A Subquadratic Algorithm for Computing the L1-distance between Two Terrains

Pankaj K. Agarwal, Boris Aronov, Guillaume Moroz

A Subquadratic Algorithm for Computing the L_1-distance between Two Terrains

Pankaj K. Agarwal[1] Boris Aronov[2] Guillaume Moroz[3]

[1]Department of Computer Science, Duke University, USA
[2]Department of Computer Science and Engineering, Tandon School of Engineering, New York University, Brooklyn
[3]Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

January 23, 2025

Abstract

We study the problem of computing the L_1-distance between two piecewise-linear bivariate functions f and g, defined over a common bounded domain $\mathbb{M} \subset \mathbb{R}^2$, that is, computing the quantity $\|f-g\|_1 = \int_{\mathbb{M}} |f(x,y) - g(x,y)|\, dx\, dy$. If f and g are defined by linear interpolation over triangulations $\mathbf{T}_f$ and $\mathbf{T}_g$, respectively, of $\mathbb{M}$ with a total of n triangles, we show that $\|f-g\|_1$ can be computed in $\tilde{O}(n^{(\omega+1)/2})$ time, where $\Theta(n^\omega)$ is the time required to multiply two $n \times n$ matrices and $\tilde{O}$ notation hides polylogarithmic factors; this bound holds for the currently best known value of ω, which is approximately 2.37. The previously best known algorithm for computing $\|f-g\|_1$ takes $\Theta(n^2)$ time in the worst case.

More generally, if the complexity of the overlay of $\mathbf{T}_f$ and $\mathbf{T}_g$ is κ, then the runtime of our algorithm is $\tilde{O}(\kappa^{(\omega-1)/2} n^{(3-\omega)/2})$.

1 Introduction

Combining tools from computational geometry and computer algebra has led to new results that defy intuitive lower bounds. For example, using algebraic tools, it is possible to compute the centroid of the vertices in an arrangement of n lines in the plane in $O(n \log^2 n)$ operations, without computing the $\Theta(n^2)$ vertices [5]. One notable algebraic tool used for this problem is an algorithm to evaluate efficiently a univariate polynomial on multiple points [11]. This tool was used to speed up, among others, the computation of Shapley values in the plane [8]; the location of points among wireless transmitters [6]; and the computation of the L_2-distance between two piecewise-linear bivariate functions f and g, see below for the definition [13].

Bivariate functions over a planar domain appear naturally in geographical information systems, under the name of *terrains*, where they are used to model actual geographic terrains, such as elevation in a hilly area, and more generally to represent two-dimensional datasets, such as precipitation, air pollution, population density, or quantities of wine production. One way of approximating such a function is to triangulate the underlying domain and define a linear function over each triangle, resulting in a piecewise-linear function, often referred to as a *triangulated irregular network*. In this paper, we design an efficient algorithm for computing the L_1-distance between two piecewise-linear function, defined below, by combining tools from computational geometry and computer algebra.

Problem statement. Let $\mathbb{M}$ be a bounded polygonal region of $\mathbb{R}^2$, and let $f, g : \mathbb{M} \to \mathbb{R}$ be two piecewise-linear functions defined over their common domain $\mathbb{M}$. Each of f, g is associated with a triangulation of $\mathbb{M}$, denoted by $\mathbf{T}_f, \mathbf{T}_g$, respectively, and f (resp., g) is defined to be linear over each triangle of $\mathbf{T}_f$ (resp., $\mathbf{T}_g$). Let n be the total number of vertices in the two triangulations. For any integer $p > 0$, we define the L_p-distance between f and g to be

$$\|f - g\|_p = \left(\int_{\mathbb{M}} |f - g| \, dx \, dy \right)^{1/p}. \tag{1}$$

The goal in this paper is to compute $\|f - g\|_1$.

We will work in the Real RAM model [15], where inputs and intermediate values are real numbers and algebraic operations and comparisons can be performed in constant time, but modulus or rounding to integers operations are not allowed. We assume that the coefficients of the piecewise linear functions f and g and the coordinates of the triangles in $\mathbf{T}_f$ and $\mathbf{T}_g$ are real numbers.

Related work. Piecewise-linear functions and triangulated irregular networks are common objects in computational geometry and geometric information systems, and the problem of comparing two terrains defined over the same domain has been addressed in previous work [2, 13]. One focus of that work was on identifying a linear dependence between two functions or terrains. Agarwal *et al.* [2] studied the problem of computing the minimal distance $\sigma_p(f, g) = \min_{s,t} \|sf + t - g\|_p$ for $p = 1, \infty$. They showed that $\sigma_\infty(f, g)$ can be computed in $\tilde{O}(n^{4/3})$ time (as stated in the abstract, $\tilde{O}$ notation hides polylogarithmic factors), and that an $(1 + \varepsilon)$-approximation of $\sigma_1(f, g)$ can be computed in $O(n/\varepsilon)$ time if $\mathbf{T}_f$ and $\mathbf{T}_g$ are identical. However if $\mathbf{T}_f$ and $\mathbf{T}_g$ are different, the runtime of their algorithm is $O(\kappa/\varepsilon)$, where κ is the complexity of the overlay of the two triangulations; $\kappa = \Theta(n^2)$ in the worst case.

Moroz and Aronov [13] showed that even for unaligned triangulations, $\|f - g\|_2$ and $\sigma_2(f, g)$ can be computed in $O(n \log^4 n \log \log n)$ time. This bound also applies to computing the L_{2p}-distance, where p is a constant positive integer. Unfortunately, these approaches do not extend to computing $\sigma_1(f, g)$ or $\|f - g\|_1$ in subquadratic time in the general case. One notable obstacle is that the L_1-distance involves the absolute value function, which is not algebraic.

One key algebraic tool used in [13] to speed up the computation of $\|f - g\|_2$ is the evaluation of a *univariate* polynomial of degree n on n points in $\tilde{O}(n)$ operations. In the Real RAM model, such fast *univariate multipoint evaluation* algorithms have been known since 1972 [11]. For bivariate polynomials, it is only more recently, in 2004, that subquadratic bounds were found for their evaluation on multiple points [14]. The problem of bounding the number of operations to evaluate a multivariate polynomial on multiple points is still an active area of research, with recent results improving the best known bounds [16].

Our results. The main result of the paper is a subquadratic algorithm for computing $\|f - g\|_1$. Its runtime depends on ω, the standard matrix multiplication constant, which is known to be less than 2.371552 [17] and conjectured by some to be 2. The bound in the following theorem holds if $(\omega + 1)/2 > 8/5$.

Theorem 1. *Let f and g be two bivariate functions defined by linear interpolation over triangulations $\mathbf{T}_f$ and $\mathbf{T}_g$, respectively, over a common (bounded) domain $\mathbb{M}$ with a total of n triangles, then $\|f - g\|_1$ can be computed in $\tilde{O}(n^{(\omega+1)/2})$ time, where $\Theta(n^\omega)$ is the time required to multiply two $n \times n$ matrices; currently the best known value of ω is around 2.37. If the complexity of the overlay of $\mathbf{T}_f$ and $\mathbf{T}_g$ is κ, then $\|f - g\|_1$ can be computed in $\tilde{O}(\kappa^{(\omega-1)/2} n^{(3-\omega)/2})$ operations.*

This result is based on three main ideas: we first show that the problem can be reduced to computing a double sums of rational functions, where the denominators depend only on two variables of the outer sum (Section 2). Those sums are defined over pairs of geometric objects that satisfy specific algebraic inequalities. Next, we use geometric cuttings to obtain a sign-invariant biclique decomposition of those pairs of geometric objects (Section 3). Finally, we extend the fast multipoint polynomial evaluation (Section 4) to evaluate those sums efficiently.

Very recently, we have become aware of a preprint improving the bound on fast multipoint polynomial evaluation [16]. This seems to improve the bound of Theorem 1 to $O(n^{8/5+\varepsilon})$ for any $\varepsilon > 0$. We discuss this (possible) improvement at the end of this paper.

2 Reduction to Multi-point Evaluation

In this section, we first express the desired quantity $\|f - g\|_1$ as a sum of integrals associated with vertices of the overlay of $\mathbf{T}_f$ and $\mathbf{T}_g$ (in Section 2.1) and ultimately as a sum of low-degree rational expressions of the parameters of f and g (in Section 2.2).

Without loss of generality, we can rotate the (x, y)-coordinate system such that none of the edges of $\mathbf{T}_f$ and $\mathbf{T}_g$ is parallel to the vertical line $x = 0$. Moreover, for two triangles $T_1 \in \mathbf{T}_f$ and $T_2 \in \mathbf{T}_g$, let π_1 be the plane in $\mathbb{R}^3$ supporting the triangle $f(T_1)$, and π_2 be the plane supporting the triangle $g(T_2)$. If π_1 and π_2 are not parallel, for clarity of presentation, and with no loss of generality, we assume that their intersection line crosses the vertical plane $x = 0$.

