Certified Approximation Algorithms for the Fermat Point

Kolja Junginger[1], Ioannis Mantas[1], Evanthia Papadopoulou[1], Martin Suderland[1], and Chee Yap[2]

1 **Faculty of Informatics, Università della Svizzera italiana, Lugano, Switzerland**
`kolja.junginger@usi.ch,ioannis.mantas@usi.ch,`
`evanthia.papadopoulou@usi.ch,martin.suderland@usi.ch`

2 **Courant Institute, NYU, New York, NY, USA**
`yap@cs.nyu.edu`

Abstract

Given a set of k points $A \subseteq \mathbb{R}^d$ together with a positive weight function $w : A \to \mathbb{R}_{>0}$, the *Fermat distance function* is $\varphi(\boldsymbol{x}) = \sum_{\boldsymbol{a} \in A} w(\boldsymbol{a}) \|\boldsymbol{x} - \boldsymbol{a}\|$. A classic problem in facility location is finding the *Fermat point* $\boldsymbol{x}^*$, the point that minimizes the function φ. We present algorithms to compute an ε-approximation of the Fermat point $\boldsymbol{x}^*$, that is, a point $\widetilde{\boldsymbol{x}}^*$ satisfying $\|\widetilde{\boldsymbol{x}}^* - \boldsymbol{x}^*\| < \varepsilon$. Our algorithms are based on the *subdivision paradigm*, which we combine with Newton methods, used for speed-ups and certification. Our algorithms are certified in the sense of interval methods.

1 Introduction

A classic problem in Facility Location [14, 29] is the placement of a facility to serve a given set of demand points or customers so that the total transportation costs are minimized. The total cost at any point is interpreted as the sum of the distances to the demand points. The point that minimizes this sum is called the *Fermat Point*, see Fig. 1.

A *weighted foci set* is a non-empty finite set of (demand) points $A = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k\} \subseteq \mathbb{R}^d$ associated with a positive weight function $w : A \to \mathbb{R}_{>0}$. Each $\boldsymbol{a} \in A$ is called a *focus* with weight $w(\boldsymbol{a})$. Let $W = \sum_{\boldsymbol{a} \in A} w(\boldsymbol{a})$. The *Fermat distance function* of A is given by

$$\varphi(\boldsymbol{x}) := \sum_{\boldsymbol{a} \in A} w(\boldsymbol{a}) \|\boldsymbol{x} - \boldsymbol{a}\|,$$

where $\|\boldsymbol{x}\|$ is the Euclidean norm in $\mathbb{R}^d$. The global minimum value of φ is called the *Fermat radius* of A and denoted r^*; any point $\boldsymbol{x} \in \mathbb{R}^d$ that achieves this minimum, $\varphi(\boldsymbol{x}) = r^*$, is called a *Fermat point* and denoted $\boldsymbol{x}^* = \boldsymbol{x}^*(A)$. If A is not collinear then $\boldsymbol{x}^*$ is unique [22, 24]. We also consider the closely related problem of computing *k-ellipses* of A: for any $r > r^*(A)$, the k-ellipsoid of A of radius r is the level set $\varphi^{-1}(r) := \{\boldsymbol{x} \in \mathbb{R}^d : \varphi(\boldsymbol{x}) = r\}$. When $d = 2$, they are called k-ellipses motivated by classical ellipses being 2-ellipses. Figure 1 shows some 28-ellipses with different radii computed by our algorithm.

The question of approximating the Fermat point is of great interest as its coordinates are the solution of a polynomial with exponentially high degree [3, 28]. An ε-approximation $\widetilde{\boldsymbol{x}}^*$ to the Fermat point $\boldsymbol{x}^*$ can be interpreted in 3 senses: (A) $\|\widetilde{\boldsymbol{x}}^* - \boldsymbol{x}^*\| \leq \varepsilon$, (B) $\varphi(\widetilde{\boldsymbol{x}}^*) \leq \varphi(\boldsymbol{x}^*) + \varepsilon$, and (C) $\varphi(\widetilde{\boldsymbol{x}}^*) \leq (1 + \varepsilon)\varphi(\boldsymbol{x}^*)$. In this paper, we consider approximations in the sense (A) which are stronger than senses (B) and (C). To the best of our knowledge, only senses (B) and (C) have been studied so far; they are actually approximations of the Fermat radius.

There is a plethora of results for the Fermat point including, among others, approximations algorithms [2, 6, 9, 11, 15, 18, 30, 38] and special configurations and other variants [1, 5, 8, 10, 13, 14]. A smaller, but equally old, literature also exists for the k-ellipses [25, 28, 33, 35].

