Stochastics: An International Journal of Probability and Stochastic Processes
iFirst article, 2011, 1–29

The GAEP algorithm for the fast computation of the distribution of a function of dependent random variables

Philipp Arbenz[a]*, Paul Embrechts[b] and Giovanni Puccetti[c]

[a]*Department of Mathematics, ETH Zurich, 8092 Zurich, Switzerland;* [b]*Swiss Finance Institute, Department of Mathematics, ETH Zurich, 8092 Zurich, Switzerland;* [c]*Department of Mathematics for Decisions, University of Firenze, 50134 Firenze, Italy*

(*Received 2 September 2010; final version received 22 February 2011*)

We introduce a new algorithm for numerically computing the distribution of an increasing function of d dependent, non-negative random variables with a given joint distribution. We prove the convergence of the algorithm and give convergence rates under regularity conditions.

Keywords: convolution; distribution functions; dependent random variables

AMS Subject Classification: 62E17; 65C20; 65C50

1. Introduction

In this paper, we introduce the *GAEP algorithm* (generalised AEP algorithm) in order to compute $\mathbb{P}[\varphi(\mathbf{X}) \leq s]$, where $\mathbf{X}$ is a random vector in $(0, \infty)^d$ and φ is a continuous function, strictly increasing in each coordinate. The algorithm is based on the decomposition of the set $\{\mathbf{x} \in (0, \infty)^d : \varphi(\mathbf{x}) \leq s\}$ into a countable family of disjoint hypercubes. The corresponding probability $\mathbb{P}[\varphi(\mathbf{X}) \leq s]$ is then approximated by the measure over these hypercubes.

The GAEP algorithm is similar in spirit to the AEP algorithm introduced by the same authors in Ref. [1] for the case $\varphi(\mathbf{x}) = \sum_{k=1}^{d} x_k$. As the two algorithms are based on different geometrical decompositions, GAEP is not a proper extension of AEP.

This paper is organised as follows. After some preliminaries in Section 2, we illustrate GAEP in dimension two ($d = 2$) in Section 3. Section 4 extends GAEP to arbitrary dimensions, its convergence being discussed in Sections 5–7. In Section 8, we test GAEP on some examples, and, in Section 9, we compare and contrast it to its main competitors. Section 10 illustrates the differences between GAEP and AEP, while, in Section 11, we provide a method to improve convergence rates in dimension three. Section 12 concludes the paper.

2. Notation and preliminaries

Fix $d \in \mathbb{N}$, $d \geq 2$, and define $N = 2^d$. We set $\mathbb{R}_+ = [0, +\infty)$ and $\mathbb{R}_- = (-\infty, 0]$. Throughout the paper, (row) vectors are denoted in boldface, for example, $\mathbf{e}_k \in \mathbb{R}^d$ is the kth standard unit vector, for $k \in D = \{1, \ldots, d\}$. We write $\mathbf{i}_1, \ldots, \mathbf{i}_N$ for all the 2^d vectors in $\{0, 1\}^d$, i.e. $\mathbf{i}_1 = \mathbf{0} = (0, \ldots, 0)$, $\mathbf{i}_{k+1} = \mathbf{e}_k, k \in D$, and $\mathbf{i}_N = \mathbf{1} = (1, \ldots, 1)$.

*Corresponding author. Email: philipp.arbenz@math.ethz.ch

ISSN 1744-2508 print/ISSN 1744-2516 online
© 2011 Taylor & Francis
http://dx.doi.org/10.1080/17442508.2011.566337
http://www.tandfonline.com

By $\#\mathbf{i} = \sum_{k=1}^{d} i_k$, we denote the number of 1's in the vector $\mathbf{i}$, for example, $\#\mathbf{i}_1 = 0, \#\mathbf{i}_N = d$. We define the component-wise product between two vectors $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{y} = (y_1, \ldots, y_d) \in \mathbb{R}^d$ as

$$\mathbf{x} \circ \mathbf{y} = (x_1 y_1, \ldots, x_d y_d) \in \mathbb{R}^d.$$

For instance, $\mathbf{x} \circ \mathbf{e}_k = x_k \mathbf{e}_k = (0, \ldots, 0, x_k, 0, \ldots, 0)$. Let $\geq$ denote the component-wise order between vectors, i.e. $\mathbf{x} \geq \mathbf{y}$ if and only if $x_k \geq y_k$ for all $k \in D$. The orders $\leq$, $<$, and $>$ are defined analogously.

On some probability space $(\Omega, \mathfrak{A}, \mathbb{P})$, assume that the random vector $\mathbf{X} = (X_1, \ldots, X_d)$ has joint distribution H. Throughout the paper, we assume the marginal components X_k to be non-negative, i.e. $\mathbb{P}[X_k \leq 0] = 0$ for all $k \in D$. The extension to random vectors which are component wise bounded from below is straightforward and will be illustrated in the following. The joint distribution H induces the probability measure V_H on $\mathbb{R}^d$ via

$$V_H\big[\{\mathbf{y} \in \mathbb{R}^d : \mathbf{y} \leq \mathbf{x}\}\big] = H(\mathbf{x}), \quad \text{for all } \mathbf{x} \in \mathbb{R}^d.$$

For $\mathbf{b} \in \mathbb{R}^d$ and $\mathbf{h} \in \mathbb{R}_-^d \cup \mathbb{R}_+^d$, we define the hypercube $\mathcal{Q}(\mathbf{b}, \mathbf{h}) \subset \mathbb{R}^d$ as

$$\mathcal{Q}(\mathbf{b}, \mathbf{h}) = \begin{cases} \{\mathbf{x} \in \mathbb{R}^d : \mathbf{b} < \mathbf{x} \leq \mathbf{b} + \mathbf{h}\}, & \text{if } \mathbf{h} \in \mathbb{R}_+^d, \\ \{\mathbf{x} \in \mathbb{R}^d : \mathbf{b} + \mathbf{h} < \mathbf{x} \leq \mathbf{b}\}, & \text{if } \mathbf{h} \in \mathbb{R}_-^d. \end{cases} \tag{1}$$

For $\mathbf{h} \in \mathbb{R}_+^d$, the V_H-measure of $\mathcal{Q}(\mathbf{b}, \mathbf{h})$ can be calculated easily as

$$V_H\big[\mathcal{Q}(\mathbf{b}, \mathbf{h})\big] = \mathbb{P}\big[\mathbf{X} \in \mathcal{Q}(\mathbf{b}, \mathbf{h})\big] = \sum_{j=1}^{N} (-1)^{d+\#\mathbf{i}_j} H(\mathbf{b} + \mathbf{h} \circ \mathbf{i}_j). \tag{2}$$

The case $\mathbf{h} \in \mathbb{R}_-^d$ is analogous. As a special case of (2) for $d = 2$, the probability measure of a rectangle $\mathcal{Q}(\mathbf{b}, \mathbf{h}) = (b_1, b_1 + h_1] \times (b_2, b_2 + h_2]$ can be written as

$$V_H\big[\mathcal{Q}(\mathbf{b}, \mathbf{h})\big] = H(b_1, b_2) - H(b_1 + h_1, b_2) - H(b_1, b_2 + h_2) + H(b_1 + h_1, b_2 + h_2).$$

Let $\mathcal{N}$ be the set of continuous functions $\varphi : \mathbb{R}^d \to \mathbb{R}$, which are strictly increasing in each coordinate, and such that

$$\lim_{t \to +\infty} \varphi(\mathbf{b} + t\mathbf{e}_k) = +\infty, \text{ and } \lim_{t \to -\infty} \varphi(\mathbf{b} + t\mathbf{e}_k) = -\infty, \text{ for all } \mathbf{b} \in \mathbb{R}^d \text{ and } k \in D.$$

Throughout the paper, we assume that $\varphi \in \mathcal{N}$ and fix $s \in \mathbb{R}$ such that $\varphi(\mathbf{0}) < s$. Note that if, on the contrary, $\varphi(\mathbf{0}) \geq s$, then for non-negative vectors $\mathbf{X}$, we trivially have that $\mathbb{P}[\varphi(\mathbf{X}) \leq s] = 0$.

For $\mathbf{b} \in \mathbb{R}^d$ and $\mathbf{p} \in \mathbb{R}_-^d \cup \mathbb{R}_+^d$, we also define the *quasi-simplex* $\mathcal{S}(\mathbf{b}, \mathbf{p})$ as

$$\mathcal{S}(\mathbf{b}, \mathbf{p}) = \begin{cases} \{\mathbf{x} \in \mathbb{R}^d : \mathbf{b} < \mathbf{x} \leq \mathbf{b} + \mathbf{p}, \varphi(\mathbf{x}) \leq s\}, & \text{if } \mathbf{p} \in \mathbb{R}_+^d, \\ \{\mathbf{x} \in \mathbb{R}^d : \mathbf{b} + \mathbf{p} < \mathbf{x} \leq \mathbf{b}, \varphi(\mathbf{x}) > s\}, & \text{if } \mathbf{p} \in \mathbb{R}_-^d. \end{cases} \tag{3}$$

Note that, if one or more of the components of $\mathbf{h}$ and $\mathbf{p}$ is equal to zero, then $\mathcal{Q}(\mathbf{b}, \mathbf{h})$ and $\mathcal{S}(\mathbf{b}, \mathbf{p})$ are empty.

As $\varphi \in \mathcal{N}$, there exists a unique vector $\mathbf{p}_1^1 \in \mathbb{R}_+^d$ such that

$$\varphi(\mathbf{p}_1^1 \circ \mathbf{e}_k) = s, \quad \text{for all } k \in D. \tag{4}$$

Figure 1 illustrates (3) and (1) as well as (4) for $d = 2$. Defining $\mathcal{S}_1^1 = \mathcal{S}(0, \mathbf{p}_1^1)$ and recalling that $\mathbf{X}$ is non-negative, (4) implies that

$$\mathbb{P}\left[\varphi(\mathbf{X}) \leq s\right] = V_H\left[\mathcal{S}_1^1\right].$$

3. Description of the GAEP algorithm in dimension *d* = 2

We first illustrate the GAEP algorithm in the case $d = 2$. In Section 4, we will generalise it to arbitrary dimensions d. In order to calculate $\mathbb{P}[\varphi(\mathbf{X}) \leq s] = V_H[\mathcal{S}_1^1]$, we decompose $\mathcal{S}_1^1$ into a family of disjoint hypercubes, whose probability measures can be easily calculated via (2).

Let $\mathbf{h} \in \mathbb{R}_+^2$ such that $\mathbf{h} \leq \mathbf{p}_1^1$. For ease of notation, we write $\mathbf{p} = \mathbf{p}_1^1$ throughout this section. As illustrated in Figure 2, the rectangle $\mathcal{Q}(\mathbf{0}, \mathbf{p}) \subset \mathbb{R}^2$ can be split into four disjoint rectangles along the components of $\mathbf{h}$ as

$$\begin{aligned} \mathcal{Q}(\mathbf{0}, \mathbf{p}) = {} & \mathcal{Q}(\mathbf{0}, \mathbf{h}) \cup \mathcal{Q}((h_1, 0), (p_1 - h_1, h_2)) \\ & \cup \mathcal{Q}((0, h_2), (h_1, p_2 - h_2)) \cup \mathcal{Q}(\mathbf{h}, \mathbf{p} - \mathbf{h}). \end{aligned} \tag{5}$$

A similar decomposition holds for the quasi-simplex $\mathcal{S}(\mathbf{0}, \mathbf{p})$, for which we have

$$\begin{aligned} \mathcal{S}(\mathbf{0}, \mathbf{p}) = {} & \mathcal{S}(\mathbf{0}, \mathbf{h}) \cup \mathcal{S}((h_1, 0), (p_1 - h_1, h_2)) \\ & \cup \mathcal{S}((0, h_2), (h_1, p_2 - h_2)) \cup \mathcal{S}(\mathbf{h}, \mathbf{p} - \mathbf{h}). \end{aligned} \tag{6}$$

Note that $\mathcal{S}(\mathbf{0}, \mathbf{h}) = \mathcal{Q}(\mathbf{0}, \mathbf{h}) \backslash \mathcal{S}(\mathbf{h}, -\mathbf{h})$. By setting $\mathcal{S}_1^1 = \mathcal{S}(\mathbf{0}, \mathbf{p})$, $\mathcal{Q}_1^1 = \mathcal{Q}(\mathbf{0}, \mathbf{h})$, $\mathcal{S}_2^1 = \mathcal{S}(\mathbf{h}, -\mathbf{h})$, $\mathcal{S}_2^2 = \mathcal{S}((h_1, 0), (p_1 - h_1, h_2))$, $\mathcal{S}_2^3 = \mathcal{S}((0, h_2), (h_1, p_2 - h_2))$, and $\mathcal{S}_2^4 = \mathcal{S}(\mathbf{h}, \mathbf{p} - \mathbf{h})$, we can reformulate (6) as

$$\mathcal{S}_1^1 = (\mathcal{Q}_1^1 \backslash \mathcal{S}_2^1) \cup \mathcal{S}_2^2 \cup \mathcal{S}_2^3 \cup \mathcal{S}_2^4, \tag{7}$$

as illustrated in Figure 3. Note that either $\mathcal{S}_2^1 = \emptyset$ (if $\varphi(\mathbf{h}) > s$) or $\mathcal{S}_2^4 = \emptyset$ (if $\varphi(\mathbf{h}) \leq s$).

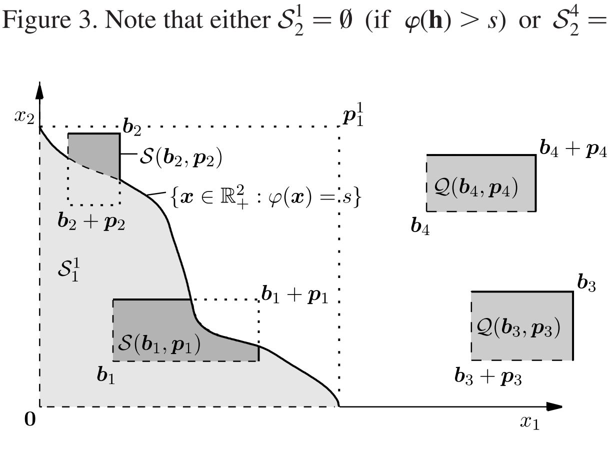
x_2
$\boldsymbol{b}_2$
$\boldsymbol{p}_1^1$
$\mathcal{S}(\boldsymbol{b}_2, \boldsymbol{p}_2)$
$\{\boldsymbol{x} \in \mathbb{R}_+^2 : \varphi(\boldsymbol{x}) = s\}$
$\boldsymbol{b}_2 + \boldsymbol{p}_2$
$\boldsymbol{b}_4 + \boldsymbol{p}_4$
$\mathcal{Q}(\boldsymbol{b}_4, \boldsymbol{p}_4)$
$\boldsymbol{b}_4$
$\mathcal{S}_1^1$
$\boldsymbol{b}_1 + \boldsymbol{p}_1$
$\boldsymbol{b}_3$
$\mathcal{Q}(\boldsymbol{b}_3, \boldsymbol{p}_3)$
$\mathcal{S}(\boldsymbol{b}_1, \boldsymbol{p}_1)$
$\boldsymbol{b}_1$
$\boldsymbol{b}_3 + \boldsymbol{p}_3$
$\mathbf{0}$
x_1

Figure 1. Three quasi-simplexes $\mathcal{S}_1^1 = \mathcal{S}(\mathbf{0}, \mathbf{p}_1^1)$, $\mathcal{S}(\mathbf{b}_1, \mathbf{p}_1)$, $\mathcal{S}(\mathbf{b}_2, \mathbf{p}_2) \subset \mathbb{R}^2$ and two hypercubes $\mathcal{Q}(\mathbf{b}_3, \mathbf{p}_3)$, $\mathcal{Q}(\mathbf{b}_4, \mathbf{p}_4) \subset \mathbb{R}^2$ with $\mathbf{p}_1^1, \mathbf{p}_1, \mathbf{p}_4 \in \mathbb{R}_+^2$ and $\mathbf{p}_2, \mathbf{p}_3 \in \mathbb{R}_-^2$. Thick and dashed lines indicate closed and open boundaries of the sets, respectively.

