Parsimonious simplicial intersection algorithm

CONOR MCCOID, McMaster University, Canada

Intersection algorithms are crucial in many applications, but they may not be robust. Without robustness these algorithms may fail to correctly identify large intersections. To prevent that, this article develops a parsimonious algorithm for the intersection of simplices. This generalizes earlier algorithms on triangle and tetrahedral intersections. This paper outlines the algorithm and its parsimony and proves its consistency. Numerical experiments confirm its applicability.

CCS Concepts: • **Theory of computation** → **Computational geometry**.

ACM Reference Format:

Conor McCoid. 2025. Parsimonious simplicial intersection algorithm. *ACM Trans. Math. Softw.* 1, 1 (September 2025), 26 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The most common application for intersection algorithms is computer graphics [2]. In this setting, one is concerned with the portion of a 'subject' object that is hidden from the viewer by a second 'clipping' object, or when the two objects collide. In the former case the intersection is between 2D objects, as the viewing plane is a projection of the 3D space. In the latter case collision detection involves intersecting 3D objects.

Other applications arise through the use of nonmatching grids, for example in multiphysics problems [9]. In highly complex models with several different types of physical behaviours, such as a ferrofluid flowing through a magnetic field, it can be useful to separate the behaviours by placing each on its own grid. In the ferrofluid example, the fluid dynamics occupies one grid while the magnetic interaction lies on another. To pass information between the grids one needs to project from one to the other, which requires calculating the intersection of the individual elements.

Intersections in 2D and 3D are also calculated for mortar methods [10], contact algorithms [6], and finite element methods [8]. Higher dimensional intersections are calculated for estimation problems [1]. In these problems the space of measurements is intersected with the space of parameters, both of which may have arbitrary dimension.

These various applications of intersection algorithms mean the problem they attempt to solve goes by many names, some of which include: mesh intersection problem; grid transfer problem; intergrid communication problem, and; polygon/polyhedron clipping problem.

The intersection of two simplices is a convex polytope. To distinguish between the vertices of the simplices and those of the intersection let the latter be referred to as corners of the polytope. Any algorithm for the intersection of two simplices must consider two types of corners: vertices of one simplex inside the other, and the points where the simplices intersect. See Figure 1 for an example in 2D.

An intersection algorithm can either be symmetric and treat the two simplices equally, or it can be asymmetric and give preferential treatment to one of them. For example, one can employ a change of coordinates that transforms one of

Author's Contact Information: Conor McCoid, McMaster University, Hamilton, ON, Canada, mccoidc@mcmaster.ca.

Manuscript submitted to ACM

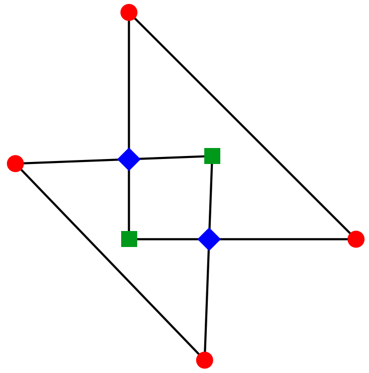

Fig. 1. The corners of the intersection of two simplices are either vertices of one simplex inside the other (green squares) or points where the two meet (blue diamonds).

the simplices into a reference simplex aligned with the coordinate axes; The two simplices are not treated equally. For asymmetric algorithms the determination of whether a vertex of one lies within the other must likewise be asymmetric. To continue with the example, a vertex in the reference simplex is found with fewer calculations than one found within a general simplex.

For dimension 2 the points where the simplices intersect is simple: They are the points where the edges meet. In dimension n things become more complicated. The corners of the polytope are now the intersections between two high dimensional faces, called k-faces, such that the sum of their dimensions is equal to n. In 3D, for example, the corners are the intersections of the edges of one simplex with the faces of the other, giving two subtypes of such corners. As the dimension increases so does the number of subtypes of corners. In symmetric algorithms these are treated in pairs, while for asymmetric algorithms each subtype must be treated differently.

The algorithm presented in this paper is an asymmetric algorithm. The pseudocode found in Algorithm 1 gives an overview of the necessary steps of the algorithm. Line 1 is dealt with in Section 2; line 2 in Section 3; lines 3 through 5 in Section 4; and line 6 in Section 5.

Algorithm 1 Asymmetric simplicial intersection algorithms

```
1: Transform simplices $U$ and $V$ to general simplex $X$ and special simplex $Y$
2: Determine which vertices of $X$ lie within $Y$
3: for $k = 1$ to $n - 1$ do
4:     Find intersections of $k$-faces of $X$ with $(n - k)$-faces of $Y$
5: end for
6: Determine which vertices of $Y$ lie within $X$
```

To ensure the algorithm is robust it has been written by adhering to the principle of parsimony, namely that a minimum number of calculations is used to arrive at the conclusion. This means that if a given calculation can be used to avoid additional calculations then the algorithm will do so, and if information pertinent to another calculation may be inferred from this given calculation then the algorithm will do so. The algorithm will therefore be referred to as the parsimonious simplicial intersection algorithm, or ParSIA.

Many other intersection algorithms have been developed over the past century, mainly for the intersection of 2D shapes [2, 5, 11–14]. Of particular note amongst these is the Sutherland-Hodgman algorithm [13], generalized to

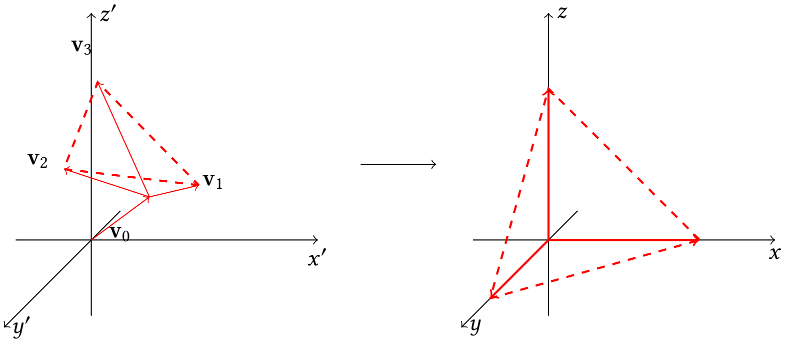

Fig. 2. Coordinate transformation of tetrahedron V into reference tetrahedron Y.

arbitrary dimension by Broman and Shensa in [1]. It is also an asymmetric algorithm, but for the intersection of one convex polytope and one object of the same dimension. The convex polytope defines a set of hyperplanes which are used to sequentially section the other object until only the intersection remains. This bears similarities to the approach in Section 4.

The algorithm presented here is a generalization of PANG2 [7], itself the successor to PANG (Projection Algorithm for Nonmatching Grids) [3, 4]. Failures of PANG related to inconsistent calculations led to the creation of PANG2, which focused on the intersection of triangles. The principles applied there are expanded upon here to encompass simplices in arbitrary dimensions.

2 Notation and coordinate system

Consider two arbitrary simplices V and U in $\mathbb{R}^n$. Each simplex has $n+1$ vertices. Denote the coordinates of one of the vertices of V as $\mathbf{v}_0$, the 0th vertex. The coordinates of the remaining vertices of V are then defined as $\mathbf{v}_0 + \mathbf{v}_i$ for $i = 1, \dots, n$. The vector $\mathbf{v}_i$ then runs along the edge of V between its 0th and i-th vertex. The vectors $\mathbf{u}_0$ and $\mathbf{u}_j$ for $j = 1, \dots, n$ are equivalently defined for the simplex U.

To simplify calculations, transform the simplex V into a reference simplex Y with the edges $\mathbf{v}_i$ aligned to the coordinate axes. That is, if $\hat{V}$ is the matrix containing the coordinates of the vertices of V then seek the affine transformation $A\mathbf{x} + \mathbf{b}$ such that

$$A\hat{V} + \mathbf{b}\mathbf{1}^\top = \begin{bmatrix} \mathbf{0} & I \end{bmatrix}.$$

This maps $\mathbf{v}_0$ to the origin and $\mathbf{v}_i$ to $\mathbf{e}_i$, a cardinal direction with unit length. It is trivial to show that $\mathbf{b} = -A\mathbf{v}_0$ and A is the inverse of $\begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_n \end{bmatrix}$. This transformation makes the algorithm asymmetric.

The coordinates of U must be subjected to the same transformation, giving coordinates of the transformed simplex X. The coordinates $\mathbf{x}_i$ for $i = 0, \dots, n$ are unknown and can be found by solving the system

$$\begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_n \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 & \dots & \mathbf{x}_n \end{bmatrix} = \hat{U} - \mathbf{v}_0\mathbf{1}^\top, \tag{2.1}$$

where $\hat{U}$ is the matrix containing the coordinates of the vertices of U.

In addition to the Euclidean coordinates of each point, $\mathbf{x} \cdot \mathbf{e}_i$, it is useful to add a 0th coordinate, $\mathbf{x} \cdot \mathbf{e}_0 = 1 - \sum_{i=1}^n \mathbf{x} \cdot \mathbf{e}_i$. With this addition the coordinates $\mathbf{x}_j$ are the barycentric coordinates of U relative to V.

To avoid unnecessary degenerate cases, use the binary-valued sign function,

$$\operatorname{sign}(x) = \begin{cases} 1 & x \geq 0, \\ 0 & x < 0, \end{cases}$$

in place of its more standard trinary-valued counterpart. All instances of sign evaluations will use this definition. In this way, a degenerate case finds the same number of vertices with the same relative positions as the non-degenerate case where the vertices of U are shifted infinitesimally towards the centre of V.

The intersection of two simplices is a polytope of the same dimension when avoiding degenerate cases. Denote the intersection of U and V as W, keeping in mind that it may be empty. The transformation of W, which is the barycentric representation of W with respect to V, is denoted as Z.

3 Vertices of X inside Y

Before proceeding, note that there are two main ways in which to define a simplex. Firstly, one can use $n+1$ hyperplanes to bound the simplex.

Definition 1 (Simplex 1). *A simplex in $\mathbb{R}^n$ is the intersection of $n+1$ half-spaces, each of which is bounded by a hyperplane with codimension 1.*

This definition will be used primarily when discussing the simplex Y. Define the $n+1$ hyperplanes as

$$P_i = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \cdot \mathbf{e}_i = 0\}, \ i = 0, \dots, n.$$

The corresponding half-space is $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \cdot \mathbf{e}_i \geq 0\}$. It is the set of points $\mathbf{x} \in \mathbb{R}^n$ such that $\operatorname{sign}(\mathbf{x} \cdot \mathbf{e}_i) = 1$. The simplex Y corresponds then exactly to the intersection of these $n+1$ half-spaces. That is, the characteristic function of Y may be expressed as

$$\chi_Y(\mathbf{x}) = \prod_{i=0}^{n} \operatorname{sign}(\mathbf{x} \cdot \mathbf{e}_i). \tag{3.1}$$

Given that the intersection Z lies in both X and Y, it is naturally true that $\chi_Z(\mathbf{x}) = \chi_Y(\mathbf{x})\chi_X(\mathbf{x})$. Since $\chi_X(\mathbf{x}_j) = 1$ (the vertices of X lie within X), $\chi_Z(\mathbf{x}_j) = 1$ if and only if $\chi_Y(\mathbf{x}_j) = 1$, which is true if and only if $\operatorname{sign}(\mathbf{x}_j \cdot \mathbf{e}_i) = 1$ for all i. Thus, the j-th vertex of X is a corner of the polytope Z if and only if all of its barycentric coordinates are non-negative.

Briefly, consider the degenerate case where $\mathbf{x} \cdot \mathbf{e}_i = 0$. Such a point lies on the hyperplane that bounds Y. Define the corresponding half-space to contain this bounding hyperplane, and so $\mathbf{x}$ lies within this half-space. Ultimately this means the boundary of Y is treated no differently than its interior. A point $\mathbf{x}$ on the boundary of Y is then equivalent to one shifted an infinitesimal distance towards the interior for the purposes of this algorithm.

4 Intersections between k-faces of X and hyperplanes of Y

The second definition of a simplex is more useful for discussing the simplex X.

Definition 2 (Simplex 2). *A simplex in $\mathbb{R}^n$ is the convex hull of $n+1$ vertices.*

However, of greater interest is the convex hull of a subset of these vertices, called a k-face. It is defined here and used in Section 4.2.

Definition 3 (k-face). *A k-face of a simplex is the convex hull of $k+1$ of its $n+1$ vertices. Let X_J denote the k-face formed by the $k+1$ vertices of X indexed by the set J.*

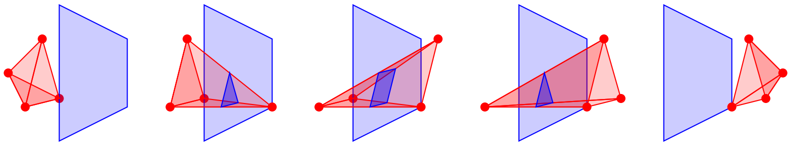

Fig. 3. A 3D example showing the five ways to partition the vertices of X between the two half-spaces created by the sectioning hyperplane P. The intersection between the two is either empty, a triangle or a quadrilateral.

4.1 Sectioning by hyperplanes

In the trivial case where all $\mathbf{x}_j$ lie within Y, $Z = X$. Otherwise, the vertices of Z (if they exist) include intersections between X and the hyperplanes P_i. Since both X and P_i bound Z, their intersection bounds Z.