36th European Workshop on Computational Geometry, Würzburg, Germany, March 16–18, 2020.
This is an extended abstract of a presentation given at EuroCG'20. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

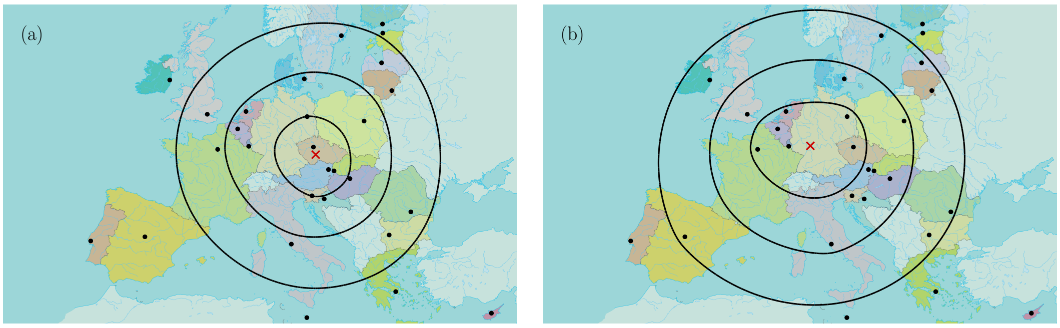

Figure 1 A set of 28 foci, corresponding to EU capitals, with the Fermat point (x) and three 28-ellipses of different radius. (a) Unweighted. (b) The weight of a focus is the population of that country.

In this work we introduce certified algorithms for approximating the Fermat point, combining a subdivision approach with interval methods, cf. [21, 31]. The approach can be formalized in the framework of "soft predicates" [36]. Our certified algorithms are fairly easy to implement, and are shown to have good performance experimentally.

Our Contributions can be summarized as follows: **(1)** We introduce a *box subdivision* scheme to compute an ε-approximation of the Fermat point. **(2)** We augment the *Weiszfeld point sequence* [37] with a Newton-operator test, that outputs an ε-approximation of the Fermat point. **(3)** We implement and experimentally evaluate our algorithms with various datasets. In the full paper, we also address the problem of computing k-ellipses.

2 Preliminaries

Let $\square\mathbb{R}^d$ (or simply $\square\mathbb{R}$ when $d = 1$) denote the set of closed d-dimensional boxes (i.e. Cartesian products of intervals) in $\mathbb{R}^d$. Our subdivision algorithms start with an initial box $B_0 \in \square\mathbb{R}^d$ and recursively split it. We organize the boxes in a *hyper-octree* data structure [32]. A box is specified by an interval in each dimension $B = I_1 \times \ldots \times I_d$. We denote: m_B the *center* of B, r_B the *radius* of B (distance between m_B and a corner), $\omega(B)$ the *width* of B (the maximum length of its defining intervals), and $c \cdot B$ the box with center m_B and the length of each I_i scaled by c. Operation SPLIT takes each interval of B and splits it in the middle. It returns 2^d *children* of B. Operation SPLIT$_2$ applies SPLIT to all of B's children and returns the 2^{2d} *grandchildren* of B. We maintain the subdivision *smooth*, i.e., the depth of any two boxes that have overlapping faces differs by a depth of at most 1. Maintaining smoothness does not increase the asymptotic costs [4].

Let P be a logical *predicate* on boxes, i.e. $P : \square\mathbb{R}^d \to \{\texttt{true}, \texttt{false}\}$. A test T looks like a predicate: $T : \square\mathbb{R}^d \to \{\texttt{success}, \texttt{failure}\}$ and it is always associated to some predicate P: Call T a *test for predicate* P if $T(B) = \texttt{success}$ implies $P(B) = \texttt{true}$. However, we conclude nothing if $T(B) = \texttt{failure}$.

▶ **Definition 2.1.** Let T be a test for a predicate P. We call T a *soft predicate* (or *soft version* of P) if it is convergent in this sense: if $(B_n)_{n\in\mathbb{N}}$ is a monotone sequence of boxes $B_{n+1} \subseteq B_n$ that converges to a point $\boldsymbol{a}$, then $P(\boldsymbol{a}) = T(B_n)$ for n large enough.

$\square P(B)$ denotes a soft version of $P(B)$. We construct soft predicates using functions of the form $F\colon \square\mathbb{R}^d \to \square(\mathbb{R} \cup \{-\infty, \infty\})$ approximating a scalar function $f\colon D \to \mathbb{R}$ with $D \subset \mathbb{R}^d$.