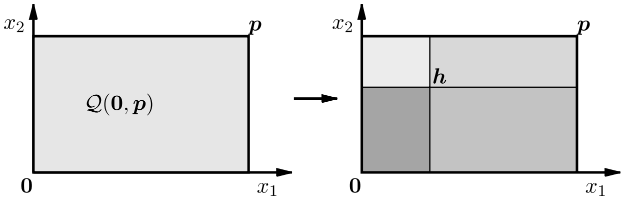

Figure 2. An illustration of (5) in which the rectangle $\mathcal{Q}(\mathbf{0},\mathbf{p}) \subset \mathbb{R}^2$ is decomposed into four disjoint rectangles.

Since $\mathcal{S}_2^t$'s are disjoint and $\mathcal{S}_2^1 \subset \mathcal{Q}_1^1$, (7) translates into the decomposition

$$V_H\big[\mathcal{S}_1^1\big] = V_H\big[\mathcal{Q}_1^1\big] - V_H\big[\mathcal{S}_2^1\big] + V_H\big[\mathcal{S}_2^2\big] + V_H\big[\mathcal{S}_2^3\big] + V_H\big[\mathcal{S}_2^4\big]. \tag{8}$$

As a first approximation of $V_H[\mathcal{S}_1^1]$, we take the value $P_1 = V_H[\mathcal{Q}_1^1]$. This results in the difference

$$V_H\big[\mathcal{S}_1^1\big] - P_1 = \sum_{t=1}^{4} \tau_2^t V_H\big[\mathcal{S}_2^t\big],$$

where $\tau_2^t \in \{-1, 1\}$ indicates whether the measure $V_H[\mathcal{S}_2^t]$ has to be added ($\tau_2^2 = \tau_2^3 = \tau_2^4 = 1$) or subtracted ($\tau_2^1 = -1$).

At this point, each of the $\mathcal{S}_2^t$ can be further decomposed via (8) into a square and four smaller quasi-simplexes, for a total of 4 squares and 16 quasi-simplexes. The measure of the four squares so obtained is to be added or subtracted to P_1 in order to define a second estimate P_2 of $V_H[\mathcal{S}_1^1]$, so that the difference between P_2 and $V_H[\mathcal{S}_1^1]$ will be given by the measure of the remaining 16 quasi-simplexes. The latter are then decomposed again in the following step of the algorithm. By iterating this procedure, we obtain a sequence P_n of estimates, which we will prove to converge to $V_H[\mathcal{S}_1^1]$ under the regularity conditions given in Section 5. In the general case, at each iteration of the algorithm, we restrict to quasi-simplexes that turn out to be non-empty.

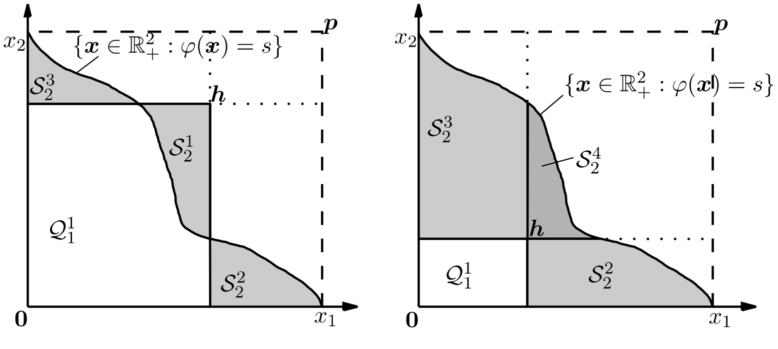

Figure 3. An illustration of the decomposition (7) for two different choices of $\mathbf{h} \in \mathbb{R}_-^d \cup \mathbb{R}_+^d$. In the left-hand side picture, we have $\varphi(\mathbf{h}) > s$ and $\mathcal{S}_2^4 = \emptyset$, while, in the right-hand side picture, we have $\varphi(\mathbf{h}) < s$ and $\mathcal{S}_2^1 = \emptyset$.

4. Description of the GAEP algorithm in arbitrary dimensions

In this section, we state the two main results of the paper. First, we will extend the measure decomposition (8) to arbitrary dimensions, by showing that the measure of an arbitrary quasi-simplex can be decomposed into the measures of a hypercube and $N = 2^d$ smaller quasi-simplexes. Then, we define a sequence P_n that will be proved to converge to $V_H[\mathcal{S}_1^1]$.

For $\mathbf{p} \in \mathbb{R}_-^d \cup \mathbb{R}_+^d$, fix $\mathbf{h} \in \mathcal{H}(\mathbf{p})$, where

$$\mathcal{H}(\mathbf{p}) = \begin{cases} \{\mathbf{x} \in \mathbb{R}^d : \mathbf{0} \leq \mathbf{x} \leq \mathbf{p}\}, & \text{if } \mathbf{p} \in \mathbb{R}_+^d, \\ \{\mathbf{x} \in \mathbb{R}^d : \mathbf{p} \leq \mathbf{x} \leq \mathbf{0}\}, & \text{if } \mathbf{p} \in \mathbb{R}_-^d. \end{cases}$$

Analogously to the case $d = 2$ (see Figure 2), a hypercube $\mathcal{Q}(\mathbf{b}, \mathbf{p}) \subset \mathbb{R}^d$ can be decomposed into N hypercubes along the components $b_k + h_k, k \in D$.

Lemma 4.1. For an arbitrary hypercube $\mathcal{Q}(\mathbf{b}, \mathbf{p}) \subset \mathbb{R}^d$, we have

$$\mathcal{Q}(\mathbf{b}, \mathbf{p}) = \bigcup_{j=1}^{N} \mathcal{Q}\big(\mathbf{b} + \mathbf{i}_j \circ \mathbf{h}, \mathbf{i}_j \circ \mathbf{p} + (\mathbf{1} - 2\mathbf{i}_j) \circ \mathbf{h}\big) \tag{9}$$

with the hypercubes on the right-hand side of (9) being disjoint.

Proof. For $j = 1, \ldots, N$, let the sets $\mathcal{C}_j$ be defined as

$$\begin{aligned} \mathcal{C}_j = \big\{\mathbf{x} \in \mathbb{R}^d : x_k \leq b_k + h_k \ \text{for all } k \text{ with } (\mathbf{i}_j)_k = 0, \\ x_k > b_k + h_k \ \text{for all } k \text{ with } (\mathbf{i}_j)_k = 1\big\}. \end{aligned}$$

Since $\cup_{j=1}^N \mathbf{i}_j = \{0, 1\}^d$, we have that the family $\{\mathcal{C}_j, j = 1, \ldots, N\}$ is a partition of $\mathbb{R}^d$, i.e. $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ and $\cup_{j=1}^N \mathcal{C}_j = \mathbb{R}^d$. Hence, we can write

$$\mathcal{Q}(\mathbf{b}, \mathbf{p}) = \bigcup_{j=1}^{N} \big(\mathcal{Q}(\mathbf{b}, \mathbf{p}) \cap \mathcal{C}_j\big).$$

If $\mathbf{p} \in \mathbb{R}_+^d$ (the case $\mathbf{p} \in \mathbb{R}_-^d$ is analogous), then note that for all $j = 1, \ldots, N$ and $k \in D$, we have

$$\big(\mathbf{b} + \mathbf{i}_j \circ \mathbf{h}\big)_k = \begin{cases} b_k, & \text{if } (\mathbf{i}_j)_k = 0, \\ b_k + h_k, & \text{if } (\mathbf{i}_j)_k = 1, \end{cases} \tag{10}$$

$$\big(\mathbf{i}_j \circ \mathbf{p} + (\mathbf{1} - 2\mathbf{i}_j) \circ \mathbf{h}\big)_k = \begin{cases} h_k, & \text{if } (\mathbf{i}_j)_k = 0, \\ p_k - h_k, & \text{if } (\mathbf{i}_j)_k = 1. \end{cases} \tag{11}$$

Thus, the result follows by observing that

$$\begin{aligned} \mathcal{Q}(\mathbf{b}, \mathbf{p}) \cap \mathcal{C}_j = \big\{\mathbf{x} \in \mathbb{R}^d : b_k < x_k \leq b_k + h_k \ \text{for all } k \text{ with } (\mathbf{i}_j)_k = 0, \\ b_k + h_k < x_k \leq p_k \ \text{for all } k \text{ with } (i_j)_k = 1\big\} \\ = \mathcal{Q}\big(\mathbf{b} + \mathbf{i}_j \circ \mathbf{h}, \mathbf{i}_j \circ \mathbf{p} + (\mathbf{1} - 2\mathbf{i}_j) \circ \mathbf{h}\big). \end{aligned}$$

□

The set decomposition (9) translates into the measure decomposition given by the following theorem.

THEOREM 4.2. Let $\mathbf{v}_1, \ldots, \mathbf{v}_N \in \{0,1\}^d$ be defined as $\mathbf{v}_1 = \mathbf{1}$ and $\mathbf{v}_k = \mathbf{i}_k$ for $k = 2, \ldots, N$. Let $\mathbf{b} \in \mathbb{R}^d$, $\mathbf{p} \in \mathbb{R}^d_- \cup \mathbb{R}^d_+$, and $\mathbf{h} \in \mathcal{H}(\mathbf{p})$. Then

$$V_H\big[\mathcal{S}(\mathbf{b},\mathbf{p})\big] = V_H\big[\mathcal{Q}(\mathbf{b},\mathbf{h})\big] + \sum_{j=1}^{N} m_j V_H\big[\mathcal{S}\big(\mathbf{b} + \mathbf{v}_j \circ \mathbf{h}, \mathbf{i}_j \circ \mathbf{p} + (\mathbf{1} - 2\mathbf{v}_j) \circ \mathbf{h}\big)\big], \qquad (12)$$

where $m_1 = -1$ and $m_j = +1$ for $j = 2, \ldots, N$.

Proof. First assume that $\varphi(\mathbf{b}) \leq s$ and $\mathbf{p} \in \mathbb{R}^d_+$. Defining the set $\Lambda = \{\mathbf{x} \in \mathbb{R}^d : \varphi(\mathbf{x}) \leq s\}$ and using (9), we can write

$$\begin{aligned} \Lambda \cap \mathcal{Q}(\mathbf{b},\mathbf{p}) = \mathcal{S}(\mathbf{b},\mathbf{p}) &= \bigcup_{j=1}^{N} \big\{\Lambda \cap \mathcal{Q}\big(\mathbf{b} + \mathbf{i}_j \circ \mathbf{h}, \mathbf{i}_j \circ \mathbf{p} + (\mathbf{1} - 2\mathbf{i}_j) \circ \mathbf{h}\big)\big\} \\ &= \bigcup_{j=1}^{N} \mathcal{S}\big(\mathbf{b} + \mathbf{i}_j \circ \mathbf{h}, \mathbf{i}_j \circ \mathbf{p} + (\mathbf{1} - 2\mathbf{i}_j) \circ \mathbf{h}\big). \qquad (13) \end{aligned}$$

As the hypercubes on the right-hand side of (9) are disjoint, the quasi-simplexes on the right-hand side of (13) are disjoint too. Thus, from (13), we get

$$V_H\big[\mathcal{S}(\mathbf{b},\mathbf{p})\big] = \sum_{j=1}^{N} V_H\big[\mathcal{S}\big(\mathbf{b} + \mathbf{i}_j \circ \mathbf{h}, \mathbf{i}_j \circ \mathbf{p} + (\mathbf{1} - 2\mathbf{i}_j) \circ \mathbf{h}\big)\big]. \qquad (14)$$

Since $\mathbf{i}_1 = \mathbf{0}$, we have

$$V_H\big[\mathcal{S}(\mathbf{b} + \mathbf{i}_1 \circ \mathbf{h}, \mathbf{i}_1 \circ \mathbf{p} + (\mathbf{1} - 2\mathbf{i}_1) \circ \mathbf{h})\big] = V_H\big[\mathcal{S}(\mathbf{b},\mathbf{h})\big],$$

and we can write (14) as

$$V_H\big[\mathcal{S}(\mathbf{b},\mathbf{p})\big] = V_H\big[\mathcal{S}(\mathbf{b},\mathbf{h})\big] + \sum_{j=2}^{N} V_H\big[\mathcal{S}\big(\mathbf{b} + \mathbf{i}_j \circ \mathbf{h}, \mathbf{i}_j \circ \mathbf{p} + (\mathbf{1} - 2\mathbf{i}_j) \circ \mathbf{h}\big)\big]. \qquad (15)$$

Noting that $\mathcal{S}(\mathbf{b},\mathbf{h}) = \Lambda \cap \mathcal{Q}(\mathbf{b},\mathbf{h})$, and using the fact that $\mathcal{Q}(\mathbf{b},\mathbf{h}) = \mathcal{Q}(\mathbf{b} + \mathbf{h}, -\mathbf{h})$, we obtain

$$\begin{aligned} V_H\big[\mathcal{S}(\mathbf{b},\mathbf{h})\big] &= V_H\big[\Lambda \cap \mathcal{Q}(\mathbf{b},\mathbf{h})\big] = V_H\big[\mathcal{Q}(\mathbf{b},\mathbf{h})\big] - V_H\big[\mathcal{Q}(\mathbf{b} + \mathbf{h}, -\mathbf{h}) \cap \Lambda^C\big] \\ &= V_H\big[\mathcal{Q}(\mathbf{b},\mathbf{h})\big] - V_H\big[\mathcal{S}(\mathbf{b} + \mathbf{1} \circ \mathbf{h}, \mathbf{0} \circ \mathbf{p} + (\mathbf{1} - 2 \cdot \mathbf{1}) \circ \mathbf{h})\big]. \qquad (16) \end{aligned}$$

Substituting (16) in (15), and recalling the definition of the $\mathbf{v}_j$'s, we finally get (12).

The case $\varphi(\mathbf{b}) > s$, $\mathbf{p} \in \mathbb{R}^d_-$ is analogous. The cases $\varphi(\mathbf{b}) \leq s$, $\mathbf{p} \in \mathbb{R}^d_-$, and $\varphi(\mathbf{b}) > s$, $\mathbf{p} \in \mathbb{R}^d_+$, are trivial, because they imply $V_H[\mathcal{S}(\mathbf{b},\mathbf{p})] = 0$ and $V_H[\mathcal{Q}(\mathbf{b},\mathbf{h})] = V_H[\mathcal{S}(\mathbf{b} + \mathbf{h}, -\mathbf{h})]$. □

Note that Theorem 4.2 holds for all measurable functions $\varphi : \mathbb{R}^d \to \mathbb{R}$.

The idea behind the GAEP algorithm is to apply the decomposition (12) recursively to the non-empty quasi-simplexes on the right-hand side, starting with $\mathcal{S}_1^1 = \mathcal{S}(\mathbf{0}, \mathbf{p}_1^1)$, $\mathbf{p}_1^1$ being the unique vector satisfying (4). Note that, by changing the conditions (4) and the vector $\mathbf{b}_1^1$, the algorithm can be applied to the case in which the random vector $\mathbf{X}$ also takes negative values but it is still bounded from below as, for example, $\mathbb{P}[\mathbf{X} \geq \mathbf{b}_1^1] = 1$.