2.1 Reduction to overlay vertices

Let $\mathsf{V}_f, \mathsf{E}_f$ (resp., $\mathsf{V}_g, \mathsf{E}_g$) be the set of vertices and edges of $\mathbf{T}_f$ (resp., $\mathbf{T}_g$); E_f and E_g do not include edges contained in the boundary of $\mathbb{M}$. Let Σ be the overlay of $\mathbf{T}_f$ and $\mathbf{T}_g$, a convex subdivision in $\mathbb{R}^2$ that is a refinement of both $\mathbf{T}_f$ and $\mathbf{T}_g$. Let V (resp., C) be the set of vertices (resp., bounded cells) of Σ, and let $\mathsf{X} \subseteq \mathsf{E}_f \times \mathsf{E}_g$ be the family of pairs of segments that intersect. Following the same argument as in [13, Corollary 4], $\|f - g\|_1$ can expressed as

$$\|f - g\|_1 = \sum_{v \in \mathsf{V}} \sum_{\tau \in \mathsf{C}: \partial\tau \ni v} F(v, \tau, h_\tau), \tag{2}$$

where the function $F(v, \tau, h_\tau)$ is defined as follows. Let e_1, e_2 be the two edges of τ incident on v, let ℓ_i be the line supporting e_i for $i = 1, 2$, and let f_τ (resp. g_τ) be the linear function of f (resp. g) over τ. By our assumption, v lies to the right of the y-axis. Let $\triangle(v)$ be the triangle defined by the y-axis and the lines ℓ_1, ℓ_2 (lying to the right of the y-axis), let $h_\tau(x, y) = |f_\tau(x, y) - g_\tau(x, y)|$, let $\delta(v, \tau) \in \{-1, +1\}$ be -1 if τ lies above or below both ℓ_1 and ℓ_2, and $+1$ if it lies between them. See Figure 1. Then

$$F(v, \tau, h_\tau) = \delta(v, \tau) \int_{\triangle(v)} h_\tau(x, y) \mathrm{d}x\mathrm{d}y. \tag{3}$$

A vertex in Σ is either an original vertex of a $\mathbf{T}_f$ or $\mathbf{T}_g$ or an intersection point of an edge of $\mathbf{T}_f$ with an edge of $\mathbf{T}_g$. We can rewrite (2) as:

$$\|f - g\|_1 = \sum_{v \in \mathsf{V}_f \cup \mathsf{V}_g} \sum_{\tau \in \mathsf{C}: \partial\tau \ni v} F(v, \tau, h_\tau) + \sum_{(e_1, e_2) \in \mathsf{X}} \sum_{\tau \in \mathsf{C}: \partial\tau \ni e_1 \cap e_2} F(e_1 \cap e_2, \tau, h_\tau), \tag{4}$$

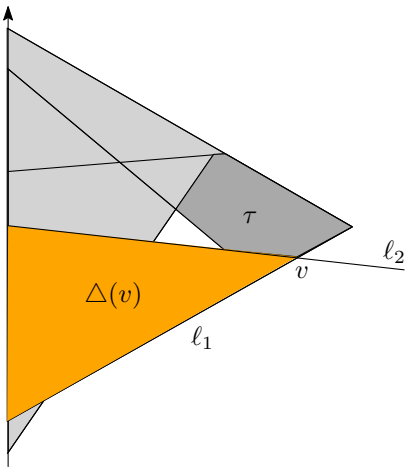

Figure 1: A typical triangle $\triangle(v)$ associated with vertex $v = \ell_1 \cap \ell_2$ in h_0 (adapted from [13])

By preprocessing $\mathbf{T}_f$ and $\mathbf{T}_g$ for planar point-location queries, the cells of C incident on the vertices of $\mathsf{V}_f \cup \mathsf{V}_g$ can be computed in a total time of $O(n \log n)$, and thus the first term in (4) can be computed in $O(n \log n)$ time; see [13] for details. We thus focus on computing the second term, which we denote by $\Phi(\mathsf{E}_f, \mathsf{E}_g)$.

For an intersecting pair of edges $(e_1, e_2) \in \mathsf{X}$, the vertex $e_1 \cap e_2 \in \mathsf{V}$ is incident on four cells of Σ, and thus it contributes to four terms in $\Phi(\mathsf{E}_f, \mathsf{E}_g)$. Each of these terms has the following form: Let f_{11}, f_{12} (resp. g_{21}, g_{22}) be the linear functions of f (resp. g) over the triangles lying above and below e_1 (resp. e_2) respectively. For $i, j \in \{1, 2\}$, define $h_{ij}(x, y) = |f_{1i}(x, y) - g_{2j}(x, y)|$, $\mathbb{1}(i = j)$ to be 1 if $i = j$ and 0 otherwise, and

$$F_{ij}(e_1, e_2) = (-1)^{\mathbb{1}(i=j)} \int_{\triangle(e_1 \cap e_2)} h_{ij}(x, y) \mathrm{d}x \mathrm{d}y;$$

in words, $F_{ij}(e_1, e_2)$ is the signed volume between the graphs of f_{1i} and g_{2j} over $\triangle(e_1 \cap e_2)$. Using these definitions, for a pair $(e_1, e_2) \in \mathsf{X}$, we can write

$$\sum_{\tau \in \mathsf{C}: \partial\tau \ni e_1 \cap e_2} F(e_1 \cap e_2, \tau, h_\tau) = \sum_{i,j \in \{1,2\}} F_{ij}(e_1, e_2) \tag{5}$$

and thus

$$\Phi(\mathsf{E}_f, \mathsf{E}_g) = \sum_{i,j \in \{1,2\}} \sum_{(e_1, e_2) \in \mathsf{X}} F_{ij}(e_1, e_2). \tag{6}$$

In the next subsection we show that each F_{ij} can be written as the sum of at most five rational functions of constant degree each.

2.2 Reduction to a sum of rational expressions

We now reexamine the integrals that appear in the previous section and express them as sums of low-degree rational functions, with appropriate signs. We first focus on a single function $F_{ij}(e_1, e_2)$.

Let h_0 be the halfplane defined by $z = 0$ and $x \geq 0$. Let $f(x, y) = a_1 x + b_1 y + c_1$ and $g(x, y) = a_2 x + b_2 y + c_2$ be two linear functions in x, y. By abuse of notation, we let $e_1(x) = \alpha_1 x + y_1$ and $e_2(x) = \alpha_2 x + y_2$ be the two linear functions in x extending the edges e_1 and e_2. Let h_1 and

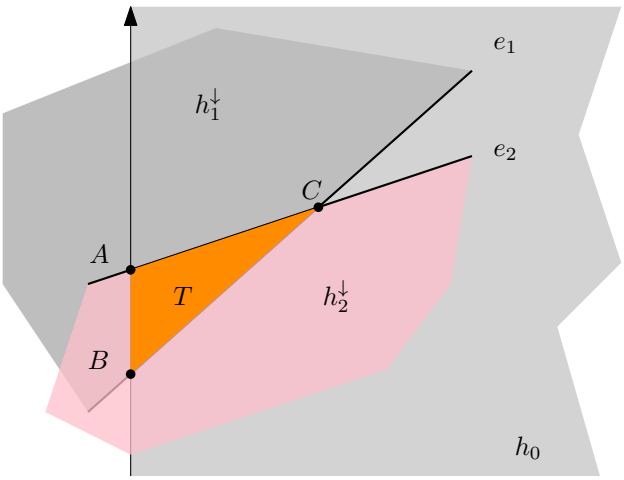

Figure 2: A typical triangle $T = \triangle ABC$ in h_0

h_2 be the two halfplanes in $\mathbb{R}^3$ defined by $z = f(x, y), y \leq e_1(x)$ and $z = g(x, y), y \geq e_2(x)$ respectively. Below we consider such pairs (h_1, h_2) that the intersection of the projections $h_1^{\downarrow}$ of h_1 and $h_2^{\downarrow}$ of h_2 to h_0 is the triangle $T = \triangle ABC$ defined as the set of points (x, y) with $x \geq 0$ and $e_2(x) \leq y \leq e_1(x)$; refer to Figure 2.

The goal of this section is to prove that the volume between h_1 and h_2 over T is a sum of rational expression in the parameters of h_1 and h_2, whose exact form depends on the sign variables $\delta_A, \delta_B, \delta_C$ (i.e., values ± 1), indicating whether h_1 is above $(+1)$ or below (-1) h_2 at the corresponding vertex of T.

