ALGORITHMS FOR HIGH DIMENSIONAL STABBING PROBLEMS

David Avis

Mike Doskas

School of Computer Science
McGill University
805 Sherbrooke St. W.
Montreal, Canada, H3A 2K6

ABSTRACT

Given a set of geometric objects in R^d, the hyperplane transversal or stabbing problem is to determine if there exists a hyperplane that simultaneously intersects all of the objects. We give algorithms based on linear programming for various hyperplane stabbing problems where the objects are line segments or convex polyhedra. A hyperplane stabber for n segments in R^d can be found in $O(n^d)$ time. In R^d, a plane stabber for a set of m polyhedra with a total of n edges can be found in $O(n^{d-1}m)$ time and space. Given a query hyperplane, a list of polyhedra intersected by the hyperplane can be determined in $O(\log n + m)$ time.

Keywords: Computational Geometry, Stabbing Problems, Transversals, Intersections, Polyhedra, Geometric Algorithms

1. Introduction

Let $\mathbf{S} = \{S_1, \ldots, S_m\}$ be a set of geometric objects in R^d. We say that the objects have a $k-$*transversal* if they are simultaneously intersected by a k-dimensional flat. The question of the existence of a k-transversal has received considerable attention in the mathematical literature, see for example [15] [4] [17] [18] [19] [24] [14] [12]. The corresponding computational issues have been considerably less studied. However, the computation of 1-transversals, or ‘stabbing lines’ as they are called in the computing literature, have appeared in the contexts of hidden line problems[8], set partitioning[2] and updating triangulations[13].

When $k = 0$, the problem reduces to that of deciding whether the given sets have a point in common. On the theoretical side, when the sets S_i are convex, the following theorem provides an answer to this question.

Helly's Theorem [15]. A family of n convex sets in R^d have a common intersection point if and only if every $d+1$ of the sets have a common intersection point.

On the computational side, when the sets are polyhedral, we can find an intersection point or determine that none exist by solving a linear program. This can be done efficiently by making use of the following result.

Megiddo-Dyer Theorem [21] [6]. In fixed dimension d, a linear programming problem with m constraints can be solved in $O(m)$ time.

When $k = 1$ we call the 1-transversal a *stabbing line* and the problems become considerably more difficult. On the theoretical side there is no analogue of Helly's theorem. There are examples in R^2 of sets of m line segments, each $m-1$ of which admit a stabbing line but for which the entire set does not admit a stabbing line[15]. Hadwiger has found an additional condition which gives a Helly-like theorem in the plane. By an *oval* we mean a bounded, closed convex set.

Hadwiger's Theorem [15]. A finite or countably infinite family of disjoint ovals in R^2 admits a stabbing line if and only if there is an ordering on the ovals such that each three ovals of the family are intersected in the given ordering.

As it stands, Hadwiger's theorem does not provide a 'good' characterization since if there is no stabbing line, the theorem does not provide an efficient way to demonstrate this, even for a finite family of sets. The theorem can however, be turned into a 'good' characterization. Consider our finite family of sets **S**. Any stabber for this family induces an ordering on the indices $1, \ldots, m$. Out of the *a priori* factorial number of stabbing orderings, Katchalski, Lewis and Zaks have shown that there can be at most $\binom{m}{2}$ such orderings for a given set of ovals[16]. When the ovals are polygons, these orderings can be efficiently computed and the condition of Hadwiger's theorem can be efficiently verified. This result has been improved by Wenger, who reduced the upper bound to $6m$[24], and further improved by Edelsbrunner and Sharir to the tight bound of $2m-2$[12]. In the case of polygonal objects, these potential orderings can be efficiently computed. As we will show in this paper, for each ordering, a linear program can determine if in fact a stabber exists which intersects the sets in the specified order.

On the computational side very little is known about stabbing lines. In the case of n line segments in the plane, Edelsbrunner et al.[8] have given an $O(n \log n)$ algorithm.

Edelsbrunner gives algorithms for stabbing translates of a convex polygon[7]. When moving to R^3 there are two interesting cases to consider: 1-transversals, or stabbing lines and 2-transversals, or stabbing planes. Somewhat surprisingly, these two cases seem to give rise to quite different types of algorithmic problems. Stabbing lines in R^3 are dealt with in a separate paper[3]. In this paper we study the problem of computing stabbing planes in space, and more generally we give an approach for finding $d-1$-transversals of line segments in R^d.