Take two vertices of X, $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$, and one hyperplane P_i that bounds Y. The edge between the two vertices intersects P_i if and only if $\mathrm{sign}(\mathbf{x}_{j_1} \cdot \mathbf{e}_i) \neq \mathrm{sign}(\mathbf{x}_{j_2} \cdot \mathbf{e}_i)$. This condition is used by ParSIA to determine if an intersection exists. Since $\mathrm{sign}(\mathbf{x})$ is binary-valued the number of intersections that can be calculated by ParSIA is limited. The following proposition ensures that a consistent number of intersections is found by ParSIA. Most importantly, it ensures the resulting intersection is an object with sufficient vertices to have non-zero volume in the hyperplane.

Proposition 4. *Let X be a simplex in $\mathbb{R}^n$, defined as the convex hull of $n + 1$ vertices, intersecting a hyperplane P with codimension 1. The intersection either does not exist or is a polytope in $\mathbb{R}^{n-1}$ with at least n vertices.*

Proof. Let the hyperplane P have normal vector $\mathbf{n}$. The simplex has $n + 1$ vertices $\mathbf{x}_j$. Each vertex has a value of $\mathrm{sign}(\mathbf{x}_j \cdot \mathbf{n})$ which is either 0 or 1.

There are $n + 2$ ways to partition $n + 1$ objects into two groups S_0 and S_1. If one does not care about the difference between S_0 and S_1, as in this case, then there are $\lceil (n + 1)/2 \rceil$ ways to partition the objects. The first of these is to put all objects into S_0. The second is to put all but one into S_0.

Every pair composed of an object from S_0 and another from S_1 represents a vertex of the intersection between X and P. The first partitioning has $S_1 = \emptyset$, meaning no such pairs exist and there is no intersection. The second partitioning has exactly one object in S_1, meaning each pair is composed of this one object and one of the n objects in S_0. This gives n pairs and therefore n vertices, which is a simplex in dimension $n - 1$. Any other partitioning has more objects in S_1 and a non-empty S_0, thus giving more pairs than this second partitioning. The intersection then has at least n vertices in dimension $n - 1$ and is thus a polytope in that dimension. □

This proposition can be reapplied to subsequent intersections. By this proposition any intersection between a simplex and a hyperplane is at least a simplex in the hyperplane. Intersecting this intersection with another hyperplane will then result again in at least a simplex. Outside of degenerate cases where the vertices of the resulting polytope are coplanar, the polytope has non-zero hypervolume.

4.2 Intersection calculation

The intersection $X \cap P_i$ is the convex hull of the set of intersections between the edges of X and P_i:

$$X \cap P_i = \mathrm{Conv}\left(\left\{X_J \cap P_i \;\middle|\; |J| = 2\right\}\right),$$

where X_J denotes a k-face of X, see Definition 3. Denote a particular intersection as $\mathbf{q}^J_{\{i\}} = X_J \cap P_i$. Those points $\mathbf{q}^J_{\{i\}}$ for which $\chi_Y(\mathbf{q}) = 1$ are vertices of Z, as they lie on the boundary of both X and Y. If $\chi_Y(\mathbf{q}) = 0$ for some of these points, then $X \cap P_i$ must be intersected with additional hyperplanes to find that portion of $X \cap P_i$ that lies within Y.

Take the intersection of $X \cap P_{i_1}$ with P_{i_2}, which is equivalent to taking the intersection of X with $P_{i_1} \cap P_{i_2}$. The intersection of the two hyperplanes is itself a hyperplane with codimension 2. Consider two vertices of $X \cap P_{i_1}$, $\mathbf{q}^J_{\{i_1\}}$ and $\mathbf{q}^K_{\{i_1\}}$ such that J and K have an index in common, which lie on opposite sides of the hyperplane P_{i_2}. Both lie on the 2-face of X indexed by $J \cup K$. The intersection between the edge connecting these two vertices and P_{i_2} then also lies on $X_{J\cup K} \cap P_{i_1} \cap P_{i_2}$, which is a single point. Keeping the same notation, this point is written as $\mathbf{q}^{J\cup K}_{\{i_1,i_2\}}$. As before, if $\chi_Y(\mathbf{q}) = 1$ then this point is a vertex of Z.

Repeat this for $k < n$ hyperplanes, arriving at

$$X \bigcap_{m=1}^{k} P_{i_m} = \left(X \bigcap_{m=1}^{k-1} P_{i_m} \right) \cap P_{i_k}.$$

As before, consider two vertices from the previous step that lie on opposite sides of P_{i_k}, $\mathbf{q}^J_\Gamma$ and $\mathbf{q}^K_\Gamma$, where $|J| = |K| = k$, J and K share all but one index, and $\Gamma = \{i_m\}_{m=1}^{k-1}$. Then both lie on the k-face of X indexed by $J \cup K$. The intersection between the edge connecting them and P_{i_k} is the point

$$\mathbf{q}^{J\cup K}_{\Gamma\cup\{i_k\}} = X_{J\cup K} \bigcap_{i \in \Gamma\cup\{i_k\}} P_i,$$

which is a vertex of Z if $\chi_Y(\mathbf{q}) = 1$.

The sets J and K are assumed to share all but one index, a property here referred to as adjacency. This ensures the vertices they index lie on the same k-face, where J and K both have cardinality k, $|J| = |K| = k$. Only these intersections appear on the exterior of X and thus Z. If J and K differ by more than one index then their vertices do not share a k-face and any intersection calculated will be interior to both X and Z and thus not useful to the calculation of the intersection.

Lemma 5. *Suppose the k-face of X indexed by J intersects the k hyperplanes P_i indexed by Γ. Then*

$$\mathbf{q}^J_\Gamma \cdot \mathbf{e}_\eta = \frac{\begin{vmatrix} \mathbf{x}_{j_0} \cdot \mathbf{e}_\eta & \mathbf{x}_{j_0} \cdot \mathbf{e}_{i_1} & \dots & \mathbf{x}_{j_0} \cdot \mathbf{e}_{i_k} \\ \vdots & \vdots & & \vdots \\ \mathbf{x}_{j_k} \cdot \mathbf{e}_\eta & \mathbf{x}_{j_k} \cdot \mathbf{e}_{i_1} & & \mathbf{x}_{j_k} \cdot \mathbf{e}_{i_k} \end{vmatrix}}{\begin{vmatrix} 1 & \mathbf{x}_{j_0} \cdot \mathbf{e}_{i_1} & \dots & \mathbf{x}_{j_0} \cdot \mathbf{e}_{i_k} \\ \vdots & \vdots & & \vdots \\ 1 & \mathbf{x}_{j_k} \cdot \mathbf{e}_{i_1} & & \mathbf{x}_{j_k} \cdot \mathbf{e}_{i_k} \end{vmatrix}} = \frac{\begin{vmatrix} \hat{X}_J^\top \mathbf{e}_\eta & \hat{X}_J^\top I_\Gamma \end{vmatrix}}{\begin{vmatrix} \mathbf{1} & \hat{X}_J^\top I_\Gamma \end{vmatrix}}, \tag{4.1}$$

where $\hat{X}_J$ are the coordinates of the vertices of X_J and I_Γ the columns of the identity matrix indexed by Γ.

Proof. See Appendix A. □

Note that this is neither the most efficient nor accurate method for calculating these intersections. It is used here only for its theoretical value.

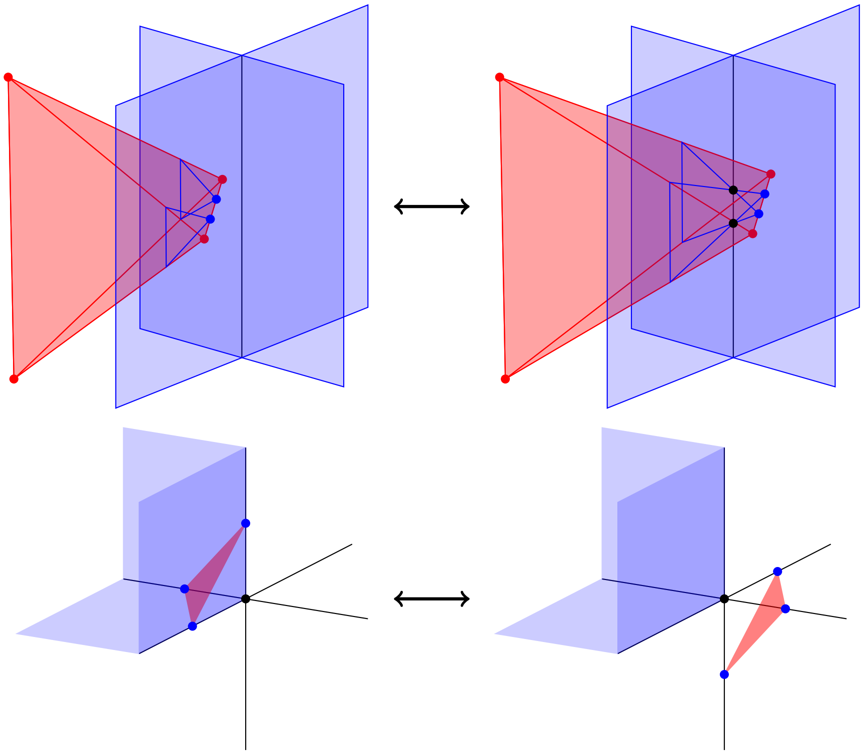

Fig. 4. Entire k-faces of tetrahedra shifting sides relative to intersections of hyperplanes. (Above) A 1-face (edge) moves from the left of the intersection of two hyperplanes (an edge) to the right of it. (Below) A portion of a 2-face (face) moves from one side of the intersection of three hyperplanes (a vertex) to the opposite side.

4.3 Relative position of the intersections

The shape of the polytope Z is determined by the position of its corners relative to the hyperplanes of Y. For example, in [7] the authors identified 10 unique shapes of Z for triangular intersections. The relative position of these corners is due to the signs of their coordinates. A change in sign will then change the shape of Z.

Corollary 6. *The numerator of* $\mathbf{q}^J_\Gamma \cdot \mathbf{e}_\eta$ *is shared with the numerators of* $\mathbf{q}^J_{\Gamma_i} \cdot \mathbf{e}_i$ *for* k *values of* i*, up to a change in sign, where* Γ *and* $\Gamma_i := \{\eta\} \cup \Gamma \setminus \{i\}$ *have cardinality* k*. If* $\mathbf{e}_i$ *is the* s_i*-th column of* Γ *and* $\mathbf{e}_\eta$ *the* s_η*-th column of* Γ_i*, then the sign change occurs if* $s_i + s_\eta$ *is even.*

Proof. For each $i \in \Gamma$, Γ_i is defined by replacing the index i with the index η. Since Γ has k indices there are k such Γ_i. For each of these the numerator of $\mathbf{q}^J_{\Gamma_i} \cdot \mathbf{e}_i$ is the same up to an exchange of columns in the determinant. It takes s_i column exchanges to move $\mathbf{e}_i$ to the front of the determinant while preserving the order of the other columns. It then takes another $s_\eta - 1$ column exchanges to move $\mathbf{e}_\eta$ to its position. Thus, there's a change in sign if $s_i + s_\eta - 1$ is odd. □

By this corollary, if there is a change in sign of $\mathbf{q}^J_\Gamma \cdot \mathbf{e}_\eta$, then the entire k-face X_J ends up on the other side of $P_\eta \bigcap_{i\in\Gamma} P_i$, see Figure 4. All intersections of a given k-face are then consistent with one another, keeping it whole and intact. If the signs were not connected across a k-face then an error in one sign could cause it to distort and break apart.

If X_J does not have $k + 1$ intersections with a given collection of hyperplanes, then this no longer holds true. While the intersections that do exist would be consistent amongst themselves for the collection $\bigcap_{i \in \Gamma} P_i$, they may not be for other collections $P_\eta \bigcap_{i \in \Gamma \setminus \{\gamma\}} P_i$. However, in this case there exist subsets of J and Γ such that

$$\begin{aligned} \text{sign}(\mathbf{q}_{\Gamma \setminus \{\gamma\}}^{J \setminus \{j_1\}} \cdot \mathbf{e}_\eta) &= \text{sign}(\mathbf{q}_{\Gamma \setminus \{\gamma\}}^{J \setminus \{j_2\}} \cdot \mathbf{e}_\eta), \\ \text{sign}(\mathbf{q}_{\Gamma \setminus \{\gamma\}}^{J \setminus \{j_1\}} \cdot \mathbf{e}_\gamma) &\neq \text{sign}(\mathbf{q}_{\Gamma \setminus \{\gamma\}}^{J \setminus \{j_2\}} \cdot \mathbf{e}_\gamma). \end{aligned}$$

That is, the k-face X_J has two intersections with the collection of hyperplanes indexed by $\Gamma \setminus \{\gamma\}$ such that they lie on the same side of P_η and different sides of P_γ. Any point on the edge between these two intersections lies on the same side of P_η as these intersections, including the intersection of this edge with P_γ. Then the sign of $\mathbf{q}_\Gamma^J \cdot \mathbf{e}_\eta$ may be determined without further calculations, and is

$$\text{sign}(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_\eta) = \text{sign}(\mathbf{q}_{\Gamma \setminus \{\gamma\}}^{J \setminus \{j_1\}} \cdot \mathbf{e}_\eta). \tag{4.2}$$

Thus, the position of the k-face relative to the collection of hyperplanes is determined either by the calculation of a single numerator or is predetermined by the coordinates of the $(k - 1)$-faces that compose the k-face.

It is now known that the numerators are strongly linked over a k-face, but what of the denominators? These are unique for each $\mathbf{q}_\Gamma^J$ and so at first glance it appears they cannot be of much use. However, the ratio between denominators of a given k-face for adjacent collections of hyperplanes provides a fundamental relation between the signs of the intersections of that k-face and those of its parent intersections.