More precisely, the halfplane h_1 is contained in the plane π_1 defined by $z = f(x, y)$ and bounded by the edge $e_1^{\uparrow}$, while h_2 is contained in π_2 defined by $z = g(x, y)$ and bounded by $e_2^{\uparrow}$ such that:

$$h_1 \left| \begin{array}{ll} \pi_1: & z = a_1 x + b_1 y + c_1 \\ e_1^{\uparrow}: & \begin{pmatrix} t \\ y_1 + \alpha_1 t \\ z_1 + \beta_1 t \end{pmatrix} \end{array} \right. \qquad h_2 \left| \begin{array}{ll} \pi_2: & z = a_2 x + b_2 y + c_2 \\ e_2^{\uparrow}: & \begin{pmatrix} t \\ y_2 + \alpha_2 t \\ z_2 + \beta_2 t \end{pmatrix} \end{array} \right. \tag{7}$$

Moreover, since $e_1^{\uparrow} \subset \pi_1$ and $e_2^{\uparrow} \subset \pi_2$, the following relations hold:

$$\begin{cases} z_1 = b_1 y_1 + c_1 \\ \beta_1 = a_1 + b_1 \alpha_1 \end{cases} \qquad \begin{cases} z_2 = b_2 y_2 + c_2 \\ \beta_2 = a_2 + b_2 \alpha_2 \end{cases} \tag{8}$$

After the same preprocessing as in [13], we can assume that $y_1 > y_2$, $\alpha_1 < \alpha_2$, so that the projection of the edges intersect at $x = \frac{y_1 - y_2}{\alpha_2 - \alpha_1} > 0$. Thus, on the xy-plane, the triangle T has three vertices A, B, C:

$$A \left| \begin{array}{l} 0 \\ y_1 \end{array} \right. \qquad B \left| \begin{array}{l} 0 \\ y_2 \end{array} \right. \qquad C \left| \begin{array}{l} \frac{y_1 - y_2}{\alpha_2 - \alpha_1} \\ \frac{\alpha_2 y_1 - \alpha_1 y_2}{\alpha_2 - \alpha_1} \end{array} \right. \tag{9}$$

2.2.1 Explicit sign conditions

The explicit rational expressions for $\int_T |f - g|$ will depend on the value of of $\delta_A, \delta_B, \delta_C$, namely the sign of the difference $g - f$ above A, B, and C respectively. Combining eqs. (7), (8), and (9) and

using the assumption $\alpha_2 > \alpha_1$, we can explicitly expand the formula for δ_A, δ_B, and δ_C:

$$\begin{aligned}
\delta_A &= \text{sign}(b_2 y_1 + c_2 - z_1) \\
\delta_B &= \text{sign}(z_2 - b_1 y_2 - c_1) \\
\delta_C &= \text{sign}\left(\frac{(z_2 - z_1)(\alpha_2 - \alpha_1) + (\beta_2 - \beta_1)(y_1 - y_2)}{\alpha_2 - \alpha_1}\right) \\
&= \text{sign}\left((z_2 - z_1)(\alpha_2 - \alpha_1) + (\beta_2 - \beta_1)(y_1 - y_2)\right) \\
&= \text{sign}\left(\gamma_2 - \gamma_1 + \beta_2 y_1 - z_1\alpha_2 + \beta_1 y_2 - z_2\alpha_1\right)
\end{aligned} \tag{10}$$

where $\gamma_i = z_i\alpha_i - \beta_i y_i$, and $\text{sign}(x) = +1$ if $x \geq 0$ and -1 otherwise. Note that, after the introduction of γ_i, the formula for $\delta_A, \delta_B, \delta_C$ are the signs of three *bilinear* expressions in the five parameters $\alpha_i, \beta_i, y_i, z_i, \gamma_i$ describing each of the two halfplanes.

2.2.2 Explicit rational expressions

The volume $\int_T |f - g|$ will be written as a weighted sum of three rational expressions. For that, we introduce the intersection point D of the line $e_1^{\uparrow}$ with the plane π_2, the intersection point E of $e_2^{\uparrow}$ and π_1, and the intersection point F of the line DE with the vertical plane $x = 0$. These points exist due to our general position assumptions, as soon as δ_A, δ_B and δ_C don't all have the same sign.

Let $D^{\downarrow}$, $E^{\downarrow}$, and $F^{\downarrow}$ be projections of D, E, and F respectively to h_0. Finally, the points A_1, B_1, C_1 are the points above A, B, C respectively, with z-coordinates $f(x_A, y_A)$, $f(x_B, y_B)$ and $f(x_C, y_C)$, and the points A_2, B_2, C_2 are the points above A, B, C respectively, with z-coordinates $g(x_A, y_A)$, $g(x_B, y_B)$ and $g(x_C, y_C)$. Figures 3 and 4 illustrate those notations in different cases.

We will express the volume between h_1 and h_2 over T as a weighted sum of volumes of polytopes spawned by C, D, and E respectively, toward the vertical plane $x = 0$. The following expressions represent the signed volumes of the polytopes between the planes π_1 and π_2, above the triangles $T = \triangle ABC$, $\triangle AD^{\downarrow}F^{\downarrow}$, and $\triangle BE^{\downarrow}F^{\downarrow}$ respectively. More precisely, letting $R_C = \int_T (g - f)$, $R_D = \int_{\triangle AD^{\downarrow}F^{\downarrow}} (g - f)$ and $R_E = \int_{\triangle BE^{\downarrow}F^{\downarrow}} (g - f)$, we have:

$$\begin{aligned}
R_C &= \frac{1}{6} x_C (y_A - y_B) \sum_{X \in \{A,B,C\}} (g(x_X, y_X) - f(x_X, y_X)), \\
R_D &= \frac{1}{6} x_D (y_F - y_A)(g(x_A, y_A) - f(x_A, y_A)), \text{ and} \\
R_E &= \frac{1}{6} x_E (y_F - y_B)(g(x_B, y_B) - f(x_B, y_B)).
\end{aligned} \tag{11}$$

We now proceed to express R_C, R_D, and R_E as fractions in the parameters of h_1 and h_2. In particular, we will show that the denominator of each fraction depends only on either two parameters of h_1, or two parameters of h_2.

Rational representation of R_C Using the formula for the signed volume R_C in eq. (11), we expand the evaluation of f and g at A, B, C to obtain

$$\begin{aligned} R_C &= \frac{1}{6}(y_1 - y_2)\frac{y_1 - y_2}{\alpha_2 - \alpha_1}\left(\left\langle \begin{pmatrix} a_1 - a_2 \\ b_1 - b_2 \end{pmatrix}, A + B + C \right\rangle + 3(c_1 - c_2)\right) \\ &= \frac{(y_1 - y_2)^2 \begin{bmatrix} (\alpha_2 - \alpha_1)\big[(b_1 - b_2)(y_1 + y_2) + 3(c_1 - c_2)\big] \\ + (a_1 - a_2)(y_1 - y_2) + (b_1 - b_2)(\alpha_2 y_1 - \alpha_1 y_2) \end{bmatrix}}{6\,(\alpha_1 - \alpha_2)^2}, \end{aligned} \tag{12}$$

so that R_C is a rational expression with denominator $(\alpha_2 - \alpha_1)^2$.

Rational representation of R_D and R_E For quantities R_D and R_E, we need to compute the coordinates of D, E, and F. Using eq. (7), the abscissa t_D of D satisfies $z_1 + \beta_1 t_D = a_2 t_D + b_2 y_1 + b_2\alpha_1 t_D + c_2$. A similar relation for the abscissa t_E of E leads to

$$\begin{aligned} t_D &= \frac{z_1 - b_2 y_1 - c_2}{b_2\alpha_1 + a_2 - \beta_1} & t_E &= \frac{z_2 - b_1 y_2 - c_1}{b_1\alpha_2 + a_1 - \beta_2} \\ &= \frac{(b_1 - b_2)y_1 + c_1 - c_2}{(b_2 - b_1)\alpha_1 + a_2 - a_1} & &= \frac{(b_1 - b_2)y_2 + c_1 - c_2}{(b_2 - b_1)\alpha_2 + a_2 - a_1} \end{aligned}$$

Moreover, F is at the intersection of h_1, h_2, and the vertical plane $x = 0$. In particular, its coordinates are the solution of $x = 0, z = b_1 y + c_1 = b_2 y + c_2$, so we have

$$D \left| \begin{matrix} t_D \\ y_1 + \alpha_1 t_D \\ z_1 + \beta_1 t_D \end{matrix} \right. \qquad E \left| \begin{matrix} t_E \\ y_2 + \alpha_2 t_E \\ z_2 + \beta_2 t_E \end{matrix} \right. \qquad F \left| \begin{matrix} 0 \\ \frac{c_2 - c_1}{b_1 - b_2} \\ \frac{b_1 c_2 - b_2 c_1}{b_1 - b_2} \end{matrix} \right. \tag{13}$$

Also, the points A_1, A_2, B_1, B_2 have coordinates

$$A_1 \left| \begin{matrix} 0 \\ y_1 \\ b_1 y_1 + c_1 \end{matrix} \right. \qquad A_2 \left| \begin{matrix} 0 \\ y_1 \\ b_2 y_1 + c_2 \end{matrix} \right. \qquad B_1 \left| \begin{matrix} 0 \\ y_2 \\ b_1 y_2 + c_1 \end{matrix} \right. \qquad B_2 \left| \begin{matrix} 0 \\ y_2 \\ b_2 y_2 + c_2 \end{matrix} \right. \tag{14}$$

Finally, replacing the coordinates in eq. (11), this leads to:

$$\begin{aligned} R_D &= \frac{1}{6}\frac{(b_1 - b_2)y_1 + c_1 - c_2}{(b_2 - b_1)\alpha_1 + a_2 - a_1}\left(\frac{c_2 - c_1}{b_1 - b_2} - y_1\right)(b_2 y_1 + c_2 - b_1 y_1 - c_1) \\ &= \frac{1}{6}\frac{\big((b_1 - b_2)y_1 + c_1 - c_2\big)^3}{(b_2 - b_1)\alpha_1 + a_2 - a_1} \end{aligned} \tag{15}$$

and

$$\begin{aligned} R_E &= \frac{1}{6}\frac{(b_1 - b_2)y_2 + c_1 - c_2}{(b_2 - b_1)\alpha_2 + a_2 - a_1}\left(\frac{c_2 - c_1}{b_1 - b_2} - y_2\right)(b_2 y_2 + c_2 - b_1 y_2 - c_1) \\ &= \frac{1}{6}\frac{\big((b_1 - b_2)y_2 + c_1 - c_2\big)^3}{(b_2 - b_1)\alpha_2 + a_2 - a_1}. \end{aligned} \tag{16}$$

2.2.3 Case distinctions for the volume expression

Our goal is to provide a formula for the volume of the polyhedron between π_1 and π_2 above the triangle ABC, as a linear combination of fractions where the denominators depend only on two variables from the triangulation $\mathbf{T}_f$ or on two variables from $\mathbf{T}_g$. This will allow us to use fast multipoint evaluation algorithms for bivariate polynomials (see Section 4).

