



HAL
open science

Exact computation of the matching distance on 2-parameter persistence modules

Michael Kerber, Michael Lesnick, Steve Oudot

► **To cite this version:**

Michael Kerber, Michael Lesnick, Steve Oudot. Exact computation of the matching distance on 2-parameter persistence modules. SoCG 2019 - International Symposium on Computational Geometry, Jun 2019, Portland, Oregon, United States. hal-01966666

HAL Id: hal-01966666

<https://inria.hal.science/hal-01966666>

Submitted on 21 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exact computation of the matching distance on 2-parameter persistence modules

Michael Kerber¹

Graz University of Technology, Graz, Austria

kerber@tugraz.at

 <https://orcid.org/0000-0002-8030-9299>

Michael Lesnick

University at Albany, SUNY

mlesnick@albany.edu

 <https://orcid.org/0000-0003-1924-3283>

Steve Oudot

Inria Saclay – Île-de-France, Palaiseau, France

steve.oudot@inria.fr

 <https://orcid.org/0000-0003-2939-9417>

1 Abstract

2 The matching distance is a pseudometric on multi-parameter persistence modules, defined in
3 terms of the weighted bottleneck distance on the restriction of the modules to affine lines. It
4 is known that this distance is stable in a reasonable sense, and can be efficiently approximated,
5 which makes it a promising tool for practical applications. In this work, we show that in the
6 2-parameter setting, the matching distance can be computed exactly in polynomial time. Our
7 approach subdivides the space of affine lines into regions, via a line arrangement. In each region,
8 the matching distance restricts to a simple analytic function, whose maximum is easily computed.
9 As a byproduct, our analysis establishes that the matching distance is a rational number, if the
10 bigrades of the input modules are rational.

2012 ACM Subject Classification Mathematics of computing → Algebraic Topology; Mathematical Optimization → Continuous Optimization

Keywords and phrases Topological Data Analysis, Multi-Parameter Persistence, Line arrangements

Lines 504

11 **1** Introduction

12 Multi-parameter persistent homology is receiving growing attention, both from the theoret-
13 ical and computational points of view. Its motivation lies in the possibility of extending
14 the success of topological data analysis to settings where the structure of data is best cap-
15 tured by 2-parameter rather than 1-parameter constructions. The basic algebraic objects
16 of study in multi-parameter persistence are certain commutative diagrams of vector spaces
17 called *persistence modules*. In the 1-parameter setting, persistence modules decompose in an
18 essentially unique way into simple summands called *interval modules*. The decomposition
19 is specified by a discrete invariant called a *persistence diagram*. In contrast, the algebraic
20 structure of a 2-parameter persistence module (henceforth, *bipersistence module*) can be

¹ Supported by Austrian Science Fund (FWF) grant number P 29984-N35



21 far more complex. As a result, a good definition of persistence diagram is unavailable for
 22 bipersistence modules [5].

23 Nevertheless, it is still possible to define meaningful notions of distance between multi-
 24 parameter persistence modules. Distances on 1-parameter persistence modules play an es-
 25 sential role in both theory and applications. To extend such theory and applications to the
 26 multi-parameter setting, one needs to select a suitable distance on multi-parameter persis-
 27 tence modules. However, progress on finding well-behaved, efficiently computable distances
 28 on multi-parameter persistence modules has been slow, and this is been an impediment to
 29 progress in practical applications.

30 The most widely studied and applied distances in the 1-parameter setting are the *bot-*
 31 *tleneck distance* and the *Wasserstein distance* [10]. Both can be efficiently computed via
 32 publicly available code [13]. In the multi-parameter setting, the distance that has received
 33 the most attention is a generalization of the bottleneck distance called the *interleaving*
 34 *distance*. This distance is theoretically well-behaved; in particular, among all distances
 35 satisfying a certain stability condition, it is the most discriminative distance on modules
 36 over prime fields [15]. However, it was proven recently that the interleaving distance on
 37 bipersistence modules is NP-hard to compute, and even to approximate to any constant
 38 factor less than three [4]. This motivates the search for a more computable surrogate for
 39 the interleaving distance.

40 The *matching distance*, introduced by Cerri et al. [6], is a natural candidate for such a
 41 surrogate. It is a lower bound for the interleaving distance; this is implicit in [6] and shown
 42 explicitly in [14]. In the 2-parameter setting, the matching distance is defined as follows:
 43 Given a pair of bipersistence modules, we call an affine line ℓ in parameter space with
 44 positive slope a *slice*. Restricting the modules to ℓ yields a pair of 1-parameter persistence
 45 modules, which we call *slice modules*. These slice modules have a well-defined bottleneck
 46 distance, which we multiply by a positive weight depending only on the slope of ℓ . (The
 47 weights are chosen in a way that ensures that the matching distance is a lower bound for the
 48 interleaving distance.) The matching distance is defined as the supremum of these weighted
 49 bottleneck distances over all slices. See Section 3 for the precise definition. The definition
 50 generalizes readily to n -parameter persistence modules, for any $n \geq 1$; when $n = 1$, the
 51 matching distance is equal to the bottleneck distance.

52 As Cerri et al. have observed, to approximate the matching distance up to any (ab-
 53 solute) precision, it suffices to sample the space of slices sufficiently densely and to return
 54 the maximum weighted bottleneck distance encountered. For a constant number of scale
 55 parameters and approximation quality ϵ , a polynomial number of slices are sufficient in
 56 terms of module size and $\frac{1}{\epsilon}$, yielding a polynomial time approximation algorithm. [3]. This
 57 approach has been recently applied to the virtual ligand screening problem in computational
 58 chemistry [12]. To the best of our knowledge, there is no other previous work in which the
 59 problem of computing the matching distance has been considered.

60 **Our contribution.** We give an algorithm that computes the exact matching distance
 61 between a pair of bipersistence modules in time polynomial with respect to the size of the
 62 input. We assume that each persistence module is specified by a *presentation*. Concretely,
 63 this means that the module is specified by a matrix, with each row and each column labeled
 64 by a point in \mathbb{R}^2 ; see Section 2.

65 To explain our strategy for computing the matching distance, consider the function F
 66 that assigns a slice to its weighted bottleneck distance. The matching distance is then
 67 simply the supremum of F , taken over all slices. F has a rather complicated structure, since
 68 it depends on the longest edge of a perfect matching in a bipartite graph whose edges lengths

69 depend on both the slice and the two modules given as input. When the slice changes, the
 70 matching realizing the bottleneck distance undergoes combinatorial changes, making the
 71 function F difficult to treat analytically.

72 We show, however, that the space of slices can be divided into polynomially many regions
 73 so that the restriction of F to each region takes a simple closed form. Perhaps surprisingly, if
 74 we parameterize the space of slices as a subset $\Omega \subset \mathbb{R}^2$, the boundary between these regions
 75 can be expressed by the union of polynomially many lines in Ω , making each region convex
 76 and bounded by (possibly unbounded) line segments. (This is analogous to the observation of
 77 [17] that for a single persistence module, the locus of lines where the combinatorial structure
 78 underlying the slice module can change is described by a line arrangement.) Moreover, the
 79 restriction of F to each cell attains its supremum at a boundary vertex of the cell, or as the
 80 limit of an unbounded line segment if the cell touches the boundary of Ω ; this follows from
 81 a straightforward case analysis. These observations together lead to a simple polynomial
 82 time algorithm to compute the matching distance.

83 The characterization of the matching distance underlying our algorithm also makes clear
 84 that if the row and column labels of the presentations of the input modules have rational
 85 coordinates, then the matching distance is rational as well. We are not aware of a simpler
 86 argument for this property.

87 **Outline.** We introduce the underlying topological concepts in Section 2, and introduce
 88 the matching distance in Section 3. We define the line arrangement subdividing the slice
 89 space in Section 4 and give the algorithm to maximize each cell of the arrangement in
 90 Section 5. We conclude in Section 6.

91 2 Persistence modules

92 **Single-parameter modules.** Let \mathbb{K} be a fixed finite field throughout. A *persistence*
 93 *module* M over \mathbb{R} is an assignment of \mathbb{K} -vector spaces M_x to real numbers x , and linear
 94 maps $M_{x \rightarrow y} : M_x \rightarrow M_y$ to a pair of real numbers $x \leq y$, such that $M_{x \rightarrow y}$ is the identity
 95 and $M_{x \rightarrow y} \circ M_{y \rightarrow z} = M_{x \rightarrow z}$. Equivalently, in categorical terms, a persistence module is a
 96 functor from \mathbb{R} (considered as a poset category) to the category of \mathbb{K} -vector spaces.