Before giving this relation as Lemma 8, define the logical biconditional operator to compensate for the use of the binary-valued sign function.

Definition 7 (Logical biconditional operator). *The operator* $\otimes$ *that acts as* $0 \otimes 0 = 1$, $1 \otimes 1 = 1$ *and* $0 \otimes 1 = 0$ *is called the logical biconditional operator.*

Lemma 8. *Suppose* $\text{sign}(\mathbf{q}_\Gamma^{J_1} \cdot \mathbf{e}_i) \neq \text{sign}(\mathbf{q}_\Gamma^{J_2} \cdot \mathbf{e}_i)$ *and* $\text{sign}(\mathbf{q}_\Gamma^{J_1} \cdot \mathbf{e}_j) \neq \text{sign}(\mathbf{q}_\Gamma^{J_2} \cdot \mathbf{e}_j)$ *for two adjacent sets* J_1 *and* J_2*, then*

$$\text{sign}(\mathbf{q}_{\Gamma \cup \{i\}}^{J_1 \cup J_2} \cdot \mathbf{e}_j) \otimes \text{sign}(\mathbf{q}_{\Gamma \cup \{j\}}^{J_1 \cup J_2} \cdot \mathbf{e}_i) = \text{sign}(\mathbf{q}_\Gamma^{J_1} \cdot \mathbf{e}_i) \otimes \text{sign}(\mathbf{q}_\Gamma^{J_2} \cdot \mathbf{e}_j).$$

Proof. See Appendix B. □

Thus, for every intersection $\mathbf{q}_{\Gamma \cup \{i\}}^{J_1 \cup J_2}$ either there is another intersection $\mathbf{q}_{\Gamma \cup \{j\}}^{J_1 \cup J_2}$ and the two have related signs in relevant directions, or its sign in the relevant direction is inherited from the previous generation.

Once the signs of the intersections $\mathbf{q}_\Gamma^J$ have been found they can be used to identify which intersections need to be calculated for the subsequent generation and which intersections lie within Y. If $\chi_Y(\mathbf{q}) = 1$ using these sign values then $\mathbf{q}_\Gamma^J$ is a corner of the polytope Z. Its equivalent position $\mathbf{w}_\Gamma^J$ on W in the original coordinates is found through a reverse change of coordinates, namely

$$\mathbf{w}_\Gamma^J = \mathbf{v}_0 + \sum_{i \notin \Gamma \cup \{0\}} \left(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_i \right) \mathbf{v}_i. \tag{4.3}$$

Note that the 0th coordinate of $\mathbf{q}_\Gamma^J$ is not required, nor are the i-th coordinates where $i \in \Gamma$ as these are known to be zero. Note also that the magnitude of $\mathbf{q}_\Gamma^J$ is only required if it is found to lie within Y, as otherwise only the signs of its coordinates are of use.

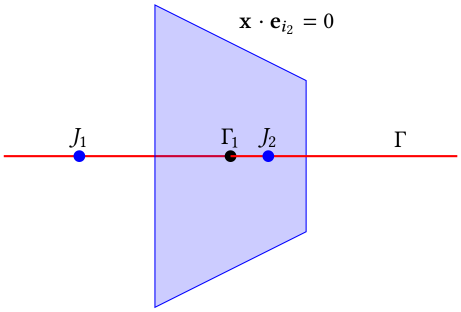

Fig. 5. The hyperplane indexed by i_2 (in blue) intersects the line which is the intersection of all hyperplanes indexed by Γ (in red). The point of their intersection is the vertex $\mathbf{e}_{i_1}$ (in black) which is the intersection of all hyperplanes indexed by Γ_1. Since it lies along the line Γ between the intersections indexed by J_1 and J_2 (in blue) it also lies within the simplex X.

5 Vertices of Y inside X

Once all intersections between X and Y have been found, the only remaining corners of the polytope Z are those vertices of Y which lie entirely within X. To understand how to determine which vertices of Y lie within X, it is easiest to consider a single edge of Y that connects two vertices. Since Y has been defined as the intersection of $n+1$ half-spaces each bounded by a hyperplane P_i, these two vertices are defined as the intersection of n of these hyperplanes. Let the index sets of these hyperplanes be Γ_1 and Γ_2. These sets contain all but one element in $\{0, \dots, n\}$, denoted i_1 and i_2 respectively, which index the hyperplanes opposite the corresponding vertices. Therefore, the two sets have an index set of cardinality $n-1$ in common, denoted Γ. Incidentally, $\Gamma_1 = \Gamma \cup \{i_2\}$ and $\Gamma_2 = \Gamma \cup \{i_1\}$. The intersection $\bigcap_{i \in \Gamma} P_i$ defines the line on which the edge sits.

Since X is convex, there are two possibilities: this infinite line passes through X and there exist two intersections $\mathbf{q}_\Gamma^{J_1}$ and $\mathbf{q}_\Gamma^{J_2}$, or; the line does not pass through X and there are no intersections between the two. In the latter case, any vertices of Y attached to this line cannot reside within X. Suppose then that the former is true, and two facets ($(n-1)$-faces) of X intersect the line indexed by Γ.

The line is a 1-dimensional object and, due to the redundant $\mathbf{e}_0$ coordinate, there are only two coordinates of $\mathbf{q}_\Gamma^{J_1}$ and $\mathbf{q}_\Gamma^{J_2}$ that are non-zero, namely $\mathbf{q}_\Gamma^{J_j} \cdot \mathbf{e}_{i_1}$ and $\mathbf{q}_\Gamma^{J_j} \cdot \mathbf{e}_{i_2}$. A vertex along the line lies within X if and only if these two intersections lie on opposite sides of it. Consider the vertex indexed by Γ_1, which includes i_2 but not i_1. It sits on the line where $\mathbf{x} \cdot \mathbf{e}_{i_2} = 0$. Therefore, this vertex lies within X if and only if $\operatorname{sign}(\mathbf{q}_\Gamma^{J_1} \cdot \mathbf{e}_{i_2}) \neq \operatorname{sign}(\mathbf{q}_\Gamma^{J_2} \cdot \mathbf{e}_{i_2})$, see Figure 5. Likewise, the vertex indexed by Γ_2 lies within X if and only if $\operatorname{sign}(\mathbf{q}_\Gamma^{J_1} \cdot \mathbf{e}_{i_1}) \neq \operatorname{sign}(\mathbf{q}_\Gamma^{J_2} \cdot \mathbf{e}_{i_1})$.

The test of a vertex of Y is essentially a final intersection calculation, adding one last hyperplane to the collection. Since the collection up to this point has formed a line, the resulting intersection is necessarily the 0-dimensional vertex of Y.

Every vertex has n edges extending from it, meaning this test can be performed n times. However, as long as previous steps have been parsimonious and produced consistent results, all tests will agree. Therefore, the test only needs to be performed once.

6 Parsimonious simplicial intersection algorithm

ParSIA is presented as Algorithm 2. It takes arbitrary simplices U and V in $\mathbb{R}^n$. The simplex V is given preferential treatment and ideally has an aspect ratio near 1. Line 1 is discussed in Section 2, lines 2 to 6 in Section 3, lines 9 to 13 in Section 4.1, line 21 in Section 4.2, lines 14 to 28 in Section 4.3, and lines 30 to 35 in Section 5.

Algorithm 2 W =ParSIA(U, V)

```
 1: Find coordinates of vertices of $X$                                        ▷ see equation (2.1)
 2: for $j = 0$ to $n$ do
 3:     if $\chi_Y(\mathbf{x}_j) = 1$ then                                      ▷ see equation (3.1)
 4:         $\mathbf{u}_0 + \mathbf{u}_j \in W$
 5:     end if
 6: end for
 7: Initialize all vertices of $X$ as intersections of the 0-th generation
 8: while there are intersections from the previous generation do
 9:     for each collection $\Gamma$ with intersections, each adjacent pair $(J, K)$, and each coordinate $i \notin \Gamma$ do
10:         if $\text{sign}(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_i) \neq \text{sign}(\mathbf{q}_\Gamma^K \cdot \mathbf{e}_i)$ then
11:             there exists an intersection $\mathbf{q}_{\Gamma \cup \{i\}}^{J \cup K}$
12:         end if
13:     end for
14:     for each intersection $\mathbf{q}_\Gamma^J$ found above do
15:         for each coordinate $j \notin \Gamma$ do
16:             if there is no $\mathbf{q}_\Lambda^J$ for any $\Lambda$ such that $\Lambda \cup \{i\} = \Gamma \cup \{j\}$ then
17:                 inherit $\text{sign}(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_j)$ from the previous generation       ▷ see equation (4.2)
18:             else if $\text{sign}(\mathbf{q}_\Lambda^J \cdot \mathbf{e}_i)$ is known for some $\Lambda$ such that $\Lambda \cup \{i\} = \Gamma \cup \{j\}$ then
19:                 use Lemma 8 to find $\text{sign}(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_j)$
20:             else
21:                 calculate $\mathbf{q}_\Gamma^J \cdot \mathbf{e}_j$, such as by Lemma 5
22:             end if
23:         end for
24:         if $\chi_Y(\mathbf{q}_\Gamma^J) = 1$ then
25:             calculate $\mathbf{q}_\Gamma^J$, if necessary
26:             calculate $\mathbf{w}_\Gamma^J \in W$                                ▷ see equation (4.3)
27:         end if
28:     end for
29: end while
30: for $i = 0$ to $n$ do
31:     choose $\Gamma$ such that $i \notin \Gamma$ and $|\Gamma| = n - 1$
32:     if $\text{sign}(\mathbf{q}_\Gamma^{J_1} \cdot \mathbf{e}_i) \neq \text{sign}(\mathbf{q}_\Gamma^{J_2} \cdot \mathbf{e}_i)$ then
33:         $\mathbf{v}_0 + \mathbf{v}_i \in W$
34:     end if
35: end for
```

The flowchart of this algorithm is presented in Figure 6. Red parallelograms represent inputs and outputs, green rectangles represent actions and calculations, yellow rounded boxes represent for and while loops, and blue diamonds represent tests and if statements.

All signs of intersection pairs for the previous generation must be tested before signs of the next generation can be determined. This is because inheritance can come from any parent pair of the previous generation. This prevents

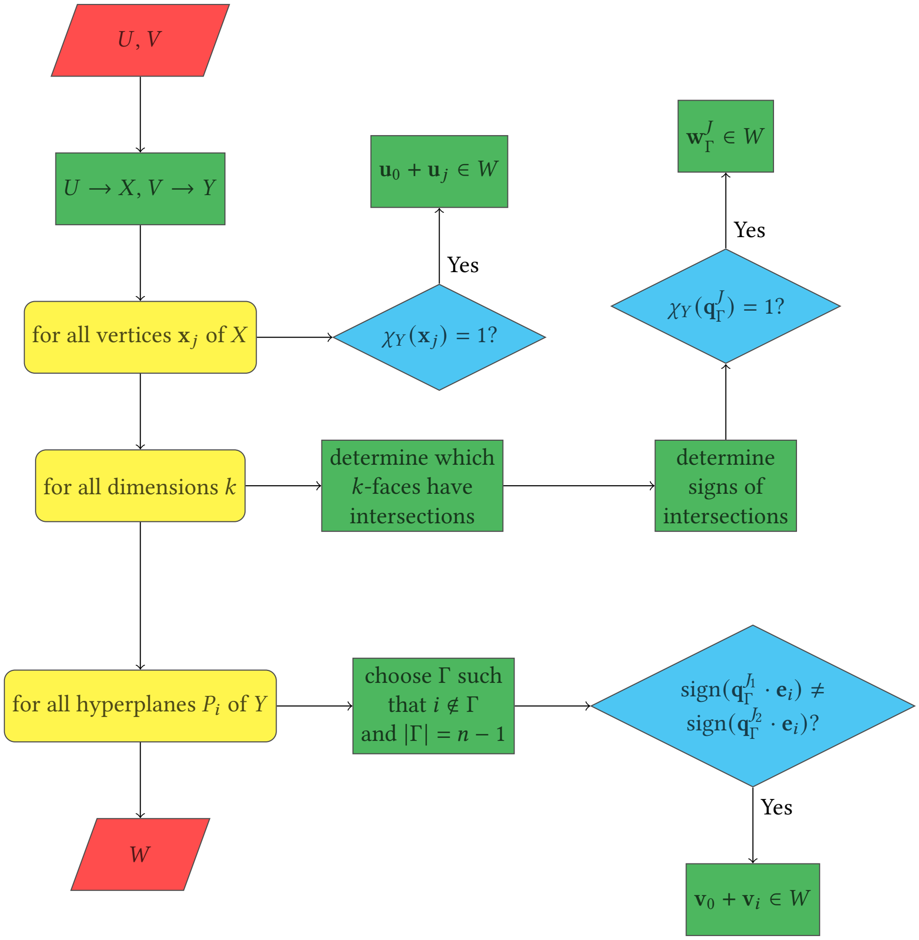
U, V
$U \to X, V \to Y$
for all vertices $\mathbf{x}_j$ of X
$\chi_Y(\mathbf{x}_j) = 1?$
Yes
$\mathbf{u}_0 + \mathbf{u}_j \in W$
for all dimensions k
determine which k-faces have intersections
determine signs of intersections
$\chi_Y(\mathbf{q}^J_\Gamma) = 1?$
Yes
$\mathbf{w}^J_\Gamma \in W$
for all hyperplanes P_i of Y
choose Γ such that $i \notin \Gamma$ and $|\Gamma| = n - 1$
$\mathrm{sign}(\mathbf{q}^{J_1}_\Gamma \cdot \mathbf{e}_i) \neq \mathrm{sign}(\mathbf{q}^{J_2}_\Gamma \cdot \mathbf{e}_i)?$
Yes
$\mathbf{v}_0 + \mathbf{v}_i \in W$
W

Fig. 6. Flowchart of the algorithm. Red parallelograms represent inputs and outputs, green rectangles represent actions and calculations, yellow rounded boxes represent for and while loops, and blue diamonds represent tests and if statements.

proceeding through the algorithm one hyperplane at a time, adding only one at each iteration, as is done in the Sutherland-Hodgman algorithm [1, 13].

