the point  $C$  has a fixed  $x$ -coordinate:  $x_C = \frac{1}{2}x_A$ . Its  $y$ -coordinate lies between the upper bound  $OA$  and the lower bound, which is the line  $OB$  if  $\frac{1}{2} \leq \alpha \leq 1$  or  $AB$  if  $0 \leq \alpha \leq \frac{1}{2}$ . We parameterize  $C$  as

$$y_C = \gamma y_A, \quad \text{with } \begin{cases} \frac{1}{2}\frac{\beta}{\alpha} \leq \gamma \leq \frac{1}{2}, & \text{if } \frac{1}{2} \leq \alpha \leq 1 \\ \frac{1}{2}\frac{1-2\alpha+\beta}{1-\alpha} \leq \gamma \leq \frac{1}{2}, & \text{if } 0 \leq \alpha \leq \frac{1}{2}. \end{cases} \quad (29)$$

We can also write  $\gamma$  in (29) as

$$\gamma = \begin{cases} \eta\frac{1}{2} + (1-\eta)\frac{1}{2}\frac{\beta}{\alpha}, & 0 \leq \eta \leq 1, & \text{if } \frac{1}{2} \leq \alpha \leq 1 \\ \eta\frac{1}{2} + (1-\eta)\frac{1}{2}\frac{1-2\alpha+\beta}{1-\alpha}, & 0 \leq \eta \leq 1, & \text{if } 0 \leq \alpha \leq \frac{1}{2}. \end{cases} \quad (30)$$

The points  $B$  and  $C$  are now fully parameterized by  $\alpha$ ,  $\beta$ , and  $\eta$ .

If  $y_C$  is fixed, the line  $AC$  is a new upper bound for the interval  $[x_A, x_C]$ , the line  $OC$  is a new upper bound for the interval  $[x_C, 0]$ . The new lower bounds for these two intervals are the lines  $AD$ ,  $CD$  en  $CE$ ,  $OE$ , where  $D$  is defined as the intersection point of  $AB$  and  $OC$  and the point  $E$  is defined as the intersection point of  $AC$  and  $OB$ . See also Figure 4.

It is easy to verify that the coordinates of point  $D$  are given by:

$$\begin{aligned} x_D &= \frac{\beta - \alpha}{2\gamma(1 - \alpha) + \beta - 1} x_A \\ y_D &= \frac{2\gamma(\beta - \alpha)}{2\gamma(1 - \alpha) + \beta - 1} y_A. \end{aligned}$$

Similarly, it is easy to verify that the coordinates of point  $E$  are given by:

$$\begin{aligned} x_E &= \frac{1 - 2(1 - \gamma)}{\frac{\beta}{\alpha} - 2(1 - \gamma)} x_A \\ y_E &= \frac{\beta}{\alpha} \cdot \frac{1 - 2(1 - \gamma)}{\frac{\beta}{\alpha} - 2(1 - \gamma)} y_A. \end{aligned}$$

We denote the area of the 'new' triangles  $\Delta ACD$  and  $\Delta OCE$  by  $A_1$  and  $A_2$  respectively.

By using (13), we can find for the areas  $A_t$ ,  $A_1$  and  $A_2$ :

$$A_t = \frac{1}{2}x_{AY_A}(\beta - \alpha) \quad (31)$$

$$A_1 = \frac{1}{4}x_{AY_A}(2\gamma - 1) \cdot \left( \frac{2(\beta - \alpha)}{2\gamma(1 - \alpha) + \beta - 1} - 1 \right) \quad (32)$$

$$A_2 = \frac{1}{4}x_{AY_A} \left( \frac{\beta}{\alpha} - 2\gamma \right) \cdot \frac{2\gamma - 1}{\frac{\beta}{\alpha} + 2\gamma - 2}. \quad (33)$$