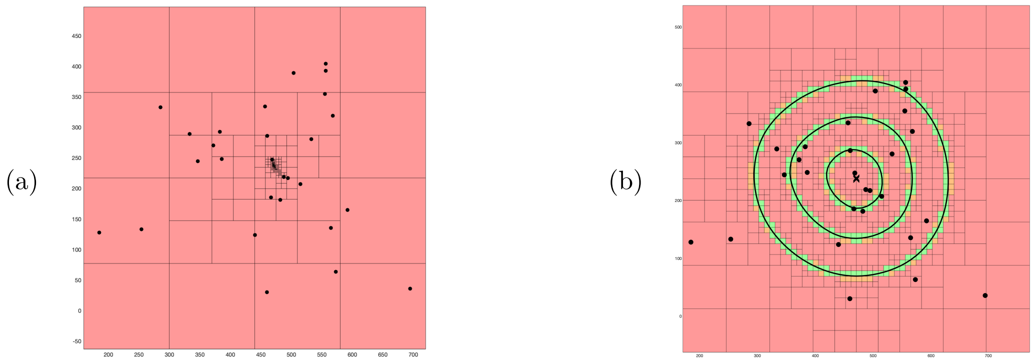

Figure 2 The resulting box subdivision for (a) the Fermat point and (b) the k-ellipses of Fig. 1a.

▶ **Definition 2.2.** Call F a *soft version* of f if it is

(i) *conservative*, i.e. for all $B \in \square\mathbb{R}^d$, $F(B)$ contains $f(B) := \{f(p) : p \in B \cap D\}$, and
(ii) *convergent*, i.e. if for monotone sequence $(B_n)_{n\in\mathbb{N}}$ that converges to a point $\boldsymbol{a} \in D$, $\lim_{n\to\infty} \omega(F(B_n)) = 0$ holds.

We denote F by $\square f$ when F is a soft version of f. There are many ways to achieve $\square f$. E.g. if f has an arithmetic expression E, we can simply evaluate E using interval arithmetic.

3 Approximate Fermat points

We assume that the Fermat point is unique and unequal to a focus. These assumptions are mild and easy to ensure. A focus $\boldsymbol{a} \in A$ is the Fermat point of A if and only if $\left\|\nabla\varphi_{A\setminus\boldsymbol{a}}(\boldsymbol{a})\right\| \le w(\boldsymbol{a})$ [37]. So, a simple $O(k^2d)$ preprocessing time suffices to verify this assumption. In both subdivision algorithms presented below the time can be reduced to $O(kd)$. The Fermat point problem reduces to computing the critical point of the gradient of φ. We now present three approximation algorithms for the Fermat point $\boldsymbol{x}^*$.

3.1 Using the Subdivision Paradigm

This paradigm requires an initial box B_0. If B_0 is not given, it is easy to find a box containing $\boldsymbol{x}^*$, as $\boldsymbol{x}^*$ lies in the convex hull of A [20]. Function Initial-Box(A), in $O(k)$ time, returns an axis-aligned box with the corners of minimum and maximum x, y coordinates.

▶ **Definition 3.1.** Given a box B, the *gradient exclusion predicate* $C^\nabla(B)$ returns true if and only if $\mathbf{0} \notin \nabla\varphi(B)$. If we replace $\nabla\varphi(B)$ by its interval form $\square\nabla\varphi(B)$ see Section 4, we obtain the corresponding interval predicate "$\square C^\nabla(B)$".

In our algorithms we keep on splitting boxes using different kinds of predicates while we exclude boxes (red in Fig. 3) that are guaranteed not to contain $\boldsymbol{x}^*$ and we keep boxes (green in Fig. 3) that might contain $\boldsymbol{x}^*$. We test whether we can already approximate $\boldsymbol{x}^*$ well enough by putting a bounding box around all (green) boxes, which we have not excluded yet.

▶ **Definition 3.2.** Given a set of boxes Q, one of which contains the Fermat point, the *stopping predicate* $C^\varepsilon(Q)$ returns true, if and only if the minimum axis-aligned bounding box containing all boxes in Q has a radius at most ε.

If C^ε returns true, then we can stop. Since the radius of the minimum bounding box is at most ε, the center of the box is an ε-approximation of the Fermat point.

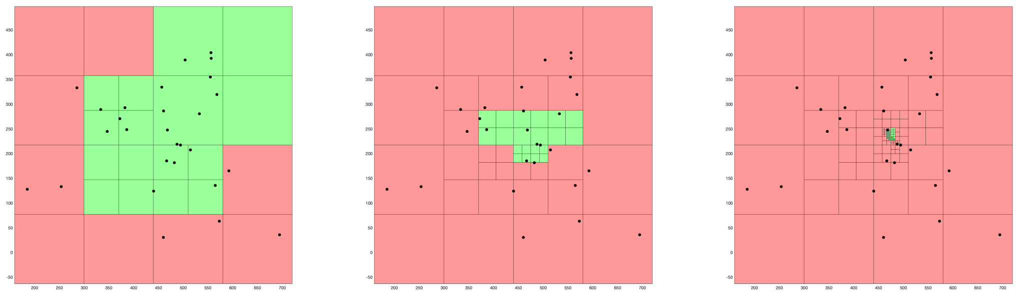

Figure 3 Illustrations of three different steps during the execution of Algorithm 1.