If a given collection of hyperplanes is found to have no intersections with X, then no collection that contains this collection as a subset will have intersections. This can be used to limit the combinations of collections checked, improving the efficiency of the algorithm.

If no intersections are found for a given generation, then the algorithm may end immediately without testing for vertices of Y inside X, as there are no intersections to bound the vertices of Y.

Additional modifications can be made to Algorithm 2 to strengthen its robustness. They are not required for Theorem 12, which gives consistency of the algorithm. However, this theorem only holds true if convexity is maintained to a certain degree, with which these modifications may assist. They may also be useful for accuracy, which is not covered by the theorem. No tests run for this article required these modifications, suggesting they are needed only for pathological examples.

	x	y	z
$\mathbf{x}_0$	+	+	+
$\cap P_x$		+	+
$\cap P_y$	-		+
$\cap P_z$	-	-	
$\mathbf{x}_1$	-	-	-

	x	y	z
$\mathbf{x}_0$	+	+	+
$\cap P_x$		+	-
$\cap P_y$	-		+
$\cap P_z$	+	-	
$\mathbf{x}_1$	-	-	-

Table 1. Ordering of intersections in a 3D example that maintain the straight line principle. The coordinates of $\mathbf{x}_0$ are all positive, while the coordinates of $\mathbf{x}_1$ are all negative. (Left) The straight line travelling from $\mathbf{x}_0$ to $\mathbf{x}_1$ passes first through P_x, then P_y, and finally P_z. (Right) Error in the sign of one intersection destroys this ordering and with it convexity of the intersection.

6.1 Straight line principle

Lemma 8 ensures the consistency of intersections between a given k-face and adjacent collections of hyperplanes in any of these hyperplanes. It does not maintain this consistency in the higher dimensional space. That is to say, if a k-face has intersections with at least three adjacent collections of hyperplanes, then it is possible for Algorithm 2 to lose convexity of the k-face through round-off error. To avoid this, one can include additional constraints when determining signs.

Suppose a k-face has intersections with $m \geq 3$ collections of hyperplanes. A straight line of intersections lies between two intersections of the previous generation. This line passes through each hyperplane one at a time. As such, the signs of its coordinates change one at a time. This means there exists an ordering of the intersections based on how similar their signs are with one of the two parent intersections.

To better explain this error, consider an example in 3D where an edge of one tetrahedron X passes near a vertex of the reference tetrahedron Y, passing through all three planes that intersect at this point. Table 1 gives such an example. On the left, the edge between the two vertices passes through P_x, then P_y, then P_z. Any two of these intersections are inextricably linked through Lemma 8, such that the i-th row and i-th column of this table correspond with one another. However, if round-off error causes a flip in sign in one coordinate, as seen on the right of the table, then the ordering of the intersections is destroyed and they are no longer consistent with a straight line passing through three planes. As a result, convexity is lost.

This ordering of the intersections, which shall be referred to as the straight line principle, reduces the degrees of freedom of the signs of the intersections. After finding a given intersection along this line, one can determine its position in the ordering by comparing its coordinates against those of its parents. The position of all other intersections in this ordering relative to this first intersection can then be determined using Lemma 8.

Remark 9 (Straight line principle). *Let $\mathbf{q}_\Gamma^J$ and $\mathbf{q}_\Gamma^K$ be two intersections such that J is adjacent to K. Suppose the line between them intersects $m \geq 3$ hyperplanes, i.e.* $\text{sign}(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_i) \neq \text{sign}(\mathbf{q}_\Gamma^K \cdot \mathbf{e}_i)$ *for m choices of i. If*

$$\text{sign}(\mathbf{q}_{\Gamma\cup\{\gamma\}}^{J\cup K} \cdot \mathbf{e}_i) = \text{sign}(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_i), \quad \text{sign}(\mathbf{q}_{\Gamma\cup\{\gamma\}}^{J\cup K} \cdot \mathbf{e}_j) \neq \text{sign}(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_j),$$

then

$$\text{sign}(\mathbf{q}_{\Gamma\cup\{j\}}^{J\cup K} \cdot \mathbf{e}_i) = \text{sign}(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_i), \quad \text{sign}(\mathbf{q}_{\Gamma\cup\{i\}}^{J\cup K} \cdot \mathbf{e}_j) = \text{sign}(\mathbf{q}_\Gamma^K \cdot \mathbf{e}_j) \neq \text{sign}(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_j).$$

Take, as a higher dimensional example, intersections $\mathbf{q}_\Gamma^J$ and $\mathbf{q}_\Gamma^K$ and seven intersections that lie between them, with planes P_{i_j} for $j = 1, \dots, 7$. Suppose one starts by finding $\mathbf{q}_{\Gamma\cup\{i_3\}}^{J\cup K}$, which shares two signs with $\mathbf{q}_\Gamma^K$. Then it is the third intersection in the straight line starting from $\mathbf{q}_\Gamma^J$ going towards $\mathbf{q}_\Gamma^K$. The two signs that have changed from $\mathbf{q}_\Gamma^J$ indicate the two intersections that appear earlier in the ordering, those with P_{i_1} and P_{i_2}. It must be these hyperplanes because

	$\mathbf{e}_{i_1}$	$\mathbf{e}_{i_2}$	$\mathbf{e}_{i_3}$	$\mathbf{e}_{i_4}$	$\mathbf{e}_{i_5}$	$\mathbf{e}_{i_6}$	$\mathbf{e}_{i_7}$
$\mathbf{q}_\Gamma^J$	+	+	+	+	+	+	+
$\cap P_{i_1}, P_{i_2}$			+	+	+	+	+
				$\vdots$	$\vdots$	$\vdots$	$\vdots$
			+				
$\cap P_{i_3}$	-	-		+	+	+	+
	$\vdots$	$\vdots$	-				
$\cap P_{i_4}, P_{i_5}, P_{i_6}, P_{i_7}$	$\vdots$	$\vdots$	$\vdots$				
	$\vdots$	$\vdots$	$\vdots$				
	-	-	-				
$\mathbf{q}_\Gamma^K$	-	-	-	-	-	-	-

Table 2. Application of the straight line principle. The intersection $\mathbf{q}_{\Gamma\cup\{i_3\}}^{J\cup K}$ is found to be the third intersection when starting from $\mathbf{q}_\Gamma^J$, dividing the remaining intersections into prior and subsequent sets.

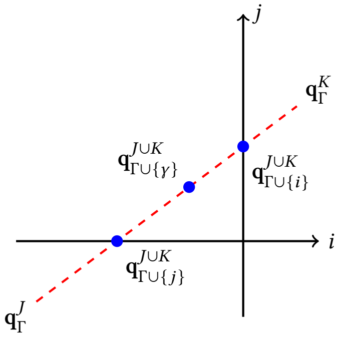
j
$\mathbf{q}_\Gamma^K$
$\mathbf{q}_{\Gamma\cup\{\gamma\}}^{J\cup K}$
$\mathbf{q}_{\Gamma\cup\{i\}}^{J\cup K}$
i
$\mathbf{q}_{\Gamma\cup\{j\}}^{J\cup K}$
$\mathbf{q}_\Gamma^J$

Fig. 7. Graphic representation of the straight line principle. The two intersections $\mathbf{q}_\Gamma^J$ and $\mathbf{q}_\Gamma^K$ are adjacent and have at least three hyperplanes between them: P_i, P_j and P_γ. The intersection $\mathbf{q}_{\Gamma\cup\{\gamma\}}^{J\cup K}$ has P_j between it and $\mathbf{q}_\Gamma^J$ and P_i between it and $\mathbf{q}_\Gamma^K$.

Lemma 8 dictates the sign in the i_3-th coordinate. For these two intersections the signs of the other coordinates are the same as $\mathbf{q}_\Gamma^J$. Likewise, those intersections with P_{i_j} for $j = 4, \ldots, 7$ have the same signs for the coordinates $\mathbf{e}_{i_1}$ and $\mathbf{e}_{i_2}$ as $\mathbf{q}_\Gamma^K$. This is summarized in Table 2.

The straight line principle may be applied after all coordinates of a given intersection have been found, such as immediately after line 35 of Algorithm 2. An additional check is required before calculating the sign of a coordinate directly to make sure it has not been found via this means. The principle is unnecessary when the line intersects fewer than three hyperplanes, as Lemma 8 completely determines consistency.

In many ways, the straight line principle supersedes Lemma 8. The lemma connects only a single row and column of these tables of intersections, while the straight line principle connects the entire off-diagonal blocks. This suggests a similar algebraic lemma and proof exists for the straight line principle that includes Lemma 8 as a special case. A geometric proof is presented in Figure 7.

6.2 Guardrails

ParSIA permits paradoxical results, due in part to the use of $\mathbf{e}_0$ as a coordinate for the purposes of determining relative positions. The absolute position of intersections along this direction can be found using the other coordinates. However, doing so decouples the signs of the intersections, removing the self-consistency of the algorithm's steps. Using $\mathbf{e}_0$ allows the paradox where an intersection has an absolute position on the negative side of P_0 but has a relative position on the positive side, or vice versa. This paradox can also arise due to sign inheritance, at least theoretically.

In most cases, these paradoxes do not present cause for concern. However, when k-faces of the tetrahedra are nearly coincident, intersection calculations become unstable and these paradoxes can induce large additions to the calculated hypervolume.

Luckily, the intersection Z must lie within the tetrahedron Y. This puts absolute limits on the positions of the corners of Z, allowing the addition of guardrails to ParSIA: If a point is determined to be a corner of Z but numerically found to lie outside of Y, then it will be mapped to a nearby point on Y.

In cases where these instabilities arise from coincident k-faces of X and Y, any point on the k-face of Y is consistent, and so it is not necessary to find the exact nearest point. This allows one to eschew a projection for a cheaper heuristic selection. Keep in mind that the coordinates are barycentric, and must be kept so for the reverse transformation, equation (4.3).

Remark 10 (Guardrails). *Suppose Algorithm 2 determines* $\chi_Y(\mathbf{q}_\Gamma^J) = 1$ *but calculates* $\operatorname{sign}(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_i) = 0$ *for some set* Λ *of* i*, i.e.* $\mathbf{q}_\Gamma^J \cdot \mathbf{e}_i < 0$ *for all* $i \in \Lambda$*. Denote*

$$e_d = \sum_{j \notin \Gamma \cup \Lambda} \mathbf{q}_\Gamma^J \cdot \mathbf{e}_j,$$

so that $1 - e_d$ *represents the amount of error that must be spread out over the coordinates so as to place* $\mathbf{q}_\Gamma^J \in Y$ *and keep the coordinates barycentric. If* $\mathbf{q}_\Gamma^J \cdot \mathbf{e}_j \in [0, 1]$ *for all* $j \notin \Gamma \cup \Lambda$*, then set*

$$\mathbf{q}_\Gamma^J \cdot \mathbf{e}_i \leftarrow \begin{cases} 0 & i \in \Lambda, \\ \frac{1}{e_d} \mathbf{q}_\Gamma^J \cdot \mathbf{e}_i & i \notin \Gamma \cup \Lambda. \end{cases}$$

Otherwise, if $\mathbf{q}_\Gamma^J \cdot \mathbf{e}_j > 1$ *for* k *values of* j*, then set* $\mathbf{q}_\Gamma^J \cdot \mathbf{e}_j \leftarrow 1/k$ *and 0 for all other coordinates, thus mapping the point to a* k*-face of* Y*.*

The effects of guardrails are exemplified in Figure 8, which examines the case when Γ contains all but three of the hyperplanes of Y, leaving a single triangular face. Effects are divided by lines where one of the coordinates is equal to one. When all coordinates are less than one, paradoxical points are mapped to k-faces of Y, with positive coordinates modulated so that the coordinates are barycentric.

If the paradoxical point has both negative coordinates and a coordinate greater than one, then the point is mapped to the nearest vertex of Y, identified by the coordinate that is greater than one. If the point has negative coordinates and k coordinates greater than one, then it is mapped to the barycentric centre of the nearest k-face. Other choices are possible, especially projections, but care must be taken to ensure the coordinates sum to one.

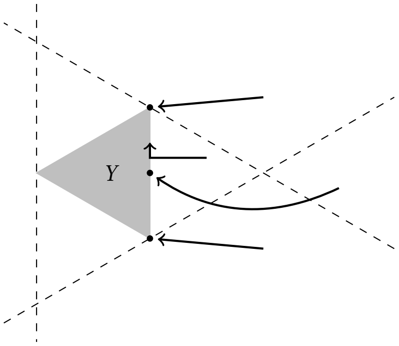

Fig. 8. Effect of guardrails. The shaded triangle represents a 2-face of Y. Dashed lines represent where one of the three non-zero coordinates is equal to one. Guardrails map paradoxical corners of Z to either vertices of this face or the midpoint of one of its edges.

