Tools for Simplicial Branch and Bound in Interval Global Optimization*

Sam Karhbet
University of Louisiana at Lafayette
skarhbet@louisiana.edu

Ralph Baker Kearfott
University of Louisiana at Lafayette
rbk@louisiana.edu

November 29, 2016

Abstract

Most branch and bound (B&B) algorithms for continuous global optimization work with hyper-rectangles, although some work in the 1970's dealt with simplexes. More recently, Žilinskas et al have considered branch and bound for Lipschitz optimization, giving examples of how symmetry can be used and how algorithms can be made efficient. Here, in the spirit of that work, we consider simplex-based branch and bound algorithms in which mathematically rigorous ranges on functions and constraints are computed using interval arithmetic. We tie the subdivision process neatly to the actual geometry, and we give formulas for reasonably tight bounds on ranges.

1 Introduction

We consider the general global optimization problem

$$\begin{aligned} &\text{minimize } \varphi(x) \\ &\text{subject to } c_i(x) = 0, i = 1, \ldots, m_1, \\ &\qquad\qquad g_i(x) \le 0, i = 1, \ldots, m_2, \\ &\text{where } \varphi : \mathbb{R}^n \to \mathbb{R} \text{ and } c_i, g_i : \mathbb{R}^n \to \mathbb{R}, \end{aligned} \tag{1}$$

A common general deterministic approach, that is, a class of "complete" algorithms [16] for finding the global optimum is the class of branch and bound

*Submitted: (insert date); Revised: (insert date); Accepted:(insert date).

(B&B) methods, in which an initial domain is adaptively subdivided, and each sub-domain is analyzed. This approach has the structure outlined in Algorithm 1. One fairly general analysis of such an algorithm is [11].

Input : An initial region $\mathcal{D}^{(0)}$, the objective φ, the constraints C, a domain stopping tolerance ϵ_d, and a limit M on the maximum number of regions to be allowed to be processed.

Output: `OK` $=$ `true` and the best upper bound $\overline{\varphi}$ for the global optimum, and the list $\mathcal{C}$ within which all optimizing points must lie, if the algorithm completed with less than M regions considered, and `OK` $=$ `false` if the algorithm could not complete.

1 Initialize the list $\mathcal{L}$ of regions to be processed to contain $\mathcal{D}^{(0)}$;
2 Determine an upper bound $\overline{\varphi}$ on the global optimum;
3 $i \leftarrow 1$;
4 **while** $\mathcal{L} \neq \emptyset$ **do**
5     $i \leftarrow i + 1$;
6   **if** $i > M$ **then return** *OK* $=$ *false*;
7   Remove a region $\mathcal{D}$ from $\mathcal{L}$;
8   **Bound:** Determine if $\mathcal{D}$ is not infeasible, and if it is not proven to be infeasible, determine a lower bound $\underline{\varphi}$ on φ over the feasible part of $\mathcal{D}$;
9   **if** *$\mathcal{D}$ is infeasible or* $\underline{\varphi} > \overline{\varphi}$ **then** return to Step 7;
10   Possibly compute a better upper bound $\overline{\varphi}$;
11   **if** *a scaled radius diam of $\mathcal{D}$ satisfies diam*$(\mathcal{D}) < \epsilon_d$ **then**
12   Store $\mathcal{D}$ in $\mathcal{C}$;
13   Return to Step 7;
14   **else**
15   **Branch:** Split $\mathcal{D}$ into two or more sub-regions whose union is $\mathcal{D}$;
16   Put each of the sub-regions into $\mathcal{L}$;
17   Return to Step 7;
18   **end**
19 **end**
20 **return** *OK* $=$ *true*, $\overline{\varphi}$, *and* $\mathcal{C}$ *(possibly empty)*;

Algorithm 1: General Branch and Bound Structure

The predominant shape of region $\mathcal{D}$ in branch and bound algorithms following the structure of Algorithm 1 is a *box* $\boldsymbol{x}$, that is, a rectangular parallelepiped defined by independent lower bounds $\underline{x}_i$ and upper bounds $\overline{x}_i$ on each coordinate,

$$\boldsymbol{x} = \{x \in \mathbb{R}^n \mid \underline{x}_i \leq x_i \leq \overline{x}_i, 1 \leq i \leq n\},$$

while the predominant method of branching has been bisection of one of the coordinate directions, say the i-th one, to form two new boxes, one for which

$\underline{x}_i \le x_i \le (\underline{x}_i + \overline{x}_i)/2$ and one for which $(\underline{x}_i + \overline{x}_i)/2 \le x_i \le \overline{x}_i$. This subdivision method has been popular because of its simplicity, because a sub-region provides clear error bounds on individual parameters, and because bounds on the coordinates occur naturally in many problems.

However, other region shapes also have advantages and have been considered. One alternative region is an *n-simplex*

$$\mathcal{S} = \langle P_0, P_1, \ldots, P_n \rangle ,$$

the convex hull of $n+1$ points $P_i \in \mathbb{R}^n$ that do not lie in a hyperplane of dimension less than n. (For example, 2-simplexes are triangles, while 3-simplexes are tetrahedra.) Proceeding from century-old work at the foundation of algebraic topology, namely the use of simplicial complexes to approximate manifolds [1], Stenger [23] proposed a B&B method for computing the topological degree, a method subsequently enhanced with more elegant formulas by Stynes [25, 26] and the present author [7, 8, 9, 10]. Although designed to numerically calculate the topological degree, the basic process in these was a B&B algorithm for finding all solutions to certain related nonlinear systems of equations, using a heuristic to decide when to branch. An advantage of simplexes in that context was the simple relationship between an n-simplex and its boundary. This author nonetheless moved away from using simplexes, in favor of boxes, because of the difficulty of enclosing large volumes with simplexes for n larger and because boxes are more natural when dealing with individual coordinate bounds, and when bounding ranges with interval arithmetic.

Due to certain advantages, simplexes as domains in B&B algorithms have received renewed scrutiny in more recent work. For example, the feasible region is a subset of a simplex if the variables satisfy

$$\text{non-negativity constraints} \quad x_i \ge 0, \quad 1 \le i \le n \tag{2}$$

and a

$$\text{normalization condition} \quad \sum_{i=1}^{n} x_i = 1. \tag{3}$$

Similarly, if the problem is symmetric in the sense that, if, in addition to non-negativity constraints, $(x_1^*, \ldots, x_n^*)$ is an optimizer, then the point obtained by switching two parameters $x_i^* \leftrightarrow x_j^*$ is also an optimizer, we may impose

$$\text{symmetry-breaking constraints } 0 \le x_1 \le x_2 \le \cdots \le x_n, \tag{4}$$

and the resulting feasible set is a subset of a simplex. There are numerous variations of these conditions, such as when some of the coordinates satisfy non-negativity and normalization constraints. Such conditions in the context of simplex-based B&B have been considered recently by Žilinskas et al [18, 19, 20, 28, 30], while the Paulavčius and Žilinskas monograph [17] brings together these results.

Paulavčius and Žilinskas consider a standard (Delaunay) triangulation of a hyper-rectangle $\boldsymbol{x} \in \mathbb{R}^n$ into $n!$ simplexes whose vertices are subsets of the set of 2^n extreme points of $\boldsymbol{x}$, and use any linear constraints to eliminate those simplexes in the triangulation that are infeasible, prior to the branch and bound process. They give details for efficient B&B algorithms based on subdividing the longest edge of the simplex in the branching step and use of heuristic estimation of Lipschitz constants used in the bounding step. In particular, two function evaluation / branching schemes are proposed and implemented in the branching and bounding steps: evaluation at the barycenters of the simplexes, forming three sub-simplexes by trisecting the longest edge along with evaluation at barycenters of faces, or forming two sub-simplexes by bisecting the longest edge along with evaluation at vertices of simplexes.

Here, in the spirit of Paulavčius and Žilinskas, we investigate simplex-based B&B algorithms in contexts in which the feasible region is best described by a simplex. However, we develop representations that allow interval evaluations (and hence mathematically rigorous bounds on ranges) with relatively small overestimation. In §2, we present an appropriate way to represent the simplexes, we derive an associated relatively sharp mean-value type interval extension, and we present our branching process related to these representations.