Algorithm 1: Subdivision for Fermat Point (*SUB*)

Input: Foci set A, $\varepsilon > 0$. **Output:** Point $\boldsymbol{p}$.

$B_0 \leftarrow \textsc{Initial-Box}(A)$; $Q \leftarrow \textsc{Queue}()$; $Q.\textsc{push}(B_0)$;
while ***not*** $C^{\varepsilon}(Q)$ **do**
  $B \leftarrow Q.\textsc{pop}()$;
  if ***not*** $\square C^{\nabla}(B)$ **then** $Q.\textsc{push}(\textsc{split}(B))$;
return $\boldsymbol{p} \leftarrow$ *Center of the bounding box of* Q;

3.2 Enhancing the Subdivision Paradigm

Using a Newton inclusion predicate, we know that our algorithm will eventually converge quadratically. Other authors have also considered Newton-type algorithms, but usually independently of other methods, thus lacking global convergence. We integrate subdivision with the Newton operator (an idea based on Dekker [12]), thus ensuring global convergence.

We want to find the Fermat point, i.e. the root of $\boldsymbol{f} = \nabla\varphi$. The Newton-type predicates are well-studied in the interval literature, and they have the form $N : \square\mathbb{R}^d \to \square\mathbb{R}^d$. We use the formula by Moore [23] and Nickel [26]: $N(B) = m_B - J_{\boldsymbol{f}}^{-1}(B) \cdot \boldsymbol{f}(m_B)$, where $J_{\boldsymbol{f}}$ is the Jacobian matrix of $\boldsymbol{f}$. Since $\boldsymbol{f} = \nabla\varphi$, the matrix $J_{\boldsymbol{f}}$ is actually the Hessian of φ. There are better Newton type operators [19, 17, 16] in the sense that they return a smaller box, but they are computationally more expensive.

This Newton box operator has the following properties, stemming from [7, 27, 34]: **(1)** If $N(B) \subset B$ then $\boldsymbol{x}^* \in N(B)$. **(2)** If $\boldsymbol{x}^* \in B$ then $\boldsymbol{x}^* \in N(B)$. **(3)** If $N(B) \cap B = \emptyset$ then $\boldsymbol{x}^* \notin B$.

▶ **Definition 3.3.** Given a box B, the *Newton inclusion predicate* $\square C^N(B)$ returns true if and only if $\square N(2B) \subset 2B$, where $\square N(B) = m_B - \square J_{\boldsymbol{f}}^{-1}(B) \cdot \boldsymbol{f}(m_B)$ see Section 4.

▶ **Theorem 3.4.** *Algorithms 1 and 2 return an ε-approximation of the Fermat point $\boldsymbol{x}^*$.*

Algorithm 2 initially excludes boxes only based on $\square C^{\nabla}$ and splits the other boxes. This subdivision phase takes exponential time in d. Once the test $\square C^N(B)$ succeeds, it will also do so for one of B's grandchildren. The Newton phase needs $O(kd^2)$ time per step and converges quadratically in ε. The time necessary to transition from the subdivision phase to the Newton phase does not depend on ε.

We can reduce the $O(k^2 d)$ preprocessing time (testing if $\boldsymbol{x}^* \in A$) to $O(kd)$ in both subdivision algorithms, by doing this test only when all except a constant number of foci got excluded by $\square C^{\nabla}$. Another speedup comes by preprocessing with a *principal component analysis* and then using rectangular boxes. Experimentally, we observe that it drastically improves the runtime for *near-degenerate* inputs.

Algorithm 2: Enhanced subdivision for Fermat Point (*E-SUB*)

```
Input: Foci set A, ε > 0.        Output: Point p.
B_0 ← INITIAL-BOX(A);   Q ← QUEUE();   Q.PUSH(B_0);
while C^ε(Q) do
    B ← Q.POP();
    if not □C^∇(B) then
        if □C^N(B) then
            Q ← QUEUE();   Q.PUSH(SPLIT_2(N(2B)));
        else Q.PUSH(SPLIT(B));
return p ← Center of the bounding box of Q;
```