Using (30), (31), (32), and (33), it is easy to show that

$$\frac{A_1}{A_t} = \begin{cases} \frac{1}{2} \cdot \frac{1-\eta}{\alpha} \cdot \frac{1-\eta-\alpha(2-\eta)}{\eta(1-\alpha)-1} & \text{for } \frac{1}{2} \leq \alpha \leq 1 \\ \frac{1}{2} \cdot \frac{\eta(1-\eta)}{2-2\alpha-\eta+\eta\alpha} & \text{for } 0 \leq \alpha \leq \frac{1}{2}, \end{cases}$$

and that

$$\frac{A_2}{A_t} = \begin{cases} \frac{1}{2} \cdot \frac{\eta}{\alpha} \cdot \frac{1-\eta}{2-\eta} & \text{for } \frac{1}{2} \leq \alpha \leq 1 \\ \frac{1}{2} \cdot \frac{1-\eta}{1-\alpha} \cdot \frac{1-2\alpha+\eta\alpha}{1-\eta\alpha} & \text{for } 0 \leq \alpha \leq \frac{1}{2}. \end{cases}$$

Note that  $\frac{A_1}{A_t}$  and  $\frac{A_2}{A_t}$  are independent of  $\beta$  and only depend on  $\alpha$  and  $\eta$ . To prove the lemma we have to show that

$$\frac{A_1 + A_2}{A_t} \leq \frac{1}{2} \quad \text{for } 0 \leq \alpha \leq 1 \text{ and } 0 \leq \eta \leq 1. \quad (34)$$

A plot of  $\frac{A_1+A_2}{A_t}$  as a function of  $\eta$  and  $\alpha$  is given in Figure 5.

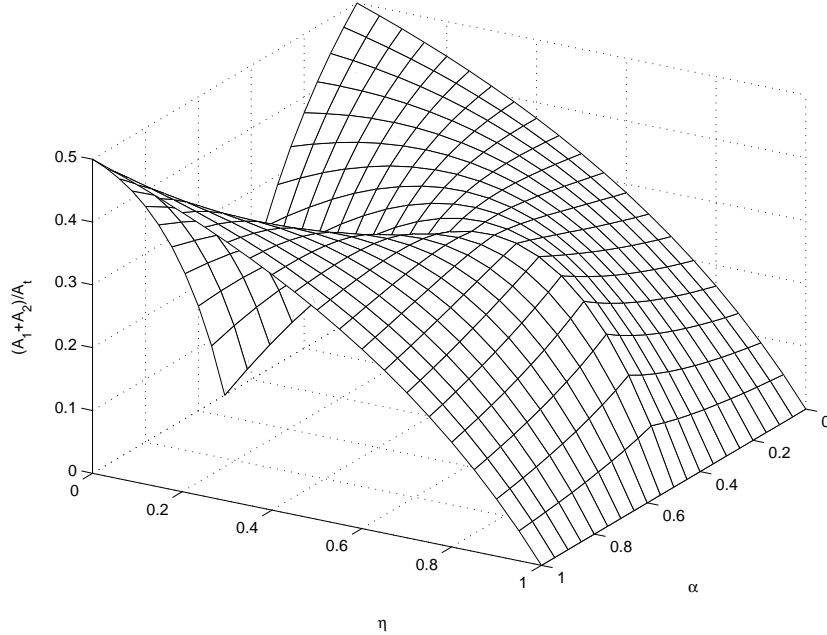


Figure 5:  $\frac{A_1+A_2}{A_t}$  as a function of  $\alpha$  and  $\eta$ .

By rewriting (34), for the case that  $\frac{1}{2} \leq \alpha \leq 1$  and  $0 \leq \eta \leq 1$ , we obtain that we must show that

$$-6\eta + 6\eta^2 - 5\eta^2\alpha - 2\alpha + 5\eta\alpha + 2 - 2\eta^3 + 2\eta^3\alpha + 2\eta\alpha^2 - \eta^2\alpha^2 \geq 0. \quad (35)$$

For the case that  $0 \leq \alpha \leq \frac{1}{2}$  and  $0 \leq \eta \leq 1$ , we must show that

$$2\alpha - 9\eta\alpha + 7\eta^2\alpha - 2\eta^3\alpha + \eta + 2\eta\alpha^2 - \eta^2\alpha^2 \geq 0. \quad (36)$$