97 A common way to arrive at a persistence module is to consider a nested sequence of
 98 simplicial complexes

$$99 \quad X_1 \subseteq X_2 \subseteq \dots \subseteq X_n$$

100 and to apply homology with respect to a fixed dimension and base field \mathbb{K} . This yields a
 101 sequence

$$102 \quad H_p(X_1, \mathbb{K}) \rightarrow H_p(X_2, \mathbb{K}) \rightarrow \dots \rightarrow H_p(X_n, \mathbb{K})$$

103 of vector spaces and linear maps. To obtain a persistence module over \mathbb{R} , we pick *grades*
 104 $s_1 < s_2 < \dots < s_n$ and set $M_x := 0$ if $x < s_1$ and $M_x := H_p(X_i, \mathbb{K})$ with $i = \max\{j \mid s_j \leq x\}$
 105 otherwise. For $y \geq x$ and $M_y = H_p(X_j, \mathbb{K})$, we define $M_{x \rightarrow y}$ as the map $H_p(X_i, \mathbb{K}) \rightarrow$
 106 $H_p(X_j, \mathbb{K})$ induced by the inclusion map $X_i \rightarrow X_j$.

107 **Finite presentations.** In this work, we restrict our attention to persistence modules
 108 that are finitely presented in the following sense. A *finite presentation* is an $\ell \times m$ matrix P
 109 over \mathbb{K} , where each row and each column is labeled by a number in \mathbb{R} , called the *grade*, such
 110 that if $P_{ij} \neq 0$, then $\text{gr}(\text{row}_i) \leq \text{gr}(\text{col}_j)$; here $\text{gr}(-)$ denotes the grade of a row or column.
 111 We refer to the multiset of grades of all rows and columns of P simply as the *set of grades of*
 112 P , and denote this set as $\text{gr}(P)$. A finite presentation gives rise to a persistence module, as

XX:4 Exact computation of the matching distance on 2-parameter persistence modules

113 we describe next. The rows of P represent the generators of the module, while the columns
 114 of P encode relations (or syzygies) on the generators. Concretely, let e_1, \dots, e_l denote the
 115 standard basis of \mathbb{K}^l , and for $x \in \mathbb{R}$ define the subspace

$$116 \quad \text{Gen}_x := \{e_i \mid \text{gr}(\text{row}_i) \leq x\}$$

117 Likewise, define

$$118 \quad \text{Rel}_x := \{\text{col}_j \mid \text{gr}(\text{col}_j) \leq x\}$$

119 Then, we define

$$120 \quad M_x^P := \text{span}(\text{Gen}_x) / \text{span}(\text{Rel}_x)$$

121 and $M_{x \rightarrow y}^P$ simply as the map induced by the inclusion map $\text{span}(\text{Gen}_x) \rightarrow \text{span}(\text{Gen}_y)$. It
 122 can be checked easily that this indeed defines a persistence module M^P . If a persistence
 123 module N is isomorphic to M^P , we say that P is a *presentation of N* . We call a persistence
 124 module N *finitely presented* if there exists a finite presentation of N . For instance, persis-
 125 tence modules as above arising from a finite simplicial filtration are finitely presented. Also,
 126 the representation theorem of persistence [19, 7] states that the category of persistence mod-
 127 ules over \mathbb{K} is isomorphic to the category of graded \mathcal{R} -modules with an appropriately chosen
 128 ring \mathcal{R} . With that, a persistence module is finitely presented if and only if the corresponding
 129 \mathcal{R} -module is finitely presented (in the sense of a module).

130 For finite presentation via an $\ell \times m$ matrix as above, we call $n := \ell \cdot m$ the *size* of that
 131 presentation.

132 **Persistence diagrams.** A *persistence diagram* is a finite multi-set of points of the form
 133 $(b, d) \in \mathbb{R} \times (\mathbb{R} \cup \{\infty\})$ with $b < d$. A well-known structure theorem tells us that we can
 134 associate to any finitely presented persistence module M a persistence diagram $D(M)$, and
 135 this determines M up to isomorphism [8].

136 Given a presentation of M , the persistence diagram can be computed by bringing the
 137 presentation matrix into echelon form. This process takes cubic time in the size of the
 138 presentation using Gaussian elimination [10, 19], or $O(n^\omega)$ time using fast matrix multi-
 139 plication, where $\omega \leq 2.373$ [18]. Elimination-based approaches to computing persistent
 140 homology perform very well in practice, and are routinely used to study real data.

141 **► Lemma 1.** *For P a finite presentation of a persistence module M ,*

- 142 1. *The x -coordinates of the points of $D(M)$ form a sub-multiset of the row grades of P ,*
- 143 2. *The y -coordinates of the points of $D(M)$ form a sub-multiset of the column grades of P .*

144 **Proof.** This follows from the correctness of the basic matrix reduction algorithm for com-
 145 puting persistent homology, as described in [19]. ◀

146 **Bottleneck distance.** Consider two persistence diagrams D_1 and D_2 and a bijection
 147 $\sigma : D'_1 \rightarrow D'_2$ for some $D'_1 \subseteq D_1$ and $D'_2 \subseteq D_2$. For $\delta > 0$, we define $\text{cost}(\sigma) := \max(A, B)$,
 148 where

$$149 \quad A = \max \{ \max(|a - c|, |b - d|) \mid \sigma((a, b)) = (c, d) \},$$

$$150 \quad B = \max \{ (b - a)/2 \mid (a, b) \in (D_1 \setminus D'_1) \cup (D_2 \setminus D'_2) \},$$

151 and it is understood that $\infty - \infty = 0$. We define the bottleneck distance d_B by

$$152 \quad d_B(D_1, D_2) = \min \{ \epsilon \mid \text{there exists a matching of cost } \epsilon \text{ between } D_1 \text{ and } D_2 \}.$$

153 For persistence modules M and N , we write $d_B(D(M), D(N))$ simply as $d_B(M, N)$.

154 ► **Lemma 2.** Let P^M and P^N be finite presentations of persistence modules M and N ,
 155 respectively. $d_B(M, N)$ is realized by

- 156 1. The difference of a grade of P^M and a grade of P^N ,
- 157 2. or half the difference of two grades in P^M ,
- 158 3. or half the difference of two grades in P^N .

159 **Proof.** This follows immediately from Lemma 1 and the definition of d_B . ◀

160 Given two finite persistence diagrams D, D' , we can compute $d_B(D, D')$ in time to
 161 $O(n^{1.5} \log n)$ [11]; see [13] for details, including a report on practical efficiency. Thus, the
 162 complexity of computing the bottleneck distance of two persistence modules is dominated
 163 by the computation of the persistence diagrams, and has worst-case complexity $O(n^\omega)$.

164 **Bipersistence modules.** The definitions of persistence modules and presentations
 165 extend to higher dimensions without problems. In the 2-parameter setting, this goes as
 166 follows: Define a partial order \leq on \mathbb{R}^2 by $p \leq q$ if $p_x \leq q_x$ and $p_y \leq q_y$. A *bipersistence*
 167 *module* is an assignment of \mathbb{K} -vector spaces M_p to points $p \in \mathbb{R}^2$, and linear maps $M_{p \rightarrow q} : M_p \rightarrow M_q$
 168 to pairs of points $p \leq q \in \mathbb{R}^2$, such that $M_{p \rightarrow p}$ is the identity and $M_{q \rightarrow r} \circ M_{p \rightarrow q} = M_{p \rightarrow r}$
 169 whenever $p \leq q \leq r$. In topological data analysis, 2-dimensional persistence modules
 170 typically arise by applying homology to a bifiltered simplicial complex

171 A *finite presentation* of a bipersistence module \mathbb{R}^2 is defined in the same way as for one-
 172 parameter persistence modules, except that the labels of each row/column are now elements
 173 of \mathbb{R}^2 , and the \leq relation appearing in the definition now means the partial order over \mathbb{R}^2 .
 174 From now on, we will assume that all bipersistence modules considered are finitely presented.