At the beginning of the nth iteration ($n \in \mathbb{N}$), the algorithm receives, as input, a family

$$\mathcal{S}_n^t = \mathcal{S}\big(\mathbf{b}_n^t, \mathbf{p}_n^t\big), \ t \in \mathbf{I}_n \subset \big\{1, \ldots, N^{n-1}\big\}$$

of non-empty quasi-simplexes. As already remarked, for $n = 1$, we have $\mathbf{I}_1 = \{1\}$ and $\mathcal{S}_1^1 = \mathcal{S}(\mathbf{0}, \mathbf{p}_1^1)$. Given a sequence of splitting points $\mathbf{h}_n^t \in \mathcal{H}(\mathbf{p}_n^t)$, $t \in \mathbf{I}_n$, each quasi-simplex $\mathcal{S}_n^t$ is then decomposed into one hypercube and N quasi-simplexes via (12):

$$\begin{aligned} V_H\big[\mathcal{S}\big(\mathbf{b}_n^t, \mathbf{p}_n^t\big)\big] &= V_H\big[\mathcal{Q}\big(\mathbf{b}_n^t, \mathbf{h}_n^t\big)\big] \\ &\quad + \sum_{j=1}^{N} m_j V_H\big[\mathcal{S}\big(\mathbf{b}_n^t + \mathbf{v}_j \circ \mathbf{h}_n^t, \mathbf{i}_j \circ \mathbf{p}_n^t + (\mathbf{1} - 2\mathbf{v}_j) \circ \mathbf{h}_n^t\big)\big] \\ &= V_H\big[\mathcal{Q}\big(\mathbf{b}_n^t, \mathbf{h}_n^t\big)\big] + \sum_{j=1}^{N} m_j V_H\big[\mathcal{S}\big(\mathbf{b}_{n+1}^{Nt-N+j}, \mathbf{p}_{n+1}^{Nt-N+j}\big)\big], \ \ t \in \mathbf{I}_n, \end{aligned} \tag{17}$$

where the sequences $\mathbf{b}_n^t$ and $\mathbf{p}_n^t$ are defined by their initial values $\mathbf{b}_1^1 = \mathbf{0}$ and $\mathbf{p}_1^1$, and by

$$\mathbf{b}_{n+1}^{Nt-N+j} = \mathbf{b}_n^t + \mathbf{v}_j \circ \mathbf{h}_n^t, \tag{18}$$

$$\mathbf{p}_{n+1}^{Nt-N+j} = \mathbf{i}_j \circ \mathbf{p}_n^t + \big(\mathbf{1} - 2\mathbf{v}_j\big) \circ \mathbf{h}_n^t, \tag{19}$$

for all $j = 1, \ldots, N$ and $t \in \mathbf{I}_n$. Recall from Theorem 4.2 that $m_1 = -1$ and $m_j = 1$ for $j = 2, \ldots, N$. At this point,

$$\mathcal{S}\big(\mathbf{b}_{n+1}^t, \mathbf{p}_{n+1}^t\big), \ t \in \mathbf{I}_{n+1}, \text{ with } \mathbf{I}_{n+1} = \big\{t \in \{1, \ldots, N^n\} : \mathcal{S}\big(\mathbf{b}_{n+1}^t, \mathbf{p}_{n+1}^t\big) \neq \emptyset\big\},$$

i.e. the family of non-empty quasi-simplexes obtained on the right-hand side of (17) is passed to the $(n+1)$th iteration of the algorithm.

As an example, we illustrate the first iteration of the algorithm in the case $d = 3$, where we denote $\mathbf{p}_1^1 = \mathbf{p}$ with $\mathbf{0} \leq \mathbf{h} \leq \mathbf{p}$. For the simplex $\mathcal{S}_1^1 = \mathcal{S}(\mathbf{0}, \mathbf{p})$, the measure decomposition (12) gives

$$\begin{aligned} V_H\big[\mathcal{S}(\mathbf{0}, \mathbf{p})\big] = {} & V_H\big[\mathcal{Q}(\mathbf{0}, \mathbf{h})\big] - V_H\big[\mathcal{S}(\mathbf{h}, -\mathbf{h})\big] + V_H\big[\mathcal{S}\big((h_1, 0, 0), (p_1 - h_1, h_2, h_3)\big)\big] \\ & + V_H\big[\mathcal{S}\big((0, h_2, 0), (h_1, p_2 - h_2, h_3)\big)\big] \\ & + V_H\big[\mathcal{S}\big((0, 0, h_3), (h_1, h_2, p_3 - h_3)\big)\big] \\ & + V_H\big[\mathcal{S}\big((h_1, h_2, 0), (p_1 - h_1, p_2 - h_2, h_3)\big)\big] \\ & + V_H\big[\mathcal{S}\big((h_1, 0, h_3), (p_1 - h_1, h_2, p_3 - h_3)\big)\big] \\ & + V_H\big[\mathcal{S}\big((0, h_1, h_2), (h_1, p_2 - h_2, p_3 - h_3)\big)\big] + V_H\big[\mathcal{S}\big(\mathbf{h}, \mathbf{p} - \mathbf{h}\big)\big]. \end{aligned}$$

In Figure 4, we illustrate the case in which $\varphi(\mathbf{p} \circ \mathbf{e}_k) = s$, $k \in D$ (see condition (4)), $\varphi(h_1, h_2, 0) \leq s$, $\varphi(h_1, 0, h_3) \leq s$, and $\varphi(0, h_2, h_3) > s$, leading to $\mathcal{S}_2^7 = \mathcal{S}_2^8 = \emptyset$. Therefore, we have $\mathbf{I}_1 = \{1\}$ and $\mathbf{I}_2 = \{1, 2, 3, 4, 5, 6\}$. Note also that, in the 2D cases described in Figure 3, we had $\mathbf{I}_2 = \{1, 2, 3\}$ (left) and $\mathbf{I}_2 = \{2, 3, 4\}$ (right).

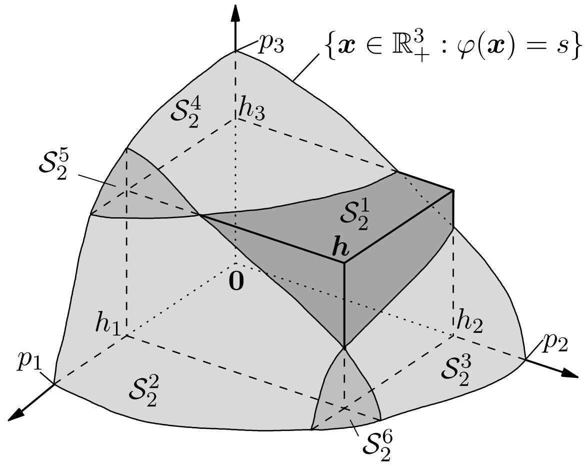

Figure 4. An illustration of the decomposition (12) of $\mathcal{S}(\mathbf{0}, \mathbf{p}) \subset \mathbb{R}^3$ for some vector $\mathbf{h} \in \mathbb{R}^d_+$.

In general, the set of indexes $\mathbf{I}_{n+1}$, which identifies the quasi-simplexes $\mathcal{S}^t_{n+1} \neq \emptyset$, depends on the vectors $\mathbf{h}^t_n$, $t \in \mathbf{I}_n$ and on the function φ, at each iteration of the algorithm. Nevertheless, for a fixed n, the quasi-simplexes $\mathcal{S}^t_{n+1}$, $t \in \mathbf{I}_{n+1}$ are always disjoint (this follows from the proof of Theorem 4.2).

Now, define the family $\mathcal{Q}^t_n = \mathcal{Q}(\mathbf{b}^t_n, \mathbf{h}^t_n)$, $t \in \mathbf{I}_n$, and the sequence P_n as the sum of the V_H-measures of all the $\mathcal{Q}^t_n$'s multiplied by the corresponding τ^t_n, as

$$P_n = P_{n-1} + \sum_{t \in \mathbf{I}_n} \tau^t_n V_H\left[\mathcal{Q}^t_n\right] = \sum_{i=1}^{n} \sum_{t \in \mathbf{I}_i} \tau^t_i V_H\left[\mathcal{Q}^t_i\right], \; n \in \mathbb{N}, \tag{20}$$

where $P_0 = 0$ and the sequence τ^t_n is defined by its initial value $\tau^1_1 = 1$ and by

$$\tau^{Nt-N+j}_{n+1} = \tau^t_n m_j, \quad \text{for all } j = 1, \ldots N \quad \text{and } t \in \mathbf{I}_n. \tag{21}$$

The value $\tau^t_n \in \{-1, 1\}$ indicates whether the measure of the quasi-simplex $\mathcal{S}^t_n$ has to be added ($\tau^t_n = 1$) or subtracted ($\tau^t_n = -1$) in order to compute an approximation of $V_H\left[\mathcal{S}^1_1\right]$.

We now show that, at each iteration of the algorithm, the error committed by taking P_n as an approximation of $V_H\left[\mathcal{S}^1_1\right]$ can be expressed in terms of the measures of the non-empty quasi-simplexes $\mathcal{S}^t_{n+1}$, $t \in \mathbf{I}_{n+1}$ passed to the $(n+1)$th iteration of the algorithm.

Theorem 4.3. With the notation introduced above, we have

$$V_H\left[\mathcal{S}^1_1\right] - P_n = \sum_{t \in \mathbf{I}_{n+1}} \tau^t_{n+1} V_H\left[\mathcal{S}^t_{n+1}\right]. \tag{22}$$

Proof. We proceed by induction on n. Recalling that $\tau^1_1 = 1$ and $P_0 = 0$, (22) corresponds, for $n = 0$, to the trivial equality $V_H\left[\mathcal{S}^1_1\right] = V_H\left[\mathcal{S}^1_1\right]$. Hence, we suppose that (22) holds for some $n \in \mathbb{N}$. Substituting (17) in (22), and using (20), yields

$$\begin{aligned} V_H\left[\mathcal{S}^1_1\right] &= P_n + \sum_{t \in \mathbf{I}_{n+1}} \tau^t_{n+1} V_H\left[\mathcal{Q}^t_{n+1}\right] + \sum_{t \in \mathbf{I}_{n+1}} \tau^t_{n+1} \left(\sum_{j=1}^{N} m_j V_H\left[\mathcal{S}^{Nt-N+j}_{n+2}\right] \right) \\ &= P_{n+1} + \sum_{t \in \mathbf{I}_{n+1}} \sum_{j=1}^{N} \tau^t_{n+1} m_j V_H\left[\mathcal{S}^{Nt-N+j}_{n+2}\right]. \end{aligned}$$

Recalling (21), we get

$$V_H\big[\mathcal{S}_1^1\big] = P_{n+1} + \sum_{t\in\mathbf{I}_{n+1}} \sum_{j=1}^{N} \tau_{n+2}^{Nt-N+j} V_H\big[\mathcal{S}_{n+2}^{Nt-N+j}\big] = P_{n+1} + \sum_{t\in\mathbf{I}_{n+2}} \tau_{n+2}^{t} V_H\big[\mathcal{S}_{n+2}^{t}\big],$$

where the last equality follows from the fact that the simplexes $\mathcal{S}_{n+2}^t,\ t \notin \mathbf{I}_{n+2}$ are empty. □

5. Convergence of the algorithm

In this section, we give sufficient conditions for the convergence of the sequence P_n, defined above, to the value $V_H\big[\mathcal{S}_1^1\big]$. First, in Lemma 5.1, we give a simple set representation of P_n. Recall that the set $\mathbf{I}_{n+1}$ identifies the non-empty quasi-simplexes $\mathcal{S}_{n+1}^t,\ t \in \mathbf{I}_{n+1}$, which are passed to the $(n+1)$th iteration of the algorithm. We partition the set $\mathbf{I}_{n+1}$ into the families

$$\mathbf{I}_{n+1}^{+} = \{t \in \mathbf{I}_n : \tau_{n+1}^t = +1\} \quad \text{and} \quad \mathbf{I}_{n+1}^{-} = \{t \in \mathbf{I}_n : \tau_{n+1}^t = -1\}.$$

Lemma 5.1. For any $n \in \mathbb{N}$, we have that $P_n = V_H[\mathcal{B}_n]$, where

$$\mathcal{B}_n = \left(\mathcal{S}_1^1 \bigcup_{t\in\mathbf{I}_{n+1}^{-}} \mathcal{S}_{n+1}^t\right) \setminus \bigcup_{t\in\mathbf{I}_{n+1}^{+}} \mathcal{S}_{n+1}^t. \tag{23}$$

Proof. Using induction, for a fixed n and all $t \in \mathbf{I}_n$, we prove that

$$\mathcal{S}_n^t \subset \begin{cases} \mathcal{S}_1^1, & \text{if } \tau_n^t = +1, \\ \mathcal{Q}(\mathbf{0}, \mathbf{p}_1^1)\setminus\mathcal{S}_1^1, & \text{if } \tau_n^t = -1. \end{cases}$$

Then, the result easily follows from (22) and the fact that, for a fixed n, the $\mathcal{S}_{n+1}^t$'s are disjoint.

From its definition (3), a quasi-simplex $\mathcal{S}(\mathbf{b}, \mathbf{p})$ lies in $\mathcal{S}_1^1$ if and only if $\mathbf{p} \in \mathbb{R}_d^+$. It lies instead in $\mathcal{Q}(\mathbf{0}, \mathbf{p}_1^1)\setminus\mathcal{S}_1^1$ if and only if $\mathbf{p} \in \mathbb{R}_d^-$. Therefore, we equivalently have to show that, for a fixed n and all $t \in \mathbf{I}_n$,

$$\mathbf{p}_n^t \in \begin{cases} \mathbb{R}_d^+, & \text{if } \tau_n^t = +1, \\ \mathbb{R}_d^-, & \text{if } \tau_n^t = -1. \end{cases} \tag{24}$$

To this end, assume that (24) is true for a fixed n (for $n = 1$, it trivially holds) and choose an arbitrary $\mathcal{S}_{n+1}^t,\ t \in \mathbf{I}_{n+1}$ such that $\tau_{n+1}^t = +1$ (the case $\tau_{n+1}^t = -1$ is analogous). By (21), there exist $t' \in \mathbf{I}_n$ and $j \in \{1, \ldots, N\}$, with $\tau_{n+1}^t = \tau_{n+1}^{Nt'-N+j} = \tau_n^{t'} m_j$. Since $\tau_{n+1}^t = +1$, either (case I) $\tau_n^{t'} = -1$ and $m_j = -1$ (in this case, $j = 1$) or (case II) $\tau_n^{t'} = 1$ and $m_j = +1$ (in this case, $j \neq 1$).

Case I. If $j = 1$, using (19) we obtain $\mathbf{p}_{n+1}^{Nt'-N+j} = -\mathbf{h}_n^{t'}$. Since it also holds that $\tau_n^{t'} = -1$, using the induction assumption, we obtain that $\mathbf{p}_n^t \in \mathbb{R}_d^-$ and finally (recall that $\mathbf{h}_n^t \in \mathcal{H}(\mathbf{p})$) $\mathbf{p}_{n+1}^{Nt'-N+j} = -\mathbf{h}_n^t \in \mathbb{R}_d^+$.

Case II. For $j \neq 1$, (11) yields

$$\left(\mathbf{p}_{n+1}^{Nt'-N+j}\right)_k = \begin{cases} (\mathbf{h}_n^{t'})_k, & \text{if } (\mathbf{i}_j)_k = 0, \\ (\mathbf{p}_n^{t'} - \mathbf{h}_n^{t'})_k, & \text{if } (\mathbf{i}_j)_k = 1. \end{cases}$$

Since it also holds that $\tau_n^{t'} = +1$, using the induction assumption, we obtain that $\mathbf{p}_n^{t'} \in \mathbb{R}_d^+$ and finally $\mathbf{p}_{n+1}^{Nt'-N+j} \in \mathbb{R}_d^+$. □

A simple illustration of (23) in dimension $d = 2$ is given by (7) and by the corresponding Figure 3. Now, we use Lemma 5.1 to obtain bounds on P_n.