3.3 Using a Point Sequence Scheme

Weiszfeld [37] gave an iterative method to compute a sequence of points converging to the Fermat point, later corrected in [20, 29]. This scheme is defined by the following map:

$$T(\boldsymbol{x}) = \begin{cases} \tilde{T}(\boldsymbol{x}) & \text{if } \boldsymbol{x} \notin A \\ \dfrac{\sum_{\boldsymbol{a}\in A, \boldsymbol{a}\neq\boldsymbol{x}} w(\boldsymbol{a}) \frac{\boldsymbol{a}}{\|\boldsymbol{x}-\boldsymbol{a}\|}}{\sum_{\boldsymbol{a}\in A, \boldsymbol{a}\neq\boldsymbol{x}} w(\boldsymbol{a}) \frac{1}{\|\boldsymbol{x}-\boldsymbol{a}\|}} & \text{if } \boldsymbol{x} \in A \end{cases} \quad \text{where} \quad \tilde{T}(\boldsymbol{x}) = \frac{\sum_{i=1}^{k} w(\boldsymbol{a}) \frac{\boldsymbol{a}_i}{\|\boldsymbol{x}-\boldsymbol{a}_i\|}}{\sum_{i=1}^{k} w(\boldsymbol{a}) \frac{1}{\|\boldsymbol{x}-\boldsymbol{a}_i\|}} \tag{1}$$

We augment this by adding a guarantee for the computation, turning it into an ε-approximation algorithm. To verify the quality of a point $\boldsymbol{p}_i$, we use the Newton inclusion predicate. We define a small box B with $\boldsymbol{p}_i$ as center and map it to the new box $\square N(B)$. If $\square N(B) \subseteq B$, we know that $\boldsymbol{x}^*$ lies in $\square N(B)$. If $\square N(B) \nsubseteq B$, we move on to point $\boldsymbol{p}_{i+1}$ and adjust the box size. If there was a focus in box $\frac{B}{10}$, then $\square N(\frac{B}{10})$ covers the whole plane, which hinders $\square N(B) \subseteq B$ to succeed. In that case we shrink the box by a factor of 10. If $\frac{B}{10} \cap \square N(\frac{B}{10}) = \emptyset$, then the box $\frac{B}{10}$ does not contain the $\boldsymbol{x}^*$ and we therefore increase the box size. As a starting point, we take the *center of mass* $\boldsymbol{p}_0$ of A, i.e., $\boldsymbol{p}_0 = \frac{1}{W}\sum_{\boldsymbol{a}\in A} w(\boldsymbol{a})\boldsymbol{a}$, as $\varphi(\boldsymbol{p}_0)$ itself, is a 2-approximation of the Fermat radius [11].

Algorithm 3: Point sequence algorithm (*P-SEQ*)

```
Input: Foci set A, ε > 0.        Output: Point p.
p ← p_0;    w ← ε;
while TRUE do
    B ← Box B(m_B = p, ω(B) = w);    N(B) ← INTERVAL-NEWTON(B);
    if N(B) ⊆ B then return p ;
    else if N(B/10) ∩ B/10 = ∅ then  w ← min{ε, w · 10};
    else w ← w/10;
    p ← T(p)
```

The point sequence in Eq. (1) converges to the Fermat point and when sufficiently close, $\square N(B) \subset B$ holds, hence Algorithm 3 returns an ε-approximation of the Fermat point.

4 Details on the box approximations $\square\varphi$, $\square\nabla\varphi$, $\square\nabla^2\varphi$ and $\square N$

We call $L(B)$ a Lipschitz constant for box B if $\forall \boldsymbol{p}, \boldsymbol{q} \in B: \ |\varphi(\boldsymbol{p}) - \varphi(\boldsymbol{q})| \leq L(B) \cdot \|\boldsymbol{p} - \boldsymbol{q}\|$. We will later choose $L(B)$ smaller than the trivial Lipschitz constant $W = \sum_{\boldsymbol{a}\in A} w(\boldsymbol{a})$.

▶ **Lemma 4.1.** $\square\varphi(B) = [\varphi(m_B) - L(B)\cdot r_B, \varphi(m_B) + L(B)\cdot r_B]$ *is a soft version of* φ.

Proof. The $L(B)$ is a Lipschitz constant of φ on box B, i.e. for all $p \in B$ it holds $|\varphi(p) - \varphi(m_B)| \le L(B)\cdot r_B$, which implies $\varphi(p) \in [\varphi(m_B) - L\cdot r_B, \varphi(m_B) + L\cdot r_B]$.

