Pergamon

Appl. Math. Lett. Vol. 8, No. 4, pp. 13–18, 1995

0893-9659(95)00039-9
0893-9659/95 $9.50 + 0.00

# An Algorithm for the Numerical Detection of Simplex Overlap

M. L. BRODZIK
Department of Mathematics and Statistics, University of Pittsburgh
Pittsburgh, PA 15260, U.S.A.

(Received and accepted March 1995)

**Abstract**—A method is presented for testing whether or not two $n$-simplices in $\mathbb{R}^n$ intersect, and if so, deciding whether or not the intersection has a nonempty interior. The algorithm is an application of a method by Stewart for solving linear inequalities [1].

**Keywords**—Intersection detection, Simplex, Computational geometry, Numerical algorithms.

## 1. INTRODUCTION

The intersection of geometrical objects has applications in various fields including robotics, geometric modelling, interactive graphics, and computational mathematics. An important class of such geometric objects are the simplices. Many finite element algorithms use simplicial complexes as grids, and such complexes have also been used to approximate submanifolds [2]. The construction of a simplicial complex must avoid any occurrences of 'unacceptable' simplex-pair intersections defined as intersections that contain interior points of both simplices. We present here an algorithm which determines whether or not two $n$-simplices in $\mathbb{R}^n$ intersect, and if so, decides whether or not the intersection contains interior points relative to $\mathbb{R}^n$.

So far, most of the known intersection algorithms appear to apply only to geometric objects in two and three dimensions; see, e.g., [3–7]. A divide-and-conquer approach is used in [3,4] to decide whether or not two convex sets in $\mathbb{R}^2$ or $\mathbb{R}^3$, whose boundaries are algebraic curves, intersect; and a point in the intersection or a separating line or plane are computed. The algorithms in [5–7] decide whether two convex objects in $\mathbb{R}^2$ or $\mathbb{R}^3$ intersect by either finding a wedge (in $\mathbb{R}^2$) or a polygonal cone (in $\mathbb{R}^3$) which contains all of one object but nothing of the other, or by finding a point which belongs to both objects. Both of these approaches appear to be intrinsically designed for two and three dimensions.

An algorithm in [8] constructs the intersection of two polyhedra in $\mathbb{R}^n$, using an algorithm which is based on the authors' system of definitions and theorems meant to serve as a basis for the computational geometry of polyhedra in $\mathbb{R}^n$. By extracting from this approach only that which is necessary for detecting, rather than constructing, the intersection of two simplices, we are left with the need for testing each facet of one $n$-simplex against each edge of the other $n$-simplex. Since this amounts to solving up to $n(n+1)^2$ linear systems, the approach quickly becomes very costly for rising $n$.

Here, we present a new approach based on formulating the problem as a linear inequality. Then an iterative method developed by Stewart in [1] for solving linear inequalities is applied. Each

This work was supported in part by ONR-Grant N-00014-90-J-1025.

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

step requires the solution of one linear system, and numerical experiments for $n$ up to 6 have so far required no more than ten iterations to reach a conclusion for one simplex pair.

## 2. THE PROBLEM DESCRIPTION AND EQUIVALENT FORMULATIONS

Two nondegenerate $n$-simplices $\Sigma_1 = \sigma[\mathbf{x}_0, \dots, \mathbf{x}_n]$ and $\Sigma_2 = \sigma[\mathbf{y}_0, \dots, \mathbf{y}_n]$ intersect if there exist barycentric coordinates $\{\xi_i\}_{i=0}^n$ and $\{\eta_i\}_{i=0}^n$ with respect to the vertices of $\Sigma_1$ and $\Sigma_2$, respectively, which satisfy the following problem.

