

# A $\sqrt{5}/2$ -approximation algorithm for optimal piecewise linear approximations of bounded variable products

ANDREAS BÄRMANN, ROBERT BURLACU,  
LUKAS HAGER, KATJA KUTZER

ABSTRACT. We investigate the optimal piecewise linear approximation of the bivariate product  $xy$  over rectangular domains. More precisely, our aim is to minimize the number of simplices in the triangulation underlying the approximation, while respecting a prescribed approximation error. First, we show how to construct optimal triangulations consisting of up to five simplices. Using these as building blocks, we construct a triangulation scheme called *crossing swords* that requires at most  $\sqrt{5}/2$ -times the number of simplices in any optimal triangulation. In other words, we derive an approximation algorithm for the optimal triangulation problem. We also show that crossing swords yields optimal triangulations in the case that each simplex has at least one axis-parallel edge. Furthermore, we present approximation guarantees for other well-known triangulation schemes, namely for the red refinement and longest-edge bisection strategies as well as for a generalized version of K1-triangulations. Thereby, we are able to show that our novel approach dominates previous triangulation schemes from the literature, which is underlined by illustrative numerical examples.

## 1. INTRODUCTION

We consider optimal piecewise linear (pwl.) approximations of the bilinear non-convex function  $F: \mathbb{R}^2 \rightarrow \mathbb{R}, F(x, y) = xy$  over the rectangular domain  $D = [x, \bar{x}] \times [y, \bar{y}]$ . A pwl. approximation  $f: D \rightarrow \mathbb{R}$  of  $F$  is uniquely defined by its underlying triangulation  $\mathcal{T} := \{S_1, \dots, S_k\} \subseteq \mathbb{R}^2, k \in \mathbb{N}$  of the domain  $D$ , where  $f$  is linear over each simplex  $S_i$ . The approximation error between  $F$  and  $f$  is defined as the maximum pointwise deviation over  $D$ . For any prescribed approximation accuracy  $\varepsilon > 0$ , we are interested in triangulations that lead to pwl. approximations with an approximation error less than  $\varepsilon$  and that are minimal with respect to the number of simplices they contain. To the best of our knowledge, finding triangulations which are optimal in this sense is still an open problem. Furthermore, there have been no attempts so far to prove approximation guarantees for existing triangulation schemes in the literature either.

**Contribution.** In this paper, we make several important steps towards efficient triangulation schemes for the pwl. approximation of bilinear functions. At first, we construct triangulations that provably minimize the approximation error with up to five simplices. We show that these triangulations are in turn optimal for approximation accuracies up to  $(\sqrt{5}-2)/4 \approx 0.059$  over the unit box. Using these basic triangulations as building blocks, we derive an approximation algorithm for the optimal triangulation problem which we call *crossing swords*, based on

---

*Date:* January 30, 2023.

*2010 Mathematics Subject Classification.* 90C20, 90C59, 41A05, 15A63, 90C11.

*Key words and phrases.* Nonconvex Quadratic Programming, Piecewise Linear Approximations, Triangulations, Approximation Algorithm.

its underlying geometric idea. We prove that this scheme can be used to create triangulations consisting of at most  $\sqrt{5}/2$ -times the number of simplices in an optimal triangulation. To be more precise, the crossing swords algorithm is an  $\sqrt{5}/2$ -approximation algorithm for all  $\varepsilon_i = 1/16^i$  with  $i \in \mathbb{N}$ . For any intermediate value, the approximation guarantee is  $\sqrt{5}/2 + 4\sqrt{5}\varepsilon$ . In this sense, the crossing swords scheme is an asymptotic  $\sqrt{5}/2$ -approximation algorithm for small prescribed accuracies. Furthermore, we prove that crossing swords produces optimal triangulations for all  $\varepsilon_i = 1/16^i$  with  $i \in \mathbb{N}$  if we require each simplex to have one axis-parallel edge. We will show that crossing swords triangulations are superior to the most-widely used triangulation schemes in the literature, i.e. K1, J1 ([20]), longest-edge bisection ([10, 13]), maximum error bisection ([12, 19]) and red refinement ([4, 8]). Although, we show that longest-edge bisection is also a  $\sqrt{5}/2$ -approximation algorithm, it produces more simplices than crossing swords expect for certain specific values of  $\varepsilon$ , where the two are equivalent. Furthermore, we introduce a generalized version of the K1-triangulation scheme and show that it is a  $\sqrt{5}$ -approximation algorithm. This also applies to the red refinement scheme. Since the approximation accuracies in crossing swords triangulations can be adjusted more finely, our scheme is by a factor of two better than previous methods in many cases. The overall dominance of the crossing swords scheme is underlined by numerical results for an indicative example.

**Literature Review.** Pwl. approximations of bilinear functions are studied in the literature in various contexts. On the one hand, such approximations are important in many applications. For example, in computer visualization, non-linear surfaces are usually represented by pwl. functions based on triangulations. This also applies to non-linear shapes in architectural design, which in some cases can be realized in practice only via pwl. approximations, cf. [18]. On the other hand, several very efficient algorithmic frameworks for solving quadratic optimization problems rely on pwl. approximations ([1, 6, 8, 12, 13]). However, finding optimal pwl. approximations of  $xy$  over rectangular domains is still an unsolved problem. The only attempt to find provably minimal triangulations is represented by [14]. Here, the author develops a mixed-integer quadratic program which models the optimal triangulation problem. However, due to its size and inherent complexity, the presented model cannot even be solved for trivial instances by state-of-the-art solvers. Besides that, there exist several heuristic triangulation approaches in the literature. In [22], the author uses regular uniform triangulations, namely J1- and K1-triangulations, which were first presented in [20]. In [11] and [19], the authors go a step further and develop triangulations which are specifically designed for the approximation of  $xy$  over rectangular domains. In [9, 12, 13], triangulations are constructed adaptively, using red, maximum error or longest-edge refinements. For all mentioned triangulation schemes, we derive relations between the approximation accuracy and the resulting number of simplices. Most importantly, we introduce a new, more efficient triangulation scheme for which we can prove that it dominates the previously known methods.

On a more general basis, the authors of [18] are interested in optimal triangulations of  $xy$  over the whole plane  $\mathbb{R}^2$ . They show how to construct so called *optimal simplices*, i.e. triangles which fulfil a prescribed approximation accuracy while maximizing their area. They prove that  $\mathbb{R}^2$  can be tiled with optimal simplices only and thus obtain an optimal triangulation. A natural idea would be to use optimal simplices to triangulate rectangular domains as well. However, due to their geometry, it is not possible to use solely optimal simplices to triangulate rectangular

regions. Instead, we will use this idea to derive a lower bound on minimal triangulations over any given compact domain. This bound is essential in proving the above-mentioned approximation guarantees for the different triangulations schemes.

**Structure.** Our work is structured as follows. In Section 2, we introduce the general notation and concepts that are used throughout the article. After that, in Section 3 we focus on optimal triangulations for the pwl. approximation of  $xy$  over rectangular domains. We show that the problem can be reduced from general rectangular domains to the unit box. In Section 3.1, we construct optimal triangulations consisting of up to five simplices. We use them as building blocks to develop our  $\sqrt{5}/2$ -approximation algorithm called *crossing swords* in Section 3.2. It even yields optimal triangulations if we require that each simplex has one axis-parallel edge. In addition, we also prove approximation guarantees for a generalized version of K1-triangulations, the red refinement and the longest-edge bisection scheme. In Section 3.4, we prove the dominance of the crossing swords triangulation compared to the known triangulations from the literature and provide numerical results on an exemplary instance that underline the theoretical results. Finally, we conclude in Section 4.

## 2. PIECEWISE LINEAR FUNCTIONS AND APPROXIMATIONS

We start by introducing the basic concepts of triangulations and piecewise linear functions and discuss their use in approximating non-linear functions. For the sake of simplicity, we restrict ourselves to continuous functions over polytopal domains.

A function is called *piecewise linear (pwl.)* if it is linear over each element of a given domain partition. To partition a domain, it is possible to use any family of polytopes. However, in practice most often triangulations are used, see e.g. [21]. This is without loss of generality, since a pwl. function defined with respect to a partition by polytopes can always be represented by a pwl. function over a triangulation, namely by triangulating each polytope. Therefore, we define pwl. functions over triangulations. In the following, we formally introduce the relevant definitions in this context. Throughout this work, we use the notation  $V(P)$  for the vertex set and  $A(P)$  for the area of a polytope  $P \subset \mathbb{R}^d$ .

**Definition 2.1.** A  $k$ -simplex  $S$  is the convex hull of  $k + 1$  affinely independent points in  $\mathbb{R}^d$ . We call  $S$  a full-dimensional simplex if  $k = d$  holds.

A triangulation is a partition consisting of full-dimensional simplices.

**Definition 2.2.** A finite set of full-dimensional simplices  $\mathcal{T} := \{S_1, \dots, S_k\} \subseteq \mathbb{R}^d$ ,  $k \in \mathbb{N}$ , is called a triangulation of a polytope  $P \subseteq \mathbb{R}^d$  if  $P = \cup_{i=1}^k S_i$  and the intersection  $S_i \cap S_j$  of any two simplices  $S_i, S_j \in \mathcal{T}$  is a proper face of  $S_i$  and  $S_j$ . Further, we denote the nodes  $N(\mathcal{T})$  of the triangulation  $\mathcal{T}$  as  $N(\mathcal{T}) := \cup_{i=1}^k V(S_i)$ .

We use the concept of triangulations to define pwl. functions.

**Definition 2.3.** Let  $P \subset \mathbb{R}^d$  be a polytope, and let  $\mathcal{T} := \{S_1, \dots, S_k\}$ ,  $k \in \mathbb{N}$ , be a triangulation of  $P$ . A continuous function  $g: P \rightarrow \mathbb{R}$  is called piecewise linear if there exist vectors  $m_i \in \mathbb{R}^d$  and constants  $c_i \in \mathbb{R}$  for  $i = 1, \dots, k$  such that

$$g(x) = m_i^\top x + c_i \quad \text{if } x \in S_i.$$

Pwl. functions can be used to approximate non-linear functions. We consider the case where the pwl. approximation coincides with the non-linear function at the nodes of the triangulations. In this way, we can ensure continuity of the approximation if the non-linear function is continuous.

**Definition 2.4.** Let  $P \subset \mathbb{R}^d$  be a polytope, and let  $\mathcal{T} := \{S_1, \dots, S_k\}$ ,  $k \in \mathbb{N}$ , be a triangulation of  $B$ . We call a pwl. function  $g: P \rightarrow \mathbb{R}$  a pwl. approximation of a continuous function  $G: P \rightarrow \mathbb{R}$  if  $g(x) = G(x)$  holds for all  $x \in N(\mathcal{T})$ .

Usually, the error of a pwl. approximation is measured by the maximum absolute pointwise deviation between the pwl. approximation itself and the non-linear function to be approximated; see e.g. [11, 12, 17, 22]. In the following, we also use this definition of the *approximation error*.

**Definition 2.5.** Consider a triangulation  $\mathcal{T}$  of a polytope  $P \subset \mathbb{R}^d$  and let  $g: P \rightarrow \mathbb{R}$  be a pwl. approximation of a function  $G: P \rightarrow \mathbb{R}$  w.r.t. to  $\mathcal{T}$ . We call

$$E_{g,G}: P \rightarrow \mathbb{R}, \quad E_{g,G}(x) := g(x) - G(x)$$

the error function w.r.t.  $g$  and  $G$  and

$$\varepsilon_{g,G}(S) := \max_{x \in S} |E_{g,G}(x)|$$

the approximation error on a simplex  $S \in \mathcal{T}$ . Consequently, we define the approximation error of  $g$  (or, equivalently, of  $\mathcal{T}$ ) w.r.t.  $G$  over the domain  $B$  as

$$\varepsilon_{g,G}(\mathcal{T}) := \max_{S \in \mathcal{T}} \varepsilon_{g,G}(S).$$

Given some  $\varepsilon > 0$ , we call  $g$  an  $\varepsilon$ -approximation and  $\mathcal{T}$  an  $\varepsilon$ -triangulation if the approximation error is smaller than or equal to  $\varepsilon$ .