First, we consider the case that  $\frac{1}{2} \leq \alpha \leq 1$  and  $0 \leq \eta \leq 1$ . By substituting  $\alpha = (p^2+1/2)/(1+p^2)$  and  $\eta = q^2/(1+q^2)$ ; see also Parrilo and Peretz (2004), we obtain

$$\frac{1}{4} \frac{q^4 + q^6 + 2q^2p^2 + 4q^2p^4 + 6q^4p^2 + 8q^4p^4 + 4q^6p^2 + 4q^6p^4 + 4p^2 + 4}{(1+p^2)^2(1+q^2)^3} \geq 0. \quad (37)$$

Note that (35) holds for all  $\frac{1}{2} \leq \alpha \leq 1$  and  $0 \leq \eta \leq 1$  if and only if (37) holds for all  $p, q \in \mathbb{R}$ . Note that indeed (37) holds for all  $p, q \in \mathbb{R}$ .

Similarly, for the case that  $0 \leq \alpha \leq \frac{1}{2}$  and  $0 \leq \eta \leq 1$ , substituting  $\alpha = p^2/(2(1+p^2))$  and  $\eta = q^2/(1+q^2)$ , we obtain

$$\frac{1}{4} \frac{2q^2p^2 + 6q^4p^2 + 4q^6p^2 + q^4p^4 + 4p^4 + 8q^4 + 4q^6 + q^6p^4 + 4p^2 + 4q^2}{(1+p^2)^2(1+q^2)^3} \geq 0. \quad (38)$$

Again, (36) holds for all  $0 \leq \alpha \leq 1/2$  and  $0 \leq \eta \leq 1$  if and only if (38) holds for all  $p, q \in \mathbb{R}$ . Note that indeed (38) holds for all  $p, q \in \mathbb{R}$ .  $\square$

Note that due to symmetry, Theorem 6 also holds if  $y(x)$  is increasing. Using Theorem 6, we can also show that Algorithm 2 converges at least linearly using the Uncertainty area as error measure and the Interval bisection partitioning rule. Suppose that we add the data points such that we halve the areas of all triangles, instead of choosing the interval with the largest area of uncertainty. In this way, the rate of convergence can only become smaller. Suppose that we need  $k$  halvings to make the total area smaller than the prescribed  $\delta$ , then:

$$A_t^k \leq \left(\frac{1}{2}\right)^k A_t^0 < \delta, \quad (39)$$

where  $A_t^k$  is the area of uncertainty after  $k$  halvings, and  $A_t^0$  the initial area of uncertainty. Note that  $k$  halvings require  $N = \sum_{i=1}^k 2^{i-1} = 2^k - 1$  function value evaluations. Substituting this into (39), we obtain:

$$\begin{aligned} \frac{1}{N+1} A_t^0 &< \delta \\ \frac{A_t^0}{\delta} - 1 &< N \end{aligned}$$

Therefore, at most

$$N = \frac{A_t^0}{\delta} - 1$$

iterations are needed to obtain a total Uncertainty area smaller than  $\delta$ .

## 5 Numerical examples

In this section we treat some numerical examples to illustrate the methodology proposed in this paper.

### Example 5.1 (Artificial data)

In this example we apply four different iterative methods that we discussed in Section 3.2, and we compare them with the case that we choose the input variables equidistantly. In the first method, we use the Interval bisection rule in combination with the Maximum error measure. In the second method, we use the Interval bisection rule in combination with the Hausdorff distance error measure. In the third method, we select the new point such that the average Uncertainty area after addition is minimized, i.e., the value of  $x$  that solves optimization problem (11), and in the fourth method we select the new point such that the worst-case Uncertainty area is maximized, i.e., the value of  $x$  that solves optimization problem (12).