175 ► **Example 3.** Let M be the bipersistence module given by

$$176 \quad M_p = \begin{cases} \mathbb{K} & \text{if } p \in [0, 1) \times [0, 1) \\ 0 & \text{otherwise,} \end{cases} \quad M_{p \rightarrow q} = \begin{cases} \text{Id}_{\mathbb{K}} & \text{if } p, q \in [0, 1) \times [0, 1) \\ 0 & \text{otherwise.} \end{cases}.$$

177 Then a presentation of M is given by

$$178 \quad \begin{matrix} (1, 0) & (0, 1) \\ (0, 0) & \begin{bmatrix} 1 & 1 \end{bmatrix} \end{matrix}.$$

179 In topological data analysis, we do not typically have immediate access to a presentation
 180 of a bipersistence module M , but rather to a chain complex of bipersistence modules for
 181 which M is a homology module. However, it has recently been observed that from such
 182 a chain complex, a (minimal) presentation of M can be computed in cubic time [16]. The
 183 algorithm for this is practical, and has been implemented in the software package RIVET [9].

184 **3 The Matching distance**

185 **Slices.** We define a *slice* as a line $\ell : y = sx + t$ where s and t are real numbers with
 186 $s > 0$. Let $\lambda : \mathbb{R} \rightarrow \ell$ be an isometric parameterization of the slice, i.e. one such that
 187 $\|\lambda(y) - \lambda(x)\|_2 = |y - x|$ for all $x, y \in \mathbb{R}$. Concretely, such a parameterization is given
 188 by $\lambda(x) = \frac{1}{\sqrt{1+s^2}}(x, sx + t)$. Given a bipersistence module M and slice ℓ , we define a (1-
 189 parameter) persistence module M^ℓ via $M_x^\ell := M_{\lambda(x)}$, with its linear maps induced by M .
 190 We call M^ℓ a *slice module*. It is easy to check that if M is finitely presented, then so is M^ℓ .

191 **Matching distance.** For a slice $\ell : y = sx + t$, we define a weight

$$192 \quad w(\ell) := \begin{cases} \frac{1}{\sqrt{1+s^2}} & s \geq 1 \\ \frac{1}{\sqrt{1+\frac{1}{s^2}}} & 0 < s < 1 \end{cases}$$

193 Note that $w(\ell)$ is maximized for slices with slope 1, and gets smaller when the slope goes to
194 0 or to ∞ .

195 Let Λ denote the set of all slices. For two persistence modules M, N over \mathbb{R}^2 , we define
196 a function $F^{M,N} : \Lambda \rightarrow [0, \infty)$ via

$$197 \quad F^{M,N}(\ell) := w(\ell) \cdot d_B(M^\ell, N^\ell).$$

198 and we define the *matching distance* between M and N as $d_{\text{match}}(M, N) := \sup F^{M,N}$. As
199 noted in the introduction, the weights $w(\ell)$ are chosen to ensure that d_{match} is a lower bound
200 for the interleaving distance.

201 **► Lemma 4.** *Given two bipersistence modules M, N , the map $F^{M,N}$ is continuous.*

202 **Proof.** w is clearly continuous, so it suffices to show that the function $\ell \mapsto d_B(M^\ell, N^\ell)$ is
203 continuous. Let \mathcal{D} denote the metric space of all finite persistence diagrams, with metric
204 the bottleneck distance. It follows from [14, Theorem 2] that the map sending a slice ℓ to
205 the persistence diagram $D(M^\ell)$ is continuous with respect to the topology on \mathcal{D} . Thus the
206 map sending ℓ to the pair $(D(M^\ell), D(N^\ell))$ is also continuous. Moreover, the bottleneck
207 distance is clearly continuous as a map $\mathcal{D} \times \mathcal{D} \rightarrow [0, \infty)$, thanks to the triangle inequality.
208 Since the composition of continuous functions is continuous, it follows that the function
209 $\ell \mapsto d_B(M^\ell, N^\ell)$ is continuous. ◀

210 **4 The arrangement**

211 In what follows, we fix two bipersistence modules M, N and write the map $F^{M,N}$ simply as
212 F . Let $\Omega := (0, \infty) \times \mathbb{R}$ and $\alpha : \Omega \rightarrow \Lambda$ be the bijection sending (s, t) to the line $y = sx + t$.
213 Clearly, α parameterizes the set of slices. By a slight abuse of notation, we write the map
214 $F \circ \alpha$ simply as F .

215 In this section, we construct a line arrangement in Ω in such a way that it is simple
216 to compute $\sup F$ on each face. Recall that a *line arrangement* of Ω is the subdivision of
217 Ω into vertices, edges, and faces induced by a finite set of distinct lines L_1, \dots, L_n . The
218 vertices of the arrangement are the intersection points of (at least) two lines, the edges
219 are maximal connected subsets of lines not containing any vertex, and the faces are the
220 connected components of $\Omega \setminus \bigcup_{i=1}^n L_i$. Clearly, each vertex, edge, and face of the arrangement
221 is a convex set. The boundary of each face consists of a finite number of edges and vertices.

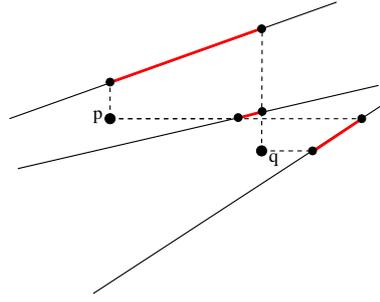
222 **A first line arrangement.** For $v \in \mathbb{R}^2$, let L_v denote the line $y = -v_x x + v_y$. Note that
223 $L_v \cap \Omega$ is exactly the set of parameterizations of slices containing v . Now, fix presentations
224 P^M and P^N of M and N . Let \mathcal{A}_0 denote the arrangement in Ω induced by the set of lines

$$225 \quad \{L_v \mid v \in \text{gr}(P^M) \cup \text{gr}(P^N)\}.$$

226 In what follows, we will refine \mathcal{A}_0 by adding more lines into the arrangement. For this
227 we first need to introduce some definitions.

228 **Pushes.** For a point $p = (p_x, p_y) \in \mathbb{R}^2$ and a slice $\ell : y = sx + t$, we define the *push of p*
229 *onto ℓ* as

$$230 \quad \text{push}(p, \ell) := \begin{cases} (p_x, s \cdot p_x + t) & \text{if } p \text{ lies below } \ell \\ (\frac{p_y - t}{s}, p_y) & \text{if } p \text{ lies on or above } \ell \end{cases}$$



234 **Figure 1** The pushes of two points p and q to three different slices. The length of the thick (red)
 235 line corresponds to the $\delta_{p,q}$ value of the corresponding slice.

231 Geometrically, $\text{push}(p, \ell)$ gives the intersection point of ℓ and a vertical upward ray emanat-
 232 ing from p in the first case, and the intersection of ℓ with a horizontal right ray emanating
 233 from p in the second case. See Figure 1 for an illustration.

236 A finite presentation of M induces a finite presentation of M^ℓ with the same underlying
 237 matrix, and each row or column grade $p \in \mathbb{R}^2$ replaced with $\lambda^{-1}(\text{push}(p, \ell))$. Clearly, this
 238 presentation can be obtained in linear time in the size of the presentation of M .

239 For $p, q \in \mathbb{R}^2$, define $\delta_{p,q} : \Omega \rightarrow [0, \infty)$ by

$$240 \quad \delta_{p,q}(s, t) := \|\text{push}(p, \ell) - \text{push}(q, \ell)\|_2 = |\lambda^{-1} \circ \text{push}(p, \ell) - \lambda^{-1} \circ \text{push}(q, \ell)|$$

241 with ℓ the slice defined by s and t . Again, see Figure 1 for an illustration.

242 We now give piecewise analytic formulae for $\delta_{p,q}$, which depend on whether the slice ℓ is
 243 above or below p and q .