We will see that for the different cases of signs of $\delta_A, \delta_B, \delta_C$, the volume above T can be expressed as a weighted signed sum of R_C, R_D, and R_E. We consider the cases where the vector of signs $(\delta_A, \delta_B, \delta_C)$ is $(+1,+1,+1), (+1,+1,-1)$, or $(+1,-1,-1)$. The other cases are deduced by symmetry.

When $\delta_A, \delta_B, \delta_C$ are positive: the halfplanes don't meet above the triangle. In this case, $V_{+++}(h_1, h_2) = R_C$, a rational expression with denominator $(\alpha_2 - \alpha_1)^2$. By symmetry, when $\delta_A, \delta_B, \delta_C$ are -1, we have $V_{---}(h_1, h_2) = -R_C$.

When δ_A, δ_B are positive and δ_C is negative. Using the notation of Section 2.2.2, the volume between the planes over T in this case is the sum of the volumes $\mathrm{vol}(A_1A_2B_2B_1DE)$ and $\mathrm{vol}(DEC_2C_1)$; refer to Figure 3. With δ_A, δ_B positive, and δ_C negative, the signed volume R_C is equal to $\mathrm{vol}(A_1A_2B_1DE) - \mathrm{vol}(DEC_2C_1)$. Thus, the volume V_{++-} can be expressed as

$$V_{++-}(h_1, h_2) = 2\,\mathrm{vol}(A_1A_2B_2B_1DE) - R_C\,.$$

The volume of the polytope $A_1A_2B_2B_1DE$ is computed as the difference of volumes of the two simplices B_1B_2FE and A_1A_2FD. When δ_A and δ_B are positive, the y-coordinate of the point F can be either (i) greater than y_A, or (ii) less than y_B.

In case (i), we have $\mathrm{vol}(A_1A_2B_2B_1DE) = \mathrm{vol}(B_1B_2FE) - \mathrm{vol}(A_1A_2FD)$. Moreover $\mathrm{vol}(A_1A_2FD) = R_D$ and $\mathrm{vol}(B_1B_2FE) = R_E$, so that $\mathrm{vol}(A_1A_2B_2B_1DE) = R_E - R_D$.

In case (ii), we have $\mathrm{vol}(A_1A_2B_2B_1DE) = \mathrm{vol}(A_1A_2FD) - \mathrm{vol}(B_1B_2FE)$, $\mathrm{vol}(A_1A_2FD) = -R_D$, and $\mathrm{vol}(B_1B_2FE) = -R_E$, so that once again $\mathrm{vol}(A_1A_2B_2B_1DE) = R_E - R_D$.

Thus, in both cases, we have $\mathrm{vol}(A_1A_2B_2B_1DE) = R_E - R_D$, and therefore

$$V_{++-}(h_1, h_2) = 2R_E - 2R_D - R_C\,.$$

Similarly, by symmetry, we have $V_{--+} = -2R_E + 2R_D + R_C$.

When δ_A is positive and δ_B, δ_C are negative. This case is illustrated in Figure 4. Using the notation of the previous section the volume between the two halfplanes is the sum of volumes of the polytopes A_1A_2FD and $B_1B_2C_2C_1DF$. First, with δ_A positive and δ_B, δ_C negative, the signed volume R_C is $\mathrm{vol}(A_1A_2FD) - \mathrm{vol}(B_1B_2C_2C_1DF)$. Moreover, since δ_A is positive and δ_B is negative, the y-coordinate of F is in the interval $[y_B, y_A]$ and we have $\mathrm{vol}(A_1A_2FD) = -R_D$. Thus, the volume V_{+--} can be expressed as

$$V_{+--}(h_1, h_2) = 2\,\mathrm{vol}(A_1A_2FD) - R_C = -2R_D - R_C\,. \tag{17}$$

By symmetry, we have $V_{-++} = 2R_D + R_C$. Moreover, the role of D and E are symmetric, and when δ_A, δ_C are negative, and δ_B is positive, we have similarly $V_{-+-} = 2\,\mathrm{vol}(B_2B_1FE) - R_C = 2R_E - R_C$ and $V_{+-+} = -2R_E + R_C$.