Let B_n be a sequence of boxes, which converges to a point. Hence $r_{B_n} \to 0$. The Lipschitz constants $L(B_n)$ can be bounded from above by W. Thus, $\omega(\square\varphi(B_n)) \le 2W\cdot r_{B_n} \to 0$. ◀

The following formulas are written for $d = 2$ but clearly generalize to higher dimensions. For any point $\boldsymbol{p} = (\boldsymbol{p}_x, \boldsymbol{p}_y)^T$, let $\sin(\boldsymbol{p}) := \boldsymbol{p}_y/\|\boldsymbol{p}\|$ and $\cos(\boldsymbol{p}) := \boldsymbol{p}_x/\|\boldsymbol{p}\|$. Clearly,

$$\nabla\varphi(p) = \begin{pmatrix} \sum_{\boldsymbol{a}\in A} w(\boldsymbol{a})\cos(\boldsymbol{p}-\boldsymbol{a}) \\ \sum_{\boldsymbol{a}\in A} w(\boldsymbol{a})\sin(\boldsymbol{p}-\boldsymbol{a}) \end{pmatrix}. \tag{2}$$

Let $Cor(B)$ denote the set of four corners of B. Then

$$\sin(B-\boldsymbol{a}) = \begin{cases} [-1,1] & \text{if } \ \boldsymbol{a}\in B, \\ [\min(\sin(Cor(B)-\boldsymbol{a})), 1] & \text{if } \ \boldsymbol{a}_x \in B_x \wedge \boldsymbol{a}_y < B_y, \\ [-1, \max(\sin(Cor(B)-\boldsymbol{a}))] & \text{if } \ \boldsymbol{a}_x \in B_x \wedge \boldsymbol{a}_y > B_y, \\ [\min(\sin(Cor(B)-\boldsymbol{a})), \max(\sin(Cor(B)-\boldsymbol{a}))] & \text{else.} \end{cases} \tag{3}$$

In other words, $\sin(B-\boldsymbol{a})$ can be computed from the sinus of at most four angles. Similarly for $\cos(B-\boldsymbol{a})$.

For any $p\in\mathbb{R}^2\setminus A$ it holds:

$$\nabla^2\varphi(\boldsymbol{p}) = \begin{pmatrix} \sum_{\boldsymbol{a}\in A} w(\boldsymbol{a})\frac{(\boldsymbol{p}_y-\boldsymbol{a}_y)^2}{\|\boldsymbol{p}-\boldsymbol{a}\|^3} & -\sum_{\boldsymbol{a}\in A} w(\boldsymbol{a})\frac{(\boldsymbol{p}_x-\boldsymbol{a}_x)(p_y-\boldsymbol{a}_y)}{\|\boldsymbol{p}-\boldsymbol{a}\|^3} \\ -\sum_{\boldsymbol{a}\in A} w(\boldsymbol{a})\frac{(\boldsymbol{p}_x-\boldsymbol{a}_x)(p_y-\boldsymbol{a}_y)}{\|\boldsymbol{p}-\boldsymbol{a}\|^3} & \sum_{\boldsymbol{a}\in A} w(\boldsymbol{a})\frac{(\boldsymbol{p}_x-\boldsymbol{a}_x)^2}{\|\boldsymbol{p}-\boldsymbol{a}\|^3} \end{pmatrix} \tag{4}$$

▶ **Lemma 4.2.** *Soft versions* $\square\nabla\varphi(B)$ *and* $\square\nabla^2\varphi(B)$ *are derived by replacing every occurrence of* p *in Eqs.* (2) *and* (4) *by* B *and evaluating with Eq.* (3) *and interval arithmetic.*

A box approximation of the length of the gradient of φ can then be achieved by:

$$\square\|\nabla\varphi(B)\| = \left\| \sum_{\boldsymbol{a}\in A\setminus B} w(\boldsymbol{a}) \begin{pmatrix}\cos(B-\boldsymbol{a}) \\ \sin(B-\boldsymbol{a})\end{pmatrix} \right\| + \left[-\sum_{\boldsymbol{a}\in A\cap B} w(\boldsymbol{a}), \sum_{\boldsymbol{a}\in A\cap B} w(\boldsymbol{a})\right]$$

where the length of an interval vector $I = (I_x, I_y)$ is computed by $\|I\| = \sqrt{I_x^2 + I_y^2}$ and the square root of an interval J by $\sqrt{J} = \left[\sqrt{\min|J|}, \sqrt{\max|J|}\right]$. We use the Lipschitz constant $L(B) = \min\{W, \max\square\|\nabla\varphi(B)\|\}$ of box B to compute $\square\varphi(B)$.

▶ **Lemma 4.3.** *The soft gradient test* $\square C^{\nabla}(B)$ *is convergent, i.e., for any monotone sequence of boxes* $(B_n)_{n\in\mathbb{N}}$ *that converges to a point* $\boldsymbol{p}$*, the point* $\boldsymbol{p}$ *is not the Fermat point iff* $\square C^{\nabla}(B_n) = \texttt{success}$ *for* n *large enough.*

We compute $\square N(B) = m_B - (\square\nabla^2\varphi(B))^{-1}\cdot\nabla\varphi(m_B)$ and the inverse of an interval matrix through interval arithmetic: $\begin{pmatrix} I & J \\ K & L\end{pmatrix}^{-1} = \frac{1}{IL-JK}\begin{pmatrix} L & -J \\ -K & I\end{pmatrix}$.