A possible alternative, particular to polynomial and rational functions, to the more general techniques presented here, is use of Bernstein polynomial representations over simplices, since the range of the polynomial is bounded by the Bernstein coefficients. Nataraj et al (see [21] etc.) have used this property in B&B algorithms in which $\mathcal{D}$ is a box, and they have proposed relatively efficient ways of computing the coefficients, as in [22]. Meanwhile, Garloff et al, perhaps starting with [4] with Bernstein coefficients over a simplex but continuing to the recent analysis of Bernstein coefficients over a simplex [2, 6, 27], have shown that computation of Bernstein coefficients over a simplex can be done as simply as over a box. We will compare the sharpness and efficiency of this alternative in future work.

2 Representations and Subdivisions

The non-negativity constraints (2) combined with the normalization constraint (3) define one kind of simplex, whereas non-negativity combined with the symmetry-breaking constraints (4) and an upper bound on x_n defines another kind of simplex. The non-negativity constraints are common in practical problems, while the normalization constraint, if not explicit, is satisfied, with variable scaling, by any linear constraint all of whose coefficients are non-zero and of the same sign. Both symmetry-breaking and normalization may be present, in which case the set of all x satisfying both (2) and (3) is simply the $n-1$ dimensional simplex in $\mathbb{R}^n$ whose vertices are the coordinate vectors e_i, $1 \le i \le n$, that is, the convex hull of the points $\{e_i\}_{i=1}^n \subset \mathbb{R}^n$, a structure that can be used.

Any symmetry or normalization conditions are incorporated into the description of the initial simplex, with the feasible region defined by (2) or (3) consisting precisely of that simplex; this is easily achievable. Thus, if we subdivide this simplex, any resulting sub-simplexes will also satisfy these constraints, just as all sub-boxes of a box satisfying bound constraints satisfy the bound constraints, and we wish to take advantage of special methods for subdividing simplexes. Ideally, for computational purposes we would choose a representation for this simplex that simultaneously sharply describes the geometrical region (without overestimation), leads to little overestimation when used in interval evaluation, and allows sharp description of the subregions formed from branching, using the same fixed coordinate system. There are various possibilities, although none we have envisioned to-date leads simultaneously to a sharp description of the geometry, easy branching, and little additional overestimation in the interval evaluations.

Here, we have elected to represent each simplex in the subdivision locally, with its own barycentric coordinates, and we use a special scheme to limit overestimation. Our technique is general, not depending on how the initial simplex is constructed, and is relatively computationally efficient. A simple implementation stores $\mathcal{O}(n^2)$ information for each element in the list $\mathcal{L}$ in the branch and bound process, but, using ideas and data structures such as in [7], this storage can be reduced at the cost of additional overhead in processing the $\mathcal{L}$ and possible additional difficulties in parallelization.

2.1 Notation

Let $\mathcal{S}$ denote an arbitrary simplex. The *vertex representation* of $\mathcal{S}$ is written

$$\mathcal{S} = \langle P_0, \ldots, P_n \rangle \tag{5}$$

where the P_i are the set of n-vectors $P_i = (p_{i,1}, \ldots, p_{i,n}\}_{i=0}^{n} \subset \mathbb{R}^n$ consisting of the vertices of $\mathcal{S}$; that is, $\mathcal{S}$ consists of the convex hull of the $n+1$ points P_i.

Given a simplex $\mathcal{S}$, the (i,j)-th *edge* $\sigma_{i,j}$ of $\mathcal{S}$ is the convex hull of P_i and P_j, that is,

$$\sigma_{i,j} = \{tP_i + (1-t)P_j \mid 0 \le t \le 1\}. \tag{6}$$

A simplex $\mathcal{S} = \langle P_0, \ldots, P_n \rangle$ consists precisely of the set of points

$$\mathcal{S} = \left\{ \sum_{i=0}^{n} \mu_i P_i \quad \middle| \quad \sum_{i=0}^{n} \mu_i = 1,\ \mu_i \ge 0,\ 0 \le i \le n \right\}, \tag{7}$$

where the μ_i are the *barycentric coordinates.*

We denote the unit cube $[0,1]^n$ by $\mathcal{C}^{(n)}$ and we denote $\mathcal{C}^{(n)} \setminus (0,\ldots,0)$ by $\mathcal{C}_0^{(n)}$.

We use the common notation for intervals and interval vectors, as [13].

2.2 Interval Bounds for a Function over a Simplex

The set of points in a simplex $\mathcal{S} \subset \mathbb{R}^n$ is described sharply using barycentric coordinates. However, objectives and constraints are most commonly expressed in terms of rectangular coordinates, corresponding to hyper-rectangles in $\mathbb{R}^n$, whereas the barycentric coordinates, although sharply describing the simplex, have limits defined by an equality constraint and n inequality constraints. Use of interval arithmetic to obtain ranges of a function f over a simplex would suffer from overestimation, not only due to interval dependency, but also due to evaluation over a region including more than just the simplex. One approach would be to use bounds $\mu_i \in [0, 1]$ along with constraint propagation, but limiting values from the constraint propagation still overestimate in general. Here, we propose a many-to-one nonlinear map $\beta : \mathcal{C}_0^{(n+1)} \to \mathcal{S}$ that is onto $\mathcal{S}$, and we compute relatively sharp bounds on the range $f(\mathcal{S})$ using special mean-value extensions of $f(\beta(\mathcal{C}_0^{(n+1)}))$ combined with an alternate extension of that portion of $\mathcal{C}^{(n+1)}$ around $(0, \ldots, 0)$.

Given $(\lambda_0, \ldots, \lambda_n) \in \mathcal{C}_0^{(n+1)}$, we map it to the set of barycentric coordinates by

$$(\mu_0(\lambda), \ldots, \mu_n(\lambda)) = \gamma(\lambda) = \frac{1}{\sum_{i=0}^{n} \lambda_i} \big(\lambda_0, \ldots, \lambda_n\big), \tag{8}$$

and define β by

$$\beta(\lambda) = \sum_{i=0}^{n} \mu_i(\lambda) P_i. \tag{9}$$

The image of β is exactly $\mathcal{S}$, although it is not injective. The preciseness of the image gives β the potential to avoid excessive overestimation when using interval arithmetic to bound the range of f over $\mathcal{S}$, while lack of injectivity does not necessarily result in overestimation. Essentially we will use an interval extension of $f \circ \beta$ over a subset of $\mathcal{C}_0^{(n+1)}$, and separate extensions near inverse images of the vertices P_i ($\lambda_i = 0$, $\lambda_j \geq 1/n$ for $j \neq i$).

The basic extension is the well-known

$$\textit{mean value extension: } f(x) \in f(\check{x}) + \sum_{i=1}^{n} \frac{\boldsymbol{\partial f}}{\boldsymbol{\partial x}_i}(\boldsymbol{x})(\boldsymbol{x} - \check{x}) \ \forall x \in \boldsymbol{x} \text{ when } \check{x} \in \boldsymbol{x}, \tag{10}$$

where $\boldsymbol{\partial f}/\boldsymbol{\partial x}_i(\boldsymbol{x})$ is an interval extension for the i-th partial derivative of f over the box $\boldsymbol{x}$. This extension is exact when f is quadratic, and provides sharp bounds to second-order as the diameter of $\boldsymbol{x}$ tends to 0; see, for example, the monograph [15, p. 47] or the text [14, p. 68ff] for additional explanation and references to earlier work.

To obtain reasonable bounds for a function f over a simplex $\mathcal{S}$, we combine (8), (9), and (10). We begin with a box $\boldsymbol{\lambda} = ([\underline{\lambda}_1, \overline{\lambda}_1], \ldots, [\underline{\lambda}_n, \overline{\lambda}_n])$, where $[\underline{\lambda}_i, \overline{\lambda}_i] \subseteq [0, 1]$ for $1 \leq i \leq n$. Observe that γ maps $((0, 1], \ldots, (0, 1])$ in a many-to-one fashion precisely onto the entire set of admissible barycentric coordinates μ, but with a singularity at $(0, \ldots, 0)$.