PROBLEM P1.

$$\sum_{j=0}^{n} \xi_j \mathbf{x}_j = \sum_{j=0}^{n} \eta_j \mathbf{y}_j, \qquad \xi_j \geq 0, \quad \eta_j \geq 0, \quad j = 0, \dots, n, \tag{1}$$

$$\sum_{j=0}^{n} \xi_j = 1, \quad \sum_{j=0}^{n} \eta_j = 1. \tag{2}$$

The intersection is unacceptable if there exists a solution to $P1$ which is strictly positive.

THEOREM 1. *Problem $P1$ has a solution if and only if Problem $P2$, defined by (1) and*

$$\sum_{j=0}^{n} \xi_j = \sum_{j=0}^{n} \eta_j, \tag{3}$$

*has a nontrivial solution.*

PROOF. Clearly, if $P1$ has a solution then that solution is nontrivial, and it is also a solution of $P2$. If $P2$ has a nontrivial solution $\{\alpha_j, \beta_j\}_{j=0}^n$, then with

$$\mu_j = \frac{\alpha_j}{\Sigma_{j=0}^n \alpha_j}, \quad \nu_j = \frac{\beta_j}{\Sigma_{j=0}^n \beta_j},$$

it is easily checked that $\{\mu_j, \nu_j\}_{j=0}^n$ satisfies (1) and (2). ∎

By Theorem 1, it suffices to determine if $P2$ has a nontrivial solution. Moreover, if $P2$ has a strictly positive solution, then so will $P1$. A matrix formulation of $P2$ is

$$\mathbf{B}\,\mathbf{w} = \mathbf{0}, \qquad \mathbf{w} \geq \mathbf{0}, \tag{4}$$

where

$$\mathbf{B} = (\mathbf{B}_1 \quad \mathbf{B}_2), \qquad \mathbf{w} = (\alpha_0 \ \ \dots \ \ \alpha_n \ \ \beta_0 \ \ \dots \ \ \beta_n)^\top$$
$$\mathbf{B}_1 = \begin{pmatrix} \mathbf{x}_0 & \dots & \mathbf{x}_n \\ 1 & \dots & 1 \end{pmatrix}, \qquad \mathbf{B}_2 = \begin{pmatrix} -y_0 & \dots & -y_n \\ -1 & \dots & -1 \end{pmatrix}.$$

Since $\mathbf{x}_0, \dots, \mathbf{x}_n$ are assumed to be the vertices of a nondegenerate simplex, they are in general position. Hence, $\mathbf{B}_1$ is nonsingular and $\mathbf{B}$ has full rank. A QR factorization of $\mathbf{B}^\top$ gives

$$\mathbf{B}^\top \mathbf{P} = (\mathbf{Q}_1 \quad \mathbf{Q}_2)\,\mathbf{R},$$

where the $n+1$ columns of $\mathbf{A} := \mathbf{Q}_2$ span the null space of $\mathbf{B}$. Clearly, (4) has a nontrivial solution if and only if the linear inequality

$$\mathbf{A}\,\mathbf{z} \geq \mathbf{0} \tag{5}$$

has a nontrivial solution $\mathbf{z} \in \mathbb{R}^{n+1}$, and, if so, then $\mathbf{w} := \mathbf{A}\mathbf{z}$ is the corresponding nontrivial solution to (4).

Define $\mathcal{P}(\mathbf{z}) := \{i : (\mathbf{A}\mathbf{z})_i > 0\}$, $\mathcal{Z}(\mathbf{z}) := \{i : (\mathbf{A}\mathbf{z})_i = 0\}$, and $\mathcal{N}(\mathbf{z}) := \{i : (\mathbf{A}\mathbf{z})_i < 0\}$. Thus, $\mathcal{P}$, $\mathcal{Z}$, and $\mathcal{N}$ comprise the indices for which the components of $\mathbf{A}\mathbf{z}$ are positive, zero, and negative, respectively. A *maximally positive* (MP) solution $\mathbf{z}^*$ of (5) is defined to be one for which the cardinality of $\mathcal{P}(\mathbf{z}^*)$ is largest. MP solutions are not unique, but they all have the same sets $\mathcal{P}^* = \mathcal{P}(\mathbf{z}^*)$ and $\mathcal{Z}^* = \mathcal{Z}(\mathbf{z}^*)$. If an MP solution of (5) has $\mathcal{Z}^* = \emptyset$, then there exists a corresponding strictly positive set of barycentric coordinates satisfying $P1$, and hence, the simplex intersection is unacceptable. If an MP solution of (5) has $\mathcal{Z}^* \neq \emptyset$, then the corresponding solution of $P1$ is not strictly positive, and the simplex intersection is acceptable.