We consider the approximation of the function  $y(x) = \frac{1}{x}$  on the interval  $[0.2, 5]$ . As initial dataset we take two data points:  $(0.2, 5)$  and  $(5, 0.2)$ . In Figure 6 the upper and lower bounds after several iterations for the worst-case area method, are given. We measure the Maximum error, the Uncertainty area, and the Hausdorff distance after each iteration. The results are shown in Table 1. As expected, all four new methods give better results than when we use the equidistant approach. Furthermore, as expected, if we use the Maximum error or the Hausdorff distance as measure to select a new point, the Maximum error or the Hausdorff distance respectively, in general decreases quicker than if we use the other criteria. Also, if we use the Average area rule or the Worst-case area rule, the total area decreases quicker than if we use the Maximum error measure.

Next, we again approximate the function  $y(x) = \frac{1}{x}$ , but now only on the interval  $[1, 2]$  with the points  $(1, 1)$  and  $(2, 0.5)$  as initial dataset. The results are given in Table 2. We can see from this table that in this case, if we look to the area, choosing the inputs equidistantly does not perform significantly worse than the four more sophisticated methods. This could be explained by the shape of the two different functions that are to be approximated. On the interval  $[0.2, 5]$ , the function has much more curvature than on the interval  $[1, 2]$ . However, if we look at the Maximum error and the Hausdorff distance, our four new methods perform better than the equidistant approach.

□

### Example 5.2 (Strategic investment model)

In this example we consider a Strategic investment model. There exist many sorts of investment

it.	ME			H			MAA		
	ME	UA	H	ME	UA	H	ME	UA	H
0	4.80	11.52	3.39	4.80	11.52	3.39	4.80	11.52	3.39
1	4.43	5.53	2.04	4.43	5.53	2.04	4.23	3.77	1.35
2	3.96	2.67	1.07	3.96	2.67	1.07	3.11	1.65	0.51
3	3.21	1.33	0.51	3.21	1.33	0.51	2.62	1.35	0.35
4	2.25	0.74	0.22	2.25	0.74	0.22	2.62	1.03	0.35
5	1.29	0.52	0.20	1.29	0.52	0.20	1.48	0.61	0.25
6	0.58	0.45	0.20	1.29	0.44	0.15	1.48	0.46	0.25
7	0.32	0.43	0.20	1.29	0.35	0.11	1.03	0.37	0.25
8	0.25	0.38	0.17	1.29	0.26	0.11	1.03	0.28	0.11
9	0.23	0.37	0.17	1.29	0.23	0.09	1.03	0.23	0.11
it.	MMA			equidistant					
	ME	UA	H	ME	UA	H			
0	4.80	11.52	3.39	4.80	11.52	3.39			
1	4.20	3.62	1.27	4.43	5.53	2.04			
2	2.99	1.64	0.53	4.18	3.52	1.42			
3	1.43	1.17	0.48	3.96	2.54	1.07			
4	1.43	0.72	0.43	3.75	1.95	0.85			
5	1.43	0.50	0.15	3.56	1.57	0.70			
6	1.43	0.40	0.15	3.38	1.30	0.59			
7	1.14	0.31	0.13	3.21	1.09	0.51			
8	0.41	0.25	0.13	3.06	0.94	0.44			
9	0.41	0.20	0.13	2.92	0.82	0.39			

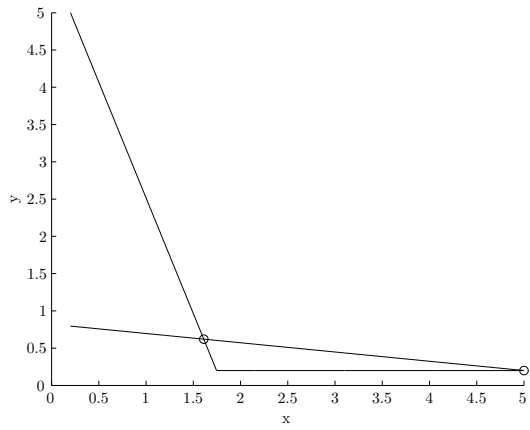
Table 1: Maximum error (ME), total Uncertainty area (UA), and Hausdorff distance (H) after each iteration in Example 5.1 on interval  $[0.2,5]$  using the Maximum error, Uncertainty area, minimal average area (MAA), minimal maximal area (MMA), and equidistant iterative strategies.