Finally, we formulate the concept of optimal triangulations for pwl. approximations of non-linear functions; see also [14].

**Definition 2.6.** Let  $P \subseteq \mathbb{R}^d$  be a polytope and  $g: P \rightarrow \mathbb{R}$  a pwl.  $\varepsilon$ -approximation of  $G: P \rightarrow \mathbb{R}$  with respect to the triangulation  $\mathcal{T}$ . We call  $\mathcal{T}$   $\varepsilon$ -optimal if  $|\mathcal{T}|$  is minimal among all  $\varepsilon$ -triangulations.

It is not obvious how to determine  $\varepsilon$ -optimal triangulations in general. We tackle this problem for bilinear functions in the following.

### 3. TRIANGULATIONS FOR THE APPROXIMATION OF XY OVER BOX DOMAINS

In this section, we focus on finding optimal triangulations for pwl. approximations of the bilinear function

$$F: \mathbb{R}^2 \rightarrow \mathbb{R}, F(x, y) = xy$$

over the rectangular domain  $D = [\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}]$ , i.e. we treat the following problem:

**Problem 1.** Given some  $\varepsilon > 0$ , find an  $\varepsilon$ -optimal triangulation of  $D$  w.r.t.  $F$ .

Our main contribution will be the derivation of a novel approximation algorithm for Problem (1). We begin by stating a known result from the literature for the approximation of  $F$  over a single simplex.

**Lemma 3.1** (Approximation error). [18] Given a pwl. approximation  $f: D \rightarrow \mathbb{R}$  of  $F$  defined by a triangulation  $\mathcal{T}$  of  $D$ , the approximation error  $\varepsilon_{f,F}(S)$  over a simplex  $S \in \mathcal{T}$  is attained at the centre of one of its facets. Further, if  $(x_0, y_0)$  and  $(x_1, y_1)$  are the endpoints of a facet, we have

$$\varepsilon_{f,F}(S) = \left| E_{f,F} \left( \frac{x_1 - x_0}{2}, \frac{y_1 - y_0}{2} \right) \right| = \frac{1}{4} |(x_1 - x_0)(y_1 - y_0)|. \quad (1)$$

Since the error on each simplex is attained on one of its facets, we know the following for the approximation error over  $D$ .

**Lemma 3.2.** Let  $f: D \rightarrow \mathbb{R}$  be a pwl. approximation of  $F$  defined by a triangulation  $\mathcal{T}$  of  $D$ . Then the approximation error  $\varepsilon_{f,F}(\mathcal{T})$  is attained on a facet of some simplex  $S$  of  $\mathcal{T}$  that is not on the boundary of  $D$ .

*Proof.* The proof follows directly from Lemma 3.1 and the fact that all boundary facets are axis-parallel. This implies that the approximation error is zero there as either  $x_1 = x_0$  or  $y_1 = y_0$  holds in Equation (1).  $\square$

A geometric view on Lemma 3.1 is sketched in Figure 1. Here, the approximation error on a facet is illustrated as one quarter of the area of the facet-enclosing axis-parallel rectangle. The yellow rectangle is the facet enclosing axis-parallel rectangle, and the area of the red rectangle equals the approximation error on this facet.

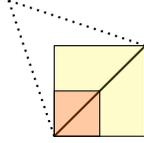


FIGURE 1. Geometrical visualization of the approximation error on an edge of a simplex.

We use Lemma 3.1 to show that we can reduce the triangulation problem over  $D$  to the unit box  $U := [0, 1] \times [0, 1]$  by scaling the prescribed accuracy with the area of the box. In particular, we show how to transform any  $\varepsilon$ -triangulation  $\mathcal{T}_U$  of  $U$  into an  $\nu\varepsilon$ -triangulation  $\mathcal{T}_D$  of  $D$  such that  $|\mathcal{T}_U| = |\mathcal{T}_D|$  holds, with  $\nu := (\bar{x} - \underline{x})(\bar{y} - \underline{y})$  being the area of  $D$ . The triangulation  $\mathcal{T}_D$  is obtained by a linear mapping of the nodes  $N(\mathcal{T}_U)$ .

**Lemma 3.3** (Invariance of triangulations under scaling and shifting). *Let  $f_U : U \rightarrow \mathbb{R}$  be a pwl.  $\varepsilon$ -approximation of  $F$  defined by a triangulation  $\mathcal{T}_U$  of the unit box  $U$ . Further, let  $L : \mathbb{R}^2 \rightarrow \mathbb{R}^2, L(x, y) = (\underline{x} + x(\bar{x} - \underline{x}), \underline{y} + y(\bar{y} - \underline{y}))$ . Let  $\mathcal{T}_D$  be the triangulation of  $D$  such that each simplex  $S_D \in \mathcal{T}_D$  corresponds to a simplex  $S_U \in \mathcal{T}_U$  via the mapping  $L$  on the respective vertex sets, i.e.  $V(S_D) = \{L(P) \mid P \in V(S_U)\}$ . Then  $\mathcal{T}_D$  is a  $\nu\varepsilon$ -triangulation of  $D$  and  $|\mathcal{T}_D| = |\mathcal{T}_U|$ .*

*Proof.* By construction it is clear that  $|\mathcal{T}_D| = |\mathcal{T}_U|$  holds. We have to show that  $\mathcal{T}_D$  is a  $\nu\varepsilon$ -triangulation of  $D$ . Let  $e_D$  be a facet of an arbitrary simplex  $S_D \in \mathcal{T}_D$  with endpoints  $(x_0^D, y_0^D)$  and  $(x_1^D, y_1^D)$ . The endpoints of  $e_D$  are nodes of  $\mathcal{T}_D$  and therefore the result of a linear mapping of two vertices  $(x_0^U, y_0^U)$  and  $(x_1^U, y_1^U)$  that are the endpoints of some face  $e_U$  in  $\mathcal{T}_U$ :

$$\begin{aligned} (x_0^D, y_0^D) &= L((x_0^U, y_0^U)) = (\underline{x} + x_0^U(\bar{x} - \underline{x}), \underline{y} + y_0^U(\bar{y} - \underline{y})), \\ (x_1^D, y_1^D) &= L((x_1^U, y_1^U)) = (\underline{x} + x_1^U(\bar{x} - \underline{x}), \underline{y} + y_1^U(\bar{y} - \underline{y})). \end{aligned}$$

Further, let  $\varepsilon_e \in [0, \varepsilon]$  be the approximation error over  $e_U$ . The approximation error over  $e_D$ , attained at its centre  $(x^*, y^*)$ , is calculated as follows:

$$\begin{aligned} E_{f_D, G}((x^*, y^*)) &= \frac{1}{4}(x_1^D - x_0^D)(y_1^D - y_0^D) \\ &= \frac{1}{4}(\underline{x} + x_1^U(\bar{x} - \underline{x}) - \underline{x} - x_0^U(\bar{x} - \underline{x})) \\ &\quad (y_1^U(\bar{y} - \underline{y}) - y_0^U(\bar{y} - \underline{y})) \\ &= \frac{1}{4} \underbrace{(x_1^U - x_0^U)}_{\varepsilon_e} \underbrace{(y_1^U - y_0^U)(\bar{x} - \underline{x})(\bar{y} - \underline{y})}_{\nu} = \varepsilon_e \nu. \end{aligned}$$

Thus, the approximation error on each facet of  $\mathcal{T}_D$  is  $\nu$  times the error on the corresponding facet in  $\mathcal{T}_U$ . Since we assumed that  $\mathcal{T}_U$  is an  $\varepsilon$ -approximation,  $\varepsilon_e \nu \leq \varepsilon \nu$  holds for each facet  $e$  of  $\mathcal{T}_D$ . This means  $\mathcal{T}_D$  is an  $\nu\varepsilon$ -triangulation.  $\square$

As a result of Lemma 3.3, we can consider the approximation of  $F$  over  $U$  w.l.o.g. in the following:

**Problem 2.** *Given some  $\varepsilon > 0$ , find an  $\varepsilon$ -optimal triangulation of  $U$  w.r.t.  $F$ .*

**3.1. Solving the optimal triangulation problem for up to five simplices and a general lower bound.** In the following, we solve Problem (2) for a fixed number of simplices up to five and give a general lower bound with respect to the approximation quality  $\varepsilon > 0$ . Finding a general scheme that solves Problem (2) for arbitrary values of  $\varepsilon$  is an open problem. However, in [14] it is shown that Problem (2) can be formulated as a mixed-integer quadratically constrained program (MIQCP). The author takes advantage of the fact that the approximation error is attained on one of the facets of the simplices, and can therefore exploit the representability of a triangulation by a fully connected planar graph. This representation allows for the modelling of Problem (2) by an MIQCP with a finite number of constraints and variables. To the best of our knowledge, this is the only work trying to determine provably  $\varepsilon$ -optimal triangulations over box domains. Unfortunately, due to the size of the resulting MIQCP, this approach is computationally intractable even for trivial instances. Nevertheless, we will later use its underlying idea of representing a triangulation as a planar graph to prove the optimality of triangulations consisting of up to five simplices.

**A lower bound for  $\varepsilon$ -optimal triangulations.** We begin our examination of Problem (2) in deriving a general lower bound for  $\varepsilon$ -optimal triangulations. This lower bound enables us later to determine approximation guarantees for specific triangulation schemes. The idea of the presented lower bound goes back to the work of Pottmann et. al ([18]), who studied optimal triangulations of the plane  $\mathbb{R}^2$ . The authors showed that an  $\varepsilon$ -optimal triangulation of  $\mathbb{R}^2$ , in the sense that the individual simplices have maximal area, is obtained by using so-called  $\varepsilon$ -optimal simplices. An  $\varepsilon$ -optimal simplex satisfies a prescribed accuracy  $\varepsilon$  while maximizing its area. The area of an  $\varepsilon$ -optimal simplex is  $2\sqrt{5}\varepsilon$ . As it is possible to tile the  $\mathbb{R}^2$  with  $\varepsilon$ -optimal simplices only, such triangulations are optimal. In Figure 2, we illustrate two different  $1/4$ -optimal simplices. For more information on  $\varepsilon$ -optimal simplices, we refer the reader to [18] and [2]. We use the following lemma from [5], to give a general lower bound for the triangulation of polytopal domains.

**Lemma 3.4** (Lower bound). [5] *Let  $P \subset \mathbb{R}^2$  be a polytopal domain with an area of  $A(P)$ . Then any  $\varepsilon$ -triangulation  $\mathcal{T}_P$  of  $P$  w.r.t.  $F$  contains at least  $\lceil A(P)/2\sqrt{5}\varepsilon \rceil$  simplices.*

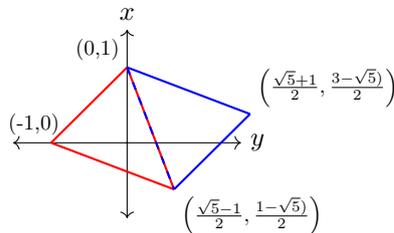


FIGURE 2. Optimal simplices with an approximation error of 0.25.

The proof is straightforward. If we assume that we can triangulate  $P$  solely by  $\varepsilon$ -optimal simplices, which have an area of  $2\sqrt{5}\varepsilon$ , we obtain the indicated lower bound immediately. For the approximation of  $F$  over  $U$ , Lemma 3.4 thus sets the following lower bound:

**Corollary 3.5.** *An  $\varepsilon$ -optimal triangulation  $\mathcal{T}$  of  $U$  w.r.t.  $F$  requires at least  $\lceil 1/2\sqrt{5}\varepsilon \rceil$  simplices.*

However, it is unclear whether or how to use  $\varepsilon$ -optimal simplices to triangulate  $U$ . Furthermore, the lower bound from Lemma 3.4 is not always tight. In [16], the author proves that one cannot triangulate a rectangle with an odd number of simplices such that all simplices have the same area. This means for those prescribed approximation accuracies  $\varepsilon > 0$  for which the lower bound is an odd number, we need at least one additional simplex. We now show that this lower bound can even be improved if we require that each simplex has at least one axis-parallel edge. This lemma has also been presented in the dissertation of the fourth author [14].