Theorem 5.2. If $\varphi \in \mathcal{N}$, then, for all $n \in \mathbb{N}$, we have

$$\mathbb{P}\big[\varphi(\mathbf{X}) \le l_n\big] \le P_n \le \mathbb{P}\big[\varphi(\mathbf{X}) \le u_n\big], \tag{25}$$

where

$$l_n = \min\left\{s, \min_{t \in \mathbf{I}_{n+1}^+} \varphi(\mathbf{b}_{n+1}^t)\right\} \text{ and } u_n = \max\left\{s, \max_{t \in \mathbf{I}_{n+1}^-} \varphi(\mathbf{b}_{n+1}^t)\right\}.$$

Proof. Suppose that, for a quasi-simplex $\mathcal{S}_{n+1}^t$, we have $\tau_{n+1}^t = -1$. Then (see (24)) $\mathbf{p}_{n+1}^t \le \mathbf{0}$. Therefore, over the quasi-simplex $\mathcal{S}_{n+1}^t$, the function $\varphi \in \mathcal{N}$ attains its maximum at $\mathbf{b}_{n+1}^t$, i.e.

$$\max_{\mathbf{x} \in \mathcal{S}_{n+1}^t} \varphi(\mathbf{x}) = \varphi(\mathbf{b}_{n+1}^t).$$

Using the above result and (23), we can write

$$\mathcal{B}_n \subset \mathcal{S}_1^1 \bigcup_{t \in \mathbf{I}_{n+1}^-} \mathcal{S}_{n+1}^t \subset \left\{\mathbf{x} \in \mathbb{R}_+^d : \varphi(\mathbf{x}) \le \max\left\{s, \max_{t \in \mathbf{I}_{n+1}^-} \varphi(\mathbf{b}_{n+1}^t)\right\}\right\},$$

which yields $\mathcal{B}_n \subset \{\mathbf{x} \in \mathbb{R}_+^d : \varphi(\mathbf{x}) \le u_n\}$. Recalling from Lemma 5.1 that $P_n = V_H[\mathcal{B}_n]$, the right-hand side of (25) follows. Analogously, if $\tau_{n+1}^t = +1$, we have that $\mathbf{p}_{n+1}^t \ge \mathbf{0}$ and

$$\inf_{\mathbf{x} \in \mathcal{S}_{n+1}^t} \varphi(\mathbf{x}) = \varphi(\mathbf{b}_{n+1}^t).$$

Recalling that $\mathbf{b}_{n+1}^t \notin \mathcal{S}_{n+1}^t$ when $\mathbf{p}_{n+1}^t \ge \mathbf{0}$, we obtain that $\varphi(\mathbf{b}_{n+1}^t) < \varphi(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{S}_{n+1}^t$. Using again (23), it follows that

$$\left\{\mathbf{x} \in \mathbb{R}_+^d : \varphi(\mathbf{x}) \le \min\left\{s, \min_{t \in \mathbf{I}_n^+} \varphi(\mathbf{b}_{n+1}^t)\right\}\right\} \subset \mathcal{S}_1^1 \setminus \bigcup_{t \in \mathbf{I}_{n+1}^+} \mathcal{S}_{n+1}^t \subset \mathcal{B}_n,$$

which yields $\{\mathbf{x} \in \mathbb{R}_+^d : \varphi(\mathbf{x}) \le l_n\} \subset \mathcal{B}_n$ and, consequently, the left-hand side of (25). □

Figure 5 illustrates Theorem 5.2 for the first two iterations of the algorithm in the case $d = 2$. Now, we define the sequence D_n, $n \in \mathbb{N}$, as

$$D_n = \max\{s - l_n, u_n - s\} = \max_{t \in \mathbf{I}_{n+1}} \left|\varphi(\mathbf{b}_{n+1}^t) - s\right|. \tag{26}$$

The following lemma will turn out to be useful in the remainder of the paper.

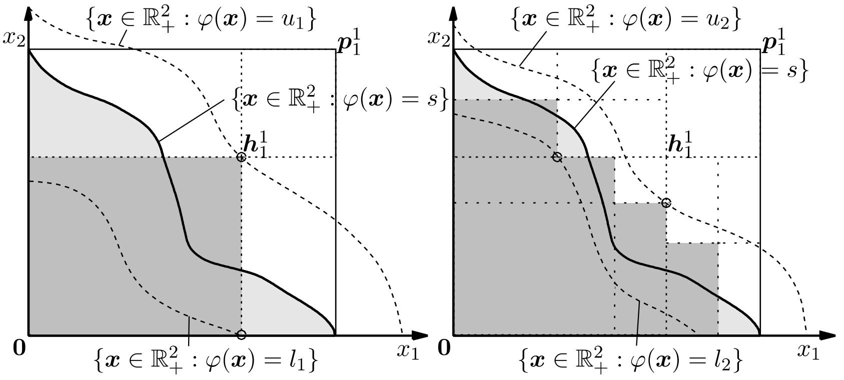
$\{\boldsymbol{x} \in \mathbb{R}^2_+ : \varphi(\boldsymbol{x}) = u_1\}$
x_2
$\boldsymbol{p}_1^1$
$\{\boldsymbol{x} \in \mathbb{R}^2_+ : \varphi(\boldsymbol{x}) = s\}$
$\boldsymbol{h}_1^1$
$\mathbf{0}$
$\{\boldsymbol{x} \in \mathbb{R}^2_+ : \varphi(\boldsymbol{x}) = l_1\}$
x_1
$\{\boldsymbol{x} \in \mathbb{R}^2_+ : \varphi(\boldsymbol{x}) = u_2\}$
x_2
$\boldsymbol{p}_1^1$
$\{\boldsymbol{x} \in \mathbb{R}^2_+ : \varphi(\boldsymbol{x}) = s\}$
$\boldsymbol{h}_1^1$
$\mathbf{0}$
$\{\boldsymbol{x} \in \mathbb{R}^2_+ : \varphi(\boldsymbol{x}) = l_2\}$
x_1

Figure 5. An illustration of Theorem 5.2 for $d = 2$, $n = 1$ (left), and $n = 2$ (right). The dark grey area identifies the sets $\mathcal{B}_1 = \mathcal{Q}_1^1$ (left) and $\mathcal{B}_2 = (\mathcal{Q}_1^1 \cup \mathcal{Q}_2^3 \cup \mathcal{Q}_2^3 \cup \mathcal{Q}_2^4) \backslash \mathcal{Q}_2^1$ (right). We set $\mathbf{h}_2^4 = \mathbf{0}$, thus $\mathcal{Q}_2^4 = \emptyset$. The circles indicate where φ attains either its maximum or its minimum.

Lemma 5.3. If $\varphi \in \mathcal{N}$, then, for all $n \in \mathbb{N}$, we have

$$\left| P_n - V_H[\mathcal{S}_1^1] \right| \leq \mathbb{P}[s - D_n < \varphi(\mathbf{X}) \leq s + D_n].$$

Proof. From (25) and (26), we have that

$$P_n \leq \mathbb{P}[\varphi(\mathbf{X}) \leq u_n] \leq \mathbb{P}[\varphi(\mathbf{X}) \leq s + D_n].$$

Since $D_n \geq 0$, we obtain

$$\begin{aligned} P_n - V_H[\mathcal{S}_1^1] &\leq \mathbb{P}[\varphi(\mathbf{X}) \leq s + D_n] - \mathbb{P}[\varphi(\mathbf{X}) \leq s] \\ &= \mathbb{P}[s < \varphi(\mathbf{X}) \leq s + D_n] \leq \mathbb{P}[s - D_n < \varphi(\mathbf{X}) \leq s + D_n]. \end{aligned}$$

Analogously, we can write

$$\begin{aligned} P_n - V_H[\mathcal{S}_1^1] &\geq \mathbb{P}[\varphi(\mathbf{X}) \leq s - D_n] - \mathbb{P}[\varphi(\mathbf{X}) \leq s] \\ &= -\mathbb{P}[s - D_n < \varphi(\mathbf{X}) \leq s] \geq -\mathbb{P}[s - D_n < \varphi(\mathbf{X}) \leq s + D_n]. \end{aligned}$$

□

Recall the definition of D_n in (26).

Theorem 5.4. Assume that $\mathbf{X}$ is absolutely continuous and $\lim_{n \to \infty} D_n = 0$. Then

$$\lim_{n \to \infty} P_n = \mathbb{P}\big[\varphi(\mathbf{X}) \leq s\big] = V_H[\mathcal{S}_1^1].$$

Proof. If $\lim_{n \to \infty} D_n = 0$, and $\varphi(\mathbf{X})$ is continuous, then the theorem follows from Lemma (5.3). Therefore, it is sufficient to show that $\mathbb{P}[\varphi(\mathbf{X}) = s] = 0$, for all $s \in \mathbb{R}$. Fix some $s \in \mathbb{R}$, and let $\Gamma_s = \{\mathbf{x} \in \mathbb{R}^d : \varphi(\mathbf{x}) = s\}$. Since $\varphi \in \mathcal{N}$, for each $\mathbf{y} \in \mathbb{R}^{d-1}$, there exists a unique $z_{\mathbf{y}} \in \mathbb{R}$ such that $\varphi(y_1, \ldots, y_{d-1}, z_{\mathbf{y}}) = s$. It is easy to see that $\Gamma_s = \{(y_1, \ldots, y_{d-1}, z_{\mathbf{y}}) : \mathbf{y} \in \mathbb{R}^{d-1}\}$ can be written as a countable union of sets with (Hausdorff) dimension $d - 1$, see [2] (pp. 54–59). Hence, the Lebesgue measure of Γ_s is zero. As $\mathbf{X}$ is absolutely continuous, V_H is absolutely continuous with respect to the Lebesgue measure. Thus, $V_H[\Gamma_s] = \mathbb{P}[\varphi(\mathbf{X}) = s] = 0$. □

Note that, in Theorem 5.4, the assumption of absolute continuity of $\mathbf{X}$ cannot be dropped. As a counterexample, take $d = 2$ and $X = (U, 1 - U)$, where U is a random variable uniformly distributed on [0, 1]. For the function $\varphi(\mathbf{x}) = x_1 + x_2$, we have that $\mathbf{X}$ is continuous, while $\mathbb{P}[\varphi(\mathbf{X}) = 1] = 1$. In this case, and depending on the sequence $\mathbf{h}^t_n$, the sequence P_n may fail to converge.

6. The choice of h^t_n: the bisection rule

Assuming the continuity of $\varphi(\mathbf{X})$, convergence of the GAEP algorithm is guaranteed by Theorem 5.4 whenever the sequence D_n goes to zero. Of course, the (speed of) convergence of the sequence D_n depends on the choice of the vectors $\{\mathbf{h}^t_n, t \in \mathbf{I}_n\}$ at each iteration. The aim of this and the following section is to find good criteria for the choice of $\mathbf{h}^t_n$ such that D_n converges (rapidly) to 0.

Note that, whenever $\mathcal{S}^t_n = \mathcal{Q}(\mathbf{b}^t_n, \mathbf{p}^t_n)$, in (17) it is convenient to set $\mathbf{h}^t_n = \mathbf{p}^t_n$, so that no simplexes are passed on to the following iteration of the algorithm. In fact, using (19), having $\mathbf{h}^t_n = \mathbf{p}^n_t$ implies that $\mathcal{S}^{Nt-N+1}_{n+1} = \mathcal{S}(\mathbf{b}^t_n + \mathbf{p}^t_n, -\mathbf{p}^t_n) = \mathcal{Q}(\mathbf{b}^t_n, \mathbf{p}^t_n) \backslash \mathcal{S}^t_n = \emptyset$. Moreover, for $j = 2, \ldots, N$, (11) implies that at least one component of $\mathbf{p}^{Nt-N+j}_{n+1}$ is zero, hence $\mathcal{S}^{Nt-N+j}_{n+1} = \emptyset$.

As a first choice for $\mathbf{h}^t_n$, we propose the so-called *bisection rule*. Using this choice, convergence of the sequence D_n is guaranteed under some extra assumptions on φ.

Theorem 6.1. Assume that $\varphi \in \mathcal{N}'$, the set of all twice differentiable functions $\varphi \in \mathcal{N}$ for which there exists a constant $r > 0$ such that $\partial_k \varphi(\mathbf{x}) > r$ for all $k \in D$ and $\mathbf{x} \in \mathbb{R}^d$. For all $n \in \mathbb{N}$ and $t \in \mathbf{I}_n$, let the sequence $\mathbf{h}^t_n$ be defined as

$$\mathbf{h}^t_n = \begin{cases} \mathbf{p}^t_n, & \text{if } \mathcal{S}^t_n = \mathcal{Q}^t_n, \\ 1/2\mathbf{p}^t_n, & \text{otherwise.} \end{cases} \tag{27}$$

Then $D_n = O(2^{-n}), n \to \infty$.

Proof. It is immediate that $\mathbf{h}^t_n \in \mathcal{H}(\mathbf{p}^t_n)$, and hence $\mathbf{h}^t_n$ is correctly defined. Since φ is twice differentiable, it is also Lipschitz continuous on $\bar{\mathcal{Q}}(\mathbf{0}, \mathbf{p}^1_1)$, the closure of $\mathcal{Q}(\mathbf{0}, \mathbf{p}^1_1)$. Thus, there exists a constant $L < \infty$ such that $|\varphi(\mathbf{x}) - \varphi(\mathbf{y})| < L\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \bar{\mathcal{Q}}(\mathbf{0}, \mathbf{p}^1_1)$.

Now, consider the non-empty quasi-simplex $\mathcal{S}^t_{n+1}, t \in \mathbf{I}_{n+1}$, for which there exist $t' \in \mathbf{I}_n$ and $j \in \{1, \ldots, N\}$ such that $\mathcal{S}^t_{n+1} = \mathcal{S}^{Nt'-N+j}_{n+1}$. For the quasi-simplex $\mathcal{S}^{t'}_n$, we have that (from the definition (3)) $\mathcal{S}^{t'}_n = \mathcal{S}(\mathbf{b}^{t'}_n, \mathbf{p}^{t'}_n) \subset \mathcal{Q}(\mathbf{b}^{t'}_n, \mathbf{p}^{t'}_n)$. As $\mathcal{S}^t_{n+1} \neq \emptyset$, (27) implies that $\mathcal{Q}(\mathbf{b}^{t'}_n, \mathbf{p}^{t'}_n) \backslash \mathcal{S}^{t'}_n \neq \emptyset$.