it.	ME			H			MAA		
	ME	UA	H	ME	UA	H	ME	UA	H
0	0.5000	0.2500	0.4472	0.5000	0.2500	0.4472	0.5000	0.2500	0.4472
1	0.1667	0.0625	0.1387	0.1667	0.0625	0.1387	0.1317	0.0638	0.1230
2	0.0667	0.0275	0.0593	0.0667	0.0275	0.0593	0.0932	0.0269	0.0741
3	0.0625	0.0206	0.0593	0.0667	0.0154	0.0521	0.0712	0.0182	0.0566
4	0.0222	0.0087	0.0181	0.0222	0.0087	0.0181	0.0302	0.0103	0.0289
5	0.0205	0.0075	0.0181	0.0222	0.0066	0.0172	0.0215	0.0062	0.0159
6	0.0179	0.0054	0.0172	0.0222	0.0046	0.0166	0.0145	0.0044	0.0108
7	0.0110	0.0035	0.0103	0.0110	0.0035	0.0103	0.0145	0.0033	0.0107
8	0.0080	0.0025	0.0065	0.0080	0.0025	0.0065	0.0145	0.0026	0.0107
9	0.0065	0.0020	0.0051	0.0065	0.0020	0.0051	0.0112	0.0022	0.0083

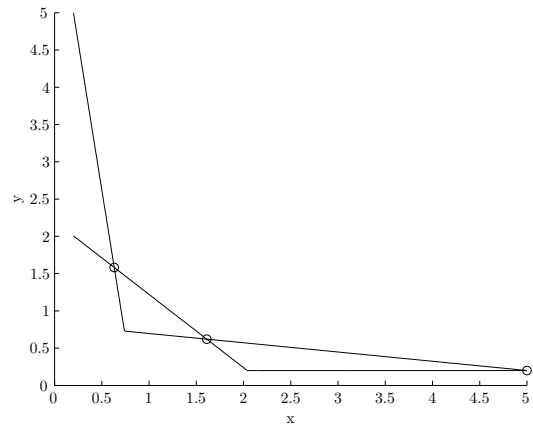
  

it.	MMA			equidistant		
	ME	UA	H	ME	UA	H
0	0.5000	0.2500	0.4472	0.5000	0.2500	0.4472
1	0.1376	0.0651	0.1283	0.1667	0.0625	0.1387
2	0.0852	0.0263	0.0673	0.1000	0.0278	0.0800
3	0.0430	0.0161	0.0411	0.0667	0.0154	0.0521
4	0.0263	0.0091	0.0237	0.0476	0.0097	0.0366
5	0.0241	0.0059	0.0180	0.0357	0.0066	0.0271
6	0.0124	0.0046	0.0110	0.0278	0.0048	0.0209
7	0.0124	0.0036	0.0102	0.0222	0.0036	0.0166
8	0.0087	0.0027	0.0080	0.0182	0.0028	0.0135
9	0.0070	0.0021	0.0054	0.0152	0.0023	0.0112

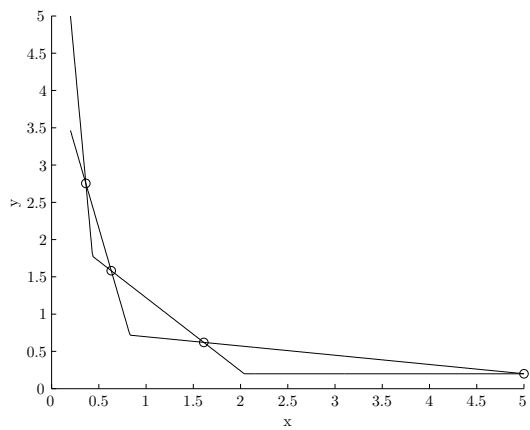
Table 2: Maximum error (ME), total Uncertainty area (UA), and Hausdorff distance (H) after each iteration in Example 5.1 on interval  $[1,2]$  using the Maximum error, Uncertainty area, minimal average area (MAA), minimal maximal area (MMA), and equidistant iterative strategies.