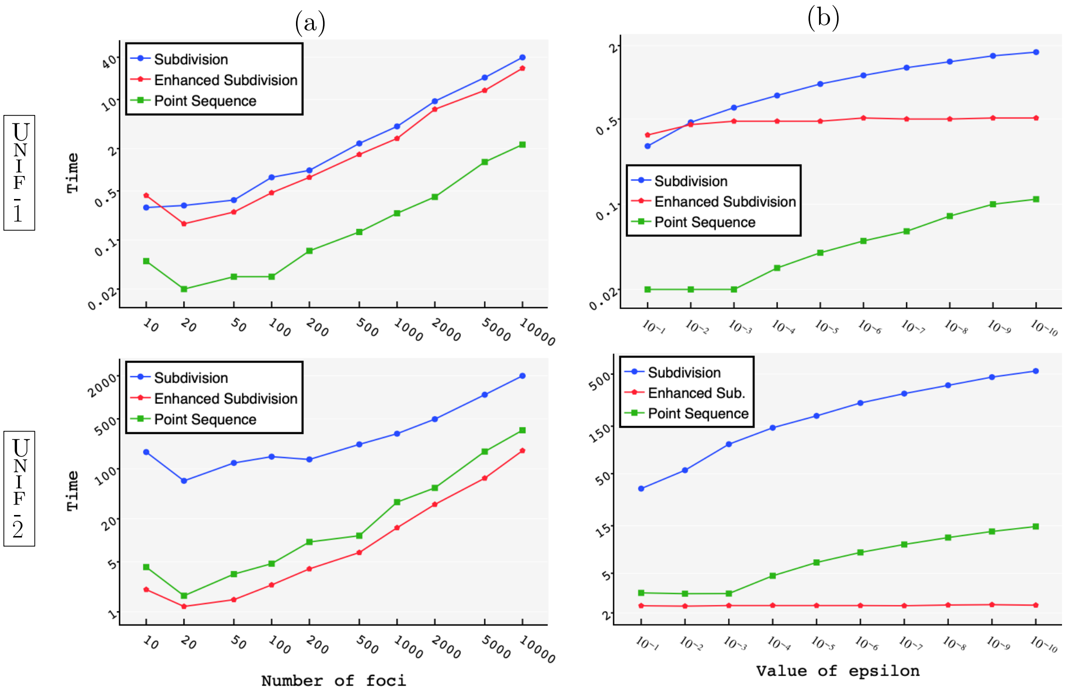

Figure 4 Experimental results. Points are, in Unif-1, uniformly sampled from a disk and, in Unif-2, uniformly sampled from two disjoint disks. Finding the Fermat point with times as function of (a) k, with $\varepsilon = 10^{-4}$ and (b) ε, with $k = 100$. Axes are in logarithmic scale.

5 Experiments and Conclusion

We have implemented our algorithms in two dimensions using Matlab. We have experimented with various datasets. An overview of experiments on synthetic datasets is shown in Fig. 4.

The runtime of all algorithms depends linearly on the number of foci. When ε is sufficiently small, then the enhanced subdivision algorithm outperforms the point sequence algorithm due to its quadratic convergence near the Fermat point.

References

1 A. Karim Abu-Affash and Matthew J. Katz. Improved bounds on the average distance to the Fermat-Weber center of a convex object. *Information Processing Letters*, 109(6):329–333, 2009.

2 Mihai Badoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proc. Symposium on Theory of Computing*, pages 250–257. ACM, 2002.

3 Chanderjit Bajaj. The algebraic degree of geometric optimization problems. *Discrete & Computational Geometry*, 3(2):177–191, 1988.

4 Huck Bennett and Chee Yap. Amortized analysis of smooth quadtrees in all dimensions. *Computational Geometry*, 63:20–39, 2017.

5 Bhaswar B. Bhattacharya. On the Fermat-Weber point of a polygonal chain and its generalizations. *Fundamenta Informaticae*, 107(4):331–343, 2011.

6 Prosenjit Bose, Anil Maheshwari, and Pat Morin. Fast approximations for sums of distances, clustering and the Fermat-Weber problem. *Computational Geometry*, 24(3):135–146, 2003.

7 Luitzen Egbertus Jan Brouwer. Über Abbildung von Mannigfaltigkeiten. *Mathematische Annalen*, 71(1):97–115, 1911.

8 Paz Carmi, Sariel Har-Peled, and Matthew J. Katz. On the Fermat-Weber center of a convex object. *Computational Geometry*, 32(3):188–195, 2005.