## 3. STEWART'S METHOD

The mentioned method of Stewart [1] is used here to find nontrivial maximally positive solutions of the homogeneous inequality in (5) when such solutions exist, and to indicate when such solutions do not exist. Its underlying idea is simple. Consider the function

$$f(\mathbf{z}) = \mathbf{1}^\top \mathbf{exp}(-\mathbf{A}\mathbf{z}),$$

where $\mathbf{1} = (\,1 \quad 1 \quad \dots \quad 1\,)^\top$, and $\mathbf{exp}(\mathbf{y}) = (\,e^{y_0} \quad e^{y_1} \quad \dots \quad e^{y_n}\,)^\top$ for any $\mathbf{y} \in \mathbb{R}^{n+1}$. In [1], it was shown that one of two things must happen if $f$ is minimized iteratively. If (5) has no nontrivial solution, then $f$ has a unique minimum, to which the iteration must converge. On the other hand, if (5) has a nontrivial solution, then the iterates grow unboundedly in such a way that a solution can be computed from them. For the numerical solution of the inequality (5), Newton's method with line search is used to produce a sequence of vectors $\{\mathbf{z}^k\}$ such that

$$\lim \mathbf{z}^k = \inf f(\mathbf{z}).$$

In [1], it was shown that this implies the method computes an MP solution when a nontrivial solution exists. In that case, the components of $\mathbf{A}\mathbf{z}^k$ divide into two classes, namely those which converge and those which grow unboundedly. The indices of the former make up the set $\mathcal{Z}^*$, while those of the latter constitute the set $\mathcal{P}^*$. This behavior of the sequence $\{\mathbf{A}\mathbf{z}^k\}$ is essential in our case because it indicates whether or not a nonempty intersection of two simplices is acceptable.

Newton's method with line search generates an iterate $\mathbf{z}^{k+1}$ from a previous iterate $\mathbf{z}^k$ as follows:

1. Calculate the descent direction $\mathbf{d}^k := -f''(\mathbf{z}^k)^{-1} f'(\mathbf{z}^k)$.
2. Set $\mathbf{z}^{k+1} := \mathbf{z}^k + \lambda_k \mathbf{d}^k$ for some $\lambda_k > 0$ that makes $\mathbf{z}^{k+1}$ an acceptable iterate.

The "line search" in Step 2 may be either a method that chooses a $\lambda_k$ as an exact or approximate solution of the one-dimensional minimization problem, or any other method that performs as well in theory. With such a line search method, the sequence $\mathbf{z}^k$ converges, ultimately quadratically, if $f$ has a minimum.

## 4. THE ALGORITHM

The line search method used here is the backtracking line search algorithm of [9]. The strategy is to start with $\lambda_k = 1$, and then, if $\mathbf{z}^k + \mathbf{d}^k$ is not acceptable, to "backtrack" (reduce $\lambda_k$) until an acceptable $\mathbf{z}^k + \lambda_k \mathbf{d}^k$ is found. This can be summarized as follows:

**Lnsrch:** **Input:** current point $\mathbf{z}$, descent direction $\mathbf{d}$, $\gamma \in (0, \frac{1}{2})$;
  Set $\lambda := 1.0$
  **while** $f(\mathbf{z} + \lambda\mathbf{d}) > f(\mathbf{z}) + \gamma\lambda f'(\mathbf{z})^\top \mathbf{d}$:
    Set $\lambda := \rho\lambda$, for some $\rho \in (0, 1)$

```
                /*ρ is chosen anew each time by the line search. See [9].*/
    end while
    Output: λ.
```