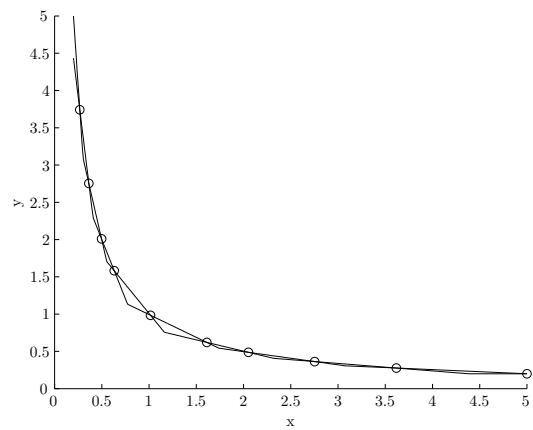
(a) iteration 2



(b) iteration 3



(c) iteration 4



(d) iteration 10

Figure 6: Upper and lower bounds of the function  $y = 1/x$  on the interval  $[0.2, 5]$  after several iterations of the iterative strategy selecting a new input point according to the worst-case area rule.



Category	$i$	$ER_i$	$\text{Cov}[R_i, R_j]$		
			$j$		
			1	2	3
stocks	1	10.8	2.250	-0.120	0.450
bonds	2	7.600	-0.120	0.640	0.336
real estate	3	9.500	0.450	0.336	1.440

Table 3: Expected returns and covariances.

categories, such as deposits, saving accounts, bonds, stocks, real estate, commodities, foreign currencies, and derivatives. Each category has its own expected return, and its own risk characteristic. The Strategic investment model models how top management could spread an overall budget over several investment categories. The objective is to minimize the portfolio risk (measured by the variance of the return), such that a certain minimal desired expected return is achieved. The model was introduced by Markowitz (1952), and is given by:

$$\begin{aligned}
y(M) := \min_x \quad & x^T V x \\
\text{s.t.} \quad & r^T x \geq M \\
& e_p^T x = 1 \\
& x \in \mathbb{R}_+^p,
\end{aligned} \tag{40}$$

where  $V$  is a positive semi-definite covariance matrix consisting of elements  $V_{ij}$  of covariances between investment categories  $i$  and  $j$ ,  $r$  is the vector consisting of elements  $r_i$  of expected return of investment category  $i$ ,  $M$  is the desired expected portfolio return,  $e_p$  is the  $p$ -dimensional all-one vector,  $x$  is the vector with elements  $x_i$  of fractions of the budget invested in each category, and  $p$  is the number of investment categories.

In Table 3, some data is given, which we took from Bisschop (2000). It contains three investment categories: stocks, bonds, and real estate. The stochastic variable  $R_i$  denotes the return of investment category  $i$ .

The optimum in (40), can be seen as a function  $y(M)$ . It can be shown that  $y$  is convex and increasing. We carried out the same experiment as in Example 5.1. We applied the same four different iterative strategies and calculated the Maximum error, the Uncertainty area, and the Hausdorff distance after each iteration. We compared the results with the case that we choose the input data points equidistantly. The results are given in Table 4. As we could expect, we can see from Table 4, all of the four iterative strategies perform better than when we choose the input data points equidistantly. In Figure 7, the upper and lower bounds are shown after iteration 9 of the Sandwich algorithm using the Hausdorff distance as error measure.  $\square$