To reduce overestimation in the range of a function f over $\mathcal{S}$, first make the following assumptions.

Assumption 1. *In bounding f over $\mathcal{S}$:*

1. *Suppose the rectangular coordinates describing $\mathcal{S}$ have been translated so*

the barycenter of $\mathcal{S}$ is at the origin

$$\frac{1}{n+1}\sum_{i=0}^{n} P_i = (0,\ldots,0) \in \mathbb{R}^n.$$

2. *Suppose the range of f has been translated so $f(0,\ldots,0) = 0$.*

With those assumptions, it is advantageous to work directly with the barycentric coordinates μ and a mean-value extension. In particular, we have

$$f\Big(\sum_{i=0}^{n}\mu_i P_i\Big) \in f(0) + \sum_{i=0}^{n}\frac{\partial f}{\partial \mu_i}(\mathcal{S})\mu_i = \sum_{i=0}^{n}\frac{\partial f}{\partial \mu_i}(\mathcal{S})\mu_i, \tag{11}$$

where $\frac{\partial f}{\partial \mu_i}(\mathcal{S})$ is an interval enclosure for $\frac{\partial f}{\partial \mu_i}$ over $\mathcal{S}$. Assuming f is given in terms of the rectangular coordinates $x \in \mathbb{R}^n$ (where the coordinates x are possibly offset to center them on the barycenter of $\mathcal{S}$), we have

$$\frac{\partial f}{\partial \mu_i}(x(\mu)) = \sum_{j=1}^{n}\frac{\partial f}{\partial x_j}(x(\mu))\frac{\partial x_j}{\partial \mu_i}(\mu) = \sum_{j=1}^{n}\frac{\partial f}{\partial x_j}(x(\mu))p_{i,j}, \tag{12}$$

where $p_{i,j}$ is the j-th coordinate of the i-th vertex P_i of $\mathcal{S}$. We let $\boldsymbol{f}_j = [\underline{f}, \overline{f}]$ represent an interval enclosure for $\frac{\partial f}{\partial x_j}$ over $\mathcal{S}$, define

$$\check{f}_j(p) = \begin{cases} \underline{f}_j & \text{if} \quad p \geq 0, \\ \overline{f}_j & \text{if} \quad p < 0, \\ \boldsymbol{f}_j & \text{if} \quad 0 \in p, \end{cases}$$

and combine (11) and (12) to obtain

$$\begin{aligned} f\Big(\sum_{i=0}^{n}\mu_i P_i\Big) &\in \sum_{i=0}^{n}\Big(\sum_{j=1}^{n}\boldsymbol{f}_j p_{i,j}\Big)\mu_i \\ &\subseteq \Big[\sum_{i=0}^{n}\sum_{j=1}^{n} p_{i,j}\check{f}_j(p_{i,j})\mu_i, \quad \sum_{i=0}^{n}\sum_{j=1}^{n} p_{i,j}\check{f}_j(-p_{i,j})\mu_i\Big] \\ &= \Big[\sum_{i=0}^{n} L_i\mu_i, \quad \sum_{i=0}^{n} U_i\mu_i\Big], \end{aligned} \tag{13}$$

where $L_i = \mathrm{Inf}\big(\sum_{j=1}^{n} p_{i,j}\check{f}_j(\mathrm{sgn}(p_{i,j}))\big)$ and $U_i = \mathrm{Sup}\big(\sum_{j=1}^{n} p_{i,j}\check{f}_j(-\mathrm{sgn}(p_{i,j}))\big)$. Thus, given good bounds on the partial derivatives of f with respect to the rectangular coordinates, we can obtain a lower bound for f over $\mathcal{S}$ by solving the linear program

$$\text{minimize} \sum_{i=0}^{n} L_i\mu_i \quad \text{subject to} \sum_{i=0}^{n}\mu_i = 1 \text{ and } \mu_i \geq 0 \text{ for } 0 \leq 0 \leq n, \tag{14}$$

and an upper bound for f over $\mathcal{S}$ can be obtained by solving

$$\text{maximize } \sum_{i=0}^{n} U_i \mu_i \quad \text{subject to } \sum_{i=0}^{n} \mu_i = 1 \text{ and } \mu_i \geq 0 \text{ for } 0 \leq i \leq n. \tag{15}$$

Notice, however, that the minimum in (14) is L_{i_ℓ}, where $i_\ell = \operatorname{argmin}_{0\leq i \leq n} L_i$, and occurs at vertex P_{i_ℓ}, while the maximum in (15) is U_{i^u}, where $i_u = \operatorname{argmax}_{0\leq i\leq n} U_i$, and occurs at vertex P_{i^u}.

In what follows, we use this terminology:

Definition 2.1. *Let $\mathcal{S} \subset \mathbb{R}^n$ be an n-simplex, and let $\boldsymbol{x} \subset \mathbb{R}^n$ be the smallest box enclosing $\mathcal{S}$ (that is, $\boldsymbol{x}$ is the interval hull of $\mathcal{S}$), and let f be a function whose range is to be enclosed over $\mathcal{S}$.*

1. *A* naive extension *or NE of f over $\mathcal{S}$ is the interval evaluation of the expression representing f with vector interval argument $\boldsymbol{x}$.*
2. *The* box mean value extension *or BMVE of f over $\mathcal{S}$ is the mean value extension* (10) *of f over $\boldsymbol{x}$.*
3. *The* simplex mean value extension *or SMVE of f over $\mathcal{S}$ is the set of bounds computed using* (13), (14) *and* (15).
4. *In examples, we compare our naive extension, the BMVE, and the SMVE to the* exact range *of f over $\mathcal{S}$, which in some instance we may compute by rigorously globally optimizing f subject to the set of linear inequality constraints defining $\mathcal{S}$.*
5. *Another alternative to bounding the range of f over $\mathcal{S}$ is to bound the range of f over $\boldsymbol{x}$ using slopes instead of interval extensions of the partial derivatives of f. In our comparisons, we use Hansen's technique [5], the* Hansen interval extension *or HSE.*

Example 1. *Let $f(x_1, x_2) = x_1^2 + x_2^3$, and let $\mathcal{S} = \langle(-1, 0), (\frac{1}{2}, -1), (\frac{1}{2}, 1)\rangle$. $\mathcal{S}$ can be described with the constraints $-\frac{2}{3}x_1 - \frac{2}{3} - x_2 \leq 0$, $-\frac{2}{3}x_1 - \frac{2}{3} + x_2 \leq 0$, $x_1 - \frac{1}{2} \leq 0$, and the range of f over $\mathcal{S}$ can be determined to be* $[-.75, 1.25]$ *(e.g. by optimizing f and $-f$ with a rigorous global optimizer such as GlobSol [12]). The enclosing rectangle for $\mathcal{S}$ is $\boldsymbol{x} = ([-1, \frac{1}{2}], [-1, 1])$, and a naive interval evaluation over $\boldsymbol{x}$ gives* $[-1, 2]$, *whereas a mean value extension over $\boldsymbol{x}$ (BMVE), using naive interval extensions[1] of the partial derivatives over $\boldsymbol{x}$ $\frac{\partial f}{\partial x_1} \in [-2, 1]$, $\frac{\partial f}{\partial x_2} \in [0, 3]$ gives $f \in [-2, 1][-1, 0.5] + [0, 3][-1, 1] = [-4, 5]$, while the HSE, holding x_2 constant in $\frac{\partial f}{\partial x_1}$, gives the same enclosure as the BMVE for this example. In contrast, using* (14) *and* (15) *(the SMVE) gives*

$$\begin{aligned}
L_0 &= (-1)(1) + 0 = -1, \qquad & L_1 &= \tfrac{1}{2}(-2) + (-1)(3) = -4,\\
L_2 &= \tfrac{1}{2}(-2) + (1)(0) = -1, & &
\end{aligned}$$

$$\begin{aligned}
U_0 &= (-1)(-2) + 0 = 2, \qquad & U_1 &= \tfrac{1}{2}(1) + (-1)(0) = \tfrac{1}{2},\\
U_2 &= \tfrac{1}{2}(1) + (1)(3) = 3.5, & &
\end{aligned}$$