The simplex overlap detection algorithm then has the following form:

```
Ovrlap: Input: matrix B in (4), start point z^0, tolerances ε_0, ε_g and ε_s;
    B^T P = ( Q_1   Q_2 ) R    /*Get QR decomposition of B^T.*/
    Set A := Q_2
    Initialize: γ := 10^{-4}
                fin := .FALSE.
                class_i := -1,   i = 1,...,2(n+1)
    while (.NOT. fin)
        Set d^k := -f''(z^k)^{-1} f'(z^k)    /*Get descent direction.*/
        λ_k := Lnsrch[z^k, d^k, γ].    /*Perform line search.*/
        Set z^{k+1} := z^k + λ_k d^k
        Set zstep^{k+1} := z^{k+1} - z^k
        Set step^{k+1} := Az^{k+1} - Az^k
        if (k ≥ 1) then
            Initialize: cntneg := 0
            for i = 1,...,2(n+1):          /*Update class.*/
                if (|step_i^{k+1}| + |step_i^k| ≤ ε_0 ) then
                    class_i = 0       /*{(Az)_i^j} converging over j.*/
                else if (step_i^{k+1} ≥ step_i^k ≥ 0.5 and Az_i^{k+1} ≥ 1.0) then
                    class_i := max(+1, class_i + 1) /*{(Az)_i^j} diverging over j.*/
                else
                    class_i := -1       /*Behavior of {(Az)_i^j} not yet known.*/
                    cntneg = cntneg + 1
                end if
            end for
        end if
        if ( (relative f'(z^{k+1})) < ε_g or (relative zstep^{k+1}) < ε_s ) then
            olap := 0   /*Sequence {z^{k+1}} converging. No overlap.*/
            fin := .TRUE.
        else if ( (class_i = 0 or class_i ≥ 3 ), for all i ) then
            if (class_i = 0, for some i) then
                olap := +1    /*Acceptable overlap.*/
            else
                olap := -1    /*Unacceptable overlap.*/
            end if
            fin := .TRUE.
        end if
    end while
    Output: olap.
```

## 5. NUMERICAL EXPERIMENTS

The algorithm **Ovrlap** was implemented in Fortran 77. We present here three examples of 6-simplex pairs whose overlap structures were numerically determined by the code.

The pair of simplices in the first example is separated by a gap of width 0.01 between two parallel facets of the pair, one facet from each simplex. Specifically, the vertices of the first simplex are $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{e}^i$, $i = 1, \ldots, 6$, where $\mathbf{e}^i$ is the $i^{\text{th}}$ standard basis vector of $\mathbb{R}^6$. The vertices of the second simplex are $\mathbf{y}_0 = -0.01\mathbf{e}^6$, $\mathbf{y}_6 = -1.01\mathbf{e}^6$, and $\mathbf{y}_i = \mathbf{y}_{i-1} + \mathbf{e}^i$, $i = 1, \dots, 5$. These two simplices do not intersect, but are very close to each other. The algorithm concluded 'no overlap' in 9 steps.

The second pair of simplices intersect in only one shared vertex. The first simplex is the same as that in the first example, and the second simplex, a translation of the first, defined by the vertices $\mathbf{y}_i = \mathbf{x}_i + \mathbf{e}^6$. The algorithm detected 'acceptable overlap' in 6 steps.

The vertices of the two simplices in the third example are the same as those in the second pair with the exception that $\mathbf{y}_5 = \mathbf{x}_5 + 0.99\mathbf{e}^6$. This third pair is a slight perturbation of the second pair; in the second simplex, the shared vertex has been moved by only 0.01 units toward the other simplex, resulting in a very small overlap with nonempty interior. The algorithm detected 'unacceptable overlap' in 8 iterations.