7 Robustness of ParSIA

7.1 Convexity

While geometrically simplices and their intersections are convex, this property is not necessarily maintained numerically by the algorithm. However, as part of the proof in Section 7.2, the calculations must maintain the convexity of the simplices near the hyperplanes. That is, suppose the sections have the following property.

Definition 11 (Cut-convex). *A polytope is cut-convex with respect to a hyperplane P if, for each face of the polytope that intersects P, exactly two edges of the face intersect P.*

Naturally, if a polytope is convex, then it is cut-convex with respect to any hyperplane. This property will ensure the necessary consequences of convexity are preserved for the sections calculated by the algorithm.

It does not appear that enforcing cut-convexity can be done within the algorithm without the addition of many redundant calculations. The property is closely related to the accuracy of the intersection calculations. One can show heuristically that failure of cut-convexity should only occur in pathological examples.

Figure 9 shows the possible configurations of a cell of a simplex with two hyperplanes P_i and P_j. Unless the intersections find themselves on the opposite sides of both hyperplanes then the algorithm fixes the signs of some of the intersections in one of the directions, stabilizing the intersection and ensuring cut-convexity.

In the remaining case, four intersections with $P_i \cap P_j$ can only occur if error causes a complete reordering of the points in the j-direction, which is orthogonal to P_j. In the example in the right of Figure 9, the intersection with the smallest j-coordinate now finds itself with the first or second largest j-coordinate. While the probability of such an error is dependant on the specific calculations used, in general these should occur only when the intersections are nearly coincident in the j-direction, implying the aspect ratio of X is significantly smaller than that of Y.

Figure 10 shows an extreme pathological example, where one tetrahedron with an aspect ratio on the order of $1e-14$ intersects a tetrahedron with an aspect ratio on the order of one. Four intersections between the former and one of the edges of the latter indicate a failure of cut-convexity. However, when the intersection between the tetrahedra is transformed under the affine transformation that maps the flat tetrahedron to the reference tetrahedron, the shape of the polytope does not appear to be unduly affected by this failure. Some erroneous and inaccurate corners can be spotted, but these cause neither significant additions nor subtractions from the volume.

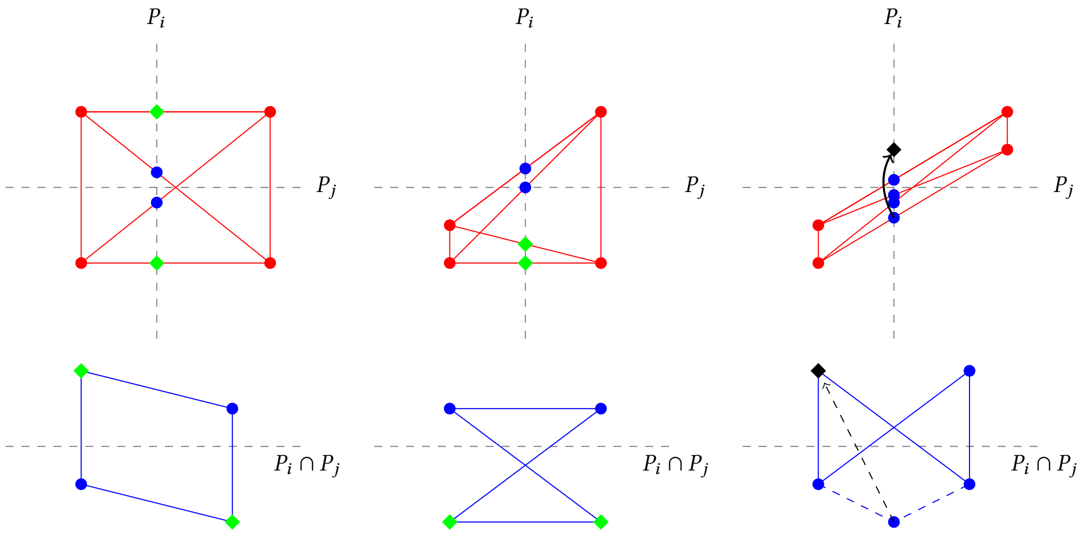

Fig. 9. Configurations of a cell of a simplex intersecting two hyperplanes P_i and P_j. Green diamond intersections have fixed sign in the j-direction. Only for pathological errors when the two simplices have very different aspect ratios does one see four intersections between X and $P_i \cap P_j$.

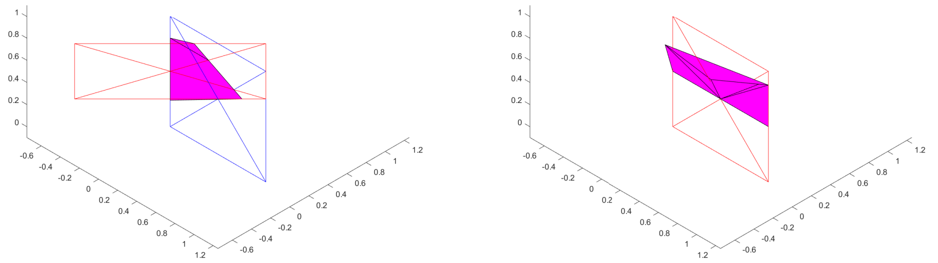

Fig. 10. Extreme pathological example displaying some properties of failure of cut-convexity. (Left) A nearly flat tetrahedron intersects a reference tetrahedron. (Right) The flat tetrahedron transformed to a reference tetrahedron, showing the polytope representing the intersection of the two tetrahedrons.

It is not clear that having this cut-convexity property is strictly necessary for the proof of consistency. The property is used when considering cells of sections, which are 3D objects, and their sectioning by two hyperplanes. Any intersections found should lie along the same 1D space and be adjacent to every other intersection. Furthermore, it is always true that sectioning by a third hyperplane will always result in a single intersection, regardless of error, thanks to the combinatoric representation of the k-faces. Thus, in all practical aspects, it appears to be identical to the same result without error.

In what follows, assume that cut-convexity is maintained, either through sufficiently accurate intersection formulas, restriction to non-pathological simplices, or that the algorithm does so naturally in a way not yet proven.

7.2 Consistency

Theorem 12. *If the intersections of Algorithm 2 are cut-convex with respect to the hyperplanes P_i for all i, then the algorithm is consistent with respect to shape. That is, the number of intersections between k-faces and collections of hyperplanes and the signs of their coordinates are internally consistent at every step with all previous steps.*

Proof. Consider two sets of intersections, one of X with the collection of hyperplanes identified by Γ_i and another with the collection Γ_j, such that Γ_i is adjacent to Γ_j. To be explicit, suppose $i \in \Gamma_i$, $i \notin \Gamma_j$, $j \in \Gamma_j$, and $j \notin \Gamma_i$. The two sets are consistent if sectioning the first with P_j gives the same set of intersections as sectioning the second with P_i. That is, if there exists an adjacent pair (J_1, J_2) such that $\text{sign}(\mathbf{q}_{\Gamma_i}^{J_1} \cdot \mathbf{e}_j) \neq \text{sign}(\mathbf{q}_{\Gamma_i}^{J_2} \cdot \mathbf{e}_j)$, then there exists an adjacent pair (J_3, J_4) such that $\text{sign}(\mathbf{q}_{\Gamma_j}^{J_3} \cdot \mathbf{e}_i) \neq \text{sign}(\mathbf{q}_{\Gamma_j}^{J_4} \cdot \mathbf{e}_i)$. We seek to prove this statement.

By Proposition 4 a convex polytope sectioned by a hyperplane results in a convex polytope. The polytopes of Algorithm 2 are assumed to be cut-convex with respect to the hyperplanes P_i rather than convex. However, if a cut-convex polytope and a convex polytope share a graph and those polytopes and their graphs are sectioned in the same way then the resulting polytopes will again share a graph. Thus, every intersection $\mathbf{q}_\Gamma^J$ is a vertex of a cut-convex polytope. If this polytope has dimension $n > 1$ then the intersection is a vertex of $n-1$ closed cycles of intersections.

Since J_1 and J_2 are adjacent, there exist pairs of adjacent sets (K_1, K_2) and (K_3, K_4) such that

$$\begin{aligned}
\text{sign}(\mathbf{q}_\Gamma^{K_1} \cdot \mathbf{e}_i) &\neq \text{sign}(\mathbf{q}_\Gamma^{K_2} \cdot \mathbf{e}_i), & K_1 \cup K_2 &= J_1, \\
\text{sign}(\mathbf{q}_\Gamma^{K_3} \cdot \mathbf{e}_i) &\neq \text{sign}(\mathbf{q}_\Gamma^{K_4} \cdot \mathbf{e}_i), & K_3 \cup K_4 &= J_2, \\
\text{sign}(\mathbf{q}_\Gamma^{K_1} \cdot \mathbf{e}_i) &= \text{sign}(\mathbf{q}_\Gamma^{K_4} \cdot \mathbf{e}_i), & \Gamma \cup \{i\} &= \Gamma_i, \\
\text{sign}(\mathbf{q}_\Gamma^{K_2} \cdot \mathbf{e}_i) &= \text{sign}(\mathbf{q}_\Gamma^{K_3} \cdot \mathbf{e}_i), & \Gamma \cup \{j\} &= \Gamma_j,
\end{aligned}$$

and these four sets lie on the same cycle and share all but two elements. Furthermore, since $\text{sign}(\mathbf{q}_{\Gamma_i}^{J_1} \cdot \mathbf{e}_j) \neq \text{sign}(\mathbf{q}_{\Gamma_i}^{J_2} \cdot \mathbf{e}_j)$, at least one of the four intersections has a different sign in the j-direction. Thus, there are two edges of the cycle that intersect the hyperplane P_j. This gives the existence of a pair (J_3, J_4). It remains to prove it satisfies the statement. This is done by proving Algorithm 2 correctly codifies the configurations of Figure 11.

If the sets J_3 and J_4 are not identical to J_1 and J_2, then the signs in both directions are inherited from those of the 'parent' intersections. For example, since J_3 is distinct from J_1, the edge it lies on does not connect K_1 to K_2. Likewise, it does not connect K_3 to K_4. However, K_4 and K_1 lie on the same cycle and so the edge of J_3 lies between them. Since $\text{sign}(\mathbf{q}_\Gamma^{K_1} \cdot \mathbf{e}_i) = \text{sign}(\mathbf{q}_\Gamma^{K_4} \cdot \mathbf{e}_i)$ all edges between them lie on the same side of P_i, and so does $\mathbf{q}_{\Gamma_j}^{J_3}$. The left of Figure 11 shows exactly this behaviour.

This leaves two cases to consider: $J_1 = J_4$ and either $J_2 \neq J_3$ or $J_2 = J_3$. This first equality, true in both cases, implies there exists intersections of the k-face J_1 with the two collections of hyperplanes Γ_i and Γ_j. By Lemma 8, $\text{sign}(\mathbf{q}_{\Gamma_j}^{J_1} \cdot \mathbf{e}_i) = \text{sign}(\mathbf{q}_{\Gamma_i}^{J_1} \cdot \mathbf{e}_j) \otimes \text{sign}(\mathbf{q}_\Gamma^{K_1} \cdot \mathbf{e}_i) \otimes \text{sign}(\mathbf{q}_\Gamma^{K_2} \cdot \mathbf{e}_j)$. Also, in both cases $\text{sign}(\mathbf{q}_\Gamma^{K_2} \cdot \mathbf{e}_j) = \text{sign}(\mathbf{q}_\Gamma^{K_3} \cdot \mathbf{e}_j)$.

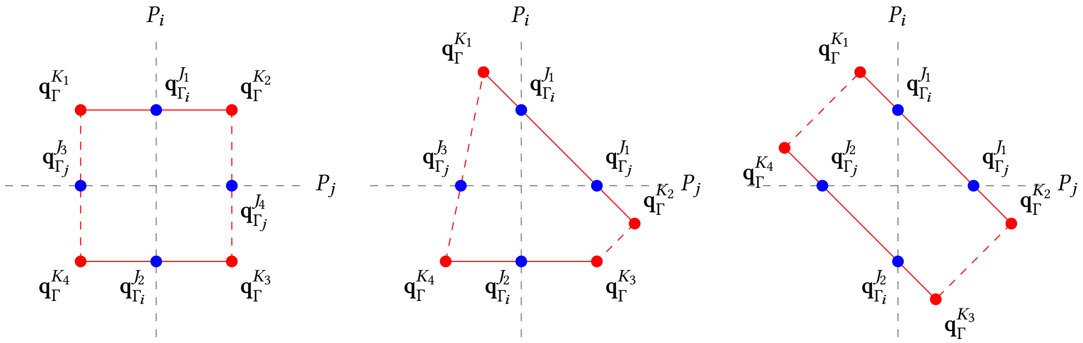
P_i
$\mathbf{q}_\Gamma^{K_1}$
$\mathbf{q}_{\Gamma_i}^{J_1}$
$\mathbf{q}_\Gamma^{K_2}$
$\mathbf{q}_{\Gamma_j}^{J_3}$
P_j
$\mathbf{q}_{\Gamma_j}^{J_4}$
$\mathbf{q}_\Gamma^{K_4}$
$\mathbf{q}_{\Gamma_i}^{J_2}$
$\mathbf{q}_\Gamma^{K_3}$
P_i
$\mathbf{q}_\Gamma^{K_1}$
$\mathbf{q}_{\Gamma_i}^{J_1}$
$\mathbf{q}_{\Gamma_j}^{J_3}$
$\mathbf{q}_{\Gamma_j}^{J_1}$
$\mathbf{q}_\Gamma^{K_2}$
P_j
$\mathbf{q}_\Gamma^{K_4}$
$\mathbf{q}_{\Gamma_i}^{J_2}$
$\mathbf{q}_\Gamma^{K_3}$
P_i
$\mathbf{q}_\Gamma^{K_1}$
$\mathbf{q}_{\Gamma_i}^{J_1}$
$\mathbf{q}_\Gamma^{K_4}$
$\mathbf{q}_{\Gamma_j}^{J_2}$
$\mathbf{q}_{\Gamma_j}^{J_1}$
$\mathbf{q}_\Gamma^{K_2}$
P_j
$\mathbf{q}_{\Gamma_i}^{J_2}$
$\mathbf{q}_\Gamma^{K_3}$

Fig. 11. Configurations of a cycle of intersections with respect to two hyperplanes. The dashed lines represent all edges between the given intersections, which may number as low as zero (though this would exclude the left-most configuration).