[1] which in this case happen to be the exact ranges

giving an enclosure $[-4, 3.5]$ *for the range of* f *over* $\mathcal{S}$*, significantly better than the mean value extension over* $\boldsymbol{x}$ *but not requiring a global optimization. For this example, however, the naive interval extension over* $\mathcal{S}$*, although not the exact range, gives the tightest bounds. Summarizing, we have:*

NE	*BMVE*	*HSE*	*SMVE*	*ER*
$[-1, 2]$	$[-4, 5]$	$[-4, 5]$	$[-4, 3.5]$	$[-.75, 1.25]$

Example 2. *Let* $f(x_1, x_2) = 0.25x_1^2 + x_1 + x_2 + 0.25x_1x_2 + 0.25x_2^3$*, and let* $\mathcal{S}$ *be as in Example 1. Computing as in Example 1, we obtain the following:*

NE	*BMVE*	*HSE*	*SMVE*	*ER*
$[-2.5, 2.25]$	$[-3.375, 2.625]$	*[-2.875, 2.375]*	$[-1.75, 2.625]$	$[-.8125, 1.9375]$

For this function, the SMVE gives a lower bound better than the naive interval extension, but a worse upper bound than the naive interval extension. The HSE gives a lower upper bound than the SMVE in this case because of the form of f *and the position of* $\mathcal{S}$ *relative to the coordinate axes.*

In fact, we expect the SMVE to always be contained in the BMVE, and for the SMVE to be better than the naive interval extension for small diameter $\mathcal{S}$, or more generally when f is approximately linear over $\mathcal{S}$. Depending on f and the shape of $\mathcal{S}$, the SMVE could provide much sharper bounds than the BMVE. (However, depending on $\mathcal{S}$, use of slopes rather than interval derivatives in the BMVE may be superior to the SMVE with interval derivatives.)

Example 3. *Let* f *be as in Example 2, but let*

$$\mathcal{S} = \langle(-0.25, 0), (0.125, -0.25), (0.125, 0.25)\rangle .$$

The bounds, rounded out, are:

NE	*BMVE*	*HSE*	*SMVE*	*ER*
$[-0.520, 0.411]$	$[-0.551, 0.411]$	$[-0.536, 0.403]$	$[-0.282, 0.411]$	$[-0.235, 0.391]$

Example 4. *Here, we look at a simplex none of whose sides are aligned with the coordinate axes. Let* f *be as in Example 2, but let* $\mathcal{S} = \langle(-2, 0), (2, -3), (0, 3)\rangle$*. The bounds are:*

NE	*BMVE*	*HSE*	*SMVE*	*ER*
$[-13.25, 14.25]$	$[-30.25, 30.25]$	$[-28.25, 28.25]$	$[-26.25, 24.75]$	$[-8.25, 9.75]$

If we take a smaller similar simplex, namely

$$\mathcal{S} = \langle(-\tfrac{1}{4}, 0), (\tfrac{1}{4}, -\tfrac{3}{8}), (0, \tfrac{3}{8})\rangle,$$

we obtain the following (rounded out):

NE	*BMVE*	*HSE*	*SMVE*	*ER*
$[-0.662, 0.678]$	$[-0.743, 0.743]$	$[-0.657, 0.657]$	$[-0.305, 0.438]$	$[-.235, .389]$

We observe a tighter enclosure for the SMVE relative to the BMVE, on both ends, although none of the enclosures approximate the exact range well. This is due to the bounds on the partial derivatives of f *over the enclosing box, rather than the simplex.*

To obtain tighter enclosures, we can apply SMVE to the partial derivatives of f, then apply SMVE to f itself using tighter bounds on the partial derivatives. or to the partial derivatives of f, although that requires use of second-order partial derivatives, and, if advantageous, we may use the Hansen slope technique to obtain initial bounds on both f and its partial derivatives.

Example 5. *Let f be as in Examples 2, 3 and 4, and let $\mathcal{S}$ be the smaller simplex in Example 4. Then $\frac{\partial f}{\partial x_1} = \frac{1}{2}x_1 + \frac{1}{4}x_2 + 1$, $\frac{\partial f}{\partial x_2} = \frac{1}{4}x_1 + \frac{3}{4}x_2^2 + 1$, and naive interval evaluations over the box $\boldsymbol{x} = ([-\frac{1}{4}, \frac{1}{4}], [-\frac{3}{8}, \frac{3}{8}])$ give $\frac{\partial f}{\partial x_1} \in [0.781, 1.22]$ and $\frac{\partial f}{\partial x_2} \in [0.937, 1.17]$. Applying SMVE to $\frac{\partial f}{\partial x_1}$ and $\frac{\partial f}{\partial x_2}$ then gives $\frac{\partial f}{\partial x_1} \in [0.875, 1.10]$ and $\frac{\partial f}{\partial x_2} \in [0.789, 1.274]$, giving an improvement in $\frac{\partial f}{\partial x_1}$ but not in $\frac{\partial f}{\partial x_2}$. Intersecting the naive and SMVE values for $\frac{\partial f}{\partial x_1}$ and $\frac{\partial f}{\partial x_2}$ and using them to compute the SMVE for f gives $f \in [-0.274, 0.438]$, a further improvement to the lower bound over the SMVE in Example 4, but no additional improvement to the upper bound.*

The SMVE may be applied recursively to higher-order partial derivatives, resulting in a trade-off between computational work and sharpness of enclosures for f over $\mathcal{S}$.

Definition 2.2. *The recursive SMVE, or RSMVE with k recursions ($k \geq 1$) is the SMVE where partial derivatives used in the SMVE are themselves obtained with the SMVE, up to order k.*

If, for example, f is quadratic, one need of course only apply recursion down to $k = 1$, and the SMVE will give exact bounds on the first order partial derivatives of f over $\mathcal{S}$; however, the RSMVE may still not give exact bounds on the range of f over $\mathcal{S}$.

Example 6. *Let $\mathcal{S}$ be the smaller simplex in Example 4, but let*

$$f(x_1, x_2) = 0.25x_1^2 + x_1 + x_2 + 0.25x_1x_2 + 0.5x_2^2,$$

a quadratic function. We obtain the following (rounded out to 3 digits):

NE	*HSE*	*SMVE*	*RSMVE*	*ER*
$[-0.649, 0.735]$	$[-0.844, 0.844]$	$[-0.344, 0.540]$	$[-0.297, 0.516]$	$[-.235, .446]$

These examples illustrate that the SMVE can help in reducing the overestimation in converting from rectangular to simplicial coordinates, although, as with other interval extensions, it does not completely do away with overestimation. Overestimation for some f is inevitable when a polynomial time algorithm is used to enclose a range.

When the simplex $\mathcal{S}$ does not have barycenter at the origin and when f is not actually zero at the barycenter of $\mathcal{S}$, roundoff error needs to be carefully taken into account, for a mathematically rigorous result, when translating the coordinates of the vertices of $\mathcal{S}$ and when evaluating f at the barycenter.

Algorithm 2 summarizes computation of the SMVE.

Input : A simplex $\mathcal{S} = \langle P_0, \ldots, P_n \rangle \subset \mathbb{R}^n$ and a function $f : \mathcal{S} \to \mathbb{R}$.
Output: Mathematically rigorous bounds $\boldsymbol{f} = [\underline{f}, \overline{f}]$ on the range of f over $\mathcal{S}$ computed with the SMVE scheme.