## 6 Conclusions and further research

In this paper we proposed piecewise linear upper and lower bounds for approximation of univariate convex functions. For the approximation of univariate convex functions we can construct

it.	ME			H			MAA		
	ME	UA	H	ME	UA	H	ME	UA	H
0	1.8018	2.8828	1.5700	1.8018	2.8828	1.5700	1.8018	2.8828	1.5700
1	1.5347	1.3261	1.0624	1.5348	1.3261	1.0624	1.1625	0.9926	0.6849
2	0.7847	0.5060	0.4287	0.7847	0.5060	0.4287	0.7899	0.5915	0.4654
3	0.4606	0.2763	0.1974	0.4606	0.2763	0.1974	0.4108	0.3279	0.1944
4	0.1738	0.2024	0.1522	0.1738	0.2024	0.1522	0.4108	0.2011	0.1674
5	0.1321	0.1454	0.1087	0.1321	0.1454	0.1087	0.2538	0.1350	0.1034
6	0.1090	0.1337	0.1087	0.1321	0.0911	0.0835	0.2538	0.1003	0.1034
7	0.0897	0.0794	0.0835	0.1321	0.0675	0.0648	0.0865	0.0689	0.0674
8	0.0846	0.0644	0.0835	0.1321	0.0524	0.0501	0.0646	0.0531	0.0361
9	0.0565	0.0409	0.0372	0.0565	0.0409	0.0372	0.0646	0.0392	0.0361

it.	MMA			equidistant		
	ME	UA	H	ME	UA	H
0	1.8018	2.8828	1.5700	1.8018	2.8828	1.5700
1	1.1203	0.9870	0.6450	1.5347	1.3261	1.0624
2	0.4994	0.5771	0.3590	1.0251	0.6979	0.6190
3	0.4994	0.2672	0.2162	0.7847	0.4397	0.4287
4	0.1792	0.1824	0.1618	0.6734	0.3064	0.3384
5	0.1137	0.1192	0.0852	0.6065	0.2298	0.2838
6	0.1137	0.0835	0.0852	0.5334	0.1779	0.2371
7	0.1100	0.0609	0.0492	0.4606	0.1407	0.1974
8	0.0676	0.0494	0.0492	0.3935	0.1127	0.1639
9	0.0676	0.0384	0.0377	0.3332	0.0911	0.1358

Table 4: Maximum error (ME), Uncertainty area (UA), and Hausdorff distance (H) after each iteration in Example 5.2 using the Maximum error, Uncertainty area, minimal average area (MAA), minimal maximal area (MMA), and equidistant iterative strategies.

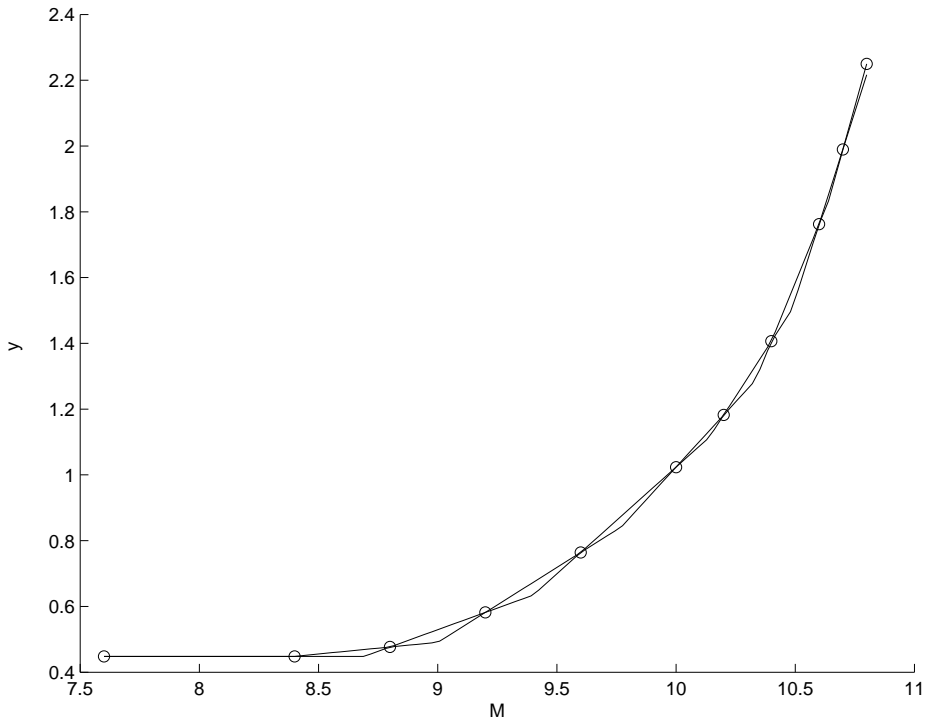


Figure 7: Upper and lower bounds of the function  $y(M)$  on the interval  $[7.6, 10.8]$  after iteration 9 of the Sandwich algorithm of using the Hausdorff distance for Example 5.2.