**Lemma 3.6.** *Let  $S$  be a non-degenerate simplex that satisfies a prescribed approximation accuracy of  $\varepsilon > 0$ . If  $S$  possesses at least one axis-parallel edge, then its maximum area is  $4\varepsilon$ .*

*Proof.* Let  $S$  be a simplex with vertices  $v_1 = (x_1, y_1)$ ,  $v_2 = (x_2, y_2)$  and  $v_3 = (x_3, y_3)$ . W.l.o.g., we assume  $e_{v_1, v_2}$  to be an axis-parallel edge. The area of  $S$  can then be calculated using the areas of the axis-parallel edge-enclosing rectangles of  $e_{v_1, v_3}$  and  $e_{v_2, v_3}$ :

$$A(S) = \frac{1}{2} |(x_3 - x_1)(y_3 - y_1) - (x_3 - x_2)(y_3 - y_2)|.$$

Further, for the edges  $e_{v_1, v_3}$  and  $e_{v_2, v_3}$ , the approximation error has to be less than or equal to  $\varepsilon$ , i.e.

$$\left| \frac{(x_3 - x_1)(y_3 - y_1)}{4} \right| \leq \varepsilon \quad \wedge \quad \left| \frac{(x_3 - x_2)(y_3 - y_2)}{4} \right| \leq \varepsilon.$$

Now, obviously  $A(S)$  attains its maximum possible value of  $4\varepsilon$  if

$$(x_3 - x_1)(y_3 - y_1) = 4\varepsilon \quad \wedge \quad (x_3 - x_2)(y_3 - y_2) = -4\varepsilon$$

holds. W.l.o.g., we assume  $e_{v_1, v_2}$  to be parallel to the  $x$ -axis. This means  $x_1 = x_2$  holds and we obtain a simplex of maximum area  $4\varepsilon$  if the vertices have the following positions

$$v_1 = (x_1, y_1), \quad v_2 = (x_2, y_1), \quad v_3 = \left( \frac{x_1 + x_2}{2}, \frac{8\varepsilon}{x_2 - x_1} + y_1 \right). \quad \square$$

**Corollary 3.7.** *An  $\varepsilon$ -optimal triangulation  $\mathcal{T}$  of  $U$  w.r.t.  $F$  where each simplex has at least one axis-parallel edge requires  $\lceil 1/4\varepsilon \rceil$  simplices.*

**Euler relations for triangulations.** The following optimality proofs exploit the representation of a triangulation by a fully connected planar graph. In this sense, the vertices and facets of the simplices become the nodes and edges of a graph. This allows us to translate the well-known Euler relations for planar graphs to triangulations, see also [3].

Let  $\mathcal{T}$  be a triangulation of some polytope  $P \subset \mathbb{R}^2$ , then it holds that

$$K = 2|N(\mathcal{T})| - |\mathcal{T}| - 2 \tag{2}$$

and

$$E = 3|N(\mathcal{T})| - K - 3, \tag{3}$$

where  $K$  is the number of vertices that lie on the boundary of  $P$  and  $E$  is the number of edges of  $\mathcal{T}$ . For  $K$  we further know that

$$K \leq |N(\mathcal{T})|, \tag{4}$$

which together with (2) gives

$$|N(\mathcal{T})| \leq |\mathcal{T}| + 2. \tag{5}$$

If  $P$  is a rectangle, then it obviously also holds that

$$K \geq 4. \quad (6)$$

**Optimal triangulations with up to five simplices.** We use the Euler relations to derive optimal triangulations of  $U$  for the approximation of  $F$  under the restriction that the number of simplices is fixed a priori. These triangulations in turn yield  $\varepsilon$ -optimal triangulations in the sense of Problem (2) for specific value ranges of  $\varepsilon$ . Let  $\mathcal{T}_n^*$  be a triangulation with a minimal approximation error among all triangulations of  $U$  consisting of  $n$  simplices, and let  $\varepsilon_n^*$  be the corresponding minimal approximation error. This means that  $\mathcal{T}_n^*$  is an  $\varepsilon$ -optimal triangulation for all  $\varepsilon \in (\varepsilon_{n+1}^*, \varepsilon_n^*]$ .

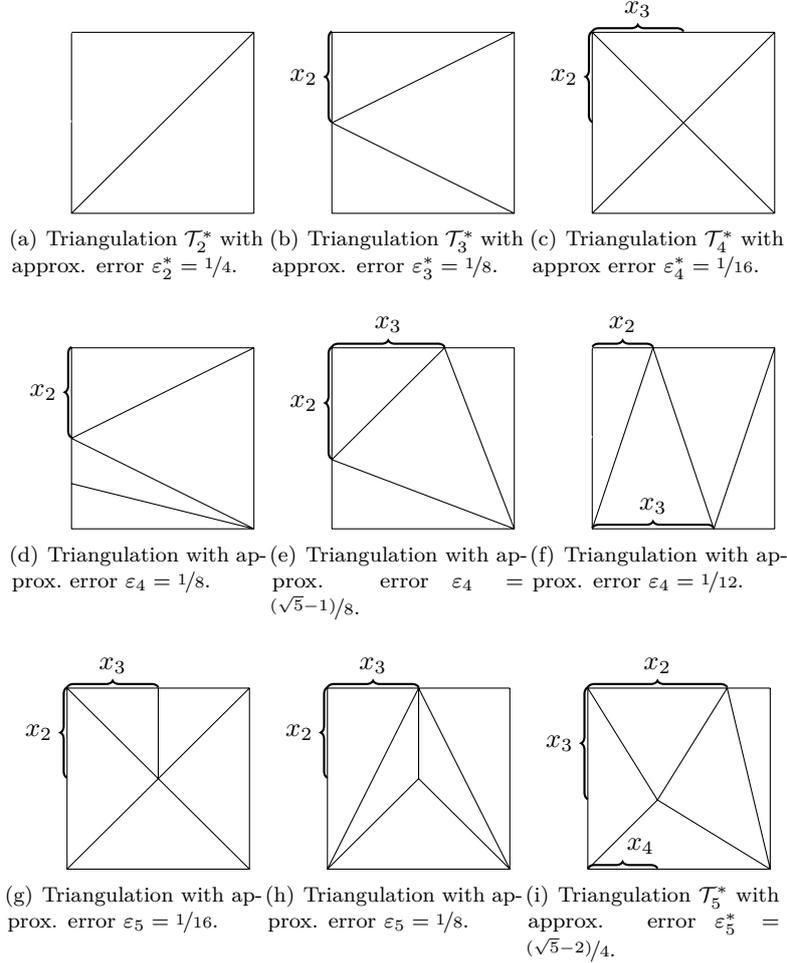


FIGURE 3. Triangulations with  $|\mathcal{T}| \leq 5$ ; (\*) Figure 3(a), Figure 3(b), Figure 3(c) and Figure 3(i) are optimal triangulations with respect to the number of simplices used.

In the following, we determine optimal triangulations  $\mathcal{T}$  of  $U$  for  $|\mathcal{T}| = 2, 3, 4, 5$ . **We start with  $|\mathcal{T}| = 2$ :** Relations (2)–(6) reduce the set of possible triangulations to those where  $|N(\mathcal{T})| = 4$ ,  $K = 4$  and  $E = 5$ . Therefore, only two triangulations of  $U$  are possible, namely the ones using one of the diagonals to triangulate  $U$ . Either

diagonal leads to an optimal triangulation as they produce the same approximation error of  $\varepsilon_2^* = 1/4$ , attained at their centre. We refer to these equivalent triangulations by  $\mathcal{T}_2^*$ , one example is shown in Figure 3(a).

**Next we consider  $|\mathcal{T}| = 3$ :** Relations (2)–(6) only allow for triangulations with  $|N(\mathcal{T})| \in \{4, 5\}$ . Assuming  $|N(\mathcal{T})| = 4$ , we obtain  $K = 3$ , which is a contradiction to Equation (4). Assuming  $|N(\mathcal{T})| = 5$  instead, we obtain  $K = 5$ . The latter implies that all nodes have to be on the boundary of  $U$ . Due to symmetry, it does not matter if the additional node is on a facet which is either parallel to the  $x$ -axis or the  $y$ -axis. In both cases, the following linear program defines the optimal position of the fifth node:

$$\begin{aligned} \varepsilon_3^* &:= \min_x 1/4 x_1 \\ \text{s.t. } &-x_1 + x_2 \leq 0 \\ &-x_1 - x_2 + 1 \leq 0 \\ &x \in [0, 1]^2. \end{aligned} \tag{7}$$

In Problem (7) and in the subsequent optimization problems used to determine optimal triangulations,  $x_1$  models four times the overall approximation error. Further variables model the coordinates of the nodes which can be chosen freely in the respective configuration, and each constraint models the approximation error on an inner edge of the considered triangulation. Figures 3(a)–3(i) illustrate which constraint models which edge. The optimal solution to Problem (7) is  $x^* = (1/2, 1/2)$  with a corresponding approximation error of  $\varepsilon_3^* = 1/8$ . We refer to these equivalent triangulation by  $\mathcal{T}_3^*$ , an example is shown in Figure 3(b).

**We continue with  $|\mathcal{T}| = 4$ :** Again, from relations (2)–(6) we know that  $|N(\mathcal{T})| \in \{4, 5, 6\}$ . If  $|N(\mathcal{T})| = 4$ , we obtain  $K = 2$ , which is a contradiction to (6). If  $|N(\mathcal{T})| = 5$ , we obtain  $K = 4$  and  $E = 8$ . Triangulations that fulfil this property have one inner vertex that is connected to all four corners of  $U$ . We can find the optimal position of the inner vertex by solving this quadratic program:

$$\begin{aligned} \varepsilon_4^* &:= \min_x 1/4 x_1 \\ \text{s.t. } &-x_1 + x_2 x_3 \leq 0 \\ &-x_1 + x_3 - x_2 x_3 \leq 0 \\ &-x_1 - x_2 - x_3 + x_2 x_3 + 1 \leq 0 \\ &x \in [0, 1]^3. \end{aligned} \tag{8}$$

The optimal solution to Problem (8) is  $x^* = (1/4, 1/2, 1/2)$ , with an approximation error of  $\varepsilon_4^* = \frac{1}{16}$ . We refer to the corresponding optimal triangulation by  $\mathcal{T}_4^*$ , see Figure 3(c). Finally, if  $|N(\mathcal{T})| = 6$ , we obtain  $K = 6$ , which means that all nodes have to be on the boundary of  $U$ . Due to symmetry, the remaining two free nodes can either be positioned on adjacent, opposite or both on the same boundary facet. If both nodes are placed on the same boundary facet of  $U$ , we have a similar case to  $|\mathcal{T}| = 3$  and the minimal possible approximation error is  $1/8$ . An example of this triangulation is shown in Figure 3(d). If the two nodes are placed on adjacent boundary facets, we can minimize the approximation error by solving the following quadratic program:

$$\begin{aligned} &\min_x 1/4 x_1 \\ \text{s.t. } &-x_1 + x_2 x_3 \leq 0 \\ &-x_1 + x_3 - x_2 x_3 \leq 0 \\ &-x_1 - x_3 + 1 \leq 0 \\ &x \in [0, 1]^3. \end{aligned} \tag{9}$$

The optimal value of Problem (9) is  $1/8$ . Thus, this configuration can also not be optimal, see Figure 3(e) for an example. If the two additional nodes are placed on opposite boundary facets, we can minimize the approximation error by solving the following linear program:

$$\begin{aligned}
& \min_x 1/4 x_1 \\
& \text{s.t. } -x_1 - x_2 \leq 0 \\
& -x_1 - x_3 + 1 \leq 0 \\
& -x_1 + x_3 - x_2 \leq 0 \\
& x \in [0, 1]^3.
\end{aligned} \tag{10}$$

The optimal solution to Problem (10) is  $x = (1/3, 1/3, 2/3)$ , with an approximation error of  $1/12$ . Thus, this configuration can also not yield an optimal triangulation, see Figure 3(f) for an example. Consequently, we know that  $\varepsilon_4^* = 1/16$  holds and that  $\mathcal{T}_4^*$  is optimal.