(I) slice is above both p and q :

$$244 \quad \delta_{p,q}(s, t) = \left\| \left(\frac{p_y - t}{s}, p_y \right) - \left(\frac{q_y - t}{s}, q_y \right) \right\|_2 = \sqrt{\left(\frac{p_y - q_y}{s} \right)^2 + (p_y - q_y)^2} = |p_y - q_y| \sqrt{1 + \frac{1}{s^2}}.$$

(II) slice is below both p and q :

$$245 \quad \delta_{p,q}(s, t) = \left\| (p_x, sp_x + t) - (q_x, sq_x + t) \right\|_2 = \sqrt{(p_x - q_x)^2 + (sp_x - sq_x)^2} = |p_x - q_x| \sqrt{1 + s^2}.$$

246 (III) slice is between p and q : There are two subcases, which we will call (IIIa) and (IIIb):

247 Assuming p lies above the slice (IIIa), the formula is

$$248 \quad \begin{aligned} \delta_{p,q}(s, t) &= \left\| \left(\frac{p_y - t}{s}, p_y \right) - (q_x, sq_x + t) \right\|_2 = \sqrt{\left(\frac{p_y - t}{s} - q_x \right)^2 + (p_y - sq_x - t)^2} \\ 249 \quad &= \sqrt{\frac{1}{s^2} (p_y - t - sq_x)^2 + (p_y - sq_x - t)^2} = |p_y - t - sq_x| \sqrt{1 + \frac{1}{s^2}}. \end{aligned}$$

250 If p lies below the slice (IIIb), the formula is the same, except with the roles of p and q
 251 exchanged.

252 The push map is easily seen to be continuous with respect to the slice ℓ , so these formulae
 253 also extend to boundaries of the cases, i.e., when the slice contains p or q .

254 **► Lemma 5.** *If $p, q \in \text{gr}(P^M) \cup \text{gr}(P^N)$, then in each face of \mathcal{A}_0 , exactly one of the conditions*
 255 *(I), (II), (IIIa), (IIIb) holds everywhere. Hence, $\delta_{p,q}$ can be expressed on the entire face by*
 256 *one of the analytic formulae above.*

257 **Proof.** Clearly, the closures of the regions in Ω described by the various cases intersect at
 258 points (s, t) such that p or q (or both) lie on the line $y = sx + t$, i.e. such that (s, t) is on
 259 the line L_p or L_q . The result follows. ◀

260 In view of Lemma 5, for $p, q \in \text{gr}(P^M) \cup \text{gr}(P^N)$ we may define the p, q -type of a face \mathcal{A}_0
 261 to be the case (I), (II), (IIIa), or (IIIb) which holds on that face.

262 **Refinement of the arrangement.** Now we further subdivide the arrangement \mathcal{A}_0 .
 263 For that, consider the set of equations of the form

$$264 \begin{aligned} \delta_{p,q}(s, t) &= 0 && \text{for } p, q \in \text{gr}(P^M) \text{ or } p, q \in \text{gr}(P^N), \\ c_{pq}\delta_{p,q}(s, t) &= c_{p'q'}\delta_{p',q'}(s, t) && \text{for } p, q, p', q' \in \text{gr}(P^M) \sqcup \text{gr}(P^N), \end{aligned} \quad (1)$$

265 where

$$266 c_{pq} := \begin{cases} \frac{1}{2} & \text{if } p, q \in \text{gr}(P^M) \text{ or } p, q \in \text{gr}(P^N), \\ 1 & \text{otherwise.} \end{cases}$$

267 ▶ **Lemma 6.** *The solution set of each of the above equations restricted to f is either the*
 268 *empty set, the entire face, the intersection of f with a line, or intersection of f with the*
 269 *union of two lines.*

270 **Proof.** First we show the statement for equations of the form $\delta_{p,q}(s, t) = 0$. There are three
 271 cases:

272 **$\delta_{p,q}$ is of type (I):** the equation becomes

$$273 |p_y - q_y| \sqrt{1 + \frac{1}{s^2}} = 0$$

274 for which either all $(s, t) \in f$ are a solution (if $p_y = q_y$), or no (s, t) is a solution.

275 **$\delta_{p,q}$ is of type (II):** the same argument holds for the equation

$$276 |p_x - q_x| \sqrt{1 + s^2} = 0.$$

277 **$\delta_{p,q}$ is of type (III):** Swapping p and q if necessary, we obtain the equation

$$278 |p_y - t - sq_x| \sqrt{1 + \frac{1}{s^2}} = 0$$

279 and the solution set is made of all $(s, t) \in f$ for which $p_y - t - sq_x = 0$, which is the
 280 equation of a line.

281 For the remaining equations, we give the proof in the special case that $c_{pq} = c_{p'q'}$;
 282 the proof in the other cases is essentially the same. For equations of the form $\delta_{p,q}(s, t) =$
 283 $\delta_{p',q'}(s, t)$, there are six cases to check, depending on the type of the δ -functions on the left
 284 and right sides of the equation:

285 **Both $\delta_{p,q}(s, t)$ and $\delta_{p',q'}(s, t)$ are of type (I):** the equation is

$$286 |p_y - q_y| \sqrt{1 + \frac{1}{s^2}} = |p'_y - q'_y| \sqrt{1 + \frac{1}{s^2}}$$

287 and the equation is satisfied if and only if $|p_y - q_y| = |p'_y - q'_y|$, independent of s and t .

288 Hence, the solution set is either f or \emptyset .

289 **Both $\delta_{p,q}(s, t)$ and $\delta_{p',q'}(s, t)$ are of type (II):** the same argument as in the previous case
 290 applies, so the solution set is either f or \emptyset .

291 $\delta_{p,q}$ is of type (I) and $\delta_{p',q'}$ of type (II): we get the equation

$$292 \quad |p_y - q_y| \sqrt{1 + \frac{1}{s^2}} = |p'_x - q'_x| \sqrt{1 + s^2}.$$

293 Since $1 + \frac{1}{s^2} = \frac{1+s^2}{s^2}$, this simplifies to

$$294 \quad |p_y - q_y| = s|p'_x - q'_x|$$

295 and the solution set is either all of f (if both absolute values vanish), the empty set (if
296 only $p'_x - q'_x = 0$), or the intersection of f with the vertical line $s = \frac{|p_y - q_y|}{|p'_x - q'_x|}$ (otherwise).

297 **Both $\delta_{p,q}$ and $\delta_{p',q'}$ are of type (III):** Swapping p, q or p', q' if necessary, we get

$$298 \quad |p_y - t - sq_x| \sqrt{1 + \frac{1}{s^2}} = |p'_y - t - sq'_x| \sqrt{1 + \frac{1}{s^2}}$$

299 hence, (s, t) is a solution if and only if $p_y - t - sq_x = p'_y - t - sq'_x$ or $p_y - t - sq_x =$
300 $-(p'_y - t - sq'_x)$. The first equations yields again either f , \emptyset , or a vertical line as solution
301 set, the second equation always defines a line. Exchanging the roles of p and q , or the
302 roles of p' and q' , or both, does not change the conclusion.

303 $\delta_{p,q}$ is of type (I) and $\delta_{p',q'}$ is of type (III): Swapping p' and q' if necessary, the formula
304 is

$$305 \quad |p_y - q_y| \sqrt{1 + \frac{1}{s^2}} = |p_y - t - sq_x| \sqrt{1 + \frac{1}{s^2}}.$$

306 $(s, t) \in f$ is a solution if $p_y - t - sq_x = p_y - q_y$ or $p_y - t - sq_x = q_y - p_y$, which is a line
307 equation in both cases.

308 $\delta_{p,q}$ is of type (II) and $\delta_{p',q'}$ is of type (III): Swapping p' and q' if necessary, we get

$$309 \quad |p_x - q_x| \sqrt{1 + s^2} = |p'_y - t - sq'_x| \sqrt{1 + \frac{1}{s^2}}$$

310 which simplifies to

$$311 \quad s|p_x - q_x| = |p'_y - t - sq'_x|$$

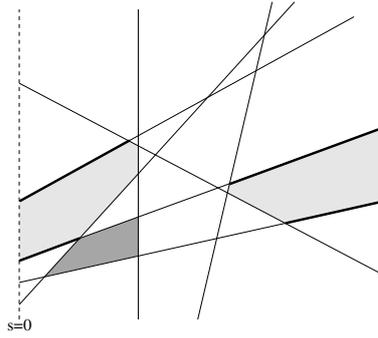
312 Here $(s, t) \in f$ is a solution if and only if $s(p_x - q_x) = p'_y - t - sq'_x$ or $s(p_x - q_x) =$
313 $-(p'_y - t - sq'_x)$. Again, we obtain a line in both cases. \blacktriangleleft

314 **► Definition 7.** Let \mathcal{A} denote the line arrangement in Ω formed by the lines in \mathcal{A}_0 , all lines
315 from the case analysis above, and the vertical line $s = 1$.