In dimension $d \geq 3$, no algorithms for hyperplane stabbing of polyhedra have previously been presented. In this paper, we develop a theory for plane stabbing of polyhedra in R^3 based on linear programming. We show how it can be used to compute a plane stabber of a set of m polyhedra with a total of n edges in $O(n^2 m)$ time. We then describe how to generalize these results to all dimensions.

2. Stabbing Segments

We begin by examining in detail the problem of stabbing segments in the plane by a line stabber. These techniques will then be generalized to higher dimensions and to polyhedra. Let S be a segment in R^2 specified by its endpoints (x_1, y_1), (x_2, y_2). For the moment we assume that S is not vertical, that is $x_1 \neq x_2$. We let

$$R_S = \{(u, b) : y = ux + b \quad \text{stabs} \quad S\}.$$

Then it is easy to see that we can write

$$R_S = \{(u, b) : y_1 - ux_1 \leq b \leq y_2 - ux_2\} \cup \{(u, b) : y_1 - ux_1 \geq b \geq y_2 - ux_2\}$$

We remark that R_S can also be thought of as the dual obtained by mapping points to lines and lines to points. For an approach along these lines, the reader is referred to chapter 15 of the book by Edelsbrunner[10]. The account given here is, however, self contained and can be read without any specific knowledge of dual transforms. For $i = 1, \ldots, n$ let S_i be a non-vertical segment with endpoints (x_1^i, y_1^i) and (x_2^i, y_2^i). Then there is a non-vertical stabbing line if and only if

$$\bigcap_{i=1}^{n} R_{S_i} \neq \emptyset.$$

Unfortunately the sets R_{S_i} are not convex, and so their intersection cannot be computed by a single linear program. The sets are in fact non-pointed polyhedral cones. Edelsbrunner et al. have given a sweep-line algorithm which computes a complete description of this intersection in time $O(n \log n)$[8].

The intersection can also be solved by a number of linear programs, by suitably partitioning the space of stabbing lines. Indeed let

$$c_i = \frac{y_2^i - y_1^i}{x_2^i - x_1^i}\,.$$

Then for any line $y = ux + b$ with $u \le c_i$, we can verify whether it is a stabber for S_i by checking the inequalities

$$y_1^i - ux_1^i \le b \le y_2^i - ux_2^i.$$

In case $u \ge c_i$, we simply test the same inequalities with directions reversed. Let us assume that the segments have been labeled so that

$$c_1 \le c_2 \le \cdots \le c_n.$$

Then to decided whether there is a stabbing line with $c_j \le u \le c_{j+1}$ it suffices to test the feasibility of the system of inequalities:

$$y_1^i - ux_1^i \le b \le y_2^i - ux_2^i, \qquad i = 1, \ldots, j$$

$$y_1^i - ux_1^i \ge b \ge y_2^i - ux_2^i, \qquad i = j+1, \ldots, n.$$

Therefore we can find a stabber by solving $n+1$ sets of inequalities of the type shown. Each such system can be solved as a linear program in two dimensions in linear time using the result of Megiddo and Dyer mentioned in the introduction.

It remains to examine the case of vertical lines and segments. To decide if there is a vertical stabbing line, it suffices to project all segments onto the x-axis and decide whether they overlap. This can be done simply in $O(n)$ time. The case of a vertical segment S causes no particular problems. We simply define c_S to be $+\infty$. We thus conclude that our algorithm takes $O(n^2)$ and is therefore slower than the $O(n \log n)$ algorithm given in[8]. However our approach generalizes easily to higher dimensions as we now show.

In R^d let segment S have endpoints $s_1, s_2 \in R^d$. For $a \in R^d$, $b \in R$, we denote by (a, b) the *hyperplane* $\{x \in R^d : ax = b\}$. In this notation, the line $y = ux + b$ in the plane is denoted $((-u, 1), b)$. As before, we let R_S denote the set of hyperplanes that stab S. Then

$$R_S = \{(a, b) \in R^{d+1} : as_1 \le b \le as_2\} \cup \{(a, b) \in R^{d+1} : as_1 \ge b \ge as_2\}.$$

As before R_S consists of a non-pointed convex polyhedral cone. If we partition the space of normals a by the hyperplane

$$a(s_1 - s_2) = 0$$

then R_S becomes a convex set in each part.