If $J_2 \neq J_3$, see middle of Figure 11, then $\mathbf{q}_\Gamma^{K_4}$ lies between $\mathbf{q}_{\Gamma_j}^{J_2}$ and $\mathbf{q}_{\Gamma_i}^{J_3}$ on the cycle. Thus,

$$\begin{aligned}
\operatorname{sign}(\mathbf{q}_{\Gamma_j}^{J_3} \cdot \mathbf{e}_i) &= \operatorname{sign}(\mathbf{q}_\Gamma^{K_4} \cdot \mathbf{e}_i) = \operatorname{sign}(\mathbf{q}_\Gamma^{K_1} \cdot \mathbf{e}_i) \\
&= \operatorname{sign}(\mathbf{q}_{\Gamma_j}^{J_1} \cdot \mathbf{e}_i) \otimes \operatorname{sign}(\mathbf{q}_{\Gamma_i}^{J_1} \cdot \mathbf{e}_j) \otimes \operatorname{sign}(\mathbf{q}_\Gamma^{K_2} \cdot \mathbf{e}_j) \\
&= \operatorname{sign}(\mathbf{q}_{\Gamma_j}^{J_1} \cdot \mathbf{e}_i) \otimes \operatorname{sign}(\mathbf{q}_{\Gamma_i}^{J_1} \cdot \mathbf{e}_j) \otimes \operatorname{sign}(\mathbf{q}_\Gamma^{K_3} \cdot \mathbf{e}_j) \\
&= \operatorname{sign}(\mathbf{q}_{\Gamma_j}^{J_1} \cdot \mathbf{e}_i) \otimes \operatorname{sign}(\mathbf{q}_{\Gamma_i}^{J_1} \cdot \mathbf{e}_j) \otimes \operatorname{sign}(\mathbf{q}_{\Gamma_i}^{J_2} \cdot \mathbf{e}_j) \\
&\neq \operatorname{sign}(\mathbf{q}_{\Gamma_j}^{J_1} \cdot \mathbf{e}_i),
\end{aligned}$$

giving the desired relationship.

If $J_2 = J_3$, see right of Figure 11, then Lemma 8 applies again and

$$\begin{aligned}
\operatorname{sign}(\mathbf{q}_{\Gamma_j}^{J_2} \cdot \mathbf{e}_i) &= \operatorname{sign}(\mathbf{q}_{\Gamma_i}^{J_2} \cdot \mathbf{e}_j) \otimes \operatorname{sign}(\mathbf{q}_\Gamma^{K_3} \cdot \mathbf{e}_j) \otimes \operatorname{sign}(\mathbf{q}_\Gamma^{K_4} \cdot \mathbf{e}_i) \\
&\neq \operatorname{sign}(\mathbf{q}_{\Gamma_i}^{J_1} \cdot \mathbf{e}_j) \otimes \operatorname{sign}(\mathbf{q}_\Gamma^{K_2} \cdot \mathbf{e}_j) \otimes \operatorname{sign}(\mathbf{q}_\Gamma^{K_1} \cdot \mathbf{e}_i) = \operatorname{sign}(\mathbf{q}_{\Gamma_j}^{J_1} \cdot \mathbf{e}_i),
\end{aligned}$$

again giving the desired relationship. □

The proof of Theorem 12 identifies those aspects of any algorithm that would provide robustness: maintaining cut-convexity; dividing vertices and intersections into negative and non-negative half-spaces, and; connecting the signs through equation (4.2), Corollary 6, and Lemma 8.

The shape remains consistent, but may differ from exact results by a small polytope. The size of this small polytope is dictated by the methods used to calculate the coordinate transformation and the intersections, as well as the aspect ratio of V and the difference in size of the two simplices.

8 Numerical examples

This section aims to stress test the algorithm on numerical examples generalized from one found in [7]. There, an integer number of congruent triangles are placed in a wheel, then a copy is made and shifted and rotated infinitesimally. The intersection algorithm is then tested on the intersections between the two wheels.

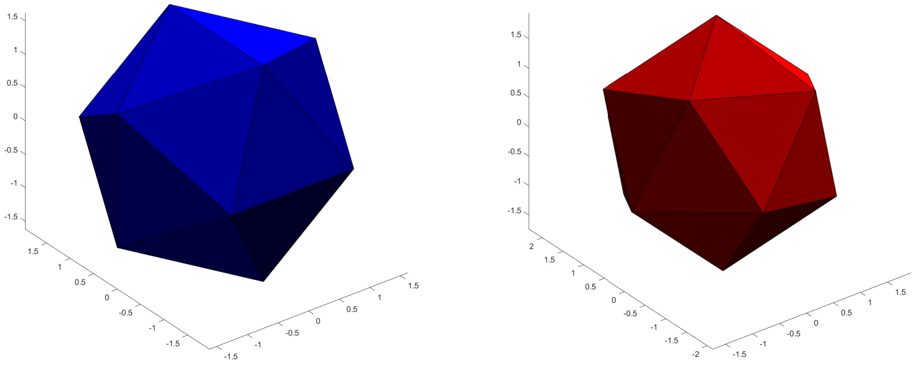

Fig. 12. Two icosahedral meshes, each containing 20 tetrahedra, to be intersected. One is a slight affine transformation of the other.

The generalization in higher dimensions is to use regular polytopes with simplicial facets, set an additional point at the centre of the polytopes and extend edges between this centre and each vertex of the polytope. This forms several congruent simplices arranged within a hypersphere of unit radius. In 3D, the icosahedron is divided into 20 tetrahedra. In 4D, the 600-cell is divided into 600 4-simplices. There are no higher dimensional equivalents beyond 4D. The copy of the mesh is rotated θ degrees in all directions and translated ϵ. The 3D example is shown in Figure 12. To avoid unnecessary intersections, the intersection algorithm is paired with an advancing front algorithm.

Figure 13 gives histograms of this example using $\theta = 0.01$ and $\epsilon = 1e - 3$, run in MATLAB. In 3D, of 400 possible tetrahedral intersections, 156 are non-empty. In 4D, of 360,000 possible 4-simplicial intersections, 24,596 are non-empty. Volumes are computed using the built-in `convhull()` function. Hypervolumes are computed using the built-in `convhulln()` function with option `'Qj'`, which 'jiggles' the inputs if they appear to be coplanar. This option limits the level of accuracy possible for very thin intersections, which are common in this example.

The exact intersections are unknown. Instead, comparison is made against the established Sutherland-Hodgman algorithm [1, 13]. Both algorithms use Lemma 5 for intersection calculations, which most likely can be improved upon. Results show the two algorithms are comparable, up to the precision of the volume and hypervolume calculations.

It is also possible to compare computation time of these two algorithms. Figure 14 shows histograms of the relative speed-up of Algorithm 2 over the Sutherland-Hodgman algorithm. The medians of these datasets are shown as solid red lines, while a speed-up of 1, representing equal time between the two algorithms, appears as dashed black lines.

On the majority of individual calculations, Sutherland-Hodgman is faster, as measured by the distance between a relative speed-up of 1 and the median of the dataset. However, this distance is small relative to the variance of the speed-ups. In 3D, the majority of speed-ups are within 133%, meaning ParSIA takes only 33% extra time for most intersections. Speed-ups were as low as 2.66% and as high as 1600%. In 4D, the majority of speed-ups are within 219%. Speed-ups were as low as 1.81×10^{-4}% and as high as 9980%.

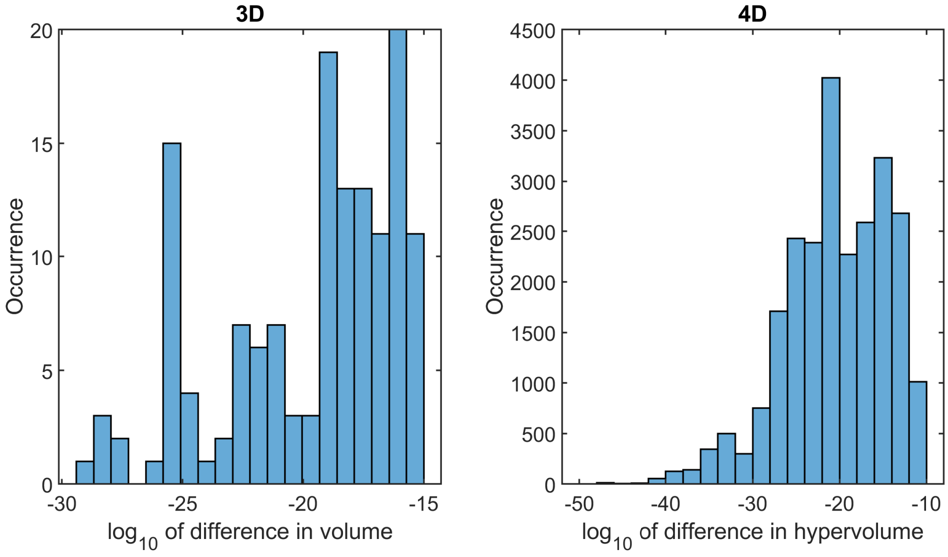

Fig. 13. Difference in volume and hypervolume between Sutherland-Hodgman and ParSIA in solving the described example with $\theta = 0.01$ and $\epsilon = 1e - 3$. (Left) 3D icosahedral mesh. (Right) 4D 600-cell mesh.

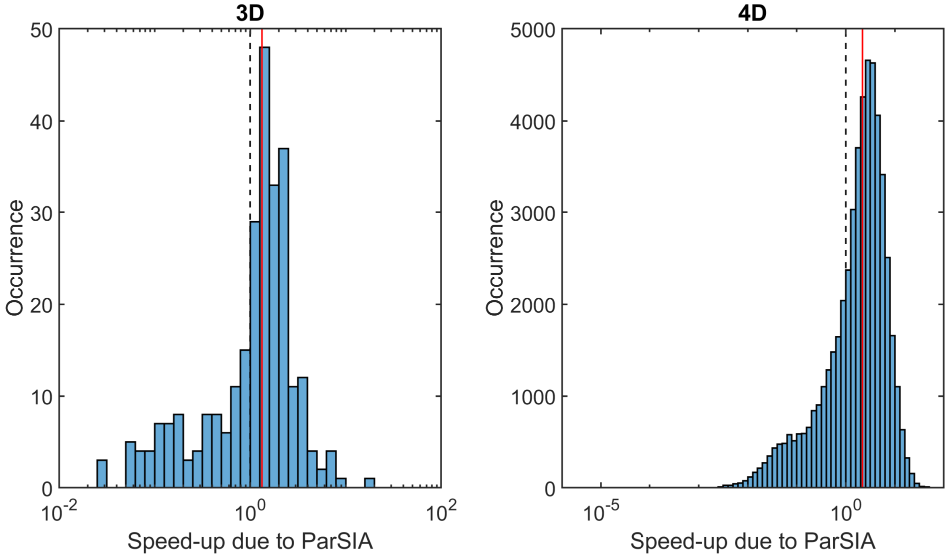

Fig. 14. Relative computation time of ParSIA over Sutherland-Hodgman in solving the same example. (Left) 3D icosahedral mesh. (Right) 4D 600-cell mesh.

This tells us that, while any given calculation is best handled by Sutherland-Hodgman, for those instances when ParSIA is faster, it tends to be significantly faster. This can change the overall speed of this example, which involves several hundred such calculations in 3D and several thousand in 4D. In 3D, the total time to compute this example

Table 3. Coordinates of sample 4-simplices U and V.

Coord.	U					Coord.	V				
1st	0	-1	-0.8090	-0.8090	-0.8090	1st	0.0010	-0.9989	-0.8161	-0.8159	-0.8519
2nd	0	0	0.5000	0	0.3090	2nd	0.0010	-0.0090	0.4897	-0.0153	0.2969
3rd	0	0	0.3090	0.5000	0	3rd	0.0010	-0.0090	0.3069	0.4897	-0.0091
4th	0	0	0	0.3090	0.5000	4th	0.0010	-0.0090	0.0010	0.3069	0.4959

Table 4. Initial information determined from vertices of X and their signs.

		j				
		0	1	2	3	4
	0	1	0	1	1	1
	1	1	1	0	0	0
i	2	0	1	1	1	1
	3	0	1	0	1	1
	4	0	1	1	1	1

(a) $\text{sign}(\mathbf{x}_j \cdot \mathbf{e}_i)$.

		J									
		01	02	12	03	13	23	04	14	24	34
	0	1		1		1			1		
	1		1	1	1	1		1	1		
Γ	2	1	1		1			1			
	3	1		1	1		1	1		1	
	4	1	1		1			1			

(b) $\exists \mathbf{q}_\Gamma^J$? 1 indicates yes.

was 134% longer for ParSIA than for Sutherland-Hodgman. In 4D, the total time for ParSIA was 7.24% the time for Sutherland-Hodgman, making it significantly faster.