316 **► Lemma 8.** *The arrangement \mathcal{A} consists of $O(n^4)$ lines.*

317 **Proof.** The case analysis in the proof of Lemma 6 was performed relative to a choice of face
318 f in \mathcal{A}_0 . However, for a fixed choice of an equation in the set of equations (1), the lines
319 which arise in the case analysis depend only on the p, q -type or p', q' -type of f . There are
320 at most $4 \times 4 = 16$ possible way of jointly choosing the p, q -type and p', q' -type of f , and for
321 a given choice, at most two lines are added to the arrangement. Hence, each of the $O(n^4)$
322 equations in the set of equations (1) contributes at most a constant number of lines to \mathcal{A} .
323 The result now follows easily. \blacktriangleleft

324 Note that the arrangement \mathcal{A} depends on the choice of presentations for M and N .



350 **Figure 2** An arrangement of lines. The lightly shaded regions show outer regions, the darkly
 351 shaded region is an example of an inner region. The outer segments of the marked outer regions
 352 are drawn more thickly.

325 **Theorem 9.** *For any face f of \mathcal{A} , there is some choice of $p, q \in \text{gr}(P^M) \cup \text{gr}(P^N)$ such*
 326 *that $d_B(M^\ell, N^\ell) = c_{pq}\delta_{p,q}(\ell)$ for all $\ell \in \alpha(f)$.*

327 The formal proof of this result is deferred to Appendix A. It can be summarized as follows:

- 328 **1.** The order of the pushes of the grades of P^M and P^N along ℓ is the same across f . This is
 329 because whenever this order changes, we need to cross one of lines of the arrangement \mathcal{A} .
 330 Since the combinatorial structure of the persistence diagram only depends on the order
 331 of the grades of the presentations, that combinatorial structure is constant across f .
- 332 **2.** Each birth or death coordinate of the combinatorial persistence diagram associated to
 333 M is indexed by an element of P^M , and similarly for N .
- 334 **3.** The order of the values of the functions $\delta_{p,q}$ remains the same across f . This is because
 335 any change in their order will result again in crossing one of the lines of the arrangement
 336 \mathcal{A} . As a result, the combinatorial bottleneck matching remains the same across f , and
 337 so do the longest edge of the matching and the pair of grades (p, q) that realizes the
 338 bottleneck. The bottleneck distance along any slice ℓ in f is $c_{pq}\delta_{p,q}(\ell)$.

339 **5 Maximization**

340 We define a *region* of \mathcal{A} as the closure of a face of \mathcal{A} within Ω . We can compute the matching
 341 distance by determining $\sup F(s, t)$ separately for each region in \mathcal{A} . We will show now that
 342 in each region, $\sup F(s, t)$ is either realized at a boundary vertex, or as the limit of an
 343 unbounded boundary edge, which can be computed easily.

344 We fix the following notation: A region $R \subseteq \Omega$ is an *inner region* if it is bounded as a set
 345 in \mathbb{R}^2 and it has a positive distance to the vertical line $s = 0$ in \mathbb{R}^2 (in other words, R does
 346 not reach the boundary of Ω). An inner region is a convex polygon. Regions that are not
 347 inner region are called *outer regions*. Outer regions have exactly two *outer segments* in their
 348 boundary, which are infinite or converge to a point on the vertical line $s = 0$. See Figure 2
 349 for illustrations of these concepts.

353 For a fixed region R of \mathcal{A} , Theorem 9 ensures that there is a pair of grades (p, q) whose
 354 δ -function realizes d_B within the interior of R (a face of \mathcal{A}). By continuity, this implies that
 355 $\delta_{p,q}$ also realizes d_B on the entire region.

356 **Lemma 10.** *The supremum of F within R is attained either at a vertex on the bound-*
 357 *ary of R , or as the limit of F along an unbounded segment. In the latter case, the limit*

358 can be expressed in simple terms based on the equation of the line segment and one of the
359 functions $\delta_{p,q}$.

360 **Proof.** We distinguish 6 cases based on the type of the δ -function and on whether $s \leq 1$ or
361 $s \geq 1$ (note that each cell belongs to one of these cases, because the line $s = 1$ is in \mathcal{A}).

362 $\delta_{p,q}$ of type (I), $s \leq 1$ In that case,

$$363 \quad F(\ell) = w(\ell)\delta_{p,q}(\ell) = \frac{1}{\sqrt{1 + \frac{1}{s^2}}} |p_y - q_y| \sqrt{1 + \frac{1}{s^2}} = |p_y - q_y|,$$

364 a constant function. Clearly, the supremum is attained everywhere, in particular at the
365 boundary vertices of R .

366 $\delta_{p,q}$ of type (I), $s \geq 1$ We get

$$367 \quad F(\ell) = \frac{1}{\sqrt{1 + s^2}} |p_y - q_y| \sqrt{1 + \frac{1}{s^2}} = \frac{1}{s} |p_y - q_y|$$

368 Clearly, this function becomes larger when s gets smaller. Moreover, because $s \geq 1$
369 within the cell, there is a leftmost vertex on the boundary, which minimizes s and
370 therefore attains the supremum within the cell.

371 $\delta_{p,q}$ of type (II), $s \leq 1$ We obtain

$$372 \quad F(\ell) = \frac{1}{\sqrt{1 + \frac{1}{s^2}}} |p_x - q_x| \sqrt{1 + s^2} = s |p_x - q_x|.$$

373 Similarly to the previous case, there exists a rightmost boundary vertex in the cell
374 (because $s \leq 1$), which realizes the supremum.

375 $\delta_{p,q}$ of type (II), $s \geq 1$ The function simplifies to

$$376 \quad F(\ell) = \frac{1}{\sqrt{1 + s^2}} |p_x - q_x| \sqrt{1 + s^2} = |p_x - q_x|,$$

377 a constant function, which attains its supremum at any boundary vertex.

378 $\delta_{p,q}$ of type (III), $s \leq 1$ Assuming that p lies above the slice, we get

$$379 \quad F(\ell) = \frac{1}{\sqrt{1 + \frac{1}{s^2}}} |p_y - t - sq_x| \sqrt{1 + \frac{1}{s^2}} = |p_y - t - sq_x|.$$

380 If R is an inner region, we are maximizing the above function over a closed convex
381 polygon, and the maximum is achieved at a boundary vertex, because $|p_y - t - sq_x|$ is
382 the maximum of two linear functions in s and t .

383 It remains to analyze the case that R is an outer region. We argue first that R is bounded
384 in t -direction from above and below: Since $\delta_{p,q}$ is of type (III), with p lying above ℓ , (s, t)
385 must be below the (non-vertical) line $t = -sp_x + p_y$ in the dual space. Likewise, since
386 q is below ℓ , (s, t) must be above $t = -sq_x + q_y$. Moreover, we have $0 < s \leq 1$. If the
387 above lines intersect at a point r with s -value in $(0, 1)$, R is contained in the triangle
388 spanned by the two lines and the vertical line $s = 0$. Otherwise, R is contained in the
389 trapezoid induced by these two lines and the vertical lines $s = 0$ and $s = 1$.

390 It follows that the two outer segments of R converge to the vertical line $s = 0$. Let
391 $(0, t_1)$ denote the limit of the lower outer segment and $(0, t_2)$ the limit of the upper outer

XX:12 Exact computation of the matching distance on 2-parameter persistence modules

392 segment. Clearly $t_1 \leq t_2$. Let \bar{R} denote the the union of R with the vertical line segment
 393 from $(0, t_1)$ to $(0, t_2)$; note that \bar{R} is the closure of R considered as a subset of \mathbb{R}^2 . Observe
 394 that $|p_y - t - sq_x|$ is continuous over \mathbb{R}^2 ; therefore F can be continuously extended to \bar{R} .
 395 It follows that the supremum of F over \bar{R} is attained at a boundary vertex, since \bar{R} is a
 396 convex closed polygon. There are two cases: either the maximum is attained at a vertex
 397 of \mathcal{A} , or at $(0, t_1)$ or $(0, t_2)$. As we can readily see, the function values at the latter two
 398 points are $|p_y - t_1|$ and $|p_y - t_2|$, respectively. The case where p is below the slice and q
 399 is above is analyzed in the same way, with the roles of p and q swapped.