Determine the bounding box $\boldsymbol{x}$ for $\mathcal{S}$ by examining the components of the P_i;
Compute a mathematically rigorous enclosure $\boldsymbol{b}$ for the barycenter of $\mathcal{S}$;
Form enclosures for the translated coordinates: $\check{\boldsymbol{P}}_i \leftarrow P_i - \boldsymbol{b}$, $0 \le i \le n$, and denote by $\check{\boldsymbol{P}}_{i,j}$, $\underline{\check{P}}_{i,j}$ and $\overline{\check{P}}_{i,j}$ the the enclosure for the j-th coordinate of $\check{\boldsymbol{P}}_i$, its lower, and its and upper bound, respectively;
Compute an enclosure $\boldsymbol{f}_b$ for $f(\boldsymbol{b})$, e.g. by intersecting a naive interval evaluation with a mean value extension over $\boldsymbol{x}$;
Compute enclosures $\boldsymbol{f}_j$ $1 \le j \le n$, for the partial derivatives of f over $\mathcal{S}$;
$\underline{f} \leftarrow \infty$; $\overline{f} \leftarrow -\infty$;
for $i = 0$ *to* n **do**
    $L_i \leftarrow 0$; $U_i \leftarrow 0$;
    for $j = 1$ *to* n **do**
        if $\underline{\check{P}}_{i,j} > 0$ **then** $f_U \leftarrow \overline{f}_j$; $f_L \leftarrow \underline{f}_j$;
        else if $\overline{\check{P}}_{i,j} < 0$ **then** $f_U \leftarrow \underline{f}_j$; $f_L \leftarrow \overline{f}_j$;
        else $f_U \leftarrow \boldsymbol{f}_j$; $f_L \leftarrow \boldsymbol{f}_j$;
        $L_i \leftarrow L_i + \check{\boldsymbol{P}}_{i,j} f_L$; $U_i \leftarrow U_i + \check{\boldsymbol{P}}_{i,j} f_U$;
    end
    $\underline{f} \leftarrow \min\{\underline{f}, L_i\}$; $\overline{f} \leftarrow \max\{\overline{f}, U_i\}$;
end
return $\boldsymbol{f} = [\underline{f}, \overline{f}]$;

Algorithm 2: Computing an SMVE for a function

For simplicity (and possibly for a balance between computation effort and sharpness of enclosure), the recursive SVME is not incorporated into Algoritnm 2.

Example 7. *Let f be as in Example 6, but let*

$$\mathcal{S} = \langle (1,3), (1.2, 3.4), (0.6, 3.8) \rangle .$$

We obtain:

NE	*HSE*	*SMVE*	*ER*
$[8.64, 13.72]$	$[8.27, 13.83]$	$[9.22, 12.49]$	$[9.50, 12.28]$

In these computations, the NE, and hence the HSE, SMVE, and HSMVE all depend on the form in which the expression for f is written, due to the subdistributivity of interval arithmetic.

2.3 Conversion Between Simplex Representations

A simplex $\mathcal{S} \subset \mathbb{R}^n$ can be represented either in terms of its $n+1$ vertices or as the feasible set of $n+1$ inequalities of the form $Ax \geq b$ for some $A \in \mathbb{R}^{n+1 \times n}$ and $b \in \mathbb{R}^{n+1}$. During the subdivision process in a branch and bound algorithm, it is most convenient to work with the vertex representation $\mathcal{S}$. However, in constraint propagation as a filter to contract or eliminate individual subregions encountered during the B&B process, it is useful to include the condition that a point x belong to the subregion $\mathcal{S}$ being considered to do so, the constraint propagation itself most easily uses the inequality, or *halfspace* representation $Ax \geq b$. Thus, conversion between the two representations is useful. Furthermore, in mathematically rigorous algorithms, it is important that the floating point data computed for either representation correspond to a simplex that contains the actual simplex. Here, we present formulas and algorithms for such mathematically rigorous conversions.

We denote the vertex representation, or $\mathcal{V}$-representation of the simplex $\mathcal{S}$ by $\mathcal{S}_{\mathcal{V}}$ (the convex hull of its vertices), and the halfspace represention, or $\mathcal{H}$-representation of the simplex, by $\mathcal{S}_{\mathcal{H}}$. Each row $a_i x \geq b_i$ of $Ax \geq b$ in the halfspace representation represents a half space corresponding to the side of a hyperplane containing a face of $\mathcal{S}$ in which $\mathcal{S}$ lies, i.e. a supporting hyperplane for $\mathcal{S}$.

More generally, this dual characterization of polytopes (and polyhedra) is explored in depth in standard texts [29]. Here, we freely refer to a simplex using either characterization. We form $n+1$ halfspaces with interval coefficients that enclose $\mathcal{S}$. We first briefly review a conversion from $\mathcal{S}_{\mathcal{V}}$ to $\mathcal{S}_{\mathcal{H}}$ in real arithmetic.

In real arithmetic, the corresponding halfspace representation of a simplex is determined as follows: Given a simplex $\mathcal{S} = \langle P_0, P_1, \ldots, P_n \rangle$, denote its i-th face by $\mathcal{S}_{\neg i} = \langle P_0, P_1, \ldots, P_{i-1}, P_{i+1}, \ldots, P_n \rangle$, where P_i is not a vertex of $\mathcal{S}_{\neg i}$. Choose an arbitrary vertex of $\mathcal{S}_{\neg i}$ and denote it by $\tilde{P}_0$, and denote the remaining vertices of $\mathcal{S}_{\neg i}$ with $\tilde{P}_j$, so

$$\mathcal{S}_{\neg i} = \langle \tilde{P}_0, \tilde{P}_1, \ldots, \tilde{P}_{n-1} \rangle.$$

Letting Π denote the hyperplane containing the face $\mathcal{S}_{\neg i}$, computing a nontrivial solution $a_i^\top = (a_{i1}, ..., a_{in})^\top$ of the system

$$\begin{pmatrix} (\tilde{P}_1 - \tilde{P}_0)^\top \\ \vdots \\ (\tilde{P}_{n-1} - \tilde{P}_0)^\top \end{pmatrix} a_i = 0$$

gives a normal vector that defines $\Pi - \tilde{P}_0$. Te offset b_i from translating Π by $\tilde{P}_0$ is $b_i = a_i^\top \tilde{P}_0$ so $\Pi = \{x \in \mathbb{R}^n : a_i^\top x = b_i\}$. The sign of the corresponding inequality constraint is then determined from the side of the hyperplane Π upon which the point P_i lies; we replace the equals sign in $a_i^\top x = b_i$ with "$\geq$" or "$\leq$" based on whether $a_i^\top (P_i - \tilde{P}_0) > 0$ or $a_i^\top (P_i - \tilde{P}_0) < 0$, respectively. (In real arithmetic, one of the strict inequalities is guaranteed provided $a_i \neq 0$ and the vertices of $\mathcal{S}$ are affinely independent.)

Negating any inequalities of the form $a_i^\top x \le b_i$, for example, one obtains the halfspace representation of $\mathcal{S}$ as $\{x \in \mathbb{R}^n : Ax \ge b\}$, where the i-th row of A is $a_i^\top$ (or $-a_i^\top$) and each entry of b is b_i (or $-b_i$). This gives the real arithmetic conversion from $\mathcal{S}_\mathcal{V}$ to $\mathcal{S}_\mathcal{H}$.

In finite-precision arithmetic, we instead seek a rigorous enclosure of $\mathcal{S} = \{x : Ax \ge b\}$, with our starting point as a collection of boxes $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, …, $\boldsymbol{P}_n$ such that for each $j = 0, 1, \ldots, n$, the exact vertex P_j of $\mathcal{S}$ is contained in the box $\boldsymbol{P}_j$. That is, $\mathcal{S}_\mathcal{V} = \langle P_0, P_1, \ldots, P_n\rangle$ is contained in the convex hull of the $n+1$ boxes $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, …, $\boldsymbol{P}_n$.

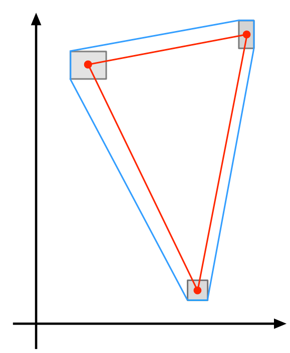


Figure 1: A simplex $\mathcal{S}$ and the convex hull of the boxes $\boldsymbol{P}_j$ containing its vertices P_j.