**Finally, we consider the case  $|\mathcal{T}| = 5$ :** From relations (2)–(6), it follows that  $|N(\mathcal{T})| \in \{6, 7\}$ . If  $|N(\mathcal{T})| = 6$ , we obtain  $K = 5$ . This means we have one free node on the boundary of  $U$  and one free inner node. In this case, there are three different ways to connect the nodes such that we have a triangulation with exactly five simplices. It is obvious that two of them are not optimal, see Figure 3(g) and Figure 3(h). The approximation error of the third configuration is minimized by the following non-convex quadratic program (QP), visualized in Figure 3(i):

$$\begin{aligned}
\varepsilon_5^* & := \min_x 1/4 x_1 \\
& \text{s.t. } -x_1 - x_2 + 1 \leq 0 \\
& -x_3x_4 + x_2x_3 - x_1 \leq 0 \\
& \quad x_3x_4 - x_1 \leq 0 \\
& -x_3x_4 + x_4 - x_1 \leq 0 \\
& x_3x_4 - x_3 - x_4 - x_1 + 1 \leq 0 \\
& x \in [0, 1]^4.
\end{aligned} \tag{11}$$

By geometric reasoning, we guess

$$x^* = (\sqrt{5} - 2, 3 - \sqrt{5}, (\sqrt{5}-1)/2, (3-\sqrt{5})/2)^T$$

as a candidate for an optimal solution, which entails an approximation error of  $\varepsilon_5^* = (\sqrt{5}-2)/4$ . In the following, we prove the optimality of  $x^*$ . In general, we can prove that a vector is globally optimal for a non-convex QP by checking two conditions which together are sufficient.

**Lemma 3.8** (Global optimality in non-convex QPs). [15] *Consider a quadratic program of the form*

$$\begin{aligned}
& \min x^T Q_0 x + c_0^T x \\
& \text{s.t. } x^T Q_i x + c_i^T x - b_i \leq 0 \quad \forall i = 1, \dots, m,
\end{aligned}$$

where all  $Q_i$  are  $n \times n$  real symmetric matrices and  $c_i \in \mathbb{R}^n$ . A vector  $x^* \in \mathbb{R}^n$  is globally optimal for this problem if there exists a vector  $\lambda^* \in \mathbb{R}^m$  such that

$$\begin{aligned}
& \left( \sum_{i=0}^m Q_i \right) x^* + c_0 + \sum_{i=0}^m \lambda_i^* c_i = 0 \\
& x^{*T} Q_i x^* + c_i^T x^* - b_i \leq 0 \quad \forall i = 1, \dots, m \\
& \quad \lambda_i^* \geq 0 \quad \forall i = 1, \dots, m \\
& \lambda_i^* (x^{*T} Q_i x^* + c_i^T x^* - b_i) = 0 \quad \forall i = 1, \dots, m.
\end{aligned} \tag{12}$$

and

$$Q_0 + \sum_{i=1}^m \lambda_i^* Q_i \succcurlyeq 0 \quad (13)$$

hold.

We use Lemma 3.8 to prove that  $x^*$  is optimal for Problem (11).

**Proposition 3.9.** *The vector  $x^* = (\sqrt{5} - 2, 3 - \sqrt{5}, (\sqrt{5}-1)/2, (3-\sqrt{5})/2)^T$  is globally optimal for the non-convex quadratic program (11).*

*Proof.* First, we show that  $x^*$  is a locally optimal solution, applying the KKT conditions from (12). This means we have to find a vector  $\lambda \in \mathbb{R}^5$  that solves the following system:

$$\begin{aligned} \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 &= 1 \\ (\sqrt{5} - 1) - 2\lambda_1 &= 0 \\ -\lambda_5 &= 0 \\ \lambda_4 - \lambda_5 &= 0 \\ \lambda_4 &= 0 \\ \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 &\geq 0. \end{aligned} \quad (14)$$

Note that we can neglect the dual variables for the constraints  $x \in [0, 1]^4$ , since they must all be zero for  $x^*$  anyway due to  $0 < x_i^* < 1$ . We can easily check that  $\lambda^* = ((\sqrt{5}-1)/2, (1+\sqrt{5})/2, 0, 0, 0)^T$  is feasible for (14), and therefore  $(x^*, \lambda^*)$  is feasible for (12). According to Lemma 3.8,  $(x^*, \lambda^*)$  is globally optimal for the Problem (11) if  $Q_0 + \sum_{i=1}^5 \lambda_i^* Q_i \succeq 0$  holds. As the matrix

$$Q^* := Q_0 + \sum_{i=1}^5 \lambda_i^* Q_i = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_2^* & 0 \\ 0 & 0 & 0 & -\lambda_2^* \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

is strictly upper triangular, its only eigenvalue is 0. It is therefore a positive semidefinite matrix, and consequently  $x^*$  is a globally optimal solution to Problem (11).  $\square$

As a result of Proposition 3.9, the minimal approximation error of the configuration modelled in Problem (11) is  $\varepsilon_5^* = (\sqrt{5}-2)/4$ . This means that the corresponding triangulation  $\mathcal{T}_5^*$  is optimal for the case of five simplices.

**3.2. Crossing swords: A  $\sqrt{5}/2$ -approximation algorithm for optimal triangulations.** We use the optimal triangulations consisting of up to five simplices from Section 3.1 to derive our novel triangulation scheme *crossing swords* stated in Algorithm 1. The algorithm takes as input an integer  $N$  and returns a triangulation of  $U$  with  $N$  simplices that has an approximation error of less than  $1/4(N-1)$ . We will use this approximation accuracy depending on the number of simplices to prove that the crossing swords scheme is a  $\sqrt{5}/2$ -approximation algorithm for Problem (2).

The general idea of crossing swords triangulations is to use the optimal triangulations consisting of up to five simplices as partial triangulations to construct arbitrarily fine triangulations of  $U$ . The algorithm takes as input a number of simplices  $N$  and starts with a partition of  $U$  into  $n$  smaller rectangles, where  $n$  depends on the modulus of  $N$  by 4. The first  $n - 1$  rectangles must have the same area and they are triangulated with four simplices in fashion of  $\mathcal{T}_4^*$ . The  $n$ -th rectangle can have a different area, and its subtriangulation is either  $\mathcal{T}_2^*$ ,  $\mathcal{T}_3^*$ ,  $\mathcal{T}_4^*$  or  $\mathcal{T}_5^*$ , also depending on the modulus of the input  $N$  by 4. The areas of the rectangles in Algorithm 1 are chosen such that the approximation error on all simplices is equal, which we use as a heuristic rule to minimize the (overall) approximation error. The repeated use of the  $\mathcal{T}_4^*$  triangulation motivates the name *crossing swords*.

**Algorithm 1:** Crossing Sword Triangulation Scheme (for the unit box  $U$ )**Input:** The number of simplices  $N \geq 4$ .**Output:** A triangulation  $\mathcal{T}_{cs}^N$  of  $U$  with  $|\mathcal{T}_{cs}^N| = N$  and a corresponding pwl. approximation  $f$  of  $F$  over  $U$  with an approximation error of

$$\varepsilon_{f,F}(\mathcal{T}_{cs}^N) = \begin{cases} \frac{1}{4} \frac{1}{N}, & \text{if } N \equiv 0 \pmod{4}, \\ \frac{1}{4} \frac{\sqrt{5}-2}{(\sqrt{5}-2)(N-5)+1}, & \text{if } N \equiv 1 \pmod{4}, \\ \frac{1}{4} \frac{1}{N-1}, & \text{if } N \equiv 2 \pmod{4}, \\ \frac{1}{4} \frac{1}{N-1}, & \text{if } N \equiv 3 \pmod{4}. \end{cases} \quad (15)$$

```

1 if  $N \equiv 0 \pmod{4}$  then
2   Set  $n = \frac{N}{4}$ 
3   Find a rectangular partition  $\bigcup_{i=1}^n U_i$  of  $U$ 
   s.t.  $U_i$  has an area of  $A(U_i) = \frac{4}{N}$ 
4   return  $\mathcal{T}_{cs}^N := \bigcup_{i=1}^n \mathcal{T}_4^*(U_i)$ 
5 else if  $N \equiv 1 \pmod{4}$  then
6   Set  $n = \frac{N-1}{4}$ 
7   Find a rectangular partition  $\bigcup_{i=1}^n U_i$  of  $U$ 
   s.t.  $U_i$  has an area of  $A(U_i) = \frac{4(\sqrt{5}-2)}{(\sqrt{5}-2)(N-5)+1}$ ,  $i = 1, \dots, n-1$ ,
   and  $U_n$  has an area of  $A(U_n) = 1/((\sqrt{5}-2)(N-5)+1)$ 
8   return  $\mathcal{T}_{cs}^N := \bigcup_{i=1}^{n-1} \mathcal{T}_4^*(U_i) \cup \mathcal{T}_5^*(U_n)$ 
9 else if  $N \equiv 2 \pmod{4}$  then
10  Set  $n = \frac{N-2}{4} + 1$ 
11  Find a rectangular partition  $\bigcup_{i=1}^n U_i$  of  $U$ 
   s.t.  $U_i$  has an area of  $A(U_i) = \frac{4}{N-1}$ ,  $i = 1, \dots, n-1$ ,
   and  $U_n$  has an area of  $A(U_n) = \frac{1}{N-1}$ 
12  return  $\mathcal{T}_{cs}^N := \bigcup_{i=1}^{n-1} \mathcal{T}_4^*(U_i) \cup \mathcal{T}_2^*(U_n)$ 
13 else if  $N \equiv 3 \pmod{4}$  then
14  Set  $n = \frac{N-3}{4} + 1$ 
15  Find a rectangular partition  $\bigcup_{i=1}^n U_i$  of  $U$ 
   s.t.  $U_i$  has an area of  $A(U_i) = \frac{4}{N-1}$ ,  $i = 1, \dots, n-1$ ,
   and  $U_n$  has an area of  $A(U_n) = \frac{2}{N-1}$ 
16  return  $\mathcal{T}_{cs}^N := \bigcup_{i=1}^{n-1} \mathcal{T}_4^*(U_i) \cup \mathcal{T}_3^*(U_n)$ 

```

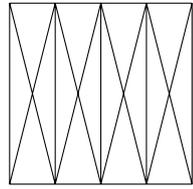
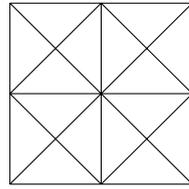
(a)  $\mathcal{T}_{cs}^{16}$  with skinny simplices(b)  $\mathcal{T}_{cs}^{16}$  with quadratic simplices.

FIGURE 4. Visualization of skinny vs. quadratic crossing swords triangulations with the same approximation error.

**Lemma 3.10.** (Crossing Sword Triangulations) *Algorithm 1 is correct, i.e. for any input  $N$  it returns a triangulation of  $U$  with an approximation error according to Equation (15).*

*Proof.* In the following, we prove the correctness of the approximation error stated in Equation (15) with respect to the modulus of  $N$  by 4. For the sake of simplicity, we speak of  $\mathcal{T}_i^*$ -triangles ( $\Delta_{\mathcal{T}_i^*}$ ) whenever a rectangle is triangulated in fashion of  $\mathcal{T}_i^*$  with  $i \in \{2, 3, 4, 5\}$ . Via the scaling formulas established in Lemma 3.3, the approximation error of these subtriangulations on arbitrary rectangles  $D$  with an area of  $A(D)$  is given as follows (cf. Figure 3.1):

$$\begin{aligned}\varepsilon(\Delta_{\mathcal{T}_2^*}) &= 1/4 \cdot A(D), & \varepsilon(\Delta_{\mathcal{T}_3^*}) &= 1/8 \cdot A(D), \\ \varepsilon(\Delta_{\mathcal{T}_4^*}) &= 1/16 \cdot A(D), & \varepsilon(\Delta_{\mathcal{T}_5^*}) &= (\sqrt{5}-2)/4 \cdot A(D).\end{aligned}$$

We now distinguish the four cases of Equation (15), examples of these with  $N = 6, 7, 8, 9$  are shown in Figure 5. To this end, we check that the areas of the rectangles sum up to one and that the approximation error postulated in Equation (15) is fulfilled by the returned triangulation  $\mathcal{T}_{cs}^N$  of Algorithm 1.