9 Hui Han Chin, Aleksander Madry, Gary L. Miller, and Richard Peng. Runtime guarantees for regression problems. In *Proc. Innovations in Theoretical Computer Science*, pages 269–282. ACM, 2013.

10 Ernest J. Cockayne and Zdzislaw A. Melzak. Euclidean constructibility in graph-minimization problems. *Mathematics Magazine*, 42(4):206–208, 1969.

11 Michael B. Cohen, Yin Tat Lee, Gary L. Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In *Proc. Symposium on Theory of Computing*, pages 9–21. ACM, 2016.

12 Theodorus Jozef Dekker. Finding a zero by means of successive linear interpolation. In *Constructive Aspects of the Fundamental Theorem of Algebra*, pages 37–48. Wiley Interscience, 1967.

13 Adrian Dumitrescu, Minghui Jiang, and Csaba D. Tóth. New bounds on the average distance from the Fermat-Weber center of a planar convex body. *Discrete Optimization*, 8(3):417–427, 2011.

14 Sándor P Fekete, Joseph SB Mitchell, and Karin Beurer. On the continuous Fermat-Weber problem. *Operations Research*, 53(1):61–76, 2005.

15 Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proc. Symposium on Theory of Computing*, pages 569–578. ACM, 2011.

16 Alexandre Goldsztejn. Comparison of the Hansen-Sengupta and the Frommer-Lang-Schnurr existence tests. *Computing*, 79(1):53–60, 2007.

17 Eldon R. Hansen and R.I. Greenberg. An interval Newton method. *Applied Math. and Computation*, 12:89–98, 1983.

18 Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. *Discrete & Computational Geometry*, 37(1):3–19, 2007.

19 Rudolf Krawczyk. Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. *Computing*, 4(3):187–201, 1969.

20 Harold W. Kuhn. A note on Fermat's problem. *Mathematical programming*, 4(1):98–107, 1973.

21 Long Lin and Chee Yap. Adaptive isotopic approximation of nonsingular curves: the parameterizability and nonlocal isotopy approach. *Discrete & Computational Geometry*, 45(4):760–795, 2011.

22 Luis Fernando Mello and Lucas Ruiz dos Santos. On the location of the minimum point in the Euclidean distance sum problem. *São Paulo Journal of Mathematical Sciences*, 12:108–120, 2018.

23 Ramon E Moore. *Interval Analysis*, volume 4. Prentice-Hall Englewood Cliffs, NJ, 1966.

24 Kent E Morrison. The fedex problem. *The College Mathematics Journal*, 41(3):222–232, 2010.

25 Gyula Sz Nagy. Tschirnhaus'sche Eiflächen und Eikurven. *Acta Mathematica Academiae Scientiarum Hungarica*, 1(1):36–45, 1950.

26 Karl Nickel. Triplex-algol and applications. *Interner Bericht des Instituts für Informatik der Universität Karlsruhe*, 1969.

27 Karl Nickel. On the Newton method in interval analysis. Technical report, Wisconsin University-Madison Mathematics Research Center, 1971.

28 Jiawang Nie, Pablo A. Parrilo, and Bernd Sturmfels. Semidefinite representation of the k-ellipse. In *Algorithms in algebraic geometry*, pages 117–132. Springer, 2008.

29 Lawrence M Ostresh Jr. Convergence and descent in the Fermat location problem. *Transportation Science*, 12(2):153–164, 1978.

30 Pablo A. Parrilo and Bernd Sturmfels. Minimizing polynomial functions. *Algorithmic and quantitative real algebraic geometry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 60:83–99, 2003.

31 Helmut Ratschek and Jon Rokne. *Computer methods for the range of functions*. Horwood, 1984.

32 Hanan Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1990.

33 Junpei Sekino. n-ellipses and the minimum distance sum problem. *The American mathematical monthly*, 106(3):193–202, 1999.

34 Sergey P Shary. Krawczyk operator revised. *Novosibirsk, Institute of Computational Technologies, Rússia*, 2004.

35 Rudolf Sturm. Über den Punkt kleinster Entfernungssumme von gegebenen Punkten. *Journal für die reine und angewandte Mathematik*, 97:49–61, 1884.

36 Cong Wang, Yi-Jen Chiang, and Chee Yap. On soft predicates in subdivision motion planning. *Computational Geometry: Theory and Applications.*, 48(8):589–605, September 2015.

37 Endre Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series*, 43:355–386, 1937.

38 Guoliang Xue and Yinyu Ye. An efficient algorithm for minimizing a sum of Euclidean norms with applications. *SIAM Journal on Optimization*, 7(4):1017–1036, 1997.