It is then possible to find $\mathbf{y}^{t'}_n \in \mathcal{S}^{t'}_n$ and $\mathbf{z}^{t'}_n \in \mathcal{Q}^{t'}_n \backslash \mathcal{S}^{t'}_n$, for which we have that either $\varphi(\mathbf{y}^{t'}_n) \leq s$, $\varphi(\mathbf{z}^{t'}_n) > s$ (when $\mathbf{p}^{t'}_n \geq \mathbf{0}$) or $\varphi(\mathbf{y}^{t'}_n) > s$, $\varphi(\mathbf{z}^{t'}_n) \leq s$ (when $\mathbf{p}^{t'}_n \leq \mathbf{0}$). In both cases, continuity of φ guarantees (by the intermediate point theorem) the existence of a vector $\mathbf{x}^{t'}_n$ on the curve $c : [0, 1] \mapsto (1 - c)\mathbf{y}^{t'}_n + c\mathbf{z}^{t'}_n$, with $\varphi(\mathbf{x}^{t'}_n) = s$. Clearly, $\mathbf{x}^{t'}_n \in \bar{\mathcal{Q}}^{t'}_n$, and by (18) and (10), also $\mathbf{b}^{Nt'-N+j}_{n+1} \in \bar{\mathcal{Q}}^{t'}_n$. As a consequence, we have that $\left\|\mathbf{b}^{Nt'-N+j}_{n+1} - \mathbf{x}^{t'}_n\right\| \leq \|\mathbf{p}^{t'}_n\|$. Therefore, for each $t \in \mathbf{I}_{n+1}$, there exists $t' \in \mathbf{I}_n$ such that

$$\begin{aligned} \left|\varphi(\mathbf{b}^t_{n+1}) - s\right| &= \left|\varphi(\mathbf{b}^{Nt'-N+j}_{n+1}) - s\right| = \left|\varphi(\mathbf{b}^{Nt'-N+j}_{n+1}) - \varphi(\mathbf{x}^{t'}_n)\right| \\ &\leq L\left\|\mathbf{b}^{Nt'-N+j}_{n+1} - \mathbf{x}^{t'}_n\right\| \leq L\left\|\mathbf{p}^{t'}_n\right\|. \end{aligned}$$

Combining the above result with the definition of D_n given in (26), it follows that

$$D_n = \max_{t\in\mathbf{I}_{n+1}} \left|\varphi(\mathbf{b}^t_{n+1}) - s\right| \leq L \max_{t\in\mathbf{I}_n} \left\|\mathbf{p}^t_n\right\|. \tag{28}$$

Note that (28) holds for a general sequence $\mathbf{h}^t_n, t \in \mathbf{I}_n$. Using the definition of $\mathbf{p}^{Nt-N+j}_{n+1}$ (see (19)) and the bisection rule (27), it follows that

$$\begin{aligned}\mathbf{p}^{Nt-N+j}_{n+1} &= \mathbf{i}_j \circ \mathbf{p}^t_n + (\mathbf{1} - 2\mathbf{v}_j) \circ \mathbf{h}^t_n = \mathbf{i}_j \circ \mathbf{p}^t_n + (\mathbf{1} - 2\mathbf{v}_j) \circ 1/2\mathbf{p}^t_n \\ &= (\mathbf{i}_j + 1/2(\mathbf{1} - 2\mathbf{v}_j)) \circ \mathbf{p}^t_n = (\mathbf{i}_j - \mathbf{v}_j + 1/2\mathbf{1}) \circ \mathbf{p}^t_n = \begin{cases} 1/2\mathbf{p}^t_n & \text{if } j \neq 1, \\ -1/2\mathbf{p}^t_n & \text{if } j = 1. \end{cases}\end{aligned}$$

Thus, we have

$$\max_{t\in\mathbf{I}_{n+1}} \left\|\mathbf{p}^t_{n+1}\right\| = \frac{1}{2} \max_{t\in\mathbf{I}_n} \left\|\mathbf{p}^t_n\right\|.$$

Using (28), we finally obtain that

$$D_n \leq L \max_{t\in\mathbf{I}_n} \left\|\mathbf{p}^t_n\right\| = \frac{1}{2} L \max_{t\in\mathbf{I}_{n-1}} \left\|\mathbf{p}^t_{n-1}\right\| = \left(\frac{1}{2}\right)^{n-1} L \max_{t\in\mathbf{I}_1} \left\|\mathbf{p}^t_1\right\| = (1/2)^{n-1} L \left\|\mathbf{p}^1_1\right\|,$$

which implies the theorem. In the proof above, note that, even if $t \in \mathbf{I}_{n+1}$, there does not necessarily exist a vector $\mathbf{x}^t_{n+1} \in \bar{\mathcal{Q}}^t_{n+1}$ such that $\varphi(\mathbf{x}^t_{n+1}) = s$. This case occurs, for example, when $\mathcal{S}^t_{n+1} = \mathcal{Q}^t_{n+1}$. □

In order to have convergence of the bisection method, we only need φ to be Lipschitz continuous on $\bar{\mathcal{Q}}(\mathbf{0}, \mathbf{p}^1_1)$. However, to keep notation simple, we defined the smaller set of functions $\mathcal{N}'$, which we will use in the following section.

7. The choice of $\mathbf{h}^t_n$: the gradient rule

In this section, we present the *gradient method*, a different way of choosing the sequence $\mathbf{h}^t_n$, which guarantees a better asymptotic convergence rate in the case $d = 2$. By $\mathbf{x} \wedge \mathbf{y}$, denote the component-wise minimum of $\mathbf{x}$ and $\mathbf{y}$, and, by $\mathbf{x} \vee \mathbf{y}$, the component-wise maximum. We keep the assumption that $\varphi \in \mathcal{N}'$, see Theorem 6.1.

First, we use Taylor's expansion to find a constant $0 \leq R < \infty$ such that

$$\left|\varphi(\mathbf{b} + \delta) - \left(\varphi(\mathbf{b}) + \nabla\varphi(\mathbf{b})^T\delta\right)\right| \leq R\|\delta\|^2_\infty,$$

for all $\mathbf{x} \in \mathcal{Q}(\mathbf{b}, \mathbf{p})$ and $\mathbf{x} + \delta \in \mathcal{Q}(\mathbf{b}, \mathbf{p})$, where $\nabla\varphi$ denotes the gradient of φ. Since φ is twice differentiable, the constant R can be chosen to be the same for every $\mathbf{b} \in \bar{\mathcal{Q}}(\mathbf{0}, \mathbf{p}^1_1)$.

Theorem 7.1. Assume that $\varphi \in \mathcal{N}'$ and fix $\alpha \in (1/d, 1)$. For some $n \in \mathbb{N}$ and all $t \in \mathbf{I}_n$, let the sequence $\mathbf{h}^t_n$ be defined as

$$\mathbf{h}^t_n = \begin{cases} \mathbf{p}^t_n, & \text{if } \mathcal{S}^t_n = \mathcal{Q}(\mathbf{b}^t_n, \mathbf{p}^t_n), \\ (\mathbf{h}^{t*}_n \wedge \mathbf{p}^t_n) \vee \mathbf{0}, & \text{if } \mathcal{S}^t_n \neq \mathcal{Q}(\mathbf{b}^t_n, \mathbf{p}^t_n), \mathbf{p}^t_n \in \mathbb{R}^d_+, \\ (\mathbf{h}^{t*}_n \vee \mathbf{p}^t_n) \wedge \mathbf{0}, & \text{if } \mathcal{S}^t_n \neq \mathcal{Q}(\mathbf{b}^t_n, \mathbf{p}^t_n), \mathbf{p}^t_n \in \mathbb{R}^d_-, \end{cases} \tag{29}$$

where

$$\left(\mathbf{h}_n^{t*}\right)_k = \alpha \frac{s - \varphi(\mathbf{b}_n^t)}{\partial_k \varphi(\mathbf{b}_n^t)}, \quad \text{for all } k \in D.$$

Then, we have that

$$D_{n+1} \leq \left(\max\{1-\alpha, \alpha d - 1\} + \alpha^2 \frac{R}{r^2} D_n\right) D_n. \tag{30}$$

Before proving Theorem 7.1, we need the following result.

Lemma 7.2. Assume that $\varphi \in \mathcal{N}'$ and fix $n \in \mathbb{N}$. Let $\mathbf{h}_n^t, t \in \mathbf{I}_n$ be defined by (29). Then, for all $t \in \mathbf{I}_n$, we have that

$$\max_{j:\mathcal{S}_{n+1}^{Nt-N+j} \neq \emptyset} \left|\varphi(\mathbf{b}_{n+1}^{Nt-N+j}) - s\right| \leq \tag{31}$$

$$\left(\max\{1-\alpha, \alpha d - 1\} + \alpha^2 \frac{R}{r^2} \left|\varphi(\mathbf{b}_n^t) - s\right|\right) \left|\varphi(\mathbf{b}_n^t) - s\right|.$$

Proof. In the case $\mathcal{S}_n^t = \mathcal{Q}(\mathbf{b}_n^t, \mathbf{p}_n^t)$, there is nothing to prove. Suppose that $\varphi(\mathbf{b}_n^t) < s$, $\mathcal{S}_n^t \neq \mathcal{Q}(\mathbf{b}_n^t, \mathbf{p}_n^t)$, and $\mathbf{p} \in \mathbb{R}_+^d$, the other non-trivial case where $\varphi(\mathbf{b}_n^t) > s$ and $\mathbf{p} \in \mathbb{R}_-^d$ being analogous. It follows from (29) that, for all $j \in 1, \ldots, N$,

$$\begin{aligned}
\varphi(\mathbf{b}_{n+1}^{Nt-N+j}) - s &= \varphi(\mathbf{b}_n^t + \mathbf{v}_j \circ \mathbf{h}_n^t) - s \\
&\leq \varphi(\mathbf{b}_n^t + \mathbf{h}_n^{t*}) - s \leq \varphi(\mathbf{b}_n^t) + \nabla\varphi(\mathbf{b}_n^t)^T \mathbf{h}_n^{t*} - s + R\left\|\mathbf{h}_n^{t*}\right\|_\infty^2 \\
&= \varphi(\mathbf{b}_n^t) + d\alpha(s - \varphi(\mathbf{b}_n^t)) - s + R\left\|\mathbf{h}_n^{t*}\right\|_\infty^2 \\
&= (\alpha d - 1)(s - \varphi(\mathbf{b}_n^t)) + R\left\|\mathbf{h}_n^{t*}\right\|_\infty^2.
\end{aligned}$$

Recalling that $\partial_k \varphi(\mathbf{x}) > r$, $k \in D$, we also have $\|\mathbf{h}_n^{t*}\|_\infty \leq \alpha/r\left|\varphi(\mathbf{b}_n^t) - s\right|$, which gives

$$\varphi(\mathbf{b}_{n+1}^{Nt-N+j}) - s \leq |\varphi(\mathbf{b}_n^t) - s|\left((\alpha d - 1) + \alpha^2 \frac{R}{r^2} |\varphi(\mathbf{b}_n^t) - s|\right). \tag{32}$$

Note that $(\mathbf{v}_j \circ \mathbf{h}_n^t)_k = p_k$ implies $\mathcal{S}_{n+1}^{Nt-N+j} = \emptyset$. Hence, for all j with $\mathcal{S}_{n+1}^{Nt-N+j} \neq \emptyset$, we have $\mathbf{v}_j \circ \mathbf{h}_n^t = \mathbf{v}_j \circ \mathbf{h}_n^{t*}$. Thus,

$$\begin{aligned}
\varphi(\mathbf{b}_{n+1}^{Nt-N+j}) - s &= \varphi(\mathbf{b}_n^t + \mathbf{v}_j \circ \mathbf{h}_n^{t*}) - s \\
&\geq \varphi(\mathbf{b}_n^t) + \nabla\varphi(\mathbf{b}_n^t)^T(\mathbf{v}_j \circ \mathbf{h}_n^{t*}) - s - R\left\|v_j \circ \mathbf{h}_n^{t*}\right\|_\infty^2 \\
&\geq \varphi(\mathbf{b}_n^t) - s + \alpha\#\mathbf{v}_j(s - \varphi(\mathbf{b}_n^t)) - R(\alpha/r)^2(\varphi(\mathbf{b}_n^t) - s)^2.
\end{aligned}$$

As $\#\mathbf{v}_j \geq 1$, for all $j = 1, \ldots, N$, we finally get

$$\varphi(\mathbf{b}_{n+1}^{Nt-N+j}) - s \geq -|\varphi(\mathbf{b}_n^t) - s|\left(1 - \alpha + \alpha^2 \frac{R}{r^2} |\varphi(\mathbf{b}_n^t) - s|\right). \tag{33}$$

Combining (32) and (33) yields (31). □

Proof. [Proof of Theorem 7.1] First of all, note that $\mathbf{h}_n^t \in \mathcal{H}(\mathbf{p}_n^t)$, hence $\mathbf{h}_n^t$ is correctly defined. Due to Lemma 7.2, and recalling that $R > 0$, we have that

$$\begin{aligned} D_{n+1} &= \max_{t \in \mathbf{I}_{n+1}} |\varphi(\mathbf{b}_{n+1}^t) - s| \\ &= \max_{t \in \mathbf{I}_n} \max_{j : \mathcal{S}_{n+1}^{Nt-N+j} \neq \emptyset} \left|\varphi(\mathbf{b}_{n+1}^{Nt-N+j}) - s\right| \\ &\leq \max_{t \in \mathbf{I}_n} \left(\max\{1-\alpha, \alpha d - 1\} + \alpha^2 \frac{R}{r^2} \left|\varphi(\mathbf{b}_n^t) - s\right| \right) \left|\varphi(\mathbf{b}_n^t) - s\right| \\ &= \left(\max\{1-\alpha, \alpha d - 1\} + \alpha^2 \frac{R}{r^2} D_n \right) D_n. \end{aligned}$$

□

Now, we are ready to prove convergence of the gradient method.

Theorem 7.3. Assume that $\varphi \in \mathcal{N}'$ and fix $\alpha \in (1/d, 1)$. For some $\hat{n} \in \mathbb{N}$ and $\xi \in (0,1)$, assume that

$$\max\{1-\alpha, \alpha d - 1\} + \alpha^2 \frac{R}{r^2} D_{\hat{n}} = 1 - \xi < 1. \tag{34}$$

For all $n \geq \hat{n}$, let the sequence $\mathbf{h}_n^t$ be defined as (29). Then, we have that $\lim_{n\to\infty} D_n = 0$ and indeed

$$D_n = O\big((\max\{1-\alpha, \alpha d - 1\})^n\big).$$

Proof. Using (34) and (30) iteratively, we obtain that $D_n \leq (1-\xi)^{n-\hat{n}} D_{\hat{n}}$ for all $n \geq \hat{n}$. Hence, $\lim_{n\to\infty} D_n = 0$ and

$$\lim_{n\to\infty} \frac{D_{n+1}}{D_n} \leq \lim_{n\to\infty} \left(\max\{1-\alpha, \alpha d - 1\} + \alpha^2 \frac{R}{r^2} D_n \right) = \max\{1-\alpha, \alpha d - 1\}.$$

□

Condition (34), which guarantees the convergence of the gradient method with a twice differentiable function φ, can always be achieved using the bisection method for the first iterations of the algorithm. In fact, if one defines $\mathbf{h}_n^t$ using (27), the sequence D_n will go to zero (Theorem 6.1) and will satisfy (34) for some integer $\hat{n}$ large enough. From that $\hat{n}$ on, one can then use the gradient method with convergence guaranteed.

Finally, observe that $\alpha^* = 2/(d+1)$ minimizes $\max\{1-\alpha, \alpha d - 1\}$ with $\max\{1-\alpha^*, \alpha^* d - 1\} = (d-1)/(d+1)$. Thus, the rate

$$D_n = O\left(\left(\frac{d-1}{d+1} \right)^n \right) \tag{35}$$

is the best possible rate attainable using (29).

8. Applications

In this section, we test the GAEP algorithm on some random vectors $\mathbf{X} = (X_1, \ldots, X_d)$ and several functions φ. For illustrative reasons, we provide the joint distribution H of $\mathbf{X}$ in terms of the marginal distributions F_{X_i}, $i = 1, \ldots, d$, and a copula C. For the theory of copulas and the definition of the Gumbel and Clayton copula families, we refer the reader to Ref. [3].