**Case  $N \equiv 0 \pmod{4}$ :** Each  $U_i$  has an area of  $A(U_i) = 4/N$ , and therefore

$$\sum_{i=1}^n A(U_i) = N/4 \cdot 4/N = 1$$

holds. The approximation error on each  $\mathcal{T}_4^*$ -triangle, and thus also the one arising from  $\mathcal{T}_{cs}^N$ , is given as

$$\varepsilon(\mathcal{T}_{cs}^N) = \varepsilon(\Delta_{\mathcal{T}_4^*}) = 1/16 \cdot 4/N = 1/4N.$$

**Case  $N \equiv 1 \pmod{4}$ :** The rectangles  $U_1, \dots, U_{n-1}$  have an area of

$$A(U_i) = \frac{4(\sqrt{5}-2)}{(\sqrt{5}-2)(N-5)+1},$$

and  $U_n$  has an area of  $A(U_n) = 1/(\sqrt{5}-2)(N-5)+1$ . This means

$$\sum_{i=1}^n A(U_i) = \frac{N-5}{4} \cdot \frac{4(\sqrt{5}-2)}{(\sqrt{5}-2)(N-5)+1} + \frac{1}{(\sqrt{5}-2)(N-5)+1} = 1.$$

The corresponding approximation errors are given as

$$\begin{aligned}\varepsilon(\Delta_{\mathcal{T}_4^*}) &= \frac{1}{16} \frac{4(\sqrt{5}-2)}{(\sqrt{5}-2)(N-5)+1} = \frac{1}{4} \frac{(\sqrt{5}-2)}{(\sqrt{5}-2)(N-5)+1}, \\ \varepsilon(\Delta_{\mathcal{T}_5^*}) &= \frac{\sqrt{5}-2}{4} \cdot \frac{1}{(\sqrt{5}-2)(N-5)+1} = \frac{1}{4} \frac{(\sqrt{5}-2)}{(\sqrt{5}-2)(N-5)+1}.\end{aligned}$$

**Case  $N \equiv 2 \pmod{4}$ :** The rectangles  $U_1, \dots, U_{n-1}$  have an area of  $A(U_i) = 4/N-1$ , and  $U_n$  has an area of  $A(U_n) = 1/N-1$ . Therefore, we have

$$\sum_{i=1}^n A(U_i) = \frac{(N-2)}{4} \cdot \frac{4}{N-1} + \frac{1}{N-1} = 1.$$

The corresponding approximation errors are given as

$$\varepsilon(\Delta_{\mathcal{T}_4^*}) = \frac{1}{16} \frac{4}{N-1} = \frac{1}{4} \frac{1}{N-1}, \quad \varepsilon(\Delta_{\mathcal{T}_2^*}) = \frac{1}{4} \frac{1}{N-1}.$$

**Case  $N \equiv 3 \pmod{4}$ :** The rectangles  $U_1, \dots, U_{n-1}$  have an area of  $A(U_i) = 4/N-1$ , and  $U_n$  has an area of  $A(U_n) = 2/N-1$ . This leads to

$$\sum_{i=1}^n A(U_i) = \frac{N-3}{4} \cdot \frac{4}{N-1} + \frac{2}{N-1} = 1.$$

The corresponding approximation errors are given as

$$\varepsilon(\Delta_{\mathcal{T}_4^*}) = \frac{1}{16} \frac{4}{N-1} = \frac{1}{4} \cdot \frac{1}{N-1}, \quad \varepsilon(\Delta_{\mathcal{T}_3^*}) = \frac{1}{8} \frac{2}{N-1} = \frac{1}{4} \frac{1}{N-1}. \quad \square$$

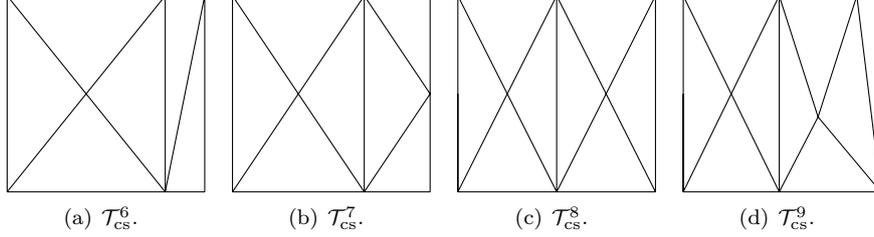


FIGURE 5. Congruence classes of crossing sword triangulations.

As a direct result of Lemma 3.10, for any approximation accuracy  $\varepsilon > 0$  there exists an  $\varepsilon$ -triangulation consisting of at most  $\lceil 1/4\varepsilon \rceil + 1$  simplices.

**Proposition 3.11** (Upper bound on  $\varepsilon$ -optimal triangulations). *For any approximation accuracy  $\varepsilon > 0$ , there exists an input  $N_\varepsilon \leq \lceil 1/4\varepsilon \rceil + 1$  to Algorithm 1 such that the returned triangulation  $\mathcal{T}_{\text{cs}}^{N_\varepsilon}$  is an  $\varepsilon$ -triangulation.*

*Proof.* The proof results directly from calling Algorithm 1 with  $N_\varepsilon := \lceil 1/4\varepsilon \rceil + 1$ .  $\square$

We use Proposition 3.11 to prove that Algorithm 1 is an approximation algorithm for Problem (2).

**Theorem 3.12.** *Given an approximation accuracy  $\varepsilon > 0$ , calling Algorithm 1 with input  $N_\varepsilon := \lceil 1/4\varepsilon \rceil + 1$  is a  $(\sqrt{5}/2 + 4\sqrt{5}\varepsilon)$ -approximation algorithm for Problem (2).*

*Proof.* From Proposition 3.11, we know that  $\mathcal{T}_{\text{cs}}^{N_\varepsilon}$  is an  $\varepsilon$ -triangulation with  $|\mathcal{T}_{\text{cs}}^{N_\varepsilon}| = \lceil 1/4\varepsilon \rceil + 1$ . Using the lower bound  $\lceil 1/2\sqrt{5}\varepsilon \rceil$  for the number of simplices in an  $\varepsilon$ -optimal triangulation from Proposition 3.5, we can prove the proclaimed approximation guarantee of crossing swords triangulations:

$$\frac{|\mathcal{T}_{\text{cs}}^\varepsilon|}{\left\lceil \frac{1}{2\sqrt{5}\varepsilon} \right\rceil} = \frac{\left\lceil \frac{1}{4\varepsilon} \right\rceil + 1}{\left\lceil \frac{1}{2\sqrt{5}\varepsilon} \right\rceil} \leq \frac{\frac{1}{4\varepsilon} + 2}{\frac{1}{2\sqrt{5}\varepsilon}} = \frac{\sqrt{5}}{2} + 4\sqrt{5}\varepsilon. \quad \square$$

This upper estimate of the approximation guarantee of crossing swords triangulations can be strengthened when only considering certain discrete values of approximation accuracies.

**Theorem 3.13.** *Let  $\varepsilon := 1/16i$  for some  $i \in \mathbb{N}$ . Then calling Algorithm 1 with input  $N_\varepsilon := 1/4\varepsilon$  is a  $\sqrt{5}/2$ -approximation algorithm for Problem (2).*

*Proof.* By the choice of  $\varepsilon$ , we have  $N_\varepsilon \in \mathbb{N}$ . Further, we know from Equation (15) that  $N_\varepsilon \equiv 0 \pmod{4}$  holds and that Algorithm 1 called with  $N_\varepsilon$  returns an  $\varepsilon$ -triangulation  $\mathcal{T}_{\text{cs}}^{N_\varepsilon}$ . Again, by using the lower bound from Proposition 3.5, we can prove the proclaimed approximation quality of crossing swords triangulations:

$$\frac{|\mathcal{T}_{\text{cs}}^{N_\varepsilon}|}{\left\lceil \frac{1}{2\sqrt{5}\varepsilon} \right\rceil} = \frac{\frac{1}{4\varepsilon}}{\left\lceil \frac{1}{2\sqrt{5}\varepsilon} \right\rceil} \leq \frac{\frac{1}{4\varepsilon}}{\frac{1}{2\sqrt{5}\varepsilon}} = \frac{\sqrt{5}}{2}. \quad \square$$

Next, we show that crossing swords produces optimal triangulations for  $\varepsilon := 1/16i$  with  $i \in \mathbb{N}$  if we additionally require that each simplex has at least one axis-parallel edge.

**Theorem 3.14.** *Let  $\varepsilon := 1/16i$  for some  $i \in \mathbb{N}$ . Then calling Algorithm 1 with input  $N_\varepsilon := 1/4\varepsilon$  yields an  $\varepsilon$ -optimal triangulation for Problem (2) if we require that each simplex have at least one axis-parallel edge.*

*Proof.* In Corollary 3.7, we showed that under the condition that each simplex has an axis-parallel edge, the lower bound for the number of simplices in an  $\varepsilon$ -optimal triangulation is  $\lceil 1/4\varepsilon \rceil$ . Further, we know from Equation (15) that  $N_\varepsilon \equiv 0 \pmod{4}$  holds and that Algorithm 1 called with  $N_\varepsilon$  returns an  $\varepsilon$ -triangulation  $\mathcal{T}_{\text{cs}}^{N_\varepsilon}$ , where  $|\mathcal{T}_{\text{cs}}^{N_\varepsilon}| = 1/4\varepsilon$ . □

Note that in Algorithm 1, we do not explicitly state how to construct the rectangular partition of the domain. In fact, a partition with given areas of the rectangles always exists as we explain now. One way to create such a partition is to line up rectangles along the  $x$ -axis that have a full height of one. In this arrangement, it is easy to choose the area of these rectangles as required based on their width, whereas it is not obvious how to do this for other arrangements. A corresponding version of Algorithm 1 is stated in the appendix as Algorithm 2. However, this arrangement inherently leads to very skinny simplices for large values of  $N$ , see Figure 4 for an example with  $N = 16$ . Skinny simplices in turn can lead to numerical difficulties in evaluating the function values of the pwl. approximation. Although irrelevant from a theoretical point of view, it would be desirable in practice to partition  $U$  into rectangles that are “as square as possible” to avoid skinny simplices. In [7], two problems are studied that address this very question of partitioning “as square as possible”. One is called *PERI-SUM*, which aims at minimizing the sum of all perimeters for a given set of rectangular areas. The second one is *PERI-MAX*, which has the goal to minimize the largest perimeter over all rectangles for a given set of rectangular areas. Both problems are NP-complete. For certain inputs, namely  $N = 4^i$  with  $i \in \mathbb{N}$ , all rectangles have the same area and a partition into squares is possible. The longest-edge bisection scheme described in Section 3.3 yields exactly such crossing swords triangulations with square partition elements for  $N = 4^i$ . For general areas as inputs to PERI-SUM, [11] gives a modelling as an MIQCP. In order to find high quality solutions fast, the authors also present a polynomial time  $3/\sqrt{2}$ -approximation algorithm.

**Remark 3.15.** *According to Lemma 3.3, we can apply Algorithm 1 and the lower bound from Lemma 3.4 to general box domains  $D = [x, \bar{x}] \times [y, \bar{y}]$ . Therefore, the approximation guarantees from Theorem 3.12 and Theorem 3.13 are also valid for arbitrary box domains.*

**3.3. Approximation qualities of widely used triangulation schemes.** In this section, we prove approximation guarantees for several popular triangulation schemes from the literature. For this purpose, we proceed analogously to Theorem 3.13 and consider only discrete values of approximation accuracies converging to zero.