The enclosure for the simplex comes from an intersection of "interval-halfspaces" $\boldsymbol{H}_i,\ i = 0, 1, \ldots, n$; each interval-halfspace contains the simplex $\mathcal{S}$, as is the case with its real arithmetic analog. An interval-halfspace for our purposes is defined as a linear inequality using interval coefficients in place of real coefficients for the normal vector defining the halfspace. We analogously represent an interval-halfspace using an interval dot product:

$$\begin{aligned}\boldsymbol{H}_i &:= \{x : \boldsymbol{a}_i^\top x \ge \underline{b}_i\}\\ &= \bigcap_{a_i \in \boldsymbol{a}_i} \{x : a_i^\top x \ge \underline{b}_i\}.\end{aligned}$$

Here, $\underline{b}_i$ is a scalar; the underline notation is to suggest that in our relaxation of $\mathcal{S}$, we seek to have $\underline{b}_i \le b_i$ where b_i is the scalar from the corresponding real halfspace $\{x : a_i^\top x \ge b_i\}$. An interval dot product between two intervals $\boldsymbol{x}$ and $\boldsymbol{y}$ is taken as an interval-valued extension $\boldsymbol{x}^\top \boldsymbol{y}$ which *contains* all of the pointwise evaluations $\{x^\top y : x \in \boldsymbol{x},\ y \in \boldsymbol{y}\}$ of the real-valued dot product. The inequality $\boldsymbol{x} \ge \boldsymbol{y}$ with the usual symbol $\ge$ overloaded for intervals is taken to mean that $\inf \boldsymbol{x} \ge \sup \boldsymbol{y}$. These notations subsume the special case of one of the intervals taken as a real scalar (the real scalar is identified with an interval whose endpoints are equal).

In real arithmetic, $\boldsymbol{H}_i$ itself is an intersection of a finite number of (real) halfspaces; we are in effect constructing a polyhedral enclosure of the simplex.

These $n+1$ interval-halfspaces $\boldsymbol{H}_i$ can then be immediately applied to obtain a floating-point-defined simplex $\mathcal{S}_{\text{fl}}$ which rigorously contains the true simplex $\mathcal{S}$, successfully accounting for the original roundoff error in specifying the coordinates of its vertices. The inclusion $\mathcal{S} \subset \mathcal{S}_{\text{fl}}$ is immediate, for if $\mathcal{S} \subset \boldsymbol{H}_i = \bigcap_{a_i \in \boldsymbol{a}_i} \{x : a_i^\top x \geq \underline{b}_i\}$ $(i = 0, 1, \ldots, n)$, we can simply take $n+1$ floating number vectors $a_{\text{fl}(i)} \in \boldsymbol{a}_i$ so that, in particular, $\mathcal{S} \subset \bigcap_{i=0}^n \{x : a_{\text{fl}(i)}^\top x \geq \underline{b}_i\} := \mathcal{S}_{\text{fl}}$. i.e., $\mathcal{S}$ is contained in a verified floating-point defined simplex.

Computing an enclosure for $\boldsymbol{a}_i$ parallels the procedure followed in real arithmetic; this is done by obtaining an enclosure of a nonzero solution a_i of

$$\begin{pmatrix} (\tilde{\boldsymbol{P}}_1 - \tilde{\boldsymbol{P}}_0)^\top \\ \vdots \\ (\tilde{\boldsymbol{P}}_{n-1} - \tilde{\boldsymbol{P}}_0)^\top \end{pmatrix} a_i = 0.$$

The boxes $\boldsymbol{P}_i \ni P_i$ for the vertices P_i of $\mathcal{S}$ are used to enclose a real numbered vertex, accounting for round-off error; therefore the boxes of the vertices should have nearly zero width, thereby making the interval coefficient matrix above regular in practice. We do not verify the regularity of the matrix here. However, we check if the interval row vectors in the coefficient matrix above are nonzero before proceeding.

Using a floating-point algorithm such as QR factorization or singular value decomposition, one can obtain an approximate nonzero solution a_i to the system above (e.g., taking a_i with unit length). Enclosing a_i in a box $\boldsymbol{a}^{(0)}$ away from the origin, apply an interval Newton iteration to the system $\boldsymbol{M}z = 0,\ z^\top z - 1 = 0$, with initial box $\boldsymbol{a}^{(0)} \ni a_i$, and where $\boldsymbol{M}$ is the coefficient matrix above. This generates an enclosure $\boldsymbol{a}_i$ for a nonzero normal vector of the halfspace corresponding to the face $\mathcal{S}_{\neg i}$. Enclosures were obtainable for simplices in dimensions as high as $n = 500$; numerical experiments in low dimensions are shown in the tables below. We remark that it also possible in low dimensions ($n < 10$) to apply the modified Gram-Schmidt procedure; orthogonalization of

$$\begin{pmatrix} (\tilde{\boldsymbol{P}}_1 - \tilde{\boldsymbol{P}}_0)^\top \\ \vdots \\ (\tilde{\boldsymbol{P}}_{n-1} - \tilde{\boldsymbol{P}}_0)^\top \\ \hline (\boldsymbol{P}_i - \tilde{\boldsymbol{P}}_0)^\top \end{pmatrix}$$

can sometimes be done by using the orthogonalized vector from the last row $(\boldsymbol{P}_i - \tilde{\boldsymbol{P}}_0)^\top$ as a candidate for $\boldsymbol{a}_i$, although a separate verification of $0 \notin \boldsymbol{a}_i$ is recommended to immediately rule out a possible degenerate case that may cause the corresponding interval-halfspace $\boldsymbol{H}_i = \{x : \boldsymbol{a}_i^\top x \geq \underline{b}_i\}$ to have measure zero in $\mathbb{R}^n$, thereby preventing a nondegenerate n-simplex from being contained in $\boldsymbol{H}_i$.

After obtaining an enclosure $\boldsymbol{a}_i$ for the normal vector a_i that excludes the zero vector, we verify the orientation of the enclosure $\boldsymbol{a_i}$ by interval computations; we check whether $\boldsymbol{a}_i^\top(\boldsymbol{P}_i - \tilde{\boldsymbol{P}}_0) > 0$ or $\boldsymbol{a_i}^\top(\boldsymbol{P}_i - \tilde{\boldsymbol{P}}_0) < 0$. We replace $\boldsymbol{a_i}$ with $-\boldsymbol{a_i}$ in the latter case.

In real arithmetic, the exact value of b_i is $b_i = a_i^\top \tilde{P}_0$. It suffices to decrease $\underline{b}_i$ until it is verified that $\boldsymbol{a}_i^\top \boldsymbol{P}_j \geq \underline{b}_i$ for $j = 0, 1, \ldots, n$. This process is guaranteed once the sign of $\boldsymbol{a_i}^\top(\boldsymbol{P}_i - \tilde{\boldsymbol{P}}_0)$ has been determined in the previous step. In practice, decreasing the infimum of $\boldsymbol{a}_i^\top \tilde{\boldsymbol{P}}_0$ several times by an amount proportional to the maximum magnitude of the coordinates of the vertices $\mathcal{S}$ will suffice if the enclosures for the vertices have near zero-width.

At this stage, the interval computations $\boldsymbol{a}_i^\top \boldsymbol{P}_j \geq \underline{b}_i$ for $j = 0, 1, \ldots, n$ prove that $\mathcal{S} \subset \boldsymbol{H}_i$. We elaborate on this point in the following proposition.

Proposition 2.1. *Let $\boldsymbol{H}_i = \{x : \boldsymbol{a}_i^\top x \geq \underline{b}_i\}$ be an interval-halfspace. Verification of $\boldsymbol{a}_i^\top \boldsymbol{P}_j \geq \underline{b}_i$ $(j = 0, 1, \ldots, n)$ implies $\mathcal{S} \subset \operatorname{conv}\{\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_n\} \subset \boldsymbol{H}_i$.*

Proof. This follows from the interpretation of the interval dot product $\boldsymbol{a}_i^\top \boldsymbol{P}_j$ as an interval which contains the set of pointwise evaluations $\{a_i^\top y : a_i \in \boldsymbol{a}_i,\ y \in \boldsymbol{P}_j\}$ along with the interval inequality $\boldsymbol{a}_i^\top \boldsymbol{P}_j \geq \underline{b}_i$ implying that for all $c \in \boldsymbol{a}_i^\top \boldsymbol{P}_j$, $c \geq \underline{b}_i$. Therefore, in particular, we have that for all $a_i \in \boldsymbol{a}_i$, and for the (true) vertex $P_j \in \boldsymbol{P}_j$, $a_i^\top P_j \geq \underline{b}_i$. i.e., $P_j \in \{x : a_i^\top x \geq \underline{b}_i\}$ for each $a_i \in \boldsymbol{a}_i$, or $P_j \in \bigcap_{a_i \in \boldsymbol{a}_i} \{x : a_i^\top x \geq \underline{b}_i\} = \boldsymbol{H}_i$. Passing to the convexity of $\boldsymbol{H}_i$, we have that $\mathcal{S} = \langle P_0, P_1, \ldots, P_n \rangle \subset \boldsymbol{H}_i$. A repeat of this argument gives us $\mathcal{S} = \langle P_0, P_1, \ldots, P_n \rangle \subset \operatorname{conv}\{\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_n\} \subset \boldsymbol{H}_i$. □