For $i = 1, \ldots, n$ let S_i be a segment with endpoints s_1^i, s_2^i. Consider the partition of R^d generated by the hyperplanes

$$a(s_1^i - s_2^i) = 0 \qquad i = 1, \ldots, n. \tag{1}$$

In each region of the partition, the existence of a stabbing hyperplane whose normal is contained in the region can be determined by the feasibility of a set of n inequalities. Indeed, each region is specified by a set of indices $I \subseteq \{1, \ldots, n\}$. The corresponding region is defined by

$$\begin{aligned} a(s_1^i - s_2^i) &\leq 0 \qquad i \in I \\ a(s_1^i - s_2^i) &\geq 0 \qquad i \notin I. \end{aligned} \tag{2}$$

To determine if there is any stabbing hyperplane (a, b) for a in the region defined by (2) it suffices to check the feasibility of the system

$$\begin{aligned} as_1^i \leq b \leq as_2^i \qquad i \in I \\ as_1^i \geq b \geq as_2^i \qquad i \notin I. \end{aligned}$$

This can be achieved by solving a linear program with $2n$ constraints and $d + 1$ variables.

A hyperplane (a, b) is *vertical* if the last coordinate of the normal vector a is zero. The procedure can be made more efficient by treating vertical stabbing hyperplanes separately, as was done in the plane for vertical stabbing lines. It is easily seen that there is a vertical stabbing hyperplane of segments in R^d if there is a stabbing hyperplane in R^{d-1} for the segments projected by making their last coordinates zero. This is the same problem in one dimension less. To find if there is a non vertical stabbing hyperplane, we may assume that the last coordinate of its normal vector is one by scaling if necessary. The arrangement associated with (2) by replacing inequalities by equalities now becomes an arrangement of n hyperplanes in R^{d-1}. This arrangement can generate at most $O(n^{d-1})$ regions[9]. It therefore suffices to solve $O(n^{d-1})$ linear programs of size $2n$ by d. Using the technique of Megiddo and Dyer mentioned before, this can be achieved in $O(n)$ time per program. Therefore a stabbing hyperplane for a set of segments in R^d can be computed in time $O(n^d)$.

3. Stabbers of Polygons and Polyhedra

In this section we develop algorithms for finding a stabbing line for a set of polygons and a stabbing plane for a set of polyhedra in space. We begin by remarking that a hyperplane intersects a polyhedron if and only if it intersects its convex hull. Therefore, given a set of arbitrary polyhedra, we may first preprocess by taking their convex hulls obtaining a set of convex polyhedra. Using an algorithm of Seidel[22] the convex hull of a set of n points in d dimensions can be found in time $O(n \log n + n^{\lfloor (d+1)/2 \rfloor})$. This will, in general, be dominated by the time required by our algorithms to compute hyperplane stabbers. Therefore, in what follows we assume that all polyhedra are convex. We assume that the total number of edges of the input is n. As in the previous section we begin with the planar case. The approach will be similar, we partition the space of stabbers into regions. In each region, the existence of a stabber can be determined by a linear program.

Let $P = \{p_1, \ldots, p_m\}$ be a polygon in the plane given by a list of its vertices in clockwise order. For $a \in R^2$ and $b \in R$, denote the line $ax = b$ in the plane by (a, b). Then (a, b) is a *directed support line* for P at a vertex p_i if p_i is on the line and all of the vertices of the polygon lie in the half plane $ax \le b$. In this case we say that the normal a points *away* from P. Two vertices p_i, p_j of P are an *antipodal pair* if there exist parallel lines (a, b_1), $(-a, b_2)$ such that the first line supports P at p_i and the second line supports P at p_j. For $i = 1, \ldots, m$, let (a_i, b_i) be the support line of P through the edge p_i, p_{i+1}, where $p_{m+1} = p_1$. Then a line (a, b) can only support P at p_i if

$$\lambda a_{i-1} + (1 - \lambda) a_i = a$$

for some non-negative λ. In other words, a is a *positive combination* of the normals of the support lines through the two edges of P with endpoints at p_i.