#### Generalized K1-triangulations.

We start by analysing the uniform triangulation schemes J1 and K1. Furthermore, we develop a generalized version of the K1, for which we prove an approximation guarantee of  $\sqrt{5}$ . In Figure 6(a) and Figure 6(b), we show a K1- and a J1-triangulation consisting of 32 simplices each. Both triangulation schemes are defined over a grid that partitions the domain into axis-parallel squares. Each square is triangulated in fashion of  $\mathcal{T}_2^*$ . Note that J1 and K1 only differ in the orientation of the diagonals.

However, the orientation of the diagonals does not matter for the resulting approximation error. Therefore, both triangulation schemes are equally efficient in terms of the number of simplices.

**Proposition 3.16.** *For any given uniform rectangular grid, K1- and J1-triangulations have the same approximation error.*

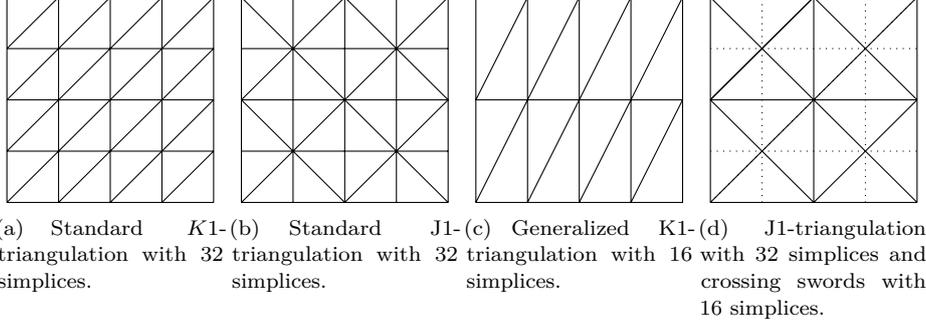


FIGURE 6. Uniform triangulations J1 and K1.

In the way K1- and J1-triangulations have been used in the literature so far, they require the same uniform partitioning along both axes. Therefore, they offer only limited flexibility to generate good triangulations for most prescribed approximation accuracies. To add somewhat more flexibility, we generalize the K1-triangulation scheme by allowing subdivisions of different granularity for each axis – see also Figure 6(c), where we show a generalized K1-triangulation over a  $4 \times 2$ -grid. The approximation error of the generalized K1-triangulation scheme is uniquely determined by the underlying grid.

**Proposition 3.17.** *Let  $L$  be a uniform grid over  $U$  with  $i$  intervals on the  $x$ -axis and  $j$  intervals on the  $y$ -axis. Further, let  $\mathcal{T}_{K1}^{ij}$  be a generalized K1-triangulation of  $U$  with respect to  $L$ . Then*

$$|\mathcal{T}_{K1}^{ij}| = 2ij$$

and

$$\varepsilon(\mathcal{T}_{K1}^{ij}) = \frac{1}{4ij} = \frac{1}{2|\mathcal{T}_{K1}^{ij}|}$$

hold.

*Proof.* Each rectangle of the underlying grid has an area of  $1/ij$ . From Lemma 3.3, we know that the approximation error of a  $\mathcal{T}_2^*$  triangulations is  $1/4$  times the area of the triangulated rectangle, which finishes the proof.  $\square$

We can use the lower bound from Lemma 3.4 to prove an approximation guarantee for generalized K1-triangulations.

**Theorem 3.18.** *Let  $\varepsilon := 1/4ij$  for some  $i, j \in \mathbb{N}$ . Then the generalized K1-triangulation scheme is an  $\sqrt{5}$ -approximation algorithm for Problem (2).*

*Proof.* Let  $L$  be a uniform grid over  $U$  with  $i$  intervals in  $x$ -direction and  $j$  intervals in  $y$ -direction, and let  $\mathcal{T}_{K1}^{ij}$  be the corresponding K1-triangulation. From Proposition 3.17, it follows that  $\varepsilon(\mathcal{T}_{K1}^{ij}) = 1/4ij = \varepsilon$  and  $|\mathcal{T}_{K1}^{ij}| = 2ij$ . As a result, we can prove the proclaimed approximation quality by using the lower bound  $\lceil 1/2\sqrt{5}\varepsilon \rceil$  from Proposition 3.5:

$$\frac{|\mathcal{T}_{K1}^{ij}|}{\lceil \frac{1}{2\sqrt{5}\varepsilon} \rceil} = \frac{2ij}{\lceil \frac{1}{2\sqrt{5}\varepsilon} \rceil} \leq \frac{2ij}{\frac{1}{2\sqrt{5}\varepsilon}} = \sqrt{5}.$$

□

Together with Theorem 3.13 this means that generalized K1-triangulations need twice the number of simplices than crossing swords triangulations to fulfil the same approximation accuracy. This observation is visualized in Figure 6(d), where we show how to convert a J1-triangulation with 32 simplices into a crossing swords triangulation of 16 simplices that has the same approximation accuracy by removing certain edges.

3.3.1. *Approximation quality of iterative refinement schemes starting with  $\mathcal{T}_2^*$ .* In the following, we show approximation qualities for triangulations generated by refinement schemes. A refinement scheme specifies how the simplices of a given triangulation are subdivided into smaller simplices. In the context of pwl. approximations, a refinement is performed to reduce the approximation error. In the following, we always assume that the initial triangulation of  $U$  is  $\mathcal{T}_2^*$ . Subsequently, we perform iterative refinements on all simplices where the approximation error is greater than an a prescribed accuracy  $\varepsilon > 0$  and repeat the refinements until we end up with an  $\varepsilon$ -triangulation.

**Longest-edge bisection.** The *longest-edge bisection* subdivides a simplex by adding a new vertex at the centre of its longest-edge and connects it with the opposite vertex, see Figure 7. Algorithm 3 in the appendix gives a detailed version of this procedure. We define  $\mathcal{T}_L^i$  as the triangulation resulting from the  $i$ -th longest-edge refinement of  $\mathcal{T}_L^0 := \mathcal{T}_2^*$ . This strategy also results in a  $\sqrt{5}/2$ -approximation algorithm for specific values of  $\varepsilon$ .

**Theorem 3.19.** *Let  $\varepsilon := 1/2^{i+3}$  for some odd  $i \in \mathbb{N}$ . Taking the triangulation  $\mathcal{T}_L^i$  resulting from the  $i$ -th refinement of  $\mathcal{T}_L^0$  by a longest-edge bisection yields a  $\sqrt{5}/2$ -approximation algorithm for Problem (2).*

*Proof.* The error on an edge  $e$  with endpoints  $(x_0, y_0)$  and  $(x_1, y_1)$  is always attained at its midpoint and has a value of  $1/4|(x_1 - x_0)(y_1 - y_0)|$ . If  $e$  is now bisected, the approximation error on the resulting edges reduces by a factor of  $1/4$ . As we start with  $\mathcal{T}_L^0 = \mathcal{T}_2^*$ , the approximation error in both initial simplices is attained at the midpoint of the diagonal edge and is  $1/4$ . The longest-edge bisection is now performed on both simplices at the midpoint of the diagonal, see Figure 7(b). The resulting four simplices each have one of the outer edges of  $U$  as their longest edge, with a constant approximation error of 0 along that edge. However, the maximum approximation error on each simplex is on one of the new diagonal edges and has a value of  $1/16$ . If we now perform a longest-edge bisection again, we bisect exclusively on axis-parallel edges. This is because adding a new node on an axis-parallel edge, along which the pwl. approximation corresponds to  $F$  anyway, leaves the pwl. approximation unchanged. The resulting eight simplices shown Figure 7(c) therefore have the same approximation error as the four from the previous iteration. In this way, we continue the refinement and obtain an increase in the number of simplices by a factor of two in each refinement iteration, while the approximation error is reduced by a factor of  $1/4$  in the odd refinement iterations only. This leads to the fact that the triangulation  $\mathcal{T}_L^i$  from the  $i$ -th longest-edge refinement consists of  $2^{i+1}$  simplices and has an approximation error of  $1/2^{i+3}$ . Therefore, we obtain the proclaimed approximation guarantee of  $\sqrt{5}/2$  only for the resulting triangulation after odd iterations of the longest-edge refinement strategy:

$$\frac{|\mathcal{T}_L^i|}{\left\lceil \frac{1}{2\sqrt{5}\varepsilon} \right\rceil} = \frac{2^{i+1}}{\left\lceil \frac{2^{i+3}}{2\sqrt{5}} \right\rceil} \leq \frac{\sqrt{5}}{2}. \quad \square$$

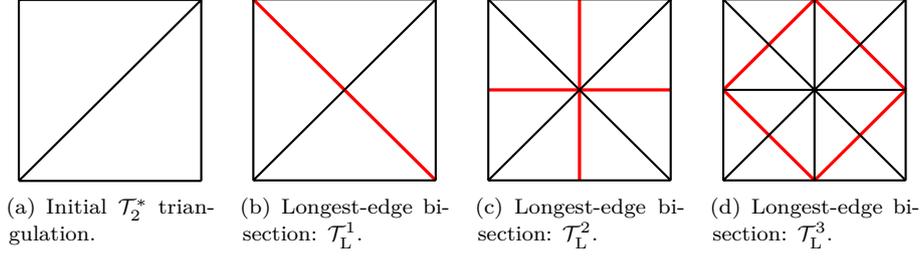


FIGURE 7. Longest-edge bisection.

**Red refinement** The *red refinement* procedure subdivides a simplex into four smaller simplices by adding a new vertex at the centre of each edge and new edges that connect these new vertices, see Figure 8. Algorithm 4 in the appendix gives a detailed version of this procedure. We define  $\mathcal{T}_R^i$  as the triangulation resulting from the  $i$ -th red refinement of  $\mathcal{T}_R^0 := \mathcal{T}_2^*$ . This strategy results in a  $\sqrt{5}$ -approximation algorithm for specific values of  $\varepsilon$ .

**Theorem 3.20.** *Let  $\varepsilon := 1/2^{2i}$  for some  $i \in \mathbb{N}_0$ . Taking the triangulation  $\mathcal{T}_R^i$  resulting from the  $i$ -th red refinement of  $\mathcal{T}_R^0 := \mathcal{T}_2^*$  yields an  $\sqrt{5}$ -approximation algorithm for problem (2).*

*Proof.* According to Lemma 3.1, the error on an edge  $e$  with endpoints  $(x_0, y_0)$  and  $(x_1, y_1)$  is attained at its midpoint and has a value of  $1/4|(x_1 - x_0)(y_1 - y_0)|$ . The red refinement strategy refines a simplex into four simplices by bisecting all edges of the initial simplex. Therefore, the approximation error reduces by a factor of  $1/4$ . As we start with  $\mathcal{T}_R^0 = \mathcal{T}_2^*$ , the approximation error in both initial simplices is attained at the midpoint of the diagonal edge and is  $1/4$ . The red refinement is then performed on both simplices at the midpoints of the respective edges. The resulting eight simplices each have an approximation error of  $1/16$ , and the triangulation can be viewed as a union of four identical  $\mathcal{T}_2^*$ -triangulations. In this pattern, we increase the number of simplices by a factor of four while the approximation error is reduced by a factor of  $1/4$  in each iteration. Consequently, the triangulation  $\mathcal{T}_R^i$  from the  $i$ -th red refinement consists of  $4^{i+1/2}$  simplices and has an approximation error of  $1/2^{2i}$ . Therefore, we obtain the proclaimed approximation guarantee of  $\sqrt{5}$  for the red refinement scheme:

$$\frac{|\mathcal{T}_R^i|}{\left\lceil \frac{1}{2\sqrt{5}\varepsilon} \right\rceil} = \frac{2^{2i+1}}{\left\lceil \frac{2^{2i}}{2\sqrt{5}} \right\rceil} \leq \sqrt{5}. \quad \square$$

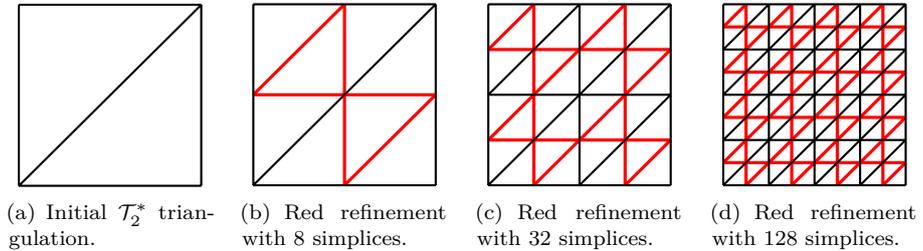


FIGURE 8. Red refinement.