Because of the advancing front algorithm and differences in their calculations, both algorithms performed calculations to confirm empty intersections that were not performed by the other. This cannot be represented in the comparison of computation time for individual intersections, but it is included in the comparison of total computation time.

It is worth noting that this implementation of ParSIA prioritizes simplifying the structure of the code rather than its efficiency, making it easier to understand. In particular, the size of the output is not pre-allocated, which accounts for the most time of any single line of code. As well, the combinatorial subroutines used to organize the index sets for both k-faces and hyperplanes make use of a global variable. These design choices leave room for improvement.

To show how ParSIA works, consider two of the 4-simplices from this example that intersect. Their coordinates are presented in Table 3. These coordinates are transformed such that V is set as the reference simplex Y and U becomes the simplex X. The binary signs of its barycentric coordinates are found in Table 4a. No vertex has a column of 1's, meaning none lie within Y. Using the signs from Table 4a, one can produce Table 4b, showing which edges of X intersect which hyperplanes of Y. Blank entries indicate no intersection and the possibility of sign inheritance.

A total of 24 intersections are found, each with four non-zero coordinates. Table 5a gives the signs for all intersections on the edge indexed by $J = \{0, 1\}$. The symbol $*$ indicates the sign is inherited, while $\dagger$ indicates the use of Lemma 8. If no symbol is displayed, the sign is found through direct calculation of the coordinate. The lower triangle of this table is found from the upper triangle through Lemma 8, while the second column is found through inheritance, since no intersection for the plane P_1 was found. This has left only six of the 16 signs to compute directly. From this table, the intersections $\mathbf{q}_0^{01}$ and $\mathbf{q}_2^{01}$ lie within Y and form part of the intersection Z. Three additional coordinates must then be computed directly and equation (4.3) applied. Table 5a is repeated for each edge J. From Table 5a and its counterparts for all edges J, one can deduce the existence of intersections between faces of X and pairs of hyperplanes of Y, see Table 5b. No intersections are found for hyperplane pairs $\Gamma = \{0, 1\}$, $\{0, 2\}$, $\{1, 2\}$ or $\{0, 4\}$, nor for faces $J = \{1, 3, 4\}$ or $\{2, 3, 4\}$.

Table 5. Information concerning first generation intersections.

		i				
		0	1	2	3	4
	0		1^*	1	1	1
	1					
Γ	2	$1^\dagger$	1^*		1	1
	3	$1^\dagger$	1^*	$0^\dagger$		0
	4	$1^\dagger$	1^*	$0^\dagger$	$1^\dagger$	

(a) $\text{sign}(\mathbf{q}_\Gamma^{01} \cdot \mathbf{e}_i)$.

		J							
		012	013	023	123	014	024	124	034
	03	1			1			1	
	13			1	1	1		1	1
	23	1		1		1			1
Γ	14	1	1				1		1
	24	1	1				1		1
	34	1		1		1			1

(b) $\exists \mathbf{q}_\Gamma^J$? 1 indicates yes.

Table 6. Information concerning second and third generation intersections.

		i				
		0	1	2	3	4
	03		1^*	1^*		1^*
	23	$1^\dagger$	1^*			0
Γ	14	1^*		1^*	0^*	
	24	1^*	1^*		$1^\dagger$	
	34	1^*	1^*	$1^\dagger$		

(a) $\text{sign}(\mathbf{q}_\Gamma^{012} \cdot \mathbf{e}_i)$.

	i				
	0	1	2	3	4
$\mathbf{q}_{134}^{0123}$	1^*		1^*		
$\mathbf{q}_{134}^{0234}$	1^*		1^*		
$\mathbf{q}_{234}^{0124}$	1^*	1^*			
$\mathbf{q}_{234}^{0234}$	1^*	1^*			

(b) $\text{sign}(\mathbf{q}_\Gamma^J \cdot \mathbf{e}_i)$.

As before, consider the specific face 012 and its five intersections. Their signs are found in Table 6a, which indicates $\mathbf{q}_{03}^{012}$, $\mathbf{q}_{24}^{012}$ and $\mathbf{q}_{34}^{012}$ are corners of Z. All nine of their coordinates must be calculated directly.

There are then the cells of X which intersect the lines formed by three hyperplanes of Y. By comparing signs, only four intersections are found: $\mathbf{q}_{134}^{0123}$, $\mathbf{q}_{134}^{0234}$, $\mathbf{q}_{234}^{0124}$, and $\mathbf{q}_{234}^{0234}$. Table 6b gives their signs, all of which were determined through inheritance. All four are corners of Z, requiring calculation of their coordinates and application of equation (4.3). Each of the two lines, indexed by $\Gamma = \{1, 3, 4\}$ and $\{2, 3, 4\}$, has two intersections with X. Neither of these pairs of intersections bounds either vertex along their respective lines, meaning no vertices of Y lie within X. This completes the intersection algorithm.

9 Conclusion

Intersection algorithms are crucial components in many applications. While efficiency is a priority for most of these applications, a lack of robustness can pose serious issues. As found in [7] an inconsistent intersection algorithm can completely invalidate results.

Algorithm 2 has been written under the principle of parsimony to ensure consistency. This consistency is proven in Theorem 12. All results from the algorithm correspond to two intersecting simplices. While these may not be exactly the two simplices given as input to the algorithm, they are within numerical error of them.

To this last point, the accuracy of the algorithm depends strongly on how the intersections between k-faces and hyperplanes are calculated. Several options are available, including the one presented in this paper, see Lemma 5, though this option is not ideal. Also important for accuracy is the coordinate transformation, see equation (2.1), which can be treated as the calculation of the barycentric coordinates of one simplex relative to the other.

The efficiency of the algorithm depends on its implementation. Some concerns in this regard have been mentioned in this paper in Section 6. Other concerns include determining which intersections are adjacent and how best to loop through the collections of hyperplanes. Additional work on using combinatorial representations to solve these problems is forthcoming.

A Proof of Lemma 5

Without loss of generality, suppose $J = \{0, \dots, k\}$. Write $\mathbf{q}_\Gamma^J$ in barycentric coordinates of the k-face of X:

$$\mathbf{q}_\Gamma^J = \hat{X}_J \mathbf{u},$$

then the vector $\mathbf{u}$ satisfies $\mathbf{1}^\top \mathbf{u} = 1$. The intersection $\mathbf{q}_\Gamma^J$ lies on all hyperplanes indexed by Γ, meaning $I_\Gamma^\top \mathbf{q}_\Gamma^J = \mathbf{0}$. Put together, $\mathbf{u}$ solves the system

$$\begin{bmatrix} \mathbf{1}^\top \\ I_\Gamma^\top \hat{X}_J \end{bmatrix} \mathbf{u} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}.$$

Using Cramer's rule, the j-th element of u, denoted u_j and indicating the amount of $\mathbf{x}_j$ found in $\mathbf{q}_\Gamma^J$, may be written as

$$u_j = \frac{\begin{vmatrix} \dots & 1 & \dots \\ I_\Gamma^\top \hat{X}_{J<j} & \mathbf{0} & I_\Gamma^\top \hat{X}_{J>j} \end{vmatrix}}{\begin{vmatrix} \mathbf{1}^\top \\ I_\Gamma^\top \hat{X}_J \end{vmatrix}} = \frac{\begin{vmatrix} \mathbf{0}^\top & 1 & \mathbf{0}^\top \\ & I_\Gamma^\top \hat{X}_J & \end{vmatrix}}{\begin{vmatrix} \mathbf{1}^\top \\ I_\Gamma^\top \hat{X}_J \end{vmatrix}}$$

by expanding the determinant along the j-th column, where $\hat{X}_{J<j}$ are those columns of $\hat{X}_J$ whose indices are less than j and $\hat{X}_{J>j}$ those whose indices are greater than j.

Now we seek a representation of $\mathbf{q}_\Gamma^J \cdot \mathbf{e}_\eta$:

$$\mathbf{q}_\Gamma^J \cdot \mathbf{e}_\eta = \mathbf{e}_\eta^\top \hat{X}_J \mathbf{u} = \sum_{j=0}^{k} \mathbf{e}_\eta^\top \mathbf{x}_j u_j = \sum_{j=0}^{k} \frac{\begin{vmatrix} \mathbf{0}^\top & \mathbf{e}_\eta^\top \mathbf{x}_j & \mathbf{0}^\top \\ & I_\Gamma^\top \hat{X}_J & \end{vmatrix}}{\begin{vmatrix} \mathbf{1}^\top \\ I_\Gamma^\top \hat{X}_J \end{vmatrix}} = \frac{\begin{vmatrix} \mathbf{e}_\eta^\top \hat{X}_J \\ I_\Gamma^\top \hat{X}_J \end{vmatrix}}{\begin{vmatrix} \mathbf{1}^\top \\ I_\Gamma^\top \hat{X}_J \end{vmatrix}}.$$

The statement of the lemma follows from the fact that the determinant is invariant under transposition.

B Proof of Lemma 8

Let us begin by specifying some notation. Let $K = J_1 \cap J_2$. Let $k_1 \in J_1$ but $k_1 \notin J_2$, and likewise $k_2 \in J_2$, $k_2 \notin J_1$. Suppose there are $s_i - 1$ elements of Γ that are smaller than i, and $s_j - 1$ elements that are smaller than j.

By the assumptions on the i and j-coordinates of $\mathbf{q}_\Gamma^{J_1}$ and $\mathbf{q}_\Gamma^{J_2}$, one may write

$$\begin{aligned} \operatorname{sign}\left(\frac{\mathbf{q}_\Gamma^{J_1} \cdot \mathbf{e}_i - \mathbf{q}_\Gamma^{J_2} \cdot \mathbf{e}_i}{\mathbf{q}_\Gamma^{J_1} \cdot \mathbf{e}_j - \mathbf{q}_\Gamma^{J_2} \cdot \mathbf{e}_j} \right) &= \operatorname{sign}(\mathbf{q}_\Gamma^{J_1} \cdot \mathbf{e}_i) \otimes \operatorname{sign}(\mathbf{q}_\Gamma^{J_1} \cdot \mathbf{e}_j) \\ &= \operatorname{sign}(\mathbf{q}_\Gamma^{J_1} \cdot \mathbf{e}_i) \otimes \operatorname{sign}(\mathbf{q}_\Gamma^{J_2} \cdot \mathbf{e}_j) \otimes 0. \end{aligned}$$

We may expand the left hand side using Lemma 5, then simplify:

$$\begin{aligned}
\operatorname{sign}(\mathbf{q}_\Gamma^{J_1}\cdot\mathbf{e}_i)\otimes\operatorname{sign}(\mathbf{q}_\Gamma^{J_2}\cdot\mathbf{e}_j)\otimes 0 &= \operatorname{sign}\left(\frac{\dfrac{\begin{vmatrix}\hat{X}_{J_1}^\top\mathbf{e}_i & \hat{X}_{J_1}^\top I_\Gamma\end{vmatrix}}{\begin{vmatrix}\mathbf{1} & \hat{X}_{J_1}^\top I_\Gamma\end{vmatrix}} - \dfrac{\begin{vmatrix}\hat{X}_{J_2}^\top\mathbf{e}_i & \hat{X}_{J_2}^\top I_\Gamma\end{vmatrix}}{\begin{vmatrix}\mathbf{1} & \hat{X}_{J_2}^\top I_\Gamma\end{vmatrix}}}{\dfrac{\begin{vmatrix}\hat{X}_{J_1}^\top\mathbf{e}_j & \hat{X}_{J_1}^\top I_\Gamma\end{vmatrix}}{\begin{vmatrix}\mathbf{1} & \hat{X}_{J_1}^\top I_\Gamma\end{vmatrix}} - \dfrac{\begin{vmatrix}\hat{X}_{J_2}^\top\mathbf{e}_j & \hat{X}_{J_2}^\top I_\Gamma\end{vmatrix}}{\begin{vmatrix}\mathbf{1} & \hat{X}_{J_2}^\top I_\Gamma\end{vmatrix}}}\right) \\
&= \operatorname{sign}\left(\frac{\begin{vmatrix}\hat{X}_{J_1}^\top\mathbf{e}_i & \hat{X}_{J_1}^\top I_\Gamma\end{vmatrix}\begin{vmatrix}\mathbf{1} & \hat{X}_{J_2}^\top I_\Gamma\end{vmatrix} - \begin{vmatrix}\hat{X}_{J_2}^\top\mathbf{e}_i & \hat{X}_{J_2}^\top I_\Gamma\end{vmatrix}\begin{vmatrix}\mathbf{1} & \hat{X}_{J_1}^\top I_\Gamma\end{vmatrix}}{\begin{vmatrix}\hat{X}_{J_1}^\top\mathbf{e}_j & \hat{X}_{J_1}^\top I_\Gamma\end{vmatrix}\begin{vmatrix}\mathbf{1} & \hat{X}_{J_2}^\top I_\Gamma\end{vmatrix} - \begin{vmatrix}\hat{X}_{J_2}^\top\mathbf{e}_j & \hat{X}_{J_2}^\top I_\Gamma\end{vmatrix}\begin{vmatrix}\mathbf{1} & \hat{X}_{J_1}^\top I_\Gamma\end{vmatrix}}\right).
\end{aligned}$$