In Table 1, we consider the 2D case ($d = 2$) with the Pareto marginals, i.e.

$$F_{X_i}(x) = \mathbb{P}[X_i \leq x] = 1 - (1 + x)^{-\theta_i}, \ x \geq 0, i = 1, 2,$$

with tail parameters $\theta_1 = 1$ and $\theta_2 = 2$. We couple these Pareto marginals via a Gumbel copula C_γ^{Gu} with a parameter $\gamma = 1.5$. For this example, we compute the approximation P_n (see (20)) using both the bisection and the gradient methods, for different values of the thresholds s and different numbers of iterations n of the algorithm. Here, we set $\varphi(x_1, x_2) = (1 + x_1)^{2/3}(1 + x_2)^{1/3} - 1$. For the gradient method, we provide the differences $P_n - P_{16}$ and their average computation times, for all iterations n and thresholds s. This has been done to show the speed of convergence of GAEP. The choice of $n = 16$ represents the maximum number of GAEP iterations allowed by the memory (4 GB RAM) of our laptop under the gradient method. Within the same table, we give the differences $P_n - P_{18}$, but by using the bisection method. Again, $n = 18$ is the maximum number of GAEP iterations under the bisection method. Note that these numbers are different because the number of quasi-simplexes produced at each iteration of GAEP and, consequently, the memory used by the algorithm depend on the method chosen. In Table 1, we also compute the ratio

$$R_n = \frac{D_n}{s - \varphi(\mathbf{0})}. \tag{36}$$

Since, from Lemma 5.3, we have

$$\left|P_n - V_H\left[\mathcal{S}_1^1\right]\right| \leq \mathbb{P}\left[1 - R_n < \frac{\varphi(\mathbf{X}) - \varphi(\mathbf{0})}{s - \varphi(\mathbf{0})} \leq 1 + R_n\right],$$

the sequence R_n provides a relative measure of convergence of the algorithm. Indeed, the convergence of R_n to $\mathbf{0}$ implies that the algorithm converges to a certain value. Note that, since analytical values for $V_H[\mathcal{S}_1^1]$ are not available for this example, nothing can be said about the correctness of the limit. However, for a 2D portfolio, we see that the estimate P_9 (for the gradient method) and P_{13} (bisection) could be already considered reasonably accurate and are both obtained in less than 0.1 s.

In Tables 2 ($d = 3$) to 4 ($d = 5$), we perform the same analysis for different Gumbel and Clayton models in which we progressively increase the number of Pareto random variables used, and we also change the function φ. In all tables, the reference values used represent the maximum number of iterations admissible under the corresponding method.

In all examples, the sequences R_n and D_n are decreasing to 0, indicating convergence of GAEP. A deeper study of the convergence rates of GAEP will be carried out in Section 9. At this stage, we only note that for $d = 2$, the gradient method, measured in terms of R_n, is more accurate than the bisection method, whereas for $d = 4, 5$, the opposite is the case. Memory constraints made estimates for $d \geq 6$ prohibitive.

9. Convergence rates and comparison with MC and QMC methods

In this section, we compare the GAEP algorithm to its main competitors for the estimation of $V_H[\mathcal{S}_1^1]$, which are the so-called Monte Carlo and quasi-Monte Carlo methods.

Table 1. Computation times and values for P_n (see (20)) and R_n (see (36)), for a random vector having two ($d = 2$) Pareto marginals with tail indexes $\theta_i = i, i = 1, 2$, coupled by a Clayton copula with parameter $\delta = 1.5$.

Gradient method P_{16} (ref. value)		$s = 0.01$ 0.00210380880234544		$s = 1$ 0.555768909195665		$s = 100$ 0.996898953638094		$s = 10{,}000$ 0.999987682453369	
		$P_n - P_{16}$	R_n	$P_n - P_{16}$	R_n	$P_n - P_{16}$	R_n	$P_n - P_{16}$	R_n
$n = 1$	(6.2e-05 s)	− 2.46e-04	3.38e-01	− 7.15e-02	5.58e-01	− 6.80e-03	9.51e-01	− 8.77e-05	9.97e-01
$n = 9$	(4.3e-02 s)	− 9.00e-12	5.18e-05	− 1.20e-08	1.50e-04	− 5.55e-09	3.23e-03	− 1.54e-10	1.69e-02
$n = 11$	(3.2e-01 s)	− 1.19e-13	5.76e-06	− 1.82e-10	1.67e-05	− 9.02e-11	3.61e-04	− 2.87e-12	1.93e-03
$n = 15$	(2.5e + 01 s)	− 1.64e-15	7.11e-08	8.36e-14	2.06e-07	6.99e-14	4.46e-06	− 2.17e-13	2.39e-05
Bisection method P_{18} (ref. value)		$s = 0.01$ 0.00210380882007647		$s = 1$ 0.555768910521824		$s = 100$ 0.997118688644563		$s = 10, 000$ 0.999999000151297	
		$P_n - P_{18}$	R_n	$P_n - P_{18}$	R_n	$P_n - P_{18}$	R_n	$P_n - P_{18}$	R_n
$n = 1$	(5.4e-05 s)	− 9.03e-04	4.99e-01	− 8.22e-02	5.46e-01	9.15e-04	5.01e+01	− 9.99e-07	5.00e+03
$n = 9$	(1.0e-02 s)	− 2.22e-06	3.93e-03	1.25e-04	1.03e-02	1.90e-03	1.17e+01	− 1.66e-09	1.25e+03
$n = 13$	(6.2e-02 s)	− 1.68e-08	2.46e-04	1.14e-06	6.72e-04	1.89e-03	4.07e+00	5.34e-11	4.95e+02
$n = 17$	(8.7e-01 s)	− 2.47e-10	1.54e-05	− 2.30e-08	4.20e-05	4.11e-04	1.08e+00	7.63e-12	1.96e+02

Note: For the function $\varphi(\mathbf{x}) = (x_1 + 1)^{2/3}(x_2 + 1)^{1/3} - 1$, we give the difference from the reference value P_{16} under the gradient method and from P_{18} under the bisection method.

Table 2. Computation times and values for P_n (see (20)) and R_n (see (36)), for a random vector having three ($d = 3$) Pareto marginals with tail indexes $\theta_i = i, i = 1, 2, 3$, coupled by a Gumbel copula with parameter $\delta = 2$.

Gradient method		$s = 1$		$s = 100$		$s = 10,000$		$s = 1,000,000$	
P_{11} (ref. value)		0.416132297573724		0.86094404059863		0.975469286382817		0.996034283083535	
		$P_n - P_{11}$	R_n	$P_n - P_{11}$	R_n	$P_n - P_{11}$	R_n	$P_n - P_{11}$	R_n
$n = 1$	(5.2e-05 s)	− 1.42e-01	8.23e-01	− 1.02e-01	8.23e-01	− 2.33e-02	8.23e-01	− 3.93e-03	8.23e-01
$n = 5$	(1.2e-02 s)	3.08e-04	1.49e-01	− 1.75e-04	1.49e-01	− 1.18e-04	1.49e-01	− 1.60e-05	1.49e-01
$n = 7$	(1.5e-01 s)	1.01e-04	3.84e-02	− 6.63e-06	3.83e-02	− 1.57e-05	3.84e-02	− 5.30e-06	3.84e-02
$n = 10$	(2.5e+01 s)	2.59e-06	6.01e-03	3.50e-07	6.01e-03	− 4.99e-07	6.02e-03	− 2.41e-07	6.01e-03
Bisection method		$s = 1$		$s = 100$		$s = 10,000$		$s = 1,000,000$	
P_{12} (ref. value)		0.416123502687784		0.860937621800580		0.975466112029285		0.996034158165874	
		$P_n - P_{12}$	R_n	$P_n - P_{12}$	R_n	$P_n - P_{12}$	R_n	$P_n - P_{12}$	R_n
$n = 1$	(5.2e-05 s)	− 1.42e-01	8.23e-01	− 1.02e-01	8.23e-01	− 2.33e-02	8.23e-01	− 3.93e-03	8.23e-01
$n = 5$	(1.1e-02 s)	− 6.97e-03	1.16e-01	− 4.49e-03	1.16e-01	− 1.06e-03	1.15e-01	− 1.27e-04	1.15e-01
$n = 7$	(7.1e-02 s)	− 9.88e-04	2.89e-02	− 8.05e-04	2.89e-02	− 2.52e-04	2.87e-02	− 4.26e-05	2.87e-02
$n = 11$	(1.6e+01 s)	− 1.26e-05	1.80e-03	− 1.01e-05	1.80e-03	− 5.63e-06	1.80e-03	− 1.39e-06	1.80e-03

Note: For the function $\varphi(\mathbf{x}) = (x_1)^{2.5} + (x_2)^{1.9} + (x_3)^{1.3}$, we give the difference from the reference value P_{11} under the gradient method and from P_{12} under the bisection method.

Table 3. Computation times and values for P_n (see (20)) and R_n (see (36)), for a random vector having four ($d = 4$) Pareto marginals with tail indexes $\theta_i = i, i = 1, \ldots, 4$, coupled by a Clayton copula with parameter 0.5.

Gradient method		$s = 0.01$		$s = 1$		$s = 100$		$s = 10,000$	
P_8 (ref. value)		0.00071182230075511		0.171145486287125		0.989555692750983		0.99989816481562	
		$P_n - P_8$	R_n	$P_n - P_8$	R_n	$P_n - P_8$	R_n	$P_n - P_8$	R_n
$n = 1$	(7.8e-05 s)	− 1.22e-05	8.92e-01	− 3.57e-02	6.94e-01	− 1.45e-02	6.52e-01	− 1.48e-04	6.43e-01
$n = 3$	(2.5e-03 s)	− 8.48e-05	5.65e-01	− 3.36e-02	5.90e-01	− 5.08e-03	5.95e-01	− 2.58e-05	5.95e-01
$n = 5$	(4.5e-01 s)	− 1.72e-05	3.51e-01	− 3.83e-03	3.53e-01	− 6.90e-04	3.57e-01	− 6.67e-06	3.57e-01
$n = 7$	(3.5e+01 s)	− 4.12e-06	2.24e-01	− 1.15e-03	.261e-01	− 8.88e-05	2.35e-01	− 1.16e-06	2.37e-01
Bisection method		$s = 0.01$		$s = 1$		$s = 100$		$s = 10,000$	
P_8 (ref. value)		0.000729352448761877		0.174626428280207		0.989734882116519		0.999899605726889	
		$P_n - P_8$	R_n	$P_n - P_8$	R_n	$P_n - P_8$	R_n	$P_n - P_8$	R_n
$n = 1$	(5.9e-05 s)	2.00e-05	1.34e+00	1.61e-03	1.11e+00	− 9.72e-03	1.06e + 00	− 9.96e-05	1.05e+00
$n = 3$	(7.2e-04 s)	1.77e-04	6.11e-01	3.26e-02	4.63e-01	− 1.19e-03	4.82e-01	− 1.39e-05	4.86e-01
$n = 5$	(7.9e-02 s)	8.70e-05	2.43e-01	1.80e-02	1.51e-01	− 1.36e-04	1.33e-01	− 2.83e-06	1.29e-01
$n = 7$	(5.7e+00 s)	1.47e-05	1.19e-01	2.55e-03	5.07e-02	1.98e-05	3.81e-02	− 3.94e-07	3.50e-02

Note: For the function $\varphi(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + 0.1(x_1^+ x_2^+)^{1/2} + 0.1(x_3^+ x_4^+)^{1/3}$, where $a^+ = \max(a, 0)$, we give the difference from the reference value P_8 under the gradient and the bisection methods.

Table 4. Computation times and values for P$_\text{n}$ (see (20)) and R_n (see (36)), for a random vector having five ($d = 5$) Pareto marginals with tail indexes $\theta_i = i, i = 1, \ldots, 5$, coupled by a Clayton copula with parameter 1.5.

Gradient method		$s = 0.01$		$s = 1$		$s = 100$		$s = 10,000$	
P_6 (ref. value)		0.00175543908912254		0.150624115746814		0.986507366924387		0.999886742040548	
		$P_n - P_6$	R_n	$P_n - P_6$	R_n	$P_n - P_6$	R_n	$P_n - P_6$	R_n
$n = 1$	(8.7e-05 s)	− 4.25e-04	6.67e-01	− 3.67e-02	6.75e-01	− 3.89e-02	7.68e-01	− 4.12e-04	8.13e-01
$n = 3$	(3.2e-01 s)	− 3.90e-05	2.96e-01	− 4.00e-03	3.04e-01	− 4.95e-03	4.80e-01	− 4.47e-05	7.14e-01
$n = 4$	(5.0e-01 s)	− 7.54e-06	1.98e-01	− 9.72e-04	2.03e-01	− 2.13e-03	3.74e-01	− 1.98e-05	6.69e-01
$n = 5$	(1.1e+01 s)	− 9.41e-07	1.32e-01	− 1.83e-04	1.36e-01	− 7.17e-04	2.88e-01	− 7.17e-06	6.10e-01
Bisection method		$s = 0.01$		$s = 1$		$s = 100$		$s = 10,000$	
P_7 (ref. value)		0.00180950633990575		0.155043475316759		0.987950687275204		0.999898149775137	
		$P_n - P_7$	R_n	$P_n - P_7$	R_n	$P_n - P_7$	R_n	$P_n - P_7$	R_n
$n = 1$	(9.0e-05 s)	2.06e-04	1.53e+00	1.69e-02	1.68e+00	− 1.03e-02	2.71e+00	− 1.00e-04	2.98e+00
$n = 3$	(5.2e-03 s)	4.20e-04	5.37e-01	3.58e-02	6.17e-01	− 8.11e-04	1.11e+00	− 1.36e-05	1.30e+00
$n = 5$	(4.7e-01 s)	1.41e-04	1.58e-01	1.43e-02	1.68e-01	2.08e-04	3.36e-01	− 2.46e-06	5.31e-01
$n = 6$	(7.2e+00 s)	5.06e-05	7.87e-02	4.75e-03	8.42e-02	2.11e-04	1.81e-01	− 8.06e-07	3.51e-01

Note: For the function $\varphi(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 + x_5 + ((x_1 + 1)(x_2 + 1))^{1/2} + ((x_2 + 1)(x_3 + 1)(x_4 + 1))^{1/3} + ((x_1 + 1)(x_3 + 1)(x_4 + 1)(x_5 + 1))^{1/4}$, we give the difference from the reference value P_6 under the gradient method and from P_7 under the bisection method.

Given M points $\mathbf{x}_1, \ldots, \mathbf{x}_M$ in $\mathcal{S}_1^1$, it is possible to approximate $V_H[\mathcal{S}_1^1]$ by the average of the density function v_H of H evaluated at those points, i.e.

$$V_H[\mathcal{S}_1^1] = \int_{\mathcal{S}_1^1} v_H(\mathbf{x})\mathrm{d}\mathbf{x} \approx V(\mathcal{S}_1^1)\frac{1}{M}\sum_{i=1}^{M} v_H(\mathbf{x}_i),$$

where $V(\mathcal{S}_1^1)$ is the Lebesgue measure of $\mathcal{S}_1^1$. If $\mathbf{x}_i$'s are chosen to be (pseudo)randomly distributed, this is the *Monte Carlo* (MC) method. If $\mathbf{x}_i$'s are generated from the so-called low discrepancy sequence (see [4]), this is the *quasi-Monte Carlo* (QMC) method. The main features of (Q)MC methods (their convergence rates included) do not depend on the function φ. We refer to [1] for references and a more detailed discussion on both methods.

Unfortunately, we were not able to find a convergence rate for the sequence P_n, which would be necessary to compare GAEP to (Q)MC methods. However, it is possible to calculate bounds on convergence rates for D_n, which, assuming that the random variable $\varphi(\mathbf{X})$ has a density near s and has the same asymptotic behaviour of P_n. Indeed, because of Lemma 5.3, we have that

$$|P_n - V_H[\mathcal{S}_1^1]| \leq \mathbb{P}[s - D_n < \varphi(\mathbf{X}) \leq s + D_n] = O(D_n).$$

The total number $M(n)$ of evaluations of the joint distribution H performed by GAEP after the nth iteration (as well as the computation time used) is proportional to the number of simplexes needed to calculate P_n. Since the number $\mathbf{I}_n$ of quasi-simplexes passed to the nth iteration is bounded by $N = 2^d$, we have that

$$M(n) \leq BN^n,$$

where B is a constant depending only on the dimension d.