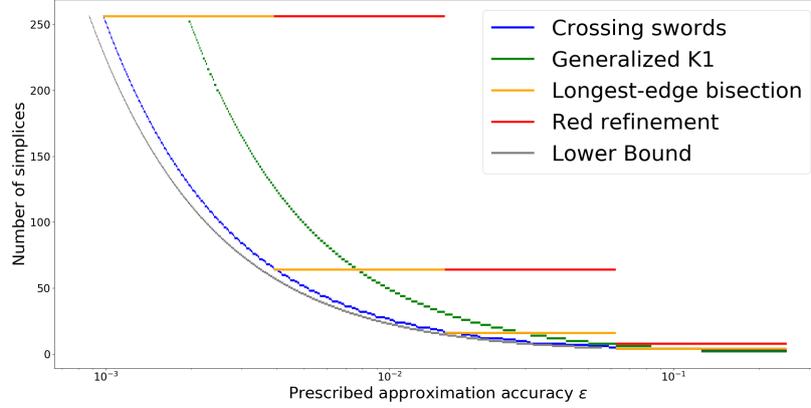


FIGURE 9. Comparison of triangulation scheme in terms of the number of simplices for prescribed approximation accuracies.

**3.4. Comparison of the different triangulation schemes.** We close this section with a comparative discussion of the presented triangulation schemes and provide exemplary numerical results on the number of simplices for several indicative approximation accuracies.

The following corollaries show that for arbitrary approximation accuracies  $\varepsilon > 0$  crossing swords triangulations always use at most as many simplices as the known methods from the literature. However, in most cases, they outperform them clearly. In Figure 3.4, these results are graphically represented as well. We see that crossing swords stays very close to the lower bound and due to its fine adjustability dominates the other triangulations most of the time.

**Corollary 3.21** (Crossing swords dominates K1-triangulations). *For any  $\varepsilon > 0$ , the crossing swords triangulation scheme produces triangulations that have at most the number of simplices than a generalized K1-triangulation. For  $\varepsilon < 1/4$ , the number is strictly lower.*

*Proof.* By Proposition 3.17, the discrete approximation errors attained exactly by the generalized K1-triangulations are all  $\varepsilon_{ij} := 1/4ij$  with  $i, j \in \mathbb{N}$ . For these approximation errors, we know that generalized K1 is an  $\sqrt{5}$ -approximation algorithm and that crossing swords is an  $\sqrt{5}(1/2 + 4\varepsilon_{ij})$ -approximation algorithm. For  $i, j = 1$ , i.e.  $\varepsilon_{1,1} = 1/4$ , both schemes produce equivalent triangulations consisting of two simplices. For  $i = 1$  and  $j = 2$  or  $i = 2$  and  $j = 1$ , i.e.  $\varepsilon_{1,2} = \varepsilon_{2,1} = 1/8$ , the generalized K1-triangulation produces four simplices, while crossing swords returns only three simplices. For all other values of  $i$  and  $j$ ,  $\varepsilon_{i,j} < 1/8$  holds, and therefore crossing swords has a guaranteed approximation quality of less than  $\sqrt{5}$ .  $\square$

**Corollary 3.22** (Crossing swords dominates longest-edge bisection). *For any  $\varepsilon > 0$ , the crossing swords triangulation scheme produces triangulations that have at most the number of simplices as a triangulation constructed by longest-edge bisection.*

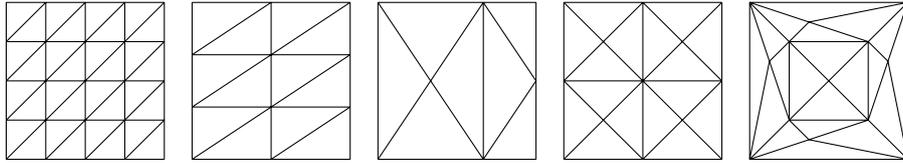
*Proof.* By Theorem 3.19, longest-edge bisection has an approximation quality of  $\sqrt{5}/2$  for all  $\varepsilon_i := 1/2^{i+3}$  with  $i \in \mathbb{N}$ . These values are a subset of the approximation errors  $\varepsilon_j := 1/16j$ , for which the crossing swords triangulation also has an approximation quality of  $\sqrt{5}/2$ . Further, for all  $\varepsilon$  with  $1/16 \cdot 2^i - 1 < \varepsilon < 1/16 \cdot 2^{i-1}$  and  $i \in \mathbb{N}$  crossing swords produces strictly less simplices.  $\square$

With a similar proof as for Corollary 3.21, we also obtain that crossing swords outperforms red-refinement.

**Corollary 3.23** (Crossing swords dominates red-refinement). *For any  $\varepsilon > 0$ , the crossing swords triangulation scheme produces triangulations that have at most the number of simplices as a triangulation constructed by longest-edge bisection. For  $\varepsilon < 1/4$ , the number is strictly lower.*

TABLE 1. Efficiencies of triangulations for  $D = [0, 6] \times [0, 2]$ .  
(\* ) General pwl. approximations (not an interpolation).

Triangulation	$\varepsilon$	$ \mathcal{T} $	$\varepsilon_{f,F}(\mathcal{T})$	Triangulation	$\varepsilon$	$ \mathcal{T} $	$\varepsilon_{f,F}(\mathcal{T})$
<b>Lower bound</b>	1.00	3		<b>Longest-edge bisection</b> [8] (starting with $\mathcal{T}_2^*$ )	1.00	4	0.7500
	0.5	6			0.5	16	0.1875
	0.25	11			0.25	16	0.1875
	0.1	27			0.1	64	0.0469
	0.05	54			0.05	64	0.0469
<b>Crossing sword</b>	1.00	3	1.0000	<b>Max. error</b> [12] (starting with $\mathcal{T}_2^*$ )	1.00	4	0.7500
	0.5	7	0.5000		0.5	20	0.2344
	0.25	12	0.2500		0.25	20	0.2344
	0.1	31	0.1000		0.1	76	0.0615
	0.05	60	0.0500		0.05	116	0.0490
<b>Generalized K1</b>	1.00	8	0.7500	<b>Rebennack et al.</b> [19] (*) (starting with $\mathcal{T}_2^*$ )	1.00	4	0.7500
	0.5	12	0.5000		0.5	12	0.4444
	0.25	24	0.2500		0.25	20	0.2344
	0.1	60	0.1000		0.1	59	0.0968
	0.05	120	0.0500		0.05	94	0.0490
<b>Red refinement</b> [8] (starting with $\mathcal{T}_2^*$ )	1.00	8	0.7500				
	0.5	32	0.1875				
	0.25	32	0.1875				
	0.1	128	0.0469				
	0.05	128	0.0469				



(a) Red refinement with 32 simplices. (b) Generalized K1-triangulation with 12 simplices. (c) Crossing swords with 7 simplices. (d) Longest-edge bisection with 16 simplices. (e) Maximum error bisection with 20 simplices.

FIGURE 10. Triangulations for a prescribed accuracy of  $\varepsilon = 0.5$ .

In addition to these theoretical proofs, we also give a numerical example that underlines the presented properties. We consider the domain  $D = [0, 2] \times [2, 6]$  and approximation accuracies of  $\{1, 0.5, 0.25, 0.1, 0.05\}$ . The results are summarized in Table 1. Here we list the prescribed approximation accuracy  $\varepsilon$ , the number of simplices and the exact approximation error of the respective triangulation. The stated results underline the efficiency and adaptability of the crossing swords triangulation scheme. For the finest approximation accuracy  $\varepsilon = 0.05$ , the crossing swords scheme

produces a triangulation with only 60 simplices, while the other schemes need a factor of at least 1.5 times more simplices. The next best triangulation scheme is that by [19], which is based on a maximum error bisection refinement strategy. However, they consider general pwl. approximations instead of interpolations. Thus, since they solve a generalization of problem (2), the results are not directly comparable. Nevertheless, the crossing swords scheme seems to provide more efficient triangulations even for this generalized problem. Crossing swords as well as generalized K1 are the only schemes which are able to fulfil the given accuracies exactly. However, the number of simplices in generalized K1-triangulations grows much faster than in crossing swords triangulations with increasing accuracy. For  $\varepsilon = 0.05$ , K1 requires even factor twice as many simplices as crossing swords. The poor adaptability to the prescribed approximation error exhibited by the refinement methods, i.e. red-refinement, longest-edge bisection and max-error refinement, can be seen from the fact that in order to satisfy an accuracy of  $\varepsilon = 0.5$ , they produce triangulations that have actual approximation errors of 0.1875 and 0.2344 respectively. To visualize the observed results, the triangulations of the considered triangulation schemes for an accuracy of  $\varepsilon = 0.5$  are shown in Figure 10(a). Besides the mere number of simplices, it is especially noticeable that the maximum error scheme produces relatively skinny simplices, which can lead to numerical problems in the evaluation of the approximation for small errors.

#### 4. CONCLUSION

In this work, we have introduced a novel triangulation scheme called crossing swords to approximate bivariate products  $xy$  over rectangular domains. We showed that a crossing swords triangulation requires at most  $\sqrt{5}/2$  times as many simplices as an  $\varepsilon$ -optimal triangulation for any approximation accuracy  $\varepsilon > 0$ . Crossing swords thus outperforms all previously known triangulation schemes in the literature, which we have proved theoretically and also underlined by exemplary numerical results. We have also proved that crossing swords triangulations are  $\varepsilon_i$ -optimal under the condition that each simplex has one axis-parallel edge and  $\varepsilon_i = 1/16^i$  with  $i \in \mathbb{N}$  holds. Supported by computational tests, we conjecture that this also holds if the simplices are not required to be axis-parallel.

#### ACKNOWLEDGEMENTS

This research was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the Center for Analytics – Data – Applications (ADA-Center) within the framework of “BAYERN DIGITAL II” (20-3410-2-9-8).

#### REFERENCES

- [1] L. Alkhalifa and H. Mittelman. “New Algorithm to Solve Mixed Integer Quadratically Constrained Quadratic Programming Problems Using Piecewise Linear Approximation.” In: *Mathematics* 10.2, 198 (2022).
- [2] D. Atariah, G. Rote, and M. Wintraecken. “Optimal triangulation of saddle surfaces.” In: *Beiträge zur Algebra und Geometrie / Contributions to Algebra and Geometry* 59.1 (2018), pp. 113–126.
- [3] F. Aurenhammer and Y.-F. Xu. “Optimal triangulations.” In: *Encyclopedia of Optimization*. Springer, 2008, pp. 2757–2764.
- [4] R. E. Bank, A. H. Sherman, and A. Weiser. “Some refinement algorithms and data structures for regular local mesh refinement.” In: *Scientific Computing, Applications of Mathematics and Computing to the Physical Sciences* 1 (1983), pp. 3–17.