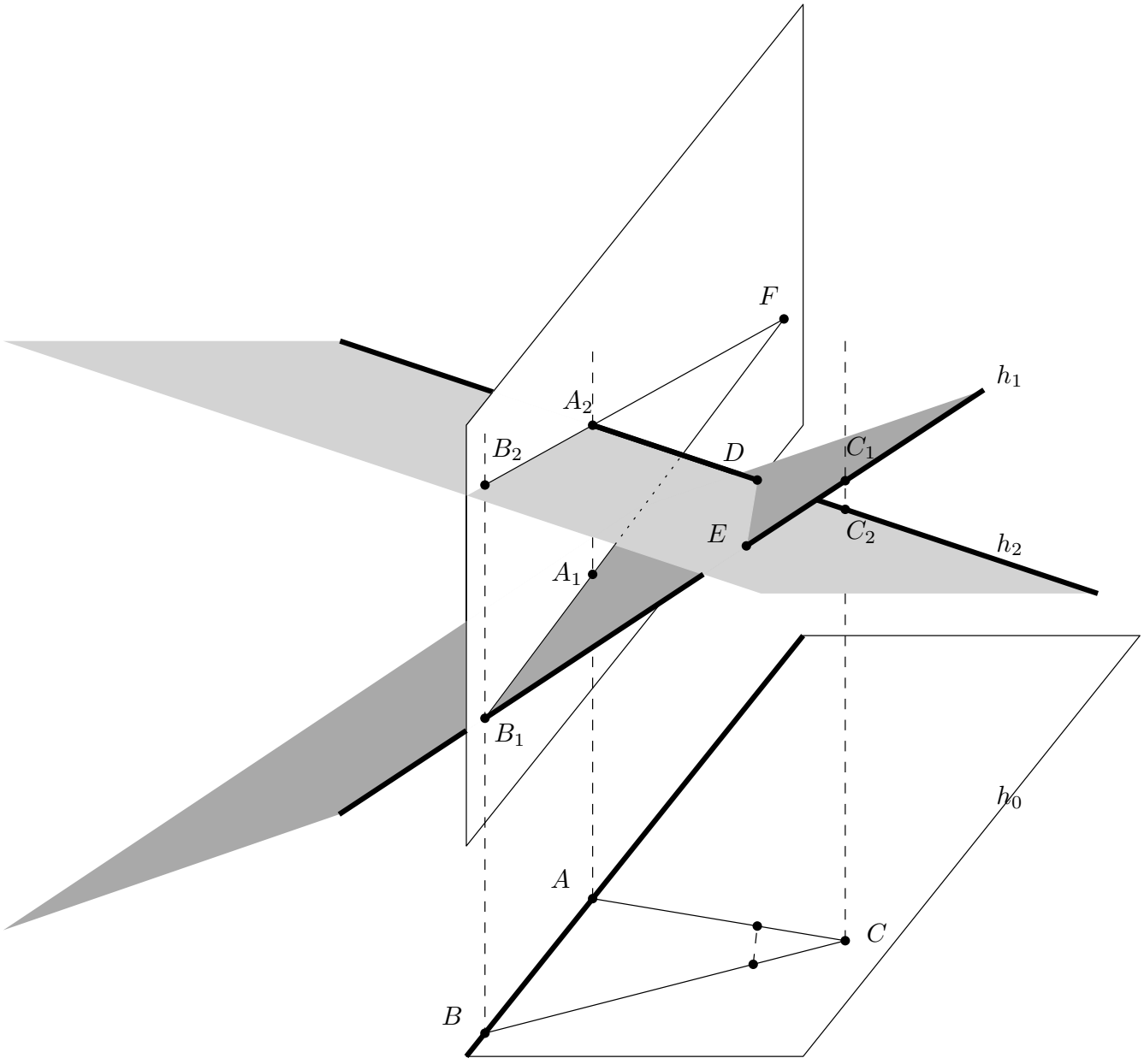

Figure 3: Case where δ_C has a sign different from δ_A and δ_B

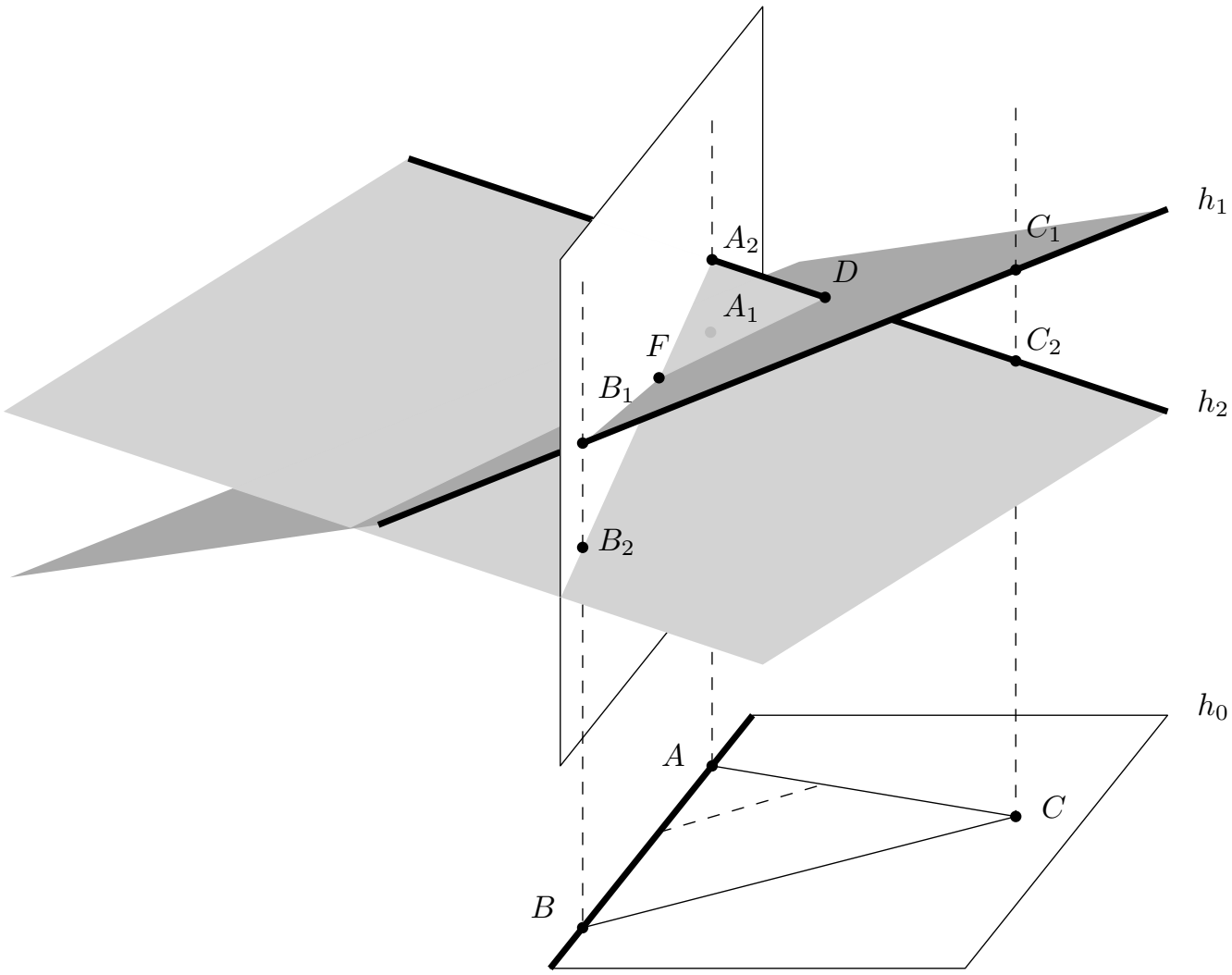

Figure 4: Case where δ_A has a sign different from δ_B and δ_C

Gathering all the cases. Depending on the signs δ_A, δ_B, and δ_C, we have different formulas for the volume between the planes over T. We can gather them all within one expression that is bilinear in $\delta_A, \delta_B, \delta_C$ on one side, and R_A, R_B, R_C on the other side.

Lemma 2. *Assume that $v_1 = (a_1, b_1, c_1, y_1, \alpha_1)$ and $v_2 = (a_2, b_2, c_2, y_2, \alpha_2)$ are vectors of real numbers such that $y_1 > y_2$ and $\alpha_1 < \alpha_2$. Let $T \subset \mathbb{R}^2$ be a triangular cell defined by the set of points (x, y) such that $x \geq 0$, $y \leq y_1 + \alpha_1 x$, and $y \geq y_2 + \alpha_2 x$. And let f and g be two linear functions defined by $f(x, y) = a_1 x + b_1 y + c_1$ and $g(x, y) = a_2 x + b_2 y + c_2$. Then the volume $\int_{(x,y)\in T} |f - g| \, dx \, dy$ can be expressed as:*

$$\int_{(x,y)\in T} |f - g| \, dx \, dy = \delta_C(R_D - R_E + R_C) - \delta_A R_D + \delta_B R_E$$

where $\delta_A, \delta_B, \delta_C$, defined in eq. (10), depend on at most four entries of v_1 and four entries of v_2, and:

$$R_C(v_1; v_2) = \frac{p_C(v_1; v_2)}{q_C(\alpha_1; \alpha_2)} \qquad R_D(v_1; v_2) = \frac{p_A(v_1, v_2)}{q_A(v_1; (a_2, b_2))} \qquad R_E(v_1; v_2) = \frac{p_B(v_1, v_2)}{q_B((a_1, b_1); v_2)}$$

are rational functions of constant degrees defined in eqs. (12), (15) and (16).

Remark. *If instead of the assumptions $y_1 > y_2$ and $\alpha_1 < \alpha_2$, we have the assumptions $y_2 < y_1$ and $\alpha_2 < \alpha_1$, then the signed volume above the triangle ABC becomes $-R_C$, and δ_C becomes $-\delta_C$ since we factored out the sign of $\alpha_2 - \alpha_1$ in eq. (10). Without assumption on the order between α_1 and α_2, letting $\sigma = \operatorname{sign}(\alpha_2 - \alpha_1)$, the formula for the volume becomes:*

$$\int_{(x,y)\in T} |f - g| \, dx \, dy = \delta_C R_C + \delta_C \sigma (R_D - R_E) - \delta_A R_D + \delta_B R_E \, .$$

Returning to expressing (6) as a sum of rational functions, recall that for $j \in 1, 2$,

$$\begin{aligned} F_{ij}(e_1, e_2) = (-1)^{\mathbb{1}(i=j)} \Big[& \delta_C(e_1^*, e_2^*) R_C(e_1^*, e_2^*) \\ & + \sigma(e_1^*, e_2^*) \delta_C(e_1^*, e_2^*)(R_D(e_1^*, e_2^*) - R_E(e_1^*, e_2^*)) \\ & - \delta_A(e_1^*, e_2^*) R_D(e_1^*, e_2^*) + \delta_B(e_1^*, e_2^*) R_E(e_1^*, e_2^*) \Big]. \end{aligned}$$

Since there are five terms in the above expression for F_{ij}, eq. (6) can be expressed as follows:

Corollary 3. *Let $\mathsf{E}_f, \mathsf{E}_g$ be the set of edges in $\mathbf{T}_f$ and $\mathbf{T}_g$ respectively, and $\mathsf{X} \subseteq \mathsf{E}_f \times \mathsf{E}_g$ be the intersecting pairs of segments. Then*

$$\Phi(\mathsf{E}_f, \mathsf{E}_g) = \sum_{k=1}^{5} \sum_{i,j\in\{1,2\}} \sum_{(e_1,e_2)\in\mathsf{X}} (-1)^{\mathbb{1}(i=j)} \sigma_{ijk}(e_{1i}^*, e_{2j}^*) \cdot \xi_{ijk}(e_{1i}^*, e_{2j}^*) \cdot F_{ijk}(e_{1i}^*, e_{2j}^*), \tag{18}$$

where σ_{ijk} is either σ or the constant function 1, $\xi_{ijk} \in \{\pm\delta_A, \pm\delta_B, \pm\delta_C\}$, and $F_{ijk} \in \{R_C, R_D, R_E\}$.

3 Computing a Biclique Partition

Using Corollary 3, $\Phi(\mathsf{E}_f, \mathsf{E}_g)$ can be computed in $O(\kappa)$ time, where $\kappa := |\mathsf{X}|$ is the number of intersecting pairs of edges in $\mathsf{E}_f \times \mathsf{E}_g$, in a straightforward manner if we have the overlay Σ of $\mathbf{T}_f$ and $\mathbf{T}_g$ at our disposal. In this section, we show how to "batch" the evaluations of Φ_{ijk}, using biclique partitions of X so that the total time spent is $\tilde{O}(\kappa^{(\omega-1)/2} n^{(3-\omega)/2})$. Recall that for an edge $e_i \in \mathsf{E}_f \cup \mathsf{E}_g$, e^*_{i1} (resp. e^*_{i2}) is the point in $\mathbb{R}^5$ corresponding to the linear function over the triangle lying below (resp. above) the line supporting e_i. We describe the algorithm for computing

$$\Phi_C(\mathsf{E}_f, \mathsf{E}_g) = \sum_{(e_1^*, e_2^*) \in \mathsf{X}} \sigma(e_1, e_2) \delta_C(e^*_{11}, e^*_{22}) R_C(e^*_{11}, e^*_{22}).$$

The other terms (those invoving R_D and R_E) can be evaluated in an analogous manner. For simplicity, we use e^*_i to denote e^*_{i1} (resp. e^*_{i2}) if e_i is a segment of E_f (resp. E_g). We compute the desired biclique partition of X in two stages. The first stage is similar to that described in [13], but the second stage is more involved and relies on geometric cuttings [9, 3].