From (35), we know that $D_n = O\big(((d-1)/(d+1))^n\big)$ is the best convergence rate attainable with GAEP, when the gradient method is used. Analogously, from Theorem 6.1, we know that $D_n = O(2^{-n})$ for the bisection method. By expressing the convergence rates for D_n in terms of $M(n)$, we find that

$$D_n = O(M(n)^{\kappa}), \qquad \kappa = \begin{cases} -\frac{1}{d\ln(2)}\ln\left(\frac{d+1}{d-1}\right), & \text{for the gradient method,} \\ -1/d, & \text{for the bisection method.} \end{cases} \tag{37}$$

Since, in general, we do not know the exact number of simplexes passed to the next iteration by GAEP, the rates provided by (37) represent only an upper bound on the real convergence rates of the GAEP algorithm. As a matter of fact, the convergence rates encountered in many numerical examples turned out to be much better than those predicted by (37).

In Figure 6, we plot absolute errors $|P_n - V_H[\mathcal{S}_1^1]|$ vs. computation time, for random vectors with independent marginals and functions φ for which $V_H[\mathcal{S}_1^1]$ is available analytically. We use linear least squares fitting on these curves in order to calculate the so-called *empirical* convergence rates for the algorithm. Here, computation time (which is proportional to $M(n)$) is used as a measure of numerical complexity. These results are collected and presented in Table 5, where the empirical convergence rates obtained from Figure 6, as well as the bounds obtained from (37), are compared with convergence rates for the MC and QMC methods.

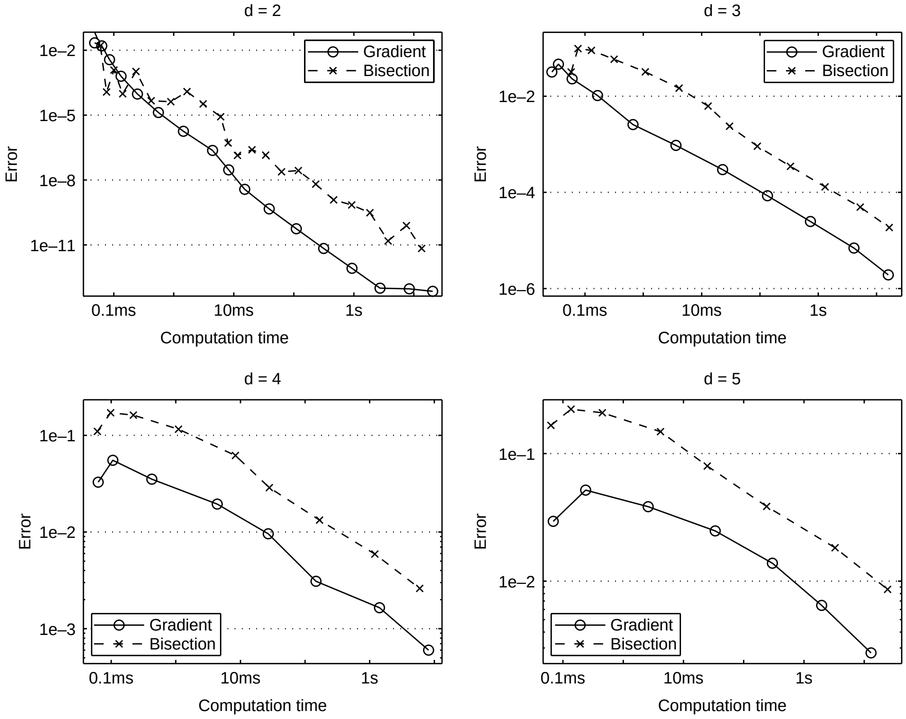

Figure 6. Errors $|P_n - V_H[\mathcal{S}(\mathbf{0}, 3)]|$ from the GAEP algorithm for random vectors of different dimensions having independent Pareto marginals with tail indexes $\theta_i = i, i = 2, \ldots, 5$. GAEP errors are plotted vs. computation time, for the function $\varphi(\mathbf{x}) = \prod_{k=1}^{d}(x_k + 1)$.

From Table 5, it is clear that the gradient method is to be preferred for 2D vectors. This is consistent with the results illustrated in Table 1. In higher dimensions, the situation is not so clear. Upper bounds on the convergence rate suggest that the bisection method, in dimension $d = 4, 5$, is slightly more competitive than the gradient, as confirmed by the results in Tables 3 and 4. However, Table 5 (and the corresponding empirical rates) indicates that the gradient rule can be better, computationally, also in higher dimensions. Here, it is important to remark that exact convergence rates for P_n are *not* available, and (empirical) convergence rates for the two methods depend on the probability model under study.

With respect to (Q)MC methods, the figures indicate that a *well-designed* QMC algorithm will perform better, asymptotically, than GAEP under a smooth probability model and for dimensions $d \geq 3$. At this point, it is, however, important to stress that GAEP and (Q)MC methods are substantially different. First of all, (Q)MC methods provide a final estimate which contains a source of *randomness*, while the GAEP algorithm, being solely based on geometric properties of a certain domain, is purely *deterministic*. This can be seen from the data in Table 6 in which we compare GAEP and MC estimates using two examples.

Also recall that (Q)MC methods need either a density (everywhere on $\mathcal{S}_1^1$) or a sampling algorithm for the distribution function of $\mathbf{X}$. Instead of this, the GAEP algorithm does not require the density of H in analytic form, nor does it have to assume overall smoothness. In order to use GAEP, one only needs that H can be evaluated numerically

Table 5. Asymptotic convergence rates of GAEP, g (gradient), b (bisection), MC, and QMC methods. Here, we use the simplified notation $M = M(n)$. The rates calculated with least squares fitting are denoted with 'emp.'

	$d = 2$	$d = 3$	$d = 4$	$d = 5$
GAEP, g. (bound)	$M^{-0.79}$	$M^{-0.33}$	$M^{-0.18}$	$M^{-0.12}$
GAEP, g. (emp.)	$M^{-2.18}$	$M^{-0.74}$	$M^{-0.40}$	$M^{-0.26}$
GAEP, b. (bound)	$M^{-0.5}$	$M^{-0.33}$	$M^{-0.25}$	$M^{-0.2}$
GAEP, b. (emp.)	$M^{-1.62}$	$M^{-0.74}$	$M^{-0.40}$	$M^{-0.28}$
MC	$M^{-0.5}$	$M^{-0.5}$	$M^{-0.5}$	$M^{-0.5}$
QMC (best)	M^{-1}	M^{-1}	M^{-1}	M^{-1}
QMC (worst)	$M^{-1}(\log M)^2$	$M^{-1}(\log M)^3$	$M^{-1}(\log M)^4$	$M^{-1}(\log M)^5$

Note: For (Q)MC methods, M is the number of samples used.

and that $\varphi \in \mathcal{N}'$. Furthermore, (Q)MC methods need to be tailored to the specific example under study, their accuracy being generally lost when the density v_H is not smooth or when $\varphi(\mathbf{X})$ has infinite first or second moments. In these cases, relative errors increase for (Q)MC for s tending to infinity. Contrary to this, GAEP does not need any adaptation to the probabilistic model under study, nor it is influenced by the heaviness of the marginal distributions used. Moreover, in Section 11, we will present another method for the 3D case, which seems to be competitive with respect to QMC.

10. GAEP vs. AEP

The GAEP algorithm is similar to the AEP algorithm introduced by the same authors in [1] for the sum operator, but cannot be seen as an extension of AEP. In the case $\varphi(\mathbf{x}) = \sum_{k=1}^{d} x_k$, the geometrical decompositions of the set

$$\mathcal{S}_1^1 = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{0} < \mathbf{x} \text{ and } x_1 + \ldots + x_k \leq s\},$$

used by AEP and GAEP (gradient method), are equivalent for $d = 2, 3$, but different for $d \geq 4$. AEP decomposes $\mathcal{S}_1^1$ in a countable family of *overlapping* simplexes, whereas the (quasi)simplexes produced by GAEP are always *disjoint*. The decomposition used by AEP has the advantage that all the simplexes generated by the algorithm are scaled copies of the simplexes from which they have been generated. On the other hand, using overlapping simplexes implies a larger numerical complexity, particularly in high dimensions, and a much more cumbersome proof of convergence. Indeed, the problem of convergence of AEP in dimensions $d \geq 9$ is still open.

This is different with GAEP, which uses disjoint simplexes, and is more efficient than AEP for $\varphi(\mathbf{x}) = \sum_{k=1}^{d} x_k$ in dimensions $d \geq 4$. Moreover, convergence (under some smoothness of φ) is easily stated in arbitrary dimensions. Unfortunately, the disjoint (quasi)simplexes produced by GAEP can be 'cut off' at the edges. This does not allow the use of the extrapolation technique described in [1], (Section 5). This is the reason why the AEP algorithm, in its extrapolated version (AEP-E), turns out to be better than GAEP for the sum operator, in all dimensions.

In Figure 7, we plot the error committed by AEP, AEP-E (see (43) below) and GAEP (gradient method) vs. computation time, when the three algorithms are applied to random vectors of different dimensions for the function $\varphi(\mathbf{x}) = \sum_{i=1}^{d} x_i$. In these examples, the value $V_H[\mathcal{S}_1^1]$ is available analytically. Again, note that AEP and GAEP are equivalent in dimensions $d = 2$ and $d = 3$. As already remarked, GAEP is more efficient than standard

Table 6. GAEP (bisection) and MC estimates for $V_H[S_1^1]$ for the example described in (a) Table 2 and (b) Table 3.

(a)	GAEP estimate	MC estimate	MC SE
s	($n = 12$, 61 s)	($M =$ 5e07, 64 s)	
10^0	0.416123502687784	0.41614132	6.97e-05
10^2	0.860937621800580	0.86099074	4.89e-05
10^4	0.975466112029285	0.97545864	2.19e-05
10^6	0.996034158165874	0.99603776	8.88e-06

(b)	GAEP estimate	MC estimate	MC SE
s	($n = 8$, 41 s)	($M =$ 4e07, 47 s)	
10^{-2}	0.000729352448762	0.000715050	4.22e-06
10^0	0.174626428280207	0.172616075	5.98e-05
10^2	0.989734882116519	0.989638175	1.60e-05
10^4	0.999899605726889	0.999899550	1.58e-06

AEP in dimensions $d \geq 4$ due to the use of a disjoint decomposition. However, AEP-E turns out to be the best algorithm to be used with the sum operator, in all dimensions d.

In order to better clarify the difference between AEP and GAEP, we want to decompose the simplex $\{\mathbf{x} \in \mathbb{R}^2 : \mathbf{0} \leq \mathbf{x} \leq \mathbf{p}, \varphi(\mathbf{x}) \leq s\} \subset \mathbb{R}^2$, where $\mathbf{0} \leq \mathbf{h} \leq \mathbf{p}$ and $\varphi(\mathbf{h}) < s$. In the case of a general function $\varphi \in \mathcal{N}$, an overlapping decomposition, analogous to the one used by AEP, can be carried out, leading to

$$\begin{aligned} V_H[\{x : \mathbf{0} \leq \mathbf{x} \leq \mathbf{p}, \varphi(\mathbf{x}) \leq s\}] =& V_H[\mathcal{Q}(\mathbf{0}, \mathbf{h})] \\ &+ V_H[\{\mathbf{x} : (0, h_2) \leq \mathbf{x} \leq \mathbf{p}, \varphi(\mathbf{x}) \leq s\}] \\ &+ V_H[\{\mathbf{x} : (h_1, 0) \leq \mathbf{x} \leq \mathbf{p}, \varphi(\mathbf{x}) \leq s\}] \\ &- V_H[\{\mathbf{x} : \mathbf{h} \leq \mathbf{x} \leq \mathbf{p}, \varphi(\mathbf{x}) \leq s\}]. \end{aligned} \tag{38}$$

For the same simplex, the disjoint decomposition used by GAEP gives, instead,

$$\begin{aligned} V_H[\{\mathbf{x} : \mathbf{0} \leq \mathbf{x} \leq \mathbf{p}, \varphi(\mathbf{x}) \leq s\}] =& V_H[\mathcal{Q}(\mathbf{0}, \mathbf{h})] \\ &+ V_H[\{\mathbf{x} : (0, h_2) \leq \mathbf{x} \leq (h_1, p_2), \varphi(\mathbf{x}) \leq s\}] \\ &+ V_H[\{\mathbf{x} : (h_1, 0) \leq \mathbf{x} \leq (p_2, h_1), \varphi(\mathbf{x}) \leq s\}] \\ &+ V_H[\{\mathbf{x} : \mathbf{h} \leq \mathbf{x} \leq \mathbf{p}, \varphi(\mathbf{x}) \leq s\}]. \end{aligned} \tag{39}$$

The two decompositions are illustrated in Figure 8.

In conclusion, the AEP-E algorithm is to be used with sum operator, while, for a general function φ, GAEP will perform better. The problem of finding a technique analogous to extrapolation, and working with a general functional φ, will be addressed in future research.

11. An alternative choice of $\mathbf{h}_n^t$ for $d = 3$

The (speed of) convergence of GAEP heavily depends on the choice of $\{\mathbf{h}_n^t, t \in \mathbf{I}_n\}$ at each iteration of the algorithm. In this paper, we have presented two different ways of choosing this sequence: the bisection and the gradient methods. For both these methods, we have

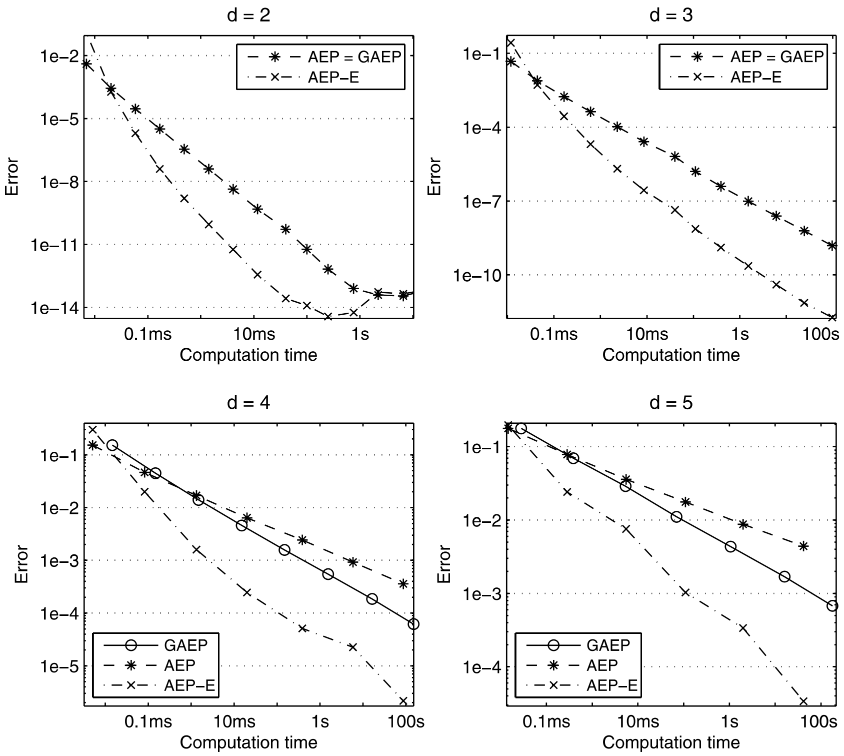

Figure 7. Errors $|P_n - V_H[\mathcal{S}(\mathbf{0}, 10)]|$ from the AEP, AEP-E and GAEP (gradient) algorithms for random vectors of different dimensions having independent Gamma marginals with scale parameter 1 and shape parameters $0.5 + 0.5i$, $\; i = 1, \ldots, 5$. GAEP errors are plotted vs. computation time for the sum operator $\varphi(\mathbf{x}) = \sum_{i=1}^{d} x_i$.

derived their mathematical properties, including convergence rates. Of course, we are aware that a different choice of the sequence $\mathbf{h}_n^t$ may provide better results on a particular model. As an example of a possible improvement of GAEP, we present an alternative approach, which, in numerical examples, yields extraordinarily fast convergence for 3D vectors. Thus, throughout this section, fix $d = 3$ and, for all $n \in \mathbb{N}$ and $t \in \mathbf{I}_n$, let the sequence $\mathbf{h}_n^t$ be defined as

$$\mathbf{h}_n^t = \begin{cases} \mathbf{p}_n^t, & \text{if } \mathcal{S}_n^t = \mathcal{Q}(\mathbf{b}_n^t, \mathbf{p}_n^t), \\ \Phi(\mathbf{b}_n^t), & \text{otherwise,} \end{cases} \tag{40}$$

where $\Phi : \mathbb{R}^3 \to \mathbb{R}^3_- \cup \mathbb{R}^3_+$ is the function that maps $\mathbf{b} \in \mathbb{R}^3$ to the unique vector $\mathbf{h} \in \mathbb{R}^3_- \cup \mathbb{R}^3_+$ satisfying

$$\varphi\big(\mathbf{b} + \mathbf{h} \circ (1, 1, 0)\big) = \varphi\big(\mathbf{b} + \mathbf{h} \circ (1, 0, 1)\big) = \varphi\big(\mathbf{b} + \mathbf{h} \circ (0, 1, 1)\big) = s. \tag{41}$$

Note that, for a fixed $\mathbf{b}$, existence and uniqueness of a vector $\mathbf{h} \in \mathbb{R}^3_- \cup \mathbb{R}^3_+$ satisfying (42) follow from the definition of the set $\mathcal{N}$. Moreover, we also have $\mathbf{h}_n^t \in \mathcal{H}(\mathbf{p}_n^t)$, for all $n \in \mathbb{N}$ and $t \in \mathbf{I}_n$. Hence, Φ is well defined.