- [5] A. Bärmann, R. Burlacu, L. Hager, and T. Kleinert. “On Piecewise Linear Approximations of Bilinear Terms: Structural Comparison of Univariate and Bivariate Mixed-Integer Programming Formulations.” In: (2021).
- [6] B. Beach, R. Hildebrand, and J. Huchette. *Compact mixed-integer programming relaxations in quadratic optimization*. arXiv preprint arXiv:2011.08823. 2020.
- [7] O. Beaumont, V. Boudet, F. Rastello, Y. Robert, et al. “Partitioning a square into rectangles: NP-completeness and approximation algorithms.” In: *Algorithmica* 34.3 (2002), pp. 217–239.
- [8] R. Burlacu. “On refinement strategies for solving MINLPs by piecewise linear relaxations: a generalized red refinement.” In: *Optimization Letters* 16 (2021), pp. 635–652.
- [9] R. Burlacu. “Adaptive Mixed-Integer Refinements for Solving Nonlinear Problems with Discrete Decisions.” PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2020.
- [10] A. Cohen, N. Dyn, F. Hecht, and J.-M. Mirebeau. “Adaptive multiresolution analysis based on anisotropic triangulations.” In: *Mathematics of Computation* 81 (2012), pp. 789–810.
- [11] A. Fügenschuh, K. Junosza-Szaniawski, and Z. Lonc. “Exact and approximation algorithms for a soft rectangle packing problem.” In: *Optimization* 63.11 (2014), pp. 1637–1663.
- [12] B. Geißler. “Towards Globally Optimal Solutions for MINLPs by Discretization Techniques with Applications in Gas Network Optimization.” PhD thesis. July 2011.
- [13] B. Geißler, A. Martin, A. Morsi, and L. Schewe. “Using piecewise linear functions for solving MINLPs.” In: *Mixed integer nonlinear programming*. Springer, 2012, pp. 287–314.
- [14] K. Kutzer. “Using Piecewise Linear Approximation Techniques to Handle Bilinear Constraints.” PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2020.
- [15] C. Lu, S.-C. Fang, Q. Jin, Z. Wang, and W. Xing. “KKT solution and conic relaxation for solving quadratically constrained quadratic programming problems.” In: *SIAM Journal on Optimization* 21.4 (2011), pp. 1475–1490.
- [16] P. Monsky. “On dividing a square into triangles.” In: *The American Mathematical Monthly* 77.2 (1970), pp. 161–164.
- [17] A. Morsi, B. Geißler, and A. Martin. “Mixed Integer Optimization of Water Supply Networks.” In: *Mathematical Optimization of Water Networks*. Vol. 162. Springer, 2012, pp. 35–54.
- [18] H. Pottmann, R. Krasauskas, B. Hamann, K. Joy, and W. Seibold. “On piecewise linear approximation of quadratic functions.” In: *Journal for Geometry and Graphics* 4.1 (2000), pp. 31–53.
- [19] S. Rebennack and J. Kallrath. “Continuous piecewise linear delta-approximations for bivariate and multivariate functions.” In: *Journal of Optimization Theory and Applications* 167.1 (2015), pp. 102–117.
- [20] M. J. Todd. “Hamiltonian triangulations of  $\mathbb{R}^n$ .” In: *Functional Differential Equations and Approximation of Fixed Points*. Ed. by H.-O. Peitgen and H.-O. Walther. Berlin, Heidelberg: Springer Berlin Heidelberg, 1979, pp. 470–483.
- [21] J. P. Vielma, S. Ahmed, and G. Nemhauser. “Mixed-Integer Models for Nonseparable Piecewise-Linear Optimization: Unifying Framework and Extensions.” In: *Operations Research* 58.2 (2010), pp. 303–315.
- [22] A. Zelmer. “Designing Coupled Energy Carrier Networks By Mixed-Integer Programming Methods.” PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2010.

APPENDIX A. CROSSING SWORD TRIANGULATION ALIGNED ALONG THE  $x$ -AXIS

Note that in the general version of Algorithm 1, only the areas of the rectangle partitioning were specified. We now provide a version of this algorithm that describes a concrete rectangle partitioning. To this end, we arrange the rectangles along the  $x$ -axis and assume that they all have a height of one.

---

**Algorithm 2:** Crossing Sword Triangulation Scheme (for the unit box  $U$  and rectangles aligned along the  $x$ -axis)

---

**Input:** The number of simplices  $N \geq 4$ .

**Output:** A triangulation  $\mathcal{T}_{cs}^N$  of  $U$  with  $|\mathcal{T}_{cs}^N| = N$  and a corresponding pwl. approximation  $f$  of  $F$  over  $U$  with an approximation error of

$$\varepsilon_{f,F}(\mathcal{T}_{cs}^N) = \begin{cases} \frac{1}{4} \frac{1}{N}, & \text{if } N \equiv 0 \pmod{4}, \\ \frac{1}{4} \frac{(\sqrt{5}-2)}{(\sqrt{5}-2)(N-5)+1}, & \text{if } N \equiv 1 \pmod{4}, \\ \frac{1}{4} \frac{1}{N-1}, & \text{if } N \equiv 2 \pmod{4}, \\ \frac{1}{4} \frac{1}{N-1}, & \text{if } N \equiv 3 \pmod{4}. \end{cases} \quad (16)$$

```

1 if  $N \equiv 0 \pmod{4}$  then
2   Set  $n = \frac{N}{4}$  and  $h = \frac{1}{n}$ 
3   for  $i = 1, \dots, n$  : do
4      $U_i := [(i-1)h, ih] \times [0, 1]$ 
5   return  $\mathcal{T}_{cs}^N := \bigcup_{i=1}^n \mathcal{T}_4^*(U_i)$ 
6 else if  $N \equiv 1 \pmod{4}$  then
7   Set  $n = \frac{N-1}{4}$  and  $h = \frac{4(\sqrt{5}-2)}{(\sqrt{5}-2)(N-5)+1}$ 
8   for  $i = 1, \dots, n-1$  : do
9      $U_i := [(i-1)h, ih] \times [0, 1]$ 
10   $U_n := [(n-1)h, 1] \times [0, 1]$ 
11  return  $\mathcal{T}_{cs}^N := \bigcup_{i=1}^{n-1} \mathcal{T}_4^*(U_i) \cup \mathcal{T}_5^*(U_n)$ 
12 else if  $N \equiv 2 \pmod{4}$  then
13  Set  $n = \frac{N-2}{4} + 1$  and  $h = \frac{4}{N-1}$ 
14  for  $i = 1, \dots, n-1$  : do
15     $U_i := [(i-1)h, ih] \times [0, 1]$ 
16   $U_n := [\frac{N-2}{N-1}, 1] \times [0, 1]$ 
17  return  $\mathcal{T}_{cs}^N := \bigcup_{i=1}^{n-1} \mathcal{T}_4^*(U_i) \cup \mathcal{T}_2^*(U_n)$ 
18 else if  $N \equiv 3 \pmod{4}$  then
19  Set  $n = \frac{N-3}{4} + 1$  and  $h = \frac{4}{N-1}$ 
20  for  $i = 1, \dots, n-1$  : do
21     $U_i := [(i-1)h, ih] \times [0, 1]$ 
22   $U_n := [\frac{N-3}{N-2}, 1] \times [0, 1]$ 
23  return  $\mathcal{T}_{cs}^N := \bigcup_{i=1}^{n-1} \mathcal{T}_4^*(U_i) \cup \mathcal{T}_3^*(U_n)$ 

```

---

**Lemma A.1** (Crossing Sword Triangulations). *Algorithm 2 is correct. This means that for any input  $N$ , Algorithm 2 returns a triangulation with an approximation error according to Equation (16).*

*Proof.* To show the correctness of Algorithm 2, it suffices due to Lemma 3.10 to show that the smaller rectangles  $U_1, \dots, U_n$  have the required areas from Algorithm 1. As the height ( $y$ -width) of each rectangle constructed in Algorithm 2 is one, the

area of each rectangle equals its  $x$ -width. As the  $x$ -width of the rectangles used in Algorithm 2 is chosen to be equal to the area of the corresponding rectangle, its correctness is certified.  $\square$

## APPENDIX B. REFINEMENT STRATEGIES

In this section, we give the exact refinement routines used in the longest-edge bisection and red refinement procedures for refining simplices.

The longest edge refinement takes a simplex  $S$  as input and returns two new simplices  $S'$  and  $S''$ , which come from bisecting  $S$  at its longest edge.

---

### Algorithm 3: Longest-edge bisection refinement

---

**Input:** A simplex  $S \subset \mathbb{R}^2$ , a pwl. approximation  $g: S \rightarrow \mathbb{R}$  of a non-linear function  $G: S \rightarrow \mathbb{R}$  and an approximation accuracy  $\varepsilon > 0$ .

**Output:** If the approximation error over  $S$  is greater than  $\varepsilon$ , a set of two simplices  $\{S', S''\}$  with  $S = S' \cup S''$  and  $\text{int}(S') \cap \text{int}(S'') = \emptyset$  is returned. Otherwise, no refinement is performed.

```

1 Set  $e \leftarrow$  longest edge of  $S$  with endpoints  $\bar{x}_a, \bar{x}_b \in V(S)$ .
2 if  $\varepsilon_{g,G}(S) \leq \varepsilon$  then
3   return  $V(S)$ 
4 else
5   Set  $\tilde{x} \leftarrow$  midpoint of  $e$ .
6   Set  $V(S') \leftarrow (V(S) \setminus \bar{x}_b) \cup \tilde{x}$ ;  $S' \leftarrow \text{conv}(V(S'))$ 
7   Set  $V(S'') \leftarrow (V(S) \setminus \bar{x}_a) \cup \tilde{x}$ ;  $S'' \leftarrow \text{conv}(V(S''))$ 
8   return  $V(S'), V(S'')$ 

```

---

The red-refinement takes a simplex  $S$  as input and returns four new simplices  $S^1, S^2, S^3$  and  $S^4$ , which come from bisecting  $S$  on each of its edges.

---

### Algorithm 4: Red refinement

---

**Input:** A simplex  $S \in \mathbb{R}^2$  with  $V(S) = \{s_0, s_1, s_2\}$ , a pwl. approximation  $g: S \rightarrow \mathbb{R}$  of a non-linear function  $G: \mathbb{R}^2 \rightarrow \mathbb{R}$  and an approximation accuracy  $\varepsilon$ .

**Output:** If the approximation error over  $S$  is greater than  $\varepsilon$ , a set of four simplices  $\{S^1, S^2, S^3, S^4\}$  with  $S = \bigcup_{i=1}^4 S_i$  and  $\text{int}(S^i) \cap \text{int}(S^j) = \emptyset$  for all  $i \neq j$  is returned. Otherwise, no refinement is performed.

```

1 Set  $e \leftarrow$  longest edge of  $S$  with endpoints  $\bar{x}_a, \bar{x}_b \in V(S)$ .
2 if  $\varepsilon_{g,G}(S) \leq \varepsilon$  then
3   return  $V(S)$ 
4 else
5   Set  $V(S^0) \leftarrow \{s_0, 1/2(s_0 + s_1), 1/2(s_0 + s_2)\}$ ;  $S^0 \leftarrow \text{conv}(V(S^0))$ 
6   Set  $V(S^1) \leftarrow \{1/2(s_0 + s_1), s_1, 1/2(s_1 + s_2)\}$ ;  $S^1 \leftarrow \text{conv}(V(S^1))$ 
7   Set  $V(S^2) \leftarrow \{1/2(s_1 + s_2), 1/2(s_0 + s_1), 1/2(s_0 + s_2)\}$ ;  $S^2 \leftarrow \text{conv}(V(S^2))$ 
8   Set  $V(S^3) \leftarrow \{s_0, 1/2(s_0 + s_2), 1/2(s_1 + s_2), s_2\}$ ;  $S^3 \leftarrow \text{conv}(V(S^3))$ 
9   return  $V(S^0), \dots, V(S^3)$ 

```

---

(A. Bärmann) FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG, DISCRETE OPTIMIZATION, CAUERSTR. 11, 91058 ERLANGEN, GERMANY

*Email address:* andreas.baermann@math.uni-erlangen.de

(R. Burlacu) FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG, DISCRETE OPTIMIZATION, CAUERSTR. 11, 91058 ERLANGEN, GERMANY

*Email address:* robert.burlacu@fau.de

(L. Hager) FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG, DISCRETE OPTIMIZATION, CAUERSTR. 11, 91058 ERLANGEN, GERMANY

*Email address:* `lukas.hager@fau.de`

(K. Kutzer) FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG, DISCRETE OPTIMIZATION, CAUERSTR. 11, 91058 ERLANGEN, GERMANY

*Email address:* `katja.kutzer@fau.de`