First stage. As in Chazelle *et al.* [10] (see also [13]), we compute, in time $O(n \log^2 n)$, a *biclique partition* $\mathcal{B} = \{(R_1, B_1), \ldots, (R_s, B_s)\}$ of X such that

(i) for every $k \leq s$, $R_k \subseteq \mathsf{E}_f$ and $B_k \subseteq \mathsf{E}_g$ and every segment in R_k intersects every segment in B_k;

(ii) every segment in R_k has lower slope than that of every segment in B_k, or vice-versa;

(iii) for every intersecting pair $(e_1, e_2) \in \mathsf{X}$, there exists exactly one $k \leq s$ with $e_1 \in R_k$ and $e_2 \in B_k$; and

(iv) $\sum_k (|R_k| + |B_k|) = O(n \log^2 n)$ and $\sum_k |R_k| \times |B_k| = \kappa$.

For any $1 \leq k \leq s$, $\sigma(e^*_1, e^*_2)$ is the same for all $(e_1, e_2) \in R_k \times B_k$, which we denote by $\sigma(R_k, B_k)$. Therefore, we express $\Phi_C(\mathsf{E}_f, \mathsf{E}_g)$ in terms of $\mathcal{B}$ as follows:

$$\Phi_C(\mathsf{E}_f, \mathsf{E}_g) = \sum_{k=1}^{s} \sigma(R_k, B_k) \sum_{(e_1, e_2) \in R_k \times B_k} \delta_C(e^*_1, e^*_1) \cdot R_C(e^*_1, e^*_2). \tag{19}$$

We fix a biclique $(R_k, B_k) \in \mathcal{B}$. Without loss of generality, assume that $\sigma(R_k, B_k) = 1$. We describe how to compute

$$\Phi_C(R_k, B_k) = \sum_{(e_1, e_2) \in R_k \times B_k} \delta_C(e^*_1, e^*_2) \cdot R_C(e^*_1, e^*_2).$$

Second stage. We thus have the following problem at hand. Let $\mathsf{R} \subseteq \mathsf{E}_f$ and $\mathsf{B} \subseteq \mathsf{E}_g$ be two sets of segments in $\mathbb{R}^2$ such that every pair of segments in $\mathsf{R} \times \mathsf{B}$ intersects and the slope of every segment in R is less than that of any segment in B; set $m = |\mathsf{R}|$ and $n = |\mathsf{B}|$. Without loss of generality assume that $m \leq n$. We compute a *sign-invariant* (biclique) partition $\Pi = \{(\mathsf{R}_1, \mathsf{B}_1), \ldots, (\mathsf{R}_u, \mathsf{B}_u)\}$ of $\mathsf{R} \times \mathsf{B}$, as decsribed below, such that for all $1 \leq k \leq s$, the value of $\delta_C(e^*_i, e^*_j)$ is the same for all pairs $(e_i, e_j) \in (\mathsf{R}_k, \mathsf{B}_k)$, which we denote by $\delta_C(\mathsf{R}_k, \mathsf{B}_k)$. Let $\Pi^+ = \{(\mathsf{R}_k, \mathsf{B}_k) \mid \delta(\mathsf{R}_k, \mathsf{B}_k) = 1\}$ and $\Pi^- = \Pi \setminus \Pi^+$. Then

$$\Phi_C(\mathsf{R}, \mathsf{B}) = \sum_{(\mathsf{R}_k, \mathsf{B}_k) \in \Pi^+} \sum_{(e_i, e_j) \in \mathsf{R}_k \times \mathsf{B}_k} R_C(e^*_i, e^*_j) - \sum_{(\mathsf{R}_k, \mathsf{B}_k) \in \Pi^-} \sum_{(e_i, e_j) \in \mathsf{R}_k \times \mathsf{B}_k} R_C(e^*_i, e^*_j)\,. \tag{20}$$

We show below in Section 4 how $\Phi_C(\mathsf{R}_k, \mathsf{B}_k) = \sum_{(e_i,e_j)\in \mathsf{R}_k\times \mathsf{B}_k} \Phi_C(e_i^*, e_j^*)$ can be computed efficiently. It thus suffices to describe how we compute Π and to analyze the overall run time.

By eq. (10), δ_C is the sign function of a bilinear function $P(\boldsymbol{\varphi}, \boldsymbol{\psi})$, where $\boldsymbol{\varphi}$ (resp. $\boldsymbol{\psi}$) is a set of five variables representing the parameters $y_i, z_i, \alpha_i, \beta_i, \gamma_i$ corresponding to the edges e_i of R (resp. B) used in the definition of δ_C; note that four of these parameters are independent and $\gamma_i = z_i\alpha_i - \beta_i y_i$ depends on them. Abusing the notation slightly, we continue to represent an edge $e_j \in \mathsf{R}$ as a point e_j^* in $\mathbb{R}^5$ with the new set of parameters. We also map e_j to a hyperplane $\tilde{e}_j$ in $\mathbb{R}^5$, where $\tilde{e}_j \colon P(e_j^*, \boldsymbol{\psi}) = 0$. Similarly, we map an edge $e_k \in \mathsf{B}$ to a point $e_k^* \in \mathbb{R}^5$ and a hyperplane $\tilde{e}_k \colon P(\boldsymbol{\varphi}, e_k^*) = 0$ in $\mathbb{R}^5$. For a subset A of R (or B), we define $\mathsf{A}^* = \{e_j^* \mid e_j \in \mathsf{A}\}$ and $\widetilde{\mathsf{A}} = \{\tilde{e}_j \mid e_j \in \mathsf{A}\}$. Our goal is to compute a partition Π of $\mathsf{R}\times\mathsf{B}$ such that for any pair $(\mathsf{R}_i, \mathsf{B}_i) \in \Pi$, all points of B_i^* lie on the same side of all hyperplanes in $\widetilde{\mathsf{R}}_i$, or equivalently all points of R_i^* lie on the same side of all hyperplanes in $\widetilde{\mathsf{B}}_i$.

Let H be a set of m hyperplanes in $\mathbb{R}^d$, Δ be a simplex, and χ be the number of vertices of the arrangement $\mathcal{A}(\mathsf{H})$ of H inside Δ. For a parameter $r > 1$, a partition of Δ into a family Ξ of simplices, referred to as *cells*, is called a $(1/r)$-*cutting* of H within Δ if every cell of Ξ is crossed by at most m/r hyperplanes of H. (For $r > m$, cells of Ξ are not crossed by any hyperplane of H, i.e., Ξ is a refinement of $\mathcal{A}(\mathsf{H})$.) The *conflict list* of a cell $\triangle \in \Xi$, denoted by $\mathsf{H}_\triangle$, is the subset of hyperplanes that cross $\triangle$. It is well known that a $(1/r)$-cutting of H of size $O(r^d)$ along with the conflict lists of all of its cells can be computed in $O(mr^{d-1})$ time [9, 12]. For any point set $P \subset \mathbb{R}^d$, one can compute $P \cap \triangle$ for every $\triangle \in \Xi$ in additional $O(|P| \log r)$ time.

In our context, we compute the desired sign-invariant partition Π of $\mathsf{R}\times\mathsf{B}$ using geometric cuttings in a round-robin manner as in [4, 12]. We choose r to be a sufficiently large constant, see below. We assume that the input is *balanced*, i.e., $n/2 \le m \le n$. Otherwise (i.e. $m < n/2$), we use the standard batching technique—we partition B into $\ell = \lceil n/m \rceil$ subsets $\mathsf{B}_1, \ldots, \mathsf{B}_\ell$, each of size between m and $2m$, and compute the desired partition for each pair $(\mathsf{R}, \mathsf{B}_j)$, $1 \le j \le \ell$, independently.

Each round of the algorithm works in two stages. In the first stage, we map R to the set $\widetilde{\mathsf{R}}$ of hyperplanes in $\mathbb{R}^5$ and compute a $(1/r)$-cutting Ξ of size $O(r^5)$. For each cell $\triangle \in \Xi$, let $\widetilde{\mathsf{R}}_\triangle \subseteq \widetilde{\mathsf{R}}$ be the conflict list of $\triangle$, let $\widetilde{\mathsf{R}}^+_\triangle$ (resp. $\widetilde{\mathsf{R}}^-_\triangle$) be the set of hyperplanes of $\widetilde{\mathsf{R}}$ that lie above (resp., below) $\triangle$. Let $\mathsf{R}_\triangle, \mathsf{R}^+_\triangle, \mathsf{R}^-_\triangle \subseteq \mathsf{R}$ be the sets of segments correspondig to $\widetilde{\mathsf{R}}_\triangle, \widetilde{\mathsf{R}}^+_\triangle, \widetilde{\mathsf{R}}^-_\triangle$, respectively. Let $\mathsf{B}^*_\triangle = \mathsf{B}^* \cap \triangle$ and $\mathsf{B}_\triangle = \{e_j \mid e_j^* \in \mathsf{B}^*_\triangle\}$. Set $m_\triangle = |\mathsf{R}_\triangle|$ and $n_\triangle = |\mathsf{B}_\triangle|$. If $n_\triangle \ge n/r^5$ for any cell $\triangle$, we partition $\triangle$ further into subcells each of which contains at most n/r^5 points of B^*. The number of cells even after this refinement is at most $c_0 r^5$ for some constant $c_0 > 0$. For every cell $\triangle \in \Xi_i$, we add $(\mathsf{R}^+_\triangle, \mathsf{B}_\triangle)$ and $(\mathsf{R}^-_\triangle, \mathsf{B}_\triangle)$ as bicliques to the partition Π. This completes the description of the first stage. which takes $O(mr^4 + n\log r) = O(n)$ since $r = O(1)$ and $m \le n$.