linear upper and lower bounds, based on function value evaluations only. These bounds can be given explicitly. The difference between the upper and lower bounds can be seen as a measure of accuracy. We may use so-called Sandwich algorithms to select new input values to be evaluated, to obtain good approximations. We introduced a new variant of the Sandwich algorithm, and also, we introduced two new iterative strategies, which minimize the area of uncertainty of the approximation. It can be shown that our new Sandwich algorithms that do not use derivative information are of order  $O(1/n^2)$  for the 1-norm, the  $\infty$ -norm, and the Hausdorff distance. These results require assumptions on the derivatives of  $y(x)$ . If these assumptions do not hold, it can be shown that under other conditions we have  $O(1/n)$  convergence for these Sandwich algorithms. We applied these new algorithms to an artificial example and a practical example. It turned out that our algorithms perform better than when we choose the input data points equidistantly. This is especially the case if the function to be approximated has much curvature.

For further research we are interested in generalizing this methodology to more dimensions, i.e., approximating functions of two or more variables. This is partly done in Siem et al. (2006).

## References

Bisschop, J. (2000). AIMMS optimization modeling. Technical report, Haarlem.

Boef, E. den and D. den Hertog (2007). Efficient line searching for convex functions. *SIAM*

*Journal on Optimization*. To appear.

- Burkard, R.E., H.W. Hamacher, and G. Rote (1991). Sandwich approximation of univariate convex functions with an application to separable convex programming. *Naval Research Logistics*, **38**, 911–924.
- Fruhworth, B., R.E. Burkard, and G. Rote (1989). Approximation of convex curves with application to the bi-criteria minimum cost flow problem. *European Journal of Operational Research*, **42**, 326–338.
- Guérin, J., P. Marcotte, and G. Savard (2006). An optimal adaptive algorithm for the approximation of concave functions. *Mathematical Programming* 107(3), 357–366.
- Hoffmann, A.L., A.Y.D. Siem, D. den Hertog, J.H.A.M. Kaanders, and H. Huizenga (2006). Derivative-free generation and interpolation of convex Pareto optimal IMRT plans. *Physics in Medicine and Biology*, **51**, 6349–6369.
- Markowitz, H.M. (1952). Portfolio selection. *Journal of Finance*, **7**, 77–91.
- Parrilo, P.A. and R. Peretz (2004). An inequality for circle packings proved by semidefinite programming. *Discrete and Computational Geometry*, **31**, 357–367.
- Rote, G. (1992). The convergence rate of the Sandwich algorithm for approximating convex functions. *Computing*, **48**, 337–361.
- Siem, A.Y.D., D. den Hertog, and A.L. Hoffmann (2006). Multivariate convex approximation and least-norm convex data-smoothing. In M. Gavrilova, O. Gervasi, C.J.K. Tan, D. Taniar, A. Lagunà, Y. Mun, and H. Choo (Eds.), *ICCSA 2006*, Lecture Notes in Computer Science, Berlin, pp. 812–821. Springer.
- Siem, A.Y.D., D. den Hertog, and A.L. Hoffmann (2007). The effect of transformations on the approximation of univariate (convex) functions with applications to Pareto curves. *European Journal of Operational Research*. To appear in *European Journal of Operational Research*.
- Siem, A. Y. D., E. de Klerk, and D. den Hertog (2007). Discrete least-norm approximation by nonnegative (trigonometric) polynomials and rational functions. *Structural and Multidisciplinary Optimization*. To appear.
- Yang, X.Q. and C.J. Goh (1997). A method for convex curve approximation. *European Journal of Operational Research*, **97**, 205–212.