For each of these three examples, the terminal values of $\mathbf{Az}^k$ and **class** are shown in Table 1. The array **class** is used only in the case where $\{\mathbf{z}^k\}$ is not converging, indicating simplex overlap. It is used to determine whether or not the overlap is acceptable. When $\{(\mathbf{Az}^k)_i\}$ is converging over $k$, $\mathbf{class}_i$ is set to zero. When divergence of $\{(\mathbf{Az}^k)_i\}$ over $k$ is numerically detected, $\mathbf{class}_i$ is used as a counter of the number of consecutive diverging steps of $\{(\mathbf{Az}^k)_i\}$; divergence is concluded when the count is at least 3. When neither convergence nor divergence is detected, $\mathbf{class}_i$ is set to $-1$.

Table 1. Numerical results for 3 examples.

| Example 1 | | Example 2 | | Example 3 | |
|---|---|---|---|---|---|
| $\mathbf{Az}^9$ | class | $\mathbf{Az}^6$ | class | $\mathbf{Az}^8$ | class |
| 5.08 | −1 | 0.20 | 0 | 1.37 | 1 |
| 5.15 | −1 | 0.20 | 0 | 1.37 | 1 |
| 5.15 | −1 | 0.20 | 0 | 1.37 | 1 |
| 5.15 | −1 | 0.20 | 0 | 1.37 | 1 |
| 5.15 | −1 | 0.20 | 0 | 1.37 | 1 |
| 5.23 | −1 | −0.50 | 0 | 0.68 | 1 |
| −0.15 | −1 | 6.00 | 5 | 819.56 | 7 |
| 5.23 | −1 | 0.20 | 0 | 1.37 | 1 |
| 5.15 | −1 | 0.20 | 0 | 1.37 | 1 |
| 5.15 | −1 | 0.20 | 0 | 1.37 | 1 |
| 5.15 | −1 | 0.20 | 0 | 1.37 | 1 |
| 5.15 | −1 | 0.20 | 0 | 1.37 | 1 |
| 5.08 | −1 | 6.00 | 5 | 819.56 | 7 |
| −0.15 | −1 | −0.50 | 0 | 0.67 | 1 |

## REFERENCES


1. G.W. Stewart, An iterative method for solving nonlinear inequalities, Research Report TR-1833, Department of Computer Science and Institute for Physical Science and Technology, University of Maryland, College Park, MD, (April 1987).
2. M.L. Brodzik and W.C. Rheinboldt, The computation of simplicial approximations of implicitly defined two-dimensional manifolds, *Computers Math. Applic.* **28** (9), 9–21 (1994).
3. B. Chazelle and D.P. Dobkin, Intersection of convex objects in two and three dimensions, *J. of the ACM* **34**, 1–27 (1987).
4. D.P. Dobkin and D.L. Souvaine, Detecting the intersection of convex objects in the plane, *Computer Aided Geom. Design* **8**, 181–199 (1991).
5. B. Roider and S. Stifter, Collision of convex objects, In *Proceedings of EUROCAL '87: European Conference on Computer Algebra*, (Edited by J.H. Davenport), Springer-Verlag, Berlin, (1989).
6. S. Stifter, A medley of solutions to the robot collision problem in two and three dimensions, Ph.D. Thesis, Univ. Linz, RISC-Linz Technical Rep. 88-12.0, (1988).
7. S. Stifter, A generalization of the Roider Method to solve the robot collision problem in 3D, In *Symbolic and Algebraic Computation: International Symposium ISAAC '88*, (Edited by P. Gianni), Springer-Verlag, Berlin, (1989).

8. H. Bieri and W. Nef, Elementary set operations with $d$-Dimensional Polyhedra, In *Computational Geometry with Applications, CG '88, International Workshop on Computational Geometry*, (Edited by H. Noltemeier), Springer-Verlag, Berlin, (1988).
9. J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Englewood Cliffs, NJ, (1983).