In the second stage, for each cell $\triangle \in \Xi$, we map $\mathsf{B}_\triangle$ to the set $\widetilde{\mathsf{B}}_\triangle$ of hyperplanes and compute a $(1/r)$ cutting $\Xi_\triangle$ of $\widetilde{\mathsf{B}}_\triangle$ of size $O(r^5)$. For a cell $\tau \in \Xi_\triangle$, we define $\mathsf{B}_\tau, \mathsf{B}^+_\tau, \mathsf{B}^-_\tau, \mathsf{R}_\triangle$ as in the first stage (but reversing the roles of R and B). If $|\mathsf{R}_\tau| \le m/r^6$, then we split τ into subcells each of which contains at most m/r^6 points of R_τ. Let $m_\tau = |\mathsf{R}_\tau|$ (after the refinement) and $n_\tau = |\mathsf{B}_\tau|$. By construction, $m_\tau \le m/r^6$ and $n_\tau \le n_\triangle/r \le n/r^6$ for all $\tau \in \Xi_\triangle$ and for all $\triangle \in \Xi$. For every $\triangle \in \Xi$ and $\tau \in \Xi_\triangle$, we add $(\mathsf{R}_\tau, \mathsf{B}^+_\tau)$ and $(\mathsf{R}_\tau, \mathsf{B}^-_\tau)$ as bicliques to the partition Π.

For every cell τ of a second-stage cutting, we recursively compute a partition of $(\mathsf{R}_\tau, \mathsf{B}_\tau)$. The recursion stops when $\min\{|\mathsf{R}_\tau|, |\mathsf{B}_\tau|\} < r$, in which case we add each pair $(e_i, e_j) \in \mathsf{R}_\tau \times \mathsf{B}_\tau$ as a singleton biclique to Π. This completes the description of computing Π. The correctness follows using a standard argument, see, e.g. [3].

Finally, for each biclique $(\mathsf{R}_k, \mathsf{B}_k) \in \Pi$, we compute $\Phi_C(\mathsf{R}_k, \mathsf{B}_k)$ using Corollary 7. This com-

pletes the description of the algorithm.

Analysis We now analyze the overall run time of the algorithm, including the time spent in computing $\Phi_C(\mathsf{R}_k, \mathsf{B}_k)$ for each clique $(\mathsf{R}_k, \mathsf{B}_k) \in \Pi$. First, we focus on the run time for the balanced case. For notational simplcity, assume that $|\mathsf{R}| = |\mathsf{B}| = n$. Let $T(n)$ be the total time spent in computing the partition Π plus the time spent in computing $\Phi_C(\cdot, \cdot)$ for each biclique in Π.

If $n < n_0$, where n_0 is a constant that depends on r, the algorithm takes $O(1)$ time. Otherwise, the algorithm spends $O(n)$ time in the two stages of the first round to compute the cuttings, the conflict lists, and the bicliques. Then it solves at most $c_1 r^{10}$ recursive problems, where $c_1 > 0$ is a constant independent of r, each of size at most n/r^6. Furthermore, the first round generates at most $O(r^{10})$ bicliques, and by Corollary 7, computing Φ_C for each biclique takes $\tilde{O}(n^{(\omega+1)/2})$. Thus the total time spent in the first round in computing the bicliques and Φ_C for all these is at most $c_3 n^{(\omega+1)/2} \log^{c_2} n$, where $c_2 > 0$ is a constant independent on r and $c_3 > 0$ is a constant that depends on r. Hence, we obtain the following recurrence for $T(n)$:

$$T(n) \leq \begin{cases} c_1 r^{10} T(n/r^6) + c_3 n^{(\omega+1)/2} \log^{c_2} n & n \geq n_0, \\ c_4 & n < n_0, \end{cases} \tag{21}$$

where $c_4 > 0$ and $n_0 > 0$ are constants that depend on r. Note that the recursive subproblems may not be balanced but the total time spent in a round with subproblem size $t \geq n_0$ is still at most $c_3 t^{(\omega+1)/2} \log^{c_2} t$, so the above recurrence holds.

Assuming r is chosen sufficiently large, the solution of the above recurrence is

$$T(n) \leq A n^{\alpha} \log^{c_2} n, \qquad \text{where } \alpha = \max\{5/3 + \varepsilon, (\omega+1)/2\} \tag{22}$$

for any constant $\varepsilon > 0$, where A is an appropriate constant that depends on r. Since the current best known value of ω is roughly 2.37, we simply write the solution of the recurrence as $T(n) = \tilde{O}(n^{(\omega+1)/2})$. We remark that the first term in the definition of α can be reduced to $8/5 + \varepsilon$ by observing that points of $\mathsf{R}^* \cup \mathsf{B}^*$ lie on the surface $\gamma_i = z_i \alpha_i - \beta_i y_i$ and thus only $O(r^4 \log r)$ cells of Ξ (resp. $\Xi_\triangle$) contain a point of B (resp. $\mathsf{R}_\triangle$) (see e.g. [7, 1]), and thus we recursively solve $O(r^8 \log^2 r)$ subproblems, each of size at most n/r^5, after the first round.

For general values of m and n with $m \leq n$, the batching technique implies that the total time spent in computing $\Phi_C(\mathsf{R}, \mathsf{B})$ is $(1 + n/m) \cdot \tilde{O}(m^{(\omega+1)/2}) = \tilde{O}(nm^{(\omega-1)/2})$. For $m > n$, we flip the roles of R and B, and the run time is $\tilde{O}(mn^{(\omega-1)/2})$. Hence, we obtain the following:

Lemma 4. *For a biclique (R_k, B_k) in the biclique partition computed in the first stage, with $|R_k| = m_k$ and $|B_k| = n_k$, $\Phi_C(R_k, B_k)$ can be computed in $\tilde{O}(m_k n_k^{(\omega-1)/2} + n_k m_k^{(\omega-1)/2})$ time.*

To bound the total time spent in computing $\Phi(\mathsf{E}_f, \mathsf{E}_g)$, we sum the bound in Lemma 4 over all bicliques of $\mathcal{B}$. Using Hölder's inequality and property (iv) of the biclique partition $\mathcal{B}$, we obtain the following:

$$\begin{aligned} \sum_{k=1}^{s} m_k n_k^{(\omega-1)/2} &= \sum_{k=1}^{s} (m_k n_k)^{(\omega-1)/2} m_k^{(3-\omega)/2} \\ &\leq \left(\sum_{k=1}^{s} m_k n_k \right)^{(\omega-1)/2} \cdot \left(\sum_{k=1}^{s} m_k \right)^{(3-\omega)/2} \\ &= \tilde{O}(\kappa^{(\omega-1)/2} n^{(3-\omega)/2}), \end{aligned}$$

which is $\tilde{O}(n)$ for $\kappa = n$ and $\tilde{O}(n^{(\omega+1)/2})$ for $\kappa = n^2$. Similarly, we can argue that $n_k m_k^{(\omega-1)/2} = \tilde{O}(\kappa^{(\omega-1)/2} n^{(3-\omega)/2})$. Adding the $O(n \log n)$ time spent in evaluating (4) at the vertices of $\mathsf{V}_f \cup \mathsf{V}_g$, the total time spent in computing $\|f - g\|_1$ is $\tilde{O}(\kappa^{(\omega-1)/2} n^{(3-\omega)/2})$. This proves Theorem 1.

4 Multipoint Evaluation of Bivariate Rational Functions

In this section, we show how computing the quantities $\Phi_C(\mathsf{R}, \mathsf{B})$ associated to each $m \times n$ biclique $\mathsf{R} \times \mathsf{B}$ can be reduced to computing a double sum of m rational functions $\frac{1}{q_i(x,y)}$ over n points p_j. This sum can then be evaluated using fast multipoint evaluation techniques for bivariate polynomials ([14]), based on fast modular composition ([18, Chapter 12, §12.2]). As in the previous section, the other terms appearing in $\Phi(\mathsf{E}_f, \mathsf{E}_g)$ can be computed in an analogous manner. We start with a technical lemma on the fast multipoint evaluation of bivariate polynomials.

Lemma 5. *Given n polynomials $f_i(x, y)$ of degree n and n^2 points $p_j \in \mathbb{R}^2$, it is possible to evaluate all the $f_i(p_j)$ in $\tilde{O}(n^{\omega+1})$ arithmetic operations.*

Proof. The main idea is to reduce this question to a problem of fast multipoint evaluation of univariate polynomials.

First, we interpolate two polynomials $g(x)$ and $h(x)$ of degree n^2, such that, for each root r of h, the point $(r, g(r))$ is a point p_i. More formally we require that

$$\{p_1, \ldots, p_n\} = \{(x_i, g(x_i)) \mid h(x_i) = 0\}.$$

(For clarity of presentation, we assume that all the coordinates x_i are distinct; the general case can be handled by introducing a generic shear transformation of the xy-plane at no asymptotic cost; see [14, §6].) Polynomials $g(x)$ and $h(x)$ can be computed in $\tilde{O}(n^2)$ operations using fast interpolation algorithms [18, Chapter 10.2].