400 $\delta_{p,q}$ of type (III), $s \geq 1$ Assuming that p lies above the slice, we get

$$401 \quad F(\ell) = \frac{1}{\sqrt{1+s^2}} |p_y - t - sq_x| \sqrt{1 + \frac{1}{s^2}} = \left| \frac{p_y}{s} - \frac{t}{s} - q_x \right|$$

402 We first consider the case where R is an inner region, and we show that the function is
 403 maximized at a boundary vertex of R . The function $|\frac{p_y}{s} - \frac{t}{s} - q_x|$ has no local maximum
 404 over \mathbb{R}^2 since it is the absolute value of a linear function in t for any fixed s . Hence, the
 405 supremum over R must be attained on the boundary. We have to exclude the case that
 406 the maximum lies in the interior of an edge. For vertical edges this is obvious, because
 407 for a constant s , the function simplifies to the absolute value of a linear function in t
 408 which must be maximized at a boundary vertex. For a non vertical line of the form
 409 $t = as + b$, plugging in this equation for t yields a function of the form

$$410 \quad \left| \frac{p_y}{s} - \frac{as+b}{s} - q_x \right| = \left| \frac{1}{s}(p_y - b) - a - q_x \right|.$$

411 This is the absolute value of a monotone function in s and hence has no local maximum.
 412 Again, this implies that it is maximized at a boundary vertex.

413 Consider now the case where R is an outer region. As in the previous case, R is upper
 414 and lower bounded by two non-vertical lines, because we assume type (III). Hence, the
 415 two outer segment of R cannot be vertical; the lower outer segment has a slope r_1 and
 416 the upper outer segment has a slope r_2 with $r_1 < r_2$. We argue next that the supremum
 417 of $\frac{p_y}{s} - \frac{t}{s} - q_x$ is either attained at a boundary vertex, or equal to $|r_1 + q_x|$, or equal
 418 to $|r_2 + q_x|$. Let (s_i, t_i) denote a sequence of points in R such that $F(s_i, t_i)$ converges
 419 to the supremum. If (s_i, t_i) converges to a point in R (or has at least a convergent
 420 subsequence), it follows (similarly to the case of an inner region) that the limit point is
 421 a boundary vertex. Otherwise, we can assume (by passing to a subsequence) that the
 422 sequence s_i is unbounded. Moreover, the sequence $\frac{t_i}{s_i}$ is bounded by $[r_1, r_2]$ and therefore
 423 has a convergent subsequence with limit r' . Passing to this subsequence, we obtain that

$$424 \quad \lim_{i \rightarrow \infty} F(s_i, t_i) = \lim_{i \rightarrow \infty} \left| \frac{p_y}{s_i} - \frac{t_i}{s_i} - q_x \right| = |-r' - q_x| = |r' + q_x|$$

425 Hence, the supremum must be of the form $|r' + q_x|$ for some $r' \in [r_1, r_2]$. On the other
 426 hand, this expression is clearly maximized for either $|r_1 + q_x|$ or $|r_2 + q_x|$, and there exist
 427 sequences attaining these values, for instance when choosing (s_i, t_i) on either of the outer
 428 segments. The case where p lies below the slice and q lies above is treated similarly, with
 429 the roles of p and q exchanged. ◀

430 **The algorithm.** We now give the algorithm to compute the matching distance:

- 431 ■ Compute the arrangement induced by \mathcal{A} from Definition 7.

- 432 ■ For each vertex (s, t) in the arrangement, compute $F(s, t)$. Let m be the maximum
 433 among all the values.
- 434 ■ For each outer region R , pick a point (s, t) in the interior. Compute the bottleneck
 435 distance and identify a pair (p, q) of grades that realizes the bottleneck. Determine
 436 whether p and q are above or below the slice (s, t) . If the region is of type (III) with
 437 respect to p and q , then do the following:
- 438 ■ If R is on the left of $s = 1$, compute the intersections $(0, t_1), (0, t_2)$ of the outer
 439 segments of R with the vertical line $s = 0$. Set $m := \max\{m, |p_y - t_1|, |p_y - t_2|\}$.
 - 440 ■ If R is on the right of $s = 1$, let r_1, r_2 denote the slopes of the outer segments of R .
 441 Set $m := \max\{m, |r_1 + q_x|, |r_2 + q_x|\}$.
- 442 ■ Return m .

443 By “computing the arrangement”, we mean to store the planar subdivision induced by
 444 the lines of the arrangement (e.g. [2, Ch.2]). In fact, it not too difficult to implement the
 445 algorithm without explicitly constructing the line arrangement \mathcal{A} , nor even storing its whole
 446 set of vertices. This reduces the space complexity of the algorithm by a polynomial factor,
 447 while leaving its time complexity unchanged. See Appendix B for the details.

448 ► **Theorem 11.** *The above algorithm computes the matching distance in polynomial time.*

449 **Proof.** Correctness follows from Lemma 10: as we check all vertices of the arrangement, we
 450 cover the supremum of all inner regions. The outer regions are handled separately in the
 451 last steps of the algorithm.

452 Running time: recall from Lemma 8 that we have $O(n^4)$ lines in the arrangement \mathcal{A} .
 453 Hence, the arrangement has $O(n^8)$ vertices, $O(n^4)$ outer regions, and can be computed in
 454 $O(n^8 \log n)$ time using an extension of the Bentley-Ottman sweep-line algorithm [1]. For
 455 each vertex and each outer region, we have to compute two persistence diagrams, which can
 456 be done in $O(n^3)$ time, and a bottleneck distance whose complexity can be neglected. The
 457 remaining computations are negligible. Hence, we arrive at a $O(n^{11})$ algorithm. ◀

458 We remark that the algorithm can be realized entirely with rational arithmetic if all
 459 grades are rational numbers. Indeed, all lines in the arrangement have rational coefficients,
 460 and so do their intersection points. An intersection points corresponds to a slice along which
 461 we are required to compute the bottleneck distance. Recall from Section 3 that the definition
 462 of the slice ℓ module introduces a grade of $\lambda^{-1}(\text{push}(p, \ell))$ where $\lambda^{-1}(p_x, p_y) = \sqrt{1 + s^2}p_x$.
 463 Hence, these grades are not rational numbers. However, the bottleneck distance is multiplied
 464 with the weight $w(\ell)$ of the slice afterwards, and instead of doing so, one can as well scale all
 465 grades with $w(\ell)$ in advance. A simple calculation shows that this indeed turns the grades
 466 into rational values.

467 A simple analysis also reveals that if the input coordinates are rational and of bitsize
 468 $\leq b$, all intermediate computations in the algorithm can be performed with a bitsize of $\leq cb$,
 469 with c a (small) constant. Hence, the algorithm is strongly polynomial.

470 6 Discussion

471 We have presented the first polynomial time algorithm to exactly compute the matching
 472 distance for 2-parameter persistent modules. It is natural to ask about practicality of our
 473 approach. The large exponent of n^{11} seems discouraging at first, but we mention first that
 474 the worst-case running time of $O(n^3)$ for persistent homology is usually not appearing for
 475 real instances; indeed an almost linear behavior can be expected. Still, the large number of

476 $O(n^4)$ lines in the arrangement constitutes a computational barrier in practice. There are
 477 several possibilities, however, to reduce this effect:

- 478 ■ Instead of computing the arrangement \mathcal{A} globally, we could compute the intermediate
 479 arrangement \mathcal{A}_0 and refine each face of it separately, using only those lines that affect
 480 the δ -functions within this face.
- 481 ■ As a follow-up to the previous point, it might be possible to compute a smaller arrange-
 482 ment per face adaptively. The idea is to start at some interior point in a face of \mathcal{A}_0 ,
 483 identifying a pair (p, q) that realizes the bottleneck distance and then to determine the
 484 boundary of the region where (p, q) realizes the bottleneck distance.
- 485 ■ As a preprocessing step, we can move from the input presentations to *minimal presenta-*
 486 *tions*, which represent isomorphic persistent module (that is, yielding the same matching
 487 distance), but with the smallest number of generators and relations (hence minimizing n).

488 We pose the question of whether an implementation realizing the above ideas is competitive
 489 to an approximative, sampling-based approach for computing the matching distance.