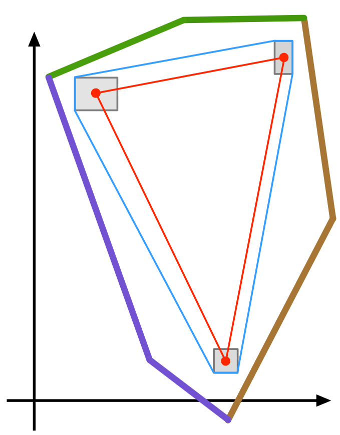
As remarked earlier, the interval-halfspace $\boldsymbol{H}_i$ can then be immediately applied to rigorously contain $\mathcal{S}$ (as well as $\operatorname{conv}\{\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_n\}$) using a floating-pointed defined halfspace.

After the verifications have completed, the simplex $\mathcal{S}$ is enclosed in an intersection of a finite number of interval-halfspaces $\boldsymbol{H}_i$ (which is, in turn, an intersection of a finite number of real halfspaces), represented by the interval system of linear inequalities; the system is compactly written as

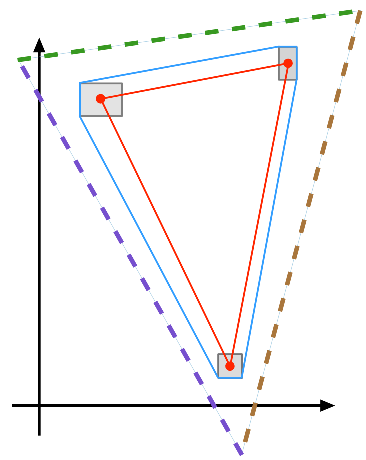
$$\left\{ x : \begin{pmatrix} \boldsymbol{a}_1^\top \\ \vdots \\ \boldsymbol{a}_{n+1}^\top \end{pmatrix} x \geq \begin{pmatrix} \underline{b}_1 \\ \vdots \\ \underline{b}_{n+1} \end{pmatrix} \right\} = \{x : \boldsymbol{A}x \geq \underline{b}\}.$$

A diagram of a simplex $\mathcal{S}$ and its interval-based enclosure is in Figure 2. For illustrative purposes, the enclosure is exaggerated using wide boxes for both the vertices and for each normal vector box $\boldsymbol{a}_i$.

We provide a couple of numerical examples that compare the results of a floating-point algorithm that generates the halfspace representation of a simplex along with an interval-based enclosure. The enclosures were generated in Matlab using the Intlab toolbox for interval computations. The implemented script follows the outline above; safechecks using return flags were incorporated throughout to indicate a failure to verify any computations. In each example, we have that $\mathcal{S} = \{Ax \geq b\} \subset \{\boldsymbol{A}x \geq \underline{b}\}$; in particular, $A \in \boldsymbol{A}$ and $b \geq \underline{b}$. The



(a) An n-simplex $\mathcal{S}$ enclosed in the polyhedron $\{\boldsymbol{A}x \geq \underline{b}\} = \bigcap_{i=0}^{n} \boldsymbol{H}_i$. Each interval-halfspace $\boldsymbol{H}_i = \{\boldsymbol{a}_i^\top x \geq \underline{b}_i\}$ corresponds to the thick borders of one color.



(b) A verified floating-point enclosure $\mathcal{S}_{\text{fl}}$ of $\mathcal{S}$.

Figure 2: Uncertain vertices and rigorous halfspace enclosure

floating-point and interval examples display approximately unit normal vectors for each row $a_i^\top$ and $\boldsymbol{a}_i^\top$ of A and $\boldsymbol{A}$, respectively.

The subscript-superscript notation used in the table refers to an interval whose tail-end digits differ. The interval $[23.456891877, 23.456891956]$, for example, is denoted by 23.456891_{877}^{956}. As a special case of this notation, we append the subscripts with an addition or subtraction symbol. The notation is analogous to the mid-radius format; this is used for some intervals whose bounds have trailing 9's. For example, $[5.99999, 6.00001]$ is denoted by $[6.00000 - 0.00001, 6.00000 + 0.00001] = 6.00000_{-1}^{+1}$, while $[-6.00001, -5.99999]$ $= -6.00000_{-1}^{+1}$. Digits that differ between the two representations are underlined.

2.4 The Subdivision Process

We subdivide by bisecting an edge. Since, upon repeated such bisections, the barycentric coordinates of the resulting sub-simplexes are not related in a simple way to the barycentric coordinates of the original simplex, we use local barycentric coordinates, particular to each sub-simplex. For this, computing the interval extensions requires knowledge of the vertices P_i of each sub-simplex. This can be done with storage of n^2 floating point numbers at the current simplex, along with information on which edge was bisected (necessitating two small integers) at each node in the search tree.

$\mathcal{S}$	$\langle(-2,0),(2,-3),(0,3)\rangle$	
A	$-0.9486832980505\underline{15}$	$-0.3162277660168\underline{38}$
	$0.8320502943378\underline{43}$	$-0.5547001962252\underline{30}$
	0.599999999999999	0.800000000000000
b	$-0.94868329805\underline{0517}$	
	$-1.66410058867\underline{5688}$	
	$-1.20000000000\underline{0000}$	
$\boldsymbol{A}$	$-0.9486832980505\,^{10}_{20}$	$-0.3162277660168\,^{30}_{40}$
	$0.8320502943378\,^{50}_{40}$	$-0.5547001962252\,^{20}_{30}$
	$0.6000000000000\,^{+1}_{-1}$	$0.8000000000000\,^{+1}_{-1}$
$\underline{b}$	$-0.94868329805\underline{1520}$	
	$-1.66410058867\underline{6690}$	
	$-1.20000000000\underline{1010}$	

Table 1: A 2-simplex $\mathcal{S} = \{Ax \geq b\} \subset \{\boldsymbol{A}x \geq \underline{b}\}$.

3 Summary, Future Work, and Perspectives

Inspired by recent work dealing with sampling algorithms based on simplicial subdivisions, we have begun an investigation of the use of simplicial subdivisions in mathematically rigorous interval-based algorithms. We have proposed and evaluated tools for bounding ranges over simplices and for rigorously converting between simplex representations, tools beneficial in simplex-based branch and bound algorithms. We have a crude implementation of such an algorithm, but have not completed this work yet.

Simplex-based branch and bound algorithms have potential advantages when the feasible set of the optimization problem is constrained to lie within a simplex. However, such advantages may be outweighed by disadvantages in the process. Use of simplices has been shown to be advantageous in branch-and-bound algorithms with function ranges based on statistical sampling; see [18]. In such sampling algorithms, simplices bring one more quickly to an upper ε-approximation to the global optimum. However, in mathematically rigorous algorithms, one still needs to obtain a rigorous lower bound, which sampling alone does not provide.