Then, denoting by $\widetilde{f_j}(x)$ the univariate polynomial $f_j(x, g(x)) \bmod h(x)$, note that $f_j(x_i, y_i) = \widetilde{f_j}(x_i)$ for all i. Hence, if we know the coefficients of $\widetilde{f_j}$, we can evaluate each $\widetilde{f_j}$ at all x_i in $\tilde{O}(n^2)$ arithmetic operations, and all $\widetilde{f_j}$ at all x_i in $\tilde{O}(n^3)$ operations.

Finally, computing all the $\widetilde{f_j}$ can be done in $\tilde{O}(n^{\omega+1})$ arithmetic operations using Lemma 10(iii) of [14], which can be seen as a variation of the fast composition algorithm 12.3 in [18, Chapter 12, §12.2]. Thus, the final complexity is in $\tilde{O}(n^{\max(3,\omega+1)}) = \tilde{O}(n^{\omega+1})$. □

We can now deduce our corollaries for computing the sum of rational functions. First we start with the case where the numerators are all 1.

Corollary 6. *Given n constant-degree polynomials $q_i(x, y)$ and n points $p_j \in \mathbb{R}^2$, the n sums $\sum_i \frac{1}{q_i(p_j)}$ for all $1 \le j \le n$ can be computed in $\tilde{O}(n^{\frac{\omega+1}{2}})$ arithmetic operations.*

Proof. First, partitioning the set of indices i into $O(\sqrt{n})$ subsets I_ℓ of size $O(\sqrt{n})$, for each ℓ, there exists a polynomial $f_\ell(x, y)$ of degree $O(\sqrt{n})$ such that $\sum_{i \in I_\ell} \frac{1}{q_i(x,y)} = \frac{f_\ell(x,y)}{\prod_{i \in I_\ell} q_i(x,y)}$. Each f_ℓ can be computed in $\tilde{O}(n)$ arithmetic operations, using fast bivariate multiplication [18, §8.4]. Then, using Lemma 5, we can evaluate all the $O(\sqrt{n})$ polynomials f_ℓ on all the n points p_j in $\tilde{O}(n^{\frac{\omega+1}{2}})$ operations. With the same approach, we can expand the denominators and evaluate them on the points p_j in $\tilde{O}(n^{\frac{\omega+1}{2}})$ operations. □

Then we generalize this result to sums of general rational functions.

Corollary 7. *For positive integers k, n, i, with $i \leq n$, let f_i and g_i be polynomials of constant degree in k and two variables, respectively. Let $A \subset \mathbb{R}^k$ contain at most n points. In the Real RAM model, the sum*

$$S = \sum_{(a_1,\ldots,a_k)\in A} \sum_{1\leq i\leq n} \frac{f_i(a_1,\ldots,a_k)}{g_i(a_1,a_2)}$$

can be computed in $\tilde{O}(n^{\frac{\omega+1}{2}})$ arithmetic operations.

Proof. Given k symbolic variables $X_1, \ldots, X_k$, all the polynomials $f_i(X_1, \ldots, X_k)$ can be written as a weighted sum of s monomials M_j, where $M_j = X_1^{e_{j,1}} \cdots X_k^{e_{j,k}}$ with $(e_{j,1}, \ldots, e_{j,k}) \in \mathbb{N}^k$. In particular, each polynomial $f_i(X_1, \ldots, X_k)$ can be expanded as

$$c_{i,1}M_1 + \cdots + c_{i,s}M_s\,.$$

According to Lemma 6, letting $q_i = \frac{g_i}{c_{i,1}}$, we can compute for all the points in $(a_1, \ldots, a_k) \in A$ the sums $\sum_{i=1}^{n} \frac{1}{q_i(a_1,a_2)}$ in $\tilde{O}(n^{\frac{\omega+1}{2}})$ operations. Since the number s of monomials is constant, we can compute with the same complexity for all $1 \leq j \leq s$ and all points $p = (a_1, \ldots, a_k) \in A$ the sums :

$$S_{p,j} := \sum_{i=1}^{n} \frac{c_{i,j}}{g_i(a_1,a_2)}\,.$$

On the other hand, since s and the degrees of the f_i are constant, we can directly compute all the monomials M_j at all the points of A in $O(n)$ operations. Using an additional number of operations linear in n, this allows us to compute, for all the points in $p \in A$,

$$S_p := \sum_{i=1}^{n} \frac{f_i(a_1,\ldots,a_k)}{g_i(a_1,a_2)} = \sum_{j=1}^{s} S_{p,j}M_j(p)\,.$$

Finally, we get the desired result by summing the quantities S_p over all $p \in A$. □

This concludes the description of the multipoint evaluation tools needed in Section 3 to complete our algorithm.

5 Conclusion

A recent unpublished preprint improves the bound on multipoint evaluation for multivariate polynomials. It shows that we could evaluate $n^{(\omega-1)}$ bivariate polynomials of degree n^2 on n^2 points in $\tilde{O}(n^{\omega+1})$ operations [16, Theorem 3.8]. Compared to Lemma 5 derived from earlier work [14], this allows us to evaluate more polynomials with the same number of operations. In turn, that would improve Corollary 7, and the complexity to evaluate the sum of rational functions on a biclique of size $n \times n$ would become $\tilde{O}(n^{2-1/\omega})$ instead of $O(n^{(\omega+1)/2})$, by summing over batches of $n^{2/\omega}$ points. With the current ω value, $2 - 1/\omega$ is approximately 1.58, which is below 8/5. Thus the cost of computing the biclique decomposition would become dominant, and the total cost of computing $\|f - g\|_1$ could be further improved to $O(n^{8/5+\varepsilon})$, for any constant $\varepsilon > 0$.

References

[1] P. K. Agarwal. Simplex range searching and its variants: A review. In *Journey through Discrete Mathematics: A Tribute to Jiří Matoušek*, pages 1–30. Springer, Berlin, 2017.

[2] Pankaj K. Agarwal, Boris Aronov, Marc Van Kreveld, Maarten Löffler, and Rodrigo I. Silveira. Computing correlation between piecewise-linear functions. *SIAM J. Comput.*, 42(5):1867–1887, 2013. URL: `semanticscholar.org/paper/bc51a9cc1639e635c4cdbb9fe421ec1c3dadf7b9`, `doi:10.1137/120900708`.

[3] Pankaj K. Agarwal, Esther Ezra, and Micha Sharir. Semi-algebraic off-line range searching and biclique partitions in the plane. In Wolfgang Mulzer and Jeff M. Phillips, editors, *40th International Symposium on Computational Geometry, SoCG 2024, June 11-14, 2024, Athens, Greece*, pages 4:1–4:15, 2024.

[4] Pankaj K. Agarwal and Micha Sharir. The number of congruent simplices in a point set. *Discret. Comput. Geom.*, 28(2):123–150, 2002.

[5] Deepak Ajwani, Saurabh Ray, Raimund Seidel, and Hans Raj Tiwary. On computing the centroid of the vertices of an arrangement and related problems. In *Algorithms and data structures. 10th international workshop, WADS 2007, Halifax, Canada, August 15–17, 2007. Proceedings.*, pages 519–528. Berlin: Springer, 2007. `doi:10.1007/978-3-540-73951-7_45`.

[6] Boris Aronov and Matthew J. Katz. Batched point location in SINR diagrams via algebraic tools. *ACM Trans. Algorithms*, 14(4), August 2018. `doi:10.1145/3209678`.

[7] Saugata Basu, Richard Pollak, and Marie-Françoise Roy. On the number of cells defined by a family of polynomials on a variety. *Mathematika*, 43(1):120–126, 1996. `doi:10.1112/S0025579300011621`.

[8] Sergio Cabello and Timothy M. Chan. Computing Shapley values in the plane. *Discrete Comput. Geom.*, 67(3):843–881, 2022. URL: `drops.dagstuhl.de/opus/volltexte/2019/10424/`, `doi:10.1007/s00454-021-00368-3`.

[9] Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discret. Comput. Geom.*, 9:145–158, 1993. `doi:10.1007/BF02189314`.

[10] Bernard Chazelle, Herbert Edelsbrunner, Leonidas J. Guibas, and Micha Sharir. Algorithms for bichromatic line-segment problems and polyhedral terrains. *Algorithmica*, pages 116–132, 1994.

[11] Charles M. Fiduccia. Polynomial evaluation via the division algorithm: The fast Fourier transform revisited. Proc. 4th ann. ACM Symp. Theory Comput., Denver 1972, 88-93 (1972)., 1972.

[12] Jiří Matoušek. Range searching with efficient hierarchical cuttings. *Discrete & Computational Geometry*, 10(2):157–182, 1993.

[13] Guillaume Moroz and Boris Aronov. Computing the distance between piecewise-linear bivariate functions. *ACM Trans. Algorithms*, 12(1), 2016.

[14] Michael Nüsken and Martin Ziegler. Fast multipoint evaluation of bivariate polynomials. In Susanne Albers and Tomasz Radzik, editors, *Algorithms – ESA 2004*, pages 544–555, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[15] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry: An Introduction*. Springer, Berlin, 1985.

[16] Joris van der Hoeven and Grégoire Lecerf. Faster multi-point evaluation over any field. working paper or preprint, November 2024. URL: `https://hal.science/hal-04774026`.

[17] Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 3792–3835. SIAM, 2024. `doi:10.1137/1.9781611977912.134`.

[18] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 3rd edition, 2013.