490 Our algorithm needs to treat the outer edges of the arrangement \mathcal{A} separately since our
 491 analysis does not rule out the possibility that the supremum is realized at the boundary of
 492 Ω . On the other hand, we are not aware of an example of two finite presentations whose
 493 matching distance is not realized by a particular slice in Ω . A proof that the supremum
 494 in the definition of the matching distance is in fact a maximum would greatly simplify our
 495 algorithm, since it would boil down to computing the intersection points of all lines and
 496 searching for the maximal F -value among them.

497 Finally, we have restricted attention to the case of two-parameter persistence modules.
 498 It is natural to conjecture that our algorithm extends to more parameters by constructing
 499 a hyperplane arrangement. It would be worthwhile to check this conjecture in future work.

500 **Acknowledgments.** This work was initiated at the Banff workshop “Multiparameter
 501 Persistent Homology” (18w55140) in Oaxaca, Mexico (Aug. 2018). We thank Jan Reining-
 502 haus and the other members of the discussion group on this topic for fruitful initial exchanges.
 503 We thank Matthew Wright for helpful discussions about line arrangements, slices, and the
 504 computational aspects of 2-parameter persistence.

505 — References —

- 506 **1** J. Bentley and T. Ottmann. Algorithms for Reporting and Counting Geometric Intersec-
 507 tions. *IEEE Transactions on Computers*, 28:643–647, 1979.
- 508 **2** M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry:*
 509 *Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed.
 510 edition, 2008.
- 511 **3** S. Biasotti, A. Cerri, P. Frosini, and D. Giorgi. A new algorithm for computing the
 512 2-dimensional matching distance between size functions. *Pattern Recognition Letters*,
 513 32(14):1735–1746, 2011.
- 514 **4** H. Bjerkevik, M. Botnan, and M. Kerber. Computing the interleaving distance is NP-hard.
 515 *arXiv*, abs/1811.09165, 2018. URL: <http://arxiv.org/abs/1811.09165>.
- 516 **5** G. Carlsson and A. Zomorodian. The theory of multidimensional persistence. *Discrete &*
 517 *Computational Geometry*, 42(1):71–93, 2009.
- 518 **6** A. Cerri, B. Di Fabio, M. Ferri, P. Frosini, and C. Landi. Betti numbers in multidimensional
 519 persistent homology are stable functions. *Mathematical Methods in the Applied Sciences*,
 520 36(12):1543–1557, 2013.

- 521 **7** R. Corbet and M. Kerber. The representation theorem of persistence revisited and gener-
522 alized. *Journal of Applied and Computational Topology*, 2(1):1–31, 2018.
- 523 **8** W. Crawley-Boevey. Decomposition of pointwise finite-dimensional persistence modules.
524 *Journal of Algebra and its Applications*, 14(05):1550066, 2015.
- 525 **9** The RIVET Developers. RIVET: Software for visualization and analysis of 2-parameter
526 persistent homology. <http://repo.rivet.online/>, 2014-2018.
- 527 **10** H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Math-
528 ematical Society, Providence, RI, USA, 2010.
- 529 **11** A. Efrat, A. Itai, and M. Katz. Geometry helps in bottleneck matching and related prob-
530 lems. *Algorithmica*, 31(1):1–28, 2001. doi:10.1007/s00453-001-0016-8.
- 531 **12** B. Keller, M. Lesnick, and T. L. Willke. Persistent Homology for Virtual Screening. *Chem-*
532 *Rxiv preprint*, 10 2018.
- 533 **13** M. Kerber, D. Morozov, and A. Nigmatov. Geometry helps to compare persistence dia-
534 grams. *Journal of Experimental Algorithms*, 22:1.4:1–1.4:20, September 2017.
- 535 **14** C. Landi. The rank invariant stability via interleavings. In *Research in Computational*
536 *Topology*, pages 1–10. Springer International Publishing, 2018.
- 537 **15** M. Lesnick. The theory of the interleaving distance on multidimensional persistence mod-
538 ules. *Foundations of Computational Mathematics*, 15(3):613–650, 2015.
- 539 **16** M. Lesnick and M. Wright. Computing a minimal presentation of 2-parameter persistent
540 homology in cubic time. *In Preparation*.
- 541 **17** M. Lesnick and M. Wright. Interactive visualization of 2-D persistence modules.
542 *arXiv:1512.00180*, 2015.
- 543 **18** N. Milosavljevic, D. Morozov, and P. Skraba. Zigzag persistent homology in matrix multi-
544 plication time. In *ACM Symposium on Computational Geometry (SoCG)*, pages 216–225,
545 2011.
- 546 **19** A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete and Computa-*
547 *tional Geometry*, 33(2):249–274, 2005.

548 **A Appendix: proof of Theorem 9**

549 Let $G^M = \text{gr}(P^M)$ and let $G^N = \text{gr}(P^N)$. As an intermediate result, we prove that the
 550 combinatorial structure of each persistence diagram stays the same across the face f :

551 **► Lemma 12.** *For each face f of \mathcal{A} , there exist multisets*

$$552 \quad \mathcal{T}_M^f \subset G^M \times (G^M \cup \{(\infty, \infty)\}),$$

$$553 \quad \mathcal{T}_N^f \subset G^N \times (G^N \cup \{(\infty, \infty)\})$$

554 *such that for any $\ell \in \alpha(f)$,*

$$555 \quad D(M^\ell) = \left\{ (\lambda^{-1} \circ \text{push}(a, \ell), \lambda^{-1} \circ \text{push}(b, \ell)) \mid (a, b) \in \mathcal{T}_M^f \right\},$$

$$556 \quad D(N^\ell) = \left\{ (\lambda^{-1} \circ \text{push}(a, \ell), \lambda^{-1} \circ \text{push}(b, \ell)) \mid (a, b) \in \mathcal{T}_N^f \right\},$$

557 *where by convention $\text{push}((\infty, \infty), \ell) := (\infty, \infty)$ and $\lambda^{-1}((\infty, \infty)) := \infty$. We call \mathcal{T}_M^f and
 558 \mathcal{T}_N^f diagram templates.*

559 **Proof.** This is a variant of [17, Theorem 4.1]. We give a succinct, algorithmically flavored
 560 proof here. We prove the result for M ; the proof for N is the same.

561 First, note that the restriction of the partial order on \mathbb{R}^2 to any $\ell \in \alpha(f)$ is a total order.
 562 This total order pulls back under the map $\text{push}(-, \ell) : \text{gr}(P^M) \rightarrow \ell$ to a totally ordered
 563 partition of $\text{gr}(P^M)$, (i.e., elements of the partition are level sets). It can be checked that,
 564 because \mathcal{A} refines \mathcal{A}_0 and also contains all lines of the form $\delta_{p,q} = 0$ for $p, q \in \text{gr}(P^M)$ with
 565 p and q incomparable in the partial order on \mathbb{R}^2 , this totally ordered partition is the same
 566 for all $\ell \in \alpha(f)$; see [17, Corollary 3.4]. Thus, we obtain a unique totally ordered partition
 567 of $\text{gr}(P^M)$ associated to all of f . Let us refine this to a fixed total order on $\text{gr}(P^M)$.

568 Permuting the row or columns of a presentation for M yields another presentation for
 569 M , so we may assume without loss of generality that the order of rows and columns for P^M
 570 is consistent with our total order on $\text{gr}(P^M)$. Applying the matrix reduction underlying the
 571 standard persistence algorithm [19] to the matrix underlying P^M yields a matching σ of row
 572 and column indices, where a non-zero column in the reduced matrix is matched to the row
 573 of its pivot. We may define

$$574 \quad \mathcal{T}_M^f := \{ \text{gr}(\text{row}_i), \text{gr}(\text{col}_j) \mid (i, j) \in \sigma, \text{gr}(\text{row}_i), \text{gr}(\text{col}_j) \text{ not together in the partition} \}$$

$$575 \quad \cup \{ (\text{gr}(\text{row}_i), (\infty, \infty)) \mid i \text{ is unmatched in } \sigma \}.$$

576 (As an aside, we remark that this is not the only way to define \mathcal{T}_M^f such that the property
 577 in the statement of the lemma is satisfied.)