Processing simplicies as individual subregions in a branch-and-bound process suffers from a volume limitation relative to rectangular subregions. Since an n-simplex contains $1/n!$ times the volume of its corresponding rectangular enclosure, we hypothesize that even if one were to obtain an exact evaluation procedure for an objective function over a simplex, the fact that a rectangular evaluation processes $n!$ times as much volume may make a simplicial interval branch-and-bound procedure not competitive with traditional rectangular in-

$\mathcal{S}$	$\langle(-15,-45,-30,0),(-15,0,0,15),(15,45,-45,30),(30,0,-75,15),(0,0,0,0)\rangle$			
A	-0.6299407883487$\underline{13}$	0.2519763153394$\underline{84}$	-0.3779644730092$\underline{28}$	-0.6299407883487$\underline{12}$
	0.3955594970624$\underline{17}$	-0.3428182307874$\underline{29}$	0.3164475976499$\underline{34}$	0.7911189941248$\underline{34}$
	-0.5874453374909$\underline{69}$	0.4307932474933$\underline{76}$	-0.3524672024945$\underline{82}$	-0.5874453374909$\underline{68}$
	0.6299407883487$\underline{11}$	-0.3779644730092$\underline{28}$	0.2519763153394$\underline{84}$	0.6299407883487$\underline{13}$
	-0.0000000000000$\underline{02}$	0.3162277660168$\underline{38}$	-0.0000000000000$\underline{02}$	-0.9486832980505$\underline{14}$
b	0.000000000000008			
	-0.0000000000000$\underline{0022}$			
	0.000000000000033			
	0.000000000000021			
	-14.23024947075$\underline{7674}$			
$\boldsymbol{A}$	-0.6299407883487$^{10}_{20}$	0.2519763153394$^{90}_{80}$	-0.3779644730092$^{20}_{30}$	-0.6299407883487$^{10}_{20}$
	0.3955594970624$^{20}_{10}$	-0.3428182307874$^{20}_{30}$	0.3164475976499$^{40}_{30}$	0.7911189941248$^{40}_{30}$
	-0.5874453374909$^{60}_{70}$	0.4307932474933$^{80}_{70}$	-0.3524672024945$^{80}_{90}$	-0.5874453374909$^{60}_{70}$
	0.6299407883487$^{20}_{10}$	-0.3779644730092$^{20}_{30}$	0.2519763153394$^{90}_{80}$	0.6299407883487$^{20}_{10}$
	0.0000000000000$^{+1}_{-1}$	0.3162277660168$^{40}_{30}$	0.0000000000000$^{+1}_{-1}$	-0.9486832980505$^{10}_{20}$
$\underline{b}$	-0.000000000000990			
	-0.00000000000$\underline{1060}$			
	-0.000000000000950			
	-0.000000000001070			
	-14.23024947075$\underline{8660}$			

Table 2: A 4-simplex $\mathcal{S} = \{Ax \geq b\} \subset \{\boldsymbol{A}x \geq \underline{b}\}$.

terval branch-and-bound algorithms.

Another possible drawback to using simplicies comes from memory usage: Each n-simplex (subregion) is defined by $n+1$ interval vectors or $2n(n+1)$ floating point numbers, whereas a rectangular subregion requires only one interval vector of length n or $2n$ floating point numbers. This could be a limiting factor for larger n.

We will be able to draw more definite conclusions after further investigation.

References

[1] Paul Alexandroff and Heinz Hopf. *Topologie.* Chelsea, 1935.

[2] Rida T. Farouki. The Bernstein polynomial basis: A centennial retrospective. *Comput. Aided Geom. Des.*, 29(6):379–419, August 2012.

[3] Rida T. Farouki, T. N. T. Goodman, and Thomas Sauer. Construction of orthogonal bases for polynomials in Bernstein form on triangular and simplex domains. *Comput. Aided Geom. Des.*, 20(4):209–230, July 2003.

[4] J. Garloff. Convergent bounds for the range of multivariate polynomials. In *Proceedings of the International Symposium on interval mathematics on Interval mathematics 1985*, pages 37–56, London, UK, 1985. Springer-Verlag.

[5] Eldon R. Hansen. Interval forms of Newton's method. *Computing*, 20:153–163, 1978.

[6] Ralf Hungerbühler and Jürgen Garloff. Bounds for the range of a bivariate polynomial over a triangle. *Reliable Computing*, 4(1):3–13, 1998.

[7] Ralph Baker Kearfott. *Computing the degree of maps and a generalized method of bisection*. Ph.D. thesis, Department of Mathematics, University of Utah, Salt Lake City, UT, USA 84112, 1977.

[8] Ralph Baker Kearfott. A proof of convergence and an error bound for the method of bisection in $\mathbb{R}^n$. *Mathematics of Computation*, 32(144):1147–1153, October 1978.

[9] Ralph Baker Kearfott. An efficient degree-computation method for a generalized method of bisection. *Numerische Mathematik*, 32(2):109–127, June 1979.

[10] Ralph Baker Kearfott. A summary of recent experiments to compute the topological degree. In V. Lakshmikantham, editor, *Applied Nonlinear Analysis: proceedings of an International Conference on Applied Nonlinear Analysis, held at the University of Texas at Arlington, Arlington, Texas, April 20–22, 1978*, pages 627–633. Academic Press, New York, 1979.

[11] Ralph Baker Kearfott. Abstract generalized bisection and a cost bound. *Mathematics of Computation*, 49(179):187–202, July 1987.

[12] Ralph Baker Kearfott. GlobSol user guide. *Optimization Methods and Software*, 24(4-5):687–708, 2009.

[13] Ralph Baker Kearfott, Mitsuhiro T. Nakao, Arnold Neumaier, Siegfried M. Rump, Sergey P. Shary, and Pascal Van Hentenryck. Standardized notation in interval analysis. In *In Proc. XIII Baikal International School-seminar Optimization methods and their applications. Vol. 4 Interval analysis. Irkutsk: Institute of Energy Systems SB RAS*, pages 106–113, 2002. `http://interval.louisiana.edu/preprints/Shary_n.pdf`.

[14] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. SIAM, Philadelphia, 2009.

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

[16] Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. In A. Iserles, editor, *Acta Numerica 2004*, pages 271–369. Cambridge University Press, 2004.

[17] Remigijus Paulavčius and Julius Žilinskas. *Simplicial Global Optimization*. Springer Verlag, 2014.

[18] Remigijus Paulavičius and Julius Žilinskas. Simplicial global optimization. *Journal of Global Optimization*, 60(4):801–802, 2014.

[19] Remigijus Paulavičius, Yaroslav Sergeyev, Dmitri Kvasov, and Julius Žilinskas. Globally-biased DISIMPL algorithm for expensive global optimization. *Journal of Global Optimization*, 59(2-3):545–567, 2014.

[20] Remigijus Paulavičius and Julius Žilinskas. Advantages of simplicial partitioning for Lipschitz optimization problems with linear constraints. *Optimization Letters*, pages 1–10, 2014.

[21] S. Ray and P. S. V. Nataraj. A new strategy for selecting subdivision point in the Bernstein approach to polynomial optimization. *Reliable Computing*, 14(1):117–137, 2010.

[22] S. Ray and P. S. V. Nataraj. A matrix method for efficient computation of Bernstein coefficients. *Reliable Computing*, 17(1):40–71, 2012.

[23] Frank Stenger. Computing the topological degree of a mapping in R^n. *Numerische Mathematik*, 25(1):23–38, March 1975.

[24] Frank Stenger. An algorithm for the topological degree of a mapping in $\mathbb{R}^n$. *Numerische Mathematik*, 25:23–38, 1976.

[25] Martin Stynes. A simplification of Stenger's topological degree formula. *Numerische Mathematik*, 33(2):147–155, June 1979.

[26] Martin Stynes. On the construction of sufficient refinements for computation of topological degree. *Numerische Mathematik*, 37(3):453–462, July 1981.

[27] Jihad Titi, Tareq Hamadneh, and Jürgen Garloff. Convergence of the simplicial rational Bernstein form. In An Hoai Le Thi, Tao Pham Dinh, and Thanh Ngoc Nguyen, editors, *Modelling, Computation and Optimization in Information Systems and Management Sciences: Proceedings of the 3rd International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences - MCO 2015 - Part I*, pages 433–441. Springer International Publishing, Cham, 2015.

[28] Julius Žilinskas and Mirjam Dur. Depth-first simplicial partition for copositivity detection, with an application to maxclique. *Optimization Methods and Software*, 26(3):499–510, 2011.

[29] Günter M. Ziegler. *Lectures on Polytopes*. Graduate texts in mathematics. Springer, New York, 1995.

[30] Julius Žilinskas. Branch and bound with simplicial partitions for global optimization. *Mathematical Modelling and Analysis*, 13(1):145–159, 2008.