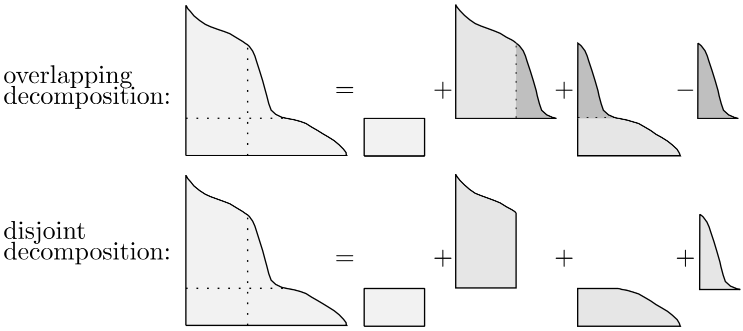

Figure 8. An illustration of the two decompositions (38) and (39).

LEMMA 11.1. Let $d = 3$ and assume that $\varphi \in \mathcal{N}$. Then, the sequence $\mathbf{h}_n^t$, as defined by (40) and (41), satisfies $\mathbf{h}_n^t \in \mathcal{H}(\mathbf{p}_n^t)$ for all $n \in \mathbb{N}$ and $t \in \mathbf{I}_n$.

Proof. Using induction on n, it is possible to show that, for all $n \in \mathbb{N}$, we have

$$\varphi(\mathbf{b}_n^t + \mathbf{p}_n^t \circ \mathbf{e}_k) = s \quad \text{for all} \quad t \in \mathbf{I}_n \quad \text{and} \quad k = 1, 2, 3. \tag{42}$$

Equation (42) is illustrated in Figure 9. Recalling that $\mathbf{b}_n^t = \mathbf{0}$, (42) follows, for $n = 1$, from the definition (4) of $\mathbf{p}_1^1$. Now assume that (42) is true for some n and choose an arbitrary $t \in \mathbf{I}_n$. First of all, it is easy to see, using (18) and (19), that the quasi-simplexes $\mathcal{S}_{n+1}^{8t-8+j}$, $j = 5, \ldots, 8$, are empty. Therefore, it is sufficient to show that

$$\varphi(\mathbf{b}_{n+1}^{Nt-N+j} + \mathbf{p}_{n+1}^{Nt-N+j} \circ \mathbf{e}_k) = s \quad \text{for} \quad j = 1, \ldots, 4 \quad \text{and} \quad k = 1, 2, 3.$$

For $j = 1$ (the proofs in the other cases are analogous), we have that $\varphi(\mathbf{b}_{n+1}^{Nt-N+1} + \mathbf{p}_{n+1}^{Nt-N+1} \circ \mathbf{e}_k) = \varphi(\mathbf{b}_n^t + \mathbf{h}_n^t - \mathbf{h}_n^t \circ \mathbf{e}_k)$ and

$$\varphi(\mathbf{b}_n^t + \mathbf{h}_n^t - \mathbf{h}_n^t \circ \mathbf{e}_k) = \begin{cases} \varphi(\mathbf{b}_n^t + \mathbf{h}_n^t \circ (0, 1, 1)), & \text{if} \ k = 1, \\ \varphi(\mathbf{b}_n^t + \mathbf{h}_n^t \circ (1, 0, 1)), & \text{if} \ k = 2, \\ \varphi(\mathbf{b}_n^t + \mathbf{h}_n^t \circ (1, 1, 0)), & \text{if} \ k = 3. \end{cases}$$

The induction step then follows from (41).

Suppose now that, for a certain n and $t \in \mathbf{I}_n$, (42) is satisfied with $\mathbf{p}_n^t \geq \mathbf{0}$. Since φ is strictly increasing, for any vector $\mathbf{x}$ with at least one coordinate, say the first, larger than $\mathbf{p}_n^t$, we have that

$$\varphi(\mathbf{b}_n^t + \mathbf{x} \circ (1, 1, 0)) \geq \varphi(\mathbf{b}_n^t + \mathbf{x} \circ (1, 0, 0)) > \varphi(\mathbf{b}_n^t + \mathbf{p}_n^t \circ (1, 0, 0)) = s.$$

Since, by (40), $\varphi(\mathbf{b}_n^t + \mathbf{h}_n^t \circ (1, 1, 0)) = s$, we must have that $\mathbf{h}_n^t \leq \mathbf{p}_n^t$. Analogously, if (42) is satisfied with $\mathbf{p}_n^t \leq \mathbf{0}$, we get $\mathbf{h}_n^t \geq \mathbf{p}_n^t$ and, finally, $\mathbf{h}_n^t \in \mathcal{H}(\mathbf{p}_n^t)$. □

Choosing the sequence $\mathbf{h}_n^t$ according to (40) has several advantages. First of all, it reduces numerical complexity of GAEP, since (see the proof of Lemma 11.1)

$$\mathcal{S}_{n+1}^{8t-8+5} = \mathcal{S}_{n+1}^{8t-8+6} = \mathcal{S}_{n+1}^{8t-8+7} = \mathcal{S}_{n+1}^{8t-8+8} = \emptyset \quad \text{for all} \quad n \in \mathbb{N} \quad \text{and} \quad t \in \mathbf{I}_n.$$

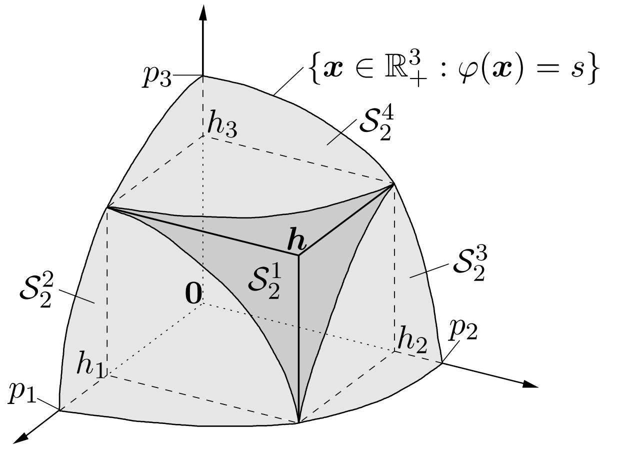

Figure 9. Decomposition (12) of the 3D simplex $\mathcal{S}_1^1 = \mathcal{S}(\mathbf{0}, \mathbf{p}_1^1)$ for the $\mathbf{h}_1^1$ satisfying (41).

As illustrated in Figure 9, the maximum number of new simplexes generated by the decomposition (12) is then reduced from 8 to 4, as, for some $n \in \mathbb{N}$ and $t \in \mathbf{I}_n$, we have

$$V_H\big[\mathcal{S}_n^t\big] = V_H\big[\mathcal{Q}_n^t\big] - V_H\big[\mathcal{S}_{n+1}^{8t-8+1}\big] + V_H\big[\mathcal{S}_{n+1}^{8t-8+2}\big] + V_H\big[\mathcal{S}_{n+1}^{8t-8+3}\big] + V_H\big[\mathcal{S}_{n+1}^{8t-8+4}\big].$$

On the other hand, solving (41) requires numerical root finding algorithms, which can be time-consuming depending on the function φ. However, for a polynomial φ, the use of Newton's method combined with an initial guess obtained from (29) proved to be extremely fast. Indeed, it is possible to show that, when $\mathbf{b}_n^t$ approaches the curve $\{\mathbf{x} : \varphi(\mathbf{x}) = s\}$, the sequence $\mathbf{h}_n^t$, as defined in (40), goes to the sequence (29) characterizing the gradient method.

In [1], the authors introduce two sequences of estimators for $V_H[\mathcal{S}_1^1]$ in the case of the sum operator $\varphi(\mathbf{x}) = \sum_{k=1}^d x_k$. The first sequence, P_n, is the standard AEP estimator. The second sequence, the so-called AEP-E(xtrapolated) estimator P_n^*, is defined as

$$P_n^* = P_{n-1} + \frac{(d+1)^d}{d!2^d} \sum_{t=1}^{N^{n-1}} \tau_n^t V_H\big[\mathcal{Q}_n^t\big], \tag{43}$$

converges significantly faster to $V_H[\mathcal{S}_1^1]$ than P_n. The idea behind AEP-E is that the simplexes generated by AEP become smaller and smaller at each iteration, so that a smooth probability distribution H can be approximated by its Taylor expansion.

Using the same philosophy, (43) can be used with the GAEP estimator in the case $d = 3$. Thus, we define the GAEP-E estimator as

$$P_n^* = P_{n-1} + 4/3 \sum_{t=1}^{I_n} \tau_n^t V_H[\mathcal{Q}_n^t],$$

where P_n is now the sequence defined in (20) and the sequence $\mathbf{h}_n^t$ as defined in (41). Unfortunately, we were not able to find theoretical results for the GAEP-E estimator for a general function φ. However, in numerical examples, we found P_n^* to be significantly more accurate and faster than P_n. Figure 10 gives an illustration of this improvement on a specific example, where the value $V_H[\mathcal{S}_1^1]$ is available analytically. Empirical

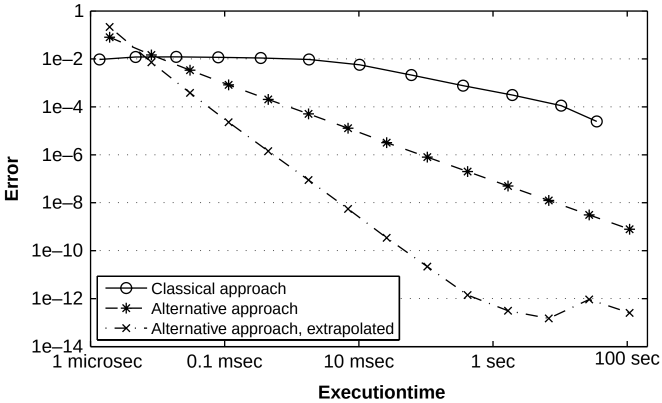

Figure 10. Errors $|P_n - V_H[\mathcal{S}(\mathbf{0}, 100)]|$ from the GAEP (gradient method and alternative approach as defined in (40)) and GAEP-E algorithm for a random vector having $d = 3$ independent Pareto marginals with tail indexes $\theta_i = i, \quad i = 1, 2, 3$. GAEP errors are plotted vs. computation time, for the function $\varphi(\mathbf{x}) = \prod_{k=1}^{3} (x_k + 1)$.

convergence rates $\left|P_n^* - V_H[\mathcal{S}_1^1]\right| = O(M(n)^\kappa)$ found in examples for the GAEP-E vary from $\kappa = -2$ to -1.6, thus making GAEP-E more than competitive with respect to QMC methods.

Unfortunately, we were not able to find the improvements in accuracy provided by the GAEP-E estimator P_n^* in any other dimension than $d = 3$.

12. Final remarks

In this paper, we have introduced the GAEP algorithm in order to numerically compute the distribution function of a function $\varphi(\mathbf{X})$ of a d-dimensional random vector $\mathbf{X}$ with given joint distribution function H. The algorithm is mainly based on two assumptions: the marginal components of $\mathbf{X}$ have to bounded from below, and φ has to be strictly increasing in each coordinate. If the vector $\mathbf{X}$ is absolutely continuous, and φ is also twice differentiable, the convergence of the algorithm is guaranteed in arbitrary dimensions, even if its numerical complexity limits any application to dimensions $d \leq 5$.

The convergence (rate) of GAEP depends heavily on the method adopted to define the sequence P_n of estimators converging to the value $\mathbb{P}[\varphi(\mathbf{X}) \leq s]$. In this paper, we have proposed three different methods, which are (very) effective for different dimensions d: the gradient (for $d = 2$), the bisection ($d \geq 4$), and the GAEP-extrapolated rule ($d = 3$). Summarizing all the results presented in the paper, we can say that the GAEP algorithm is better than its competitors, mainly MC and QMC methods, for dimensions $d \leq 3$. The GAEP algorithm behaves slightly worse in dimension $d = 4, 5$. We remark again that, contrary to (Q)MC-methods, the GAEP algorithm is deterministic and does not need any adaptation to the probabilistic model under study.

An improvement of the GAEP convergence rates in dimensions $d \geq 4$ needs an extension of the extrapolation technique introduced in [1] to general aggregating functionals φ. Alternative methods for the choice of the vectors $\{\mathbf{h}_n^t, t \in \mathbf{I}_n\}$, more efficient than those described in this paper, may also be possible. We propose to address these problems in future research.

Acknowledgements

The authors thank Guus Balkema for some useful comments on a previous version of this paper. Philipp Arbenz, as SCOR Fellow, thanks SCOR Switzerland for the financial support. Paul Embrechts, as Senior SFI Professor, acknowledges the support from the Swiss Finance Institute. Giovanni Puccetti would like to thank RiskLab and the Forschungsinstitut für Mathematik (FIM) of the Department of Mathematics, ETH Zurich, for the financial support.

References

[1] P. Arbenz, P. Embrechts, and G. Puccetti, *The AEP algorithm for the fast computation of the distribution of the sum of dependent random variables*, preprint (2011), forthcoming in Bernoulli.

[2] P. Mattila, *Geometry of Sets and Measures in Euclidean Spaces: Fractals and Rectifiability*, Cambridge University Press, Cambridge, 1995.

[3] R.B. Nelsen, *An Introduction to Copulas*, 2nd ed., Springer, New York, 2006.

[4] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, Society for Industrial Mathematics, Philadelphia, PA, 1992.