We now partition the space of normals according to which vertices of P they can generate a supporting line. For $i = 1, \ldots, n$, let

$$V^+(i) = \{a \in R^2 : (a, b) \text{ supports } p_i \text{ for some } b\}. \tag{3.1}$$

In view of the above discussion, it can be seen that the cells $V^+(i)$ form a cell decomposition Γ^+ of the plane. We form a second decomposition Γ^- as follows. For $i = 1, \ldots, n$ let

$$V^-(i) = \{a \in R^2 : (-a, b) \text{ supports } p_i \text{ for some } b\}. \tag{3.2}$$

It follows from the definitions that each cell in $\Gamma^+ \cap \Gamma^-$ corresponds to an antipodal pair for P. Indeed, a point a is contained in $V^+(i) \cap V^-(j)$ if and only if (a, ap_i) supports P at p_i and $(-a, -ap_j)$ supports P at p_j. Now if $a \in V^+(i)$ so is λa for any positive λ. The subdivision Γ^+ is therefore a collection of cones emanating from a common origin. Figure 3.1 (b) shows the subdivision Γ^+ corresponding to the polygon P shown in Figure 3.1 (a). The subdivision Γ^- is simply the mirror image formed by reflecting through the origin, as shown in Figure 3.1 (c). Superimposing the two decompositions gives a decomposition of the space of normals into antipodal pairs, as shown in Figure 3.1 (d). We will denote this decomposition Γ_P.

Let us now return to the problem of computing a stabbing line for a set of polygons $P_1, \cdots, P_m$. For each polygon we compute the decomposition Γ_{P_i}. We now superimpose all of these decompositions obtaining our final decomposition Γ^*. Consider a point a in the interior of any region in this decomposition. For each of the given polygons P_i, we now have an antipodal pair u_i, v_i that is supported by lines with normals a and $-a$. It is easy to see that if a line with normal a stabs the polygon P_i, then it must intersect the line segment $u_i v_i$ with endpoints u_i, v_i. Therefore it suffices to check, for all a in the given region of Γ^*, whether there is a stabber of the segments $u_i v_i$, $i = 1, \ldots, m$. Using the techniques of section 2 and the convexity property mentioned above, this can be done in a single linear program, using $O(m)$ time. If the set of polygons have a total of n edges, then there will be exactly n regions in Γ^*. This decomposition can readily be computed in $O(n \log n)$ time. A line stabber for m polygons with a total of n edges can therefore be computed in $O(nm + n \log n)$ time. As in the case of line segments, this is not optimal. An algorithm due to Atallah and Bajaj solves this problem in time $O(n \log n\alpha(n))$, where $\alpha(n)$ is the slow growing inverse of Ackermann's function[1]. However, as we shall see later, our approach generalizes to all dimensions.

The decomposition Γ^* can also be used to efficiently answer line stabbing queries of the form 'Which of the polygons $P_1, \ldots, P_m$ are stabbed by the line (a, b) ?' This query is handled by first locating which region of Γ^* contains the normal a, which can be done by binary search in $O(\log n)$ time. Then it is a simple matter to test which of the m segments corresponding to the m antipodal pairs the line (a, b) intersects. The query can therefore be performed in optimal time $O(\log n + m)$. Finally we note that the decomposition Γ^* is essentially one-dimensional. If we temporarily exclude vertical stabbing lines, then Γ^* can be intersected with any line parallel to, but not identical with, the a axis, producing the same regions. As in the previous sections, the case of vertical stabbing lines can be handled as a problem in one dimension.

Next we show how the above technique can be applied to stabbing polyhedra with planes in R^3. We now denote by (a, b), $a \in R^3$, $b \in R$ a plane in space. Let P be a polyhedron in space with vertices $p_1, \ldots, p_t$. We say that (a, b) is a *directed supporting plane* for P at p_i, if p_i lies on the plane and all of the vertices of the polyhedron lie in the halfspace $ax \le b$. Two vertices p_i, p_j of P are an *antipodal pair* if there exist parallel planes (a, b_1), $(-a, b_2)$ such that the first plane supports P at p_i and the second plane supports P at p_j. We define a decomposition of normal space into cells $V^+(i)$ and $V^-(i)$ analogously to equations (3.1) and (3.2). Similarly we define the cell decompositions Γ^+, Γ^-, Γ_P and Γ^*, for a set of polyhedra $P_1, \cdots, P_m$, which have a total of n edges. Each cell in Γ^* again corresponds to a set of m antipodal pairs, one for each of the polyhedra. A plane with normal in the given region stabs the set of polyhedra if and only if it stabs each of the segments whose endpoints are the given antipodal pairs. We now investigate the complexity of Γ^*.