Next, we perform elementary operations to move the rows associated with $k_1 \in J_1$ and $k_2 \in J_2$ to the top of their respective matrices. Since we do this to all matrices there is no change in sign. Then we may write

$$\begin{aligned}
&\operatorname{sign}(\mathbf{q}_\Gamma^{J_1}\cdot\mathbf{e}_i)\otimes\operatorname{sign}(\mathbf{q}_\Gamma^{J_2}\cdot\mathbf{e}_j)\otimes 0 \\
&= \operatorname{sign}\left(\frac{\begin{vmatrix}\mathbf{x}_{k_1}^\top\mathbf{e}_i & \mathbf{x}_{k_1}^\top I_\Gamma \\ \hat{X}_K^\top\mathbf{e}_i & \hat{X}_K^\top I_\Gamma\end{vmatrix}\begin{vmatrix}1 & \mathbf{x}_{k_2}^\top I_\Gamma \\ \mathbf{1} & \hat{X}_K^\top I_\Gamma\end{vmatrix} - \begin{vmatrix}\mathbf{x}_{k_2}^\top\mathbf{e}_i & \mathbf{x}_{k_2}^\top I_\Gamma \\ \hat{X}_K^\top\mathbf{e}_i & \hat{X}_K^\top I_\Gamma\end{vmatrix}\begin{vmatrix}1 & \mathbf{x}_{k_1}^\top I_\Gamma \\ \mathbf{1} & \hat{X}_K^\top I_\Gamma\end{vmatrix}}{\begin{vmatrix}\mathbf{x}_{k_1}^\top\mathbf{e}_j & \mathbf{x}_{k_1}^\top I_\Gamma \\ \hat{X}_K^\top\mathbf{e}_j & \hat{X}_K^\top I_\Gamma\end{vmatrix}\begin{vmatrix}1 & \mathbf{x}_{k_2}^\top I_\Gamma \\ \mathbf{1} & \hat{X}_K^\top I_\Gamma\end{vmatrix} - \begin{vmatrix}\mathbf{x}_{k_2}^\top\mathbf{e}_j & \mathbf{x}_{k_2}^\top I_\Gamma \\ \hat{X}_K^\top\mathbf{e}_j & \hat{X}_K^\top I_\Gamma\end{vmatrix}\begin{vmatrix}1 & \mathbf{x}_{k_1}^\top I_\Gamma \\ \mathbf{1} & \hat{X}_K^\top I_\Gamma\end{vmatrix}}\right).
\end{aligned}$$

We may use the Schur complement to expand some of the determinants:

$$\begin{aligned}
\begin{vmatrix}1 & \mathbf{x}_{k_1}^\top I_\Gamma \\ \mathbf{1} & \hat{X}_K^\top I_\Gamma\end{vmatrix} &= \begin{vmatrix}\hat{X}_K^\top I_\Gamma\end{vmatrix}\left(1 - \mathbf{x}_{k_1}^\top I_\Gamma\left(\hat{X}_K^\top I_\Gamma\right)^{-1}\mathbf{1}\right), \\
\begin{vmatrix}1 & \mathbf{x}_{k_2}^\top I_\Gamma \\ \mathbf{1} & \hat{X}_K^\top I_\Gamma\end{vmatrix} &= \begin{vmatrix}\hat{X}_K^\top I_\Gamma\end{vmatrix}\left(1 - \mathbf{x}_{k_2}^\top I_\Gamma\left(\hat{X}_K^\top I_\Gamma\right)^{-1}\mathbf{1}\right).
\end{aligned}$$

This allows us to put the terms of the previous equation into a single determinant, with the term $\begin{vmatrix}\hat{X}_K^\top I_\Gamma\end{vmatrix}$ cancelling out, as it appears in both numerator and denominator:

$$\operatorname{sign}(\mathbf{q}_\Gamma^{J_1}\cdot\mathbf{e}_i)\otimes\operatorname{sign}(\mathbf{q}_\Gamma^{J_2}\cdot\mathbf{e}_j)\otimes 0 = \operatorname{sign}\left(\frac{\begin{vmatrix}1 - \mathbf{x}_{k_2}^\top I_\Gamma\left(\hat{X}_K^\top I_\Gamma\right)^{-1}\mathbf{1} & \mathbf{x}_{k_2}^\top\mathbf{e}_i & \mathbf{x}_{k_2}^\top I_\Gamma \\ 1 - \mathbf{x}_{k_1}^\top I_\Gamma\left(\hat{X}_K^\top I_\Gamma\right)^{-1}\mathbf{1} & \mathbf{x}_{k_1}^\top\mathbf{e}_i & \mathbf{x}_{k_1}^\top I_\Gamma \\ \mathbf{0} & \hat{X}_K^\top\mathbf{e}_i & \hat{X}_K^\top I_\Gamma\end{vmatrix}}{\begin{vmatrix}1 - \mathbf{x}_{k_2}^\top I_\Gamma\left(\hat{X}_K^\top I_\Gamma\right)^{-1}\mathbf{1} & \mathbf{x}_{k_2}^\top\mathbf{e}_j & \mathbf{x}_{k_2}^\top I_\Gamma \\ 1 - \mathbf{x}_{k_1}^\top I_\Gamma\left(\hat{X}_K^\top I_\Gamma\right)^{-1}\mathbf{1} & \mathbf{x}_{k_1}^\top\mathbf{e}_j & \mathbf{x}_{k_1}^\top I_\Gamma \\ \mathbf{0} & \hat{X}_K^\top\mathbf{e}_j & \hat{X}_K^\top I_\Gamma\end{vmatrix}}\right).$$

Add a linear combination of the columns to cancel out part of the Schur complement:

$$
\begin{aligned}
&\operatorname{sign}(\mathbf{q}_\Gamma^{J_1}\cdot\mathbf{e}_i)\otimes\operatorname{sign}(\mathbf{q}_\Gamma^{J_2}\cdot\mathbf{e}_j)\otimes 0\\
&=\operatorname{sign}\left(\frac{\begin{vmatrix} 1 & \mathbf{x}_{k_2}^\top\mathbf{e}_i & \mathbf{x}_{k_2}^\top I_\Gamma \\ 1 & \mathbf{x}_{k_1}^\top\mathbf{e}_i & \mathbf{x}_{k_1}^\top I_\Gamma \\ \hat{X}_K^\top I_\Gamma\left(\hat{X}_K^\top I_\Gamma\right)^{-1}\mathbf{1} & \hat{X}_K^\top\mathbf{e}_i & \hat{X}_K^\top I_\Gamma \end{vmatrix}}{\begin{vmatrix} 1 & \mathbf{x}_{k_2}^\top\mathbf{e}_j & \mathbf{x}_{k_2}^\top I_\Gamma \\ 1 & \mathbf{x}_{k_1}^\top\mathbf{e}_j & \mathbf{x}_{k_1}^\top I_\Gamma \\ \hat{X}_K^\top I_\Gamma\left(\hat{X}_K^\top I_\Gamma\right)^{-1}\mathbf{1} & \hat{X}_K^\top\mathbf{e}_j & \hat{X}_K^\top I_\Gamma \end{vmatrix}}\right)=\operatorname{sign}\left(\frac{\begin{vmatrix} 1 & \mathbf{x}_{k_2}^\top\mathbf{e}_i & \mathbf{x}_{k_2}^\top I_\Gamma \\ 1 & \mathbf{x}_{k_1}^\top\mathbf{e}_i & \mathbf{x}_{k_1}^\top I_\Gamma \\ \mathbf{1} & \hat{X}_K^\top\mathbf{e}_i & \hat{X}_K^\top I_\Gamma \end{vmatrix}}{\begin{vmatrix} 1 & \mathbf{x}_{k_2}^\top\mathbf{e}_j & \mathbf{x}_{k_2}^\top I_\Gamma \\ 1 & \mathbf{x}_{k_1}^\top\mathbf{e}_j & \mathbf{x}_{k_1}^\top I_\Gamma \\ \mathbf{1} & \hat{X}_K^\top\mathbf{e}_j & \hat{X}_K^\top I_\Gamma \end{vmatrix}}\right).
\end{aligned}
$$

It remains to perform the elementary operations that move the first two rows and the second column into their appropriate positions. The rows are done to both numerator and denominator and therefore do not cause a change in sign. To move the columns there are $s_i - 1$ exchanges to the numerator and $s_j - 1$ to the denominator. This results in

$$
\operatorname{sign}\left(\frac{\begin{vmatrix}\mathbf{1} & \hat{X}_{J_1\cup J_2}^\top I_{\Gamma\cup\{i\}}\end{vmatrix}}{\begin{vmatrix}\mathbf{1} & \hat{X}_{J_1\cup J_2}^\top I_{\Gamma\cup\{j\}}\end{vmatrix}}\right)=0^{\otimes s_i+s_j-1}\otimes\operatorname{sign}(\mathbf{q}_\Gamma^{J_1}\cdot\mathbf{e}_i)\otimes\operatorname{sign}(\mathbf{q}_\Gamma^{J_2}\cdot\mathbf{e}_j).
$$

Using Lemma 5 we can make the following relations:

$$
\begin{aligned}
\operatorname{sign}\left(\frac{\mathbf{q}_{\Gamma\cup\{i\}}^{J_1\cup J_2}\cdot\mathbf{e}_j}{\mathbf{q}_{\Gamma\cup\{j\}}^{J_1\cup J_2}\cdot\mathbf{e}_i}\right)&=\operatorname{sign}\left(\frac{\begin{vmatrix}\hat{X}_{J_1\cup J_2}^\top\mathbf{e}_j & \hat{X}_{J_1\cup J_2}^\top I_{\Gamma\cup\{i\}}\end{vmatrix}}{\begin{vmatrix}\mathbf{1} & \hat{X}_{J_1\cup J_2}^\top I_{\Gamma\cup\{i\}}\end{vmatrix}}\,\frac{\begin{vmatrix}\mathbf{1} & \hat{X}_{J_1\cup J_2}^\top I_{\Gamma\cup\{j\}}\end{vmatrix}}{\begin{vmatrix}\hat{X}_{J_1\cup J_2}^\top\mathbf{e}_i & \hat{X}_{J_1\cup J_2}^\top I_{\Gamma\cup\{j\}}\end{vmatrix}}\right)\\
&=0^{\otimes s_i+s_j-1}\otimes\operatorname{sign}\left(\frac{\begin{vmatrix}\hat{X}_{J_1\cup J_2}^\top I_{\Gamma\cup\{i,j\}}\end{vmatrix}}{\begin{vmatrix}\hat{X}_{J_1\cup J_2}^\top I_{\Gamma\cup\{i,j\}}\end{vmatrix}}\,\frac{\begin{vmatrix}\mathbf{1} & \hat{X}_{J_1\cup J_2}^\top I_{\Gamma\cup\{j\}}\end{vmatrix}}{\begin{vmatrix}\mathbf{1} & \hat{X}_{J_1\cup J_2}^\top I_{\Gamma\cup\{i\}}\end{vmatrix}}\right)\\
&=\operatorname{sign}(\mathbf{q}_\Gamma^{J_1}\cdot\mathbf{e}_i)\otimes\operatorname{sign}(\mathbf{q}_\Gamma^{J_2}\cdot\mathbf{e}_j).
\end{aligned}
$$

This is equivalent to the statement of the lemma.

References

[1] V. Broman and M. Shensa, A compact algorithm for the intersection and approximation of N-dimensional polytopes, Mathematics and Computers in Simulation, 32 (1990), pp. 469–480.

[2] M. Cyrus and J. Beck, Generalized two-and three-dimensional clipping, Computers & Graphics, 3 (1978), pp. 23–28.

[3] M. J. Gander and C. Japhet, An algorithm for non-matching grid projections with linear complexity, in Domain decomposition methods in science and engineering XVIII, Springer, Berlin, Heidelberg, 2009, pp. 185–192.

[4] ———, Algorithm 932: PANG: software for nonmatching grid projections in 2D and 3D with linear complexity, ACM Transactions on Mathematical Software, 40 (2013), p. 6.

[5] Y.-D. Liang and B. A. Barsky, A new concept and method for line clipping, ACM Transactions on Graphics, 3 (1984), pp. 1–22.

[6] H. Liu, H. Ma, Q. Liu, X. Tang, and J. Fish, An efficient and robust GPGPU-parallelized contact algorithm for the combined finite-discrete element method, Computer Methods in Applied Mechanics and Engineering, 395 (2022), p. 114981.

[7] C. McCoid and M. J. Gander, A provably robust algorithm for triangle-triangle intersections in floating-point arithmetic, ACM Transactions on Mathematical Software, 48 (2022), pp. 1–30.

[8] A. Munjiza, D. Owen, and N. Bicanic, A combined finite-discrete element method in transient dynamics of fracturing solids, Engineering computations, (1995).

[9] A. Munjiza, E. Rougier, Z. Lei, and E. E. Knight, FSIS: a novel fluid–solid interaction solver for fracturing and fragmenting solids, Computational Particle Mechanics, 7 (2020), pp. 789–805.

[10] M. A. Puso, A 3D mortar method for solid mechanics, International Journal for Numerical Methods in Engineering, 59 (2004), pp. 315–336.

[11] V. Skala, An efficient algorithm for line clipping by convex polygon, Computers & Graphics, 17 (1993), pp. 417–421.

[12] V. Skala, A brief survey of clipping and intersection algorithms with a list of references (including triangle-triangle intersections), Informatica, 34 (2023), pp. 169–198.

[13] I. E. Sutherland and G. W. Hodgman, Reentrant polygon clipping, Communications of the ACM, 17 (1974), pp. 32–42.

[14] K. Weiler and P. Atherton, Hidden surface removal using polygon area sorting, in Computer Graphics, vol. 11, New York, 1977, ACM, pp. 214–222.