578 As mentioned earlier, P^M induces a finite presentation P^ℓ for M^ℓ ; the presentation
 579 matrix remains the same, and we simply replace each row and column grade g by $\lambda^{-1} \circ$
 580 $\text{push}(g, \ell)$. From this, and the correctness of the standard algorithm for computing persistent
 581 homology [19], it follows that

$$582 \quad D(M^\ell) = \left\{ (\lambda^{-1} \circ \text{push}(a, \ell), \lambda^{-1} \circ \text{push}(b, \ell)) \mid (a, b) \in \mathcal{T}_M^f \right\},$$

583 as desired. ◀

584 Suppose we are given two persistence diagrams D and D' . For each $(x, y) \in D \cup D'$, let
 585 $w_0(x, y) := \frac{y-x}{2}$, and for each $(x, y), (x', y') \in D \times D'$, let $w_1(x, x') := \max(|x - x'|)$, and

586 $w_2(y, y') := \max(|y - y'|)$. We call these (possibly infinite) numbers the *weights* of the pair
 587 (D, D') . Denoting the set of pairs indexing these weights as $I(D, D')$, the weights define a
 588 function $w^{D, D'} : I^{D, D'} \rightarrow [0, \infty)$. Note that $d_B(D, D') = w^{D, D'}(x, y)$ for some $(x, y) \in I^{D, D'}$.

589 Clearly, a pair of bijections of persistence diagrams $\zeta_1 : D \rightarrow E$ and $\zeta_2 : D' \rightarrow E'$ induces
 590 a bijection

$$591 \quad \zeta : I(D, D') \rightarrow I(E, E').$$

592 The following is an easy consequence of the definition of d_B .

593 ► **Lemma 13.** *If $d_B(D, D') = w^{D, D'}(x, y)$ and ζ_1, ζ_2 preserve the order of the weights, in
 594 the sense that*

$$595 \quad w^{D, D'}(x, y) \leq w^{D, D'}(x', y') \quad \text{if and only if} \quad w^{E, E'}(\zeta(x, y)) \leq w^{E, E'}(\zeta(x', y')),$$

596 *then $d_B(E, E') = w^{E, E'}(\zeta(x, y))$.*

597 **Proof of Theorem 9.** Let \mathcal{T}_M^f and \mathcal{T}_N^f be diagram templates for the cell f . For any $\ell \in \alpha(f)$,
 598 Lemma 12 gives us distinguished bijections

$$599 \quad \gamma : \mathcal{T}_M^f \rightarrow D(M^\ell), \quad \gamma' : \mathcal{T}_N^f \rightarrow D(N^\ell).$$

600 It is easily checked that for $(x, y) \in D(M^\ell)$, and $(p, q) = \gamma^{-1}(x, y)$, we have

$$601 \quad w_0(x, y) = \frac{1}{2} \delta_{p, q}(\ell) = c_{pq} \delta_{p, q}(\ell). \quad (2)$$

602 Similarly, for

$$603 \quad (x, y) \in D(M^\ell), \quad (x', y') \in D(N^\ell), \quad (p, p') = \gamma^{-1}(x, y), \quad \text{and} \quad (q, q') = \gamma'^{-1}(x', y'),$$

604 we have

$$605 \quad w_1(x, x') = \delta_{p, q}(\ell) = c_{pq} \delta_{p, q}(\ell), \quad (3)$$

$$606 \quad w_2(y, y') = \delta_{p', q'}(\ell) = c_{p'q'} \delta_{p', q'}(\ell). \quad (4)$$

607 Note that as we move the slice ℓ inside f , the order of the values taken by the functions

$$608 \quad \{c_{pq} \delta_{p, q} \mid p, q \in P^M \cup P^N\}$$

609 cannot change. Indeed, the functions $\ell \rightarrow \delta_{p, q}(\ell)$ are continuous, and by Lemma 6 and the
 610 definition of the arrangement \mathcal{A} , the intersection of f with the solution set of each equation
 611 $c_{pq} \delta_{p, q} = c_{p'q'} \delta_{p', q'}$ is either f or \emptyset .

612 The diagram templates provide bijections $D(M^\ell) \rightarrow D(M^{\ell'})$ and $D(N^\ell) \rightarrow D(N^{\ell'})$ for
 613 all $\ell, \ell' \in \alpha(f)$. It now follows from equations (2-4) that for each ℓ, ℓ' , these preserve the
 614 order on weights. Thus, by Lemma 13, if

$$615 \quad d_B(D(M^\ell), D(N^\ell)) = w^{D(M^\ell), D(N^\ell)}(x, y)$$

616 for some $\ell \in \alpha(f)$, then

$$617 \quad d_B(D(M^{\ell'}), D(N^{\ell'})) = w^{D(M^{\ell'}), D(N^{\ell'})}(\zeta^{\ell, \ell'}(x, y))$$

618 for all $\ell' \in \alpha(f)$, where

$$619 \quad \zeta^{\ell, \ell'} : I^{D(M^\ell), D(N^\ell)} \rightarrow I^{D(M^{\ell'}), D(N^{\ell'})}$$

620 is the bijection induced by the barcode templates. The result now follows from equations
 621 (2-4). ◀

622 **B A more-space efficient algorithm**

623 The algorithm described in Section 5 consists of two major steps: we first find the maximal
 624 value of F over all intersection points of lines in the arrangement \mathcal{A} , and then check the
 625 convergence along outer segments for certain outer regions of \mathcal{A} . To decide whether an
 626 outer region needs to be considered, we require one interior point of that region, to decide
 627 the type of the region. The arrangement \mathcal{A} (stored as a planar subdivision as in [2, Ch.2])
 628 contains enough information to access all necessary data conveniently. However, its space
 629 complexity is $O(n^8)$ because the arrangement is induced by $O(n^4)$ lines. We show next how
 630 to implement the algorithm without constructing \mathcal{A} in memory, yielding a space complexity
 631 of $O(n^4)$:

632 Reporting the intersection points of a set of lines in the plane is one of the oldest prob-
 633 lems in computational geometry. The Bentley-Ottmann sweep-line algorithm [1] reports all
 634 intersection points in time proportional to the number of such points. Note that there are
 635 up to $O(n^8)$ intersections, so storing all such points is space consuming. However, there is
 636 no need to do so – whenever the sweep-line algorithm reports a new point, we compute F
 637 at this point and compare the value with the maximal F -value seen before, updating the
 638 maximum if the newly encountered value is larger. There is no need to store the intersection
 639 beyond this moment. In this way, we obtain the maximal F -value among all intersection
 640 points using only $O(n^4)$ space.

641 For handling outer regions, note that we can extend the sweep-line algorithm such that
 642 it also returns the leftmost intersection point with positive s -coordinate, and the rightmost
 643 intersection point, among all pairs of lines in \mathcal{A} . Let $0 < s_{min} \leq s_{max}$ denote the s -
 644 coordinates of these intersection points. Now, consider the vertical line $s = \frac{s_{min}}{2}$, which is
 645 partitioned into open intervals

$$646 \quad (-\infty, b_1), (b_1, b_2), \dots, (b_{m-1}, b_m), (b_m, \infty) \quad (5)$$

647 where m is the number of $O(n^4)$ lines in \mathcal{A} that intersect the vertical line, and b_1, \dots, b_m are
 648 the intersection points of these (non-vertical) lines with the vertical line. The intervals in the
 649 sequence can be computed in $O(n^4 \log n)$ time, just by sorting the b_i 's. The intervals in the
 650 sequence (5) are in one-to-one correspondence with the outer segments of \mathcal{A} that have the
 651 line $s = 0$ in their boundary. More precisely, each interval is contained in the corresponding
 652 outer segment, and for $i \in \{1, \dots, m-1\}$, the point $(\frac{s_{min}}{2}, \frac{b_i + b_{i+1}}{2})$ is an interior point of
 653 the corresponding outer region, whose outer segments of the lines of \mathcal{A} contain the points
 654 b_i and b_{i+1} . This information suffices to determine the type of the region, and to compute
 655 the limit values of the F -function if necessary. The two infinite regions corresponding to
 656 $(-\infty, b_1)$ and (b_m, ∞) can be ignored, because these regions must be of type (I) or type
 657 (II). The same construction provides interior points of outer regions that are unbounded in
 658 s -direction, considering the vertical line at $s = 2s_{max}$.

659 With this variant, we compute the matching distance with space complexity $O(n^4)$. The
 660 time complexity remains $O(n^{11})$ as in the original algorithm, because we still iterate over
 661 $O(n^8)$ vertices and compute the bottleneck distance in each position in $O(n^3)$ time.