Consider first a cell $V^+(i)$ corresponding to a vertex p_i of some polyhedron P. Let

$$P - p_i = \{x - p_i : x \in P\}.$$

That is, $P - p_i$ is the translate of P formed by shifting the origin to p_i. Let C_i be the smallest cone pointed at the origin that contains $P - p_i$. Let $d(i)$ be the number of faces of P incident at p_i. Assume these faces have normals $a_1, \ldots, a_{d(i)}$. Then we can write

$$C_i = \{x \in R^3 : a_i x \leq 0,\ i = 1, \ldots, d(i)\}.$$

It follows from Farkas Lemma[23] that $V^+(i)$ is the *dual* cone generated by $\{a_1, \ldots, a_{d(i)}\}$. In other words

$$V^+(i) = \{a \in R^3 : a = \sum_{i=1}^{d(i)} \lambda_i a_i,\ \lambda_i \geq 0\}.$$

It follows that $V^+(i)$ has $d(i)$ faces, each supported by a plane through the origin. By the properties of duality, each such plane is in fact normal to an edge in P adjacent to p_i[23]. Each vertex p_j of P adjacent to p_i contributes one such plane with normal $p_j - p_i$. Therefore, if P has t edges, Γ_P is the subdivision of R^3 formed by t such planes through the origin.

Let us now consider the subdivision Γ^*. Since our collection of polyhedra have n edges, Γ^* is formed by a set of n planes through the origin. By a result of Winder[25], these planes partition R^3 into at most $9n^2 + 3n$ regions. As we saw in the planar case, by excluding vertical stabbing planes, the subdivision Γ^* can be reduced to an arrangement of at most n lines in the plane. Using the techniques of Edelsbrunner, O'Rourke and Seidel[9], this arrangement can be constructed in $O(n^2)$ time. For each cell, a single linear program determines the existence of a stabbing plane in $O(m)$ time. Therefore we can determine the existence of a stabbing plane in $O(n^2 m)$ time and space. Finally, by using the point location technique of Kirkpatrick[20], or other standard point location techniques, we can determine all of the polyhedra stabbed by a given plane in $O(\log n + m)$ time, after the above preprocessing.

Finally, we discuss the situation in higher dimensions. The approach developed above in R^3 is applicable in any dimension. In order to determine the existence of a stabbing hyperplane, we first construct Γ^*. Suppose the family of polyhedra has n edges in total. Then Γ^* is a subdivision of R^d induced by n hyperplanes through the origin. Excluding vertical hyperplanes as before, it suffices to construct the arrangement of n

hyperplanes in R^{d-1}. By the previously mentioned result[9], this can be achieved in $O(n^{d-1})$ time. For each of the cells produced, a linear program is solved to check for the existence of a stabbing hyperplane. For the query problem, it is required to quickly decide which of the cells contains the normal of the query hyperplane. Unfortunately, the known efficient planar point location techniques do not readily generalize to high dimensions. However, given sufficient preprocessing time and space, the query query problem in dimension $d \geq 4$ can still be solved in $O(\log n + m)$ time by using the techniques of Dobkin and Lipton[5].

Since this paper was written, Edelsbrunner Guibas and Sharir[11] have found an improved result for plane stabbing in R^3. Using results on computing the upper envelope of a set of planes, they can find a plane stabber of m polyhedra in time $O(n^2\alpha(n))$, where $\alpha(n)$ is the inverse of Ackermann's function. For all but very small m, this improves on the result of $O(n^2m)$ given here. For the case of n segments in R^3, they can find a plane stabber in time $O(n^2)$, improving on the time of $O(n^3)$ given in section 2.

4. Acknowledgements

The authors would like to thank an anonymous referee for many helpful suggestions.

References

1. M. Atallah and C. Bajaj, "Efficient Algorithms for Common Transversals," *Information Proc. Letters*, vol. 25, pp. 87-91, 1987.
2. D. Avis, "Diameter Partitioning," *Discrete and Computational Geometry*, vol. 1, pp. 265-276, 1986. ftp://mutt.cs.mcgill.ca/pub/doc/Av86a.ps.gz
3. D. Avis and R. Wenger, "Polyhedral Line Transversals in Space," *Discrete and Computational Geometry*, vol. 3, pp. 257-265, 1988.
4. L. Danzer, B. Grünbaum, and V. Klee, "Helly's Theorem and its Relatives," *Proceedings of Symposia in Pure Mathematics*, pp. 100-181, Am. Math. Soc., 1963.
5. D. Dobkin and R. Lipton, "Multidimensional Search Problems," *Siam J. Comp.*, vol. 5, pp. 181-186, 1986.
6. M.E. Dyer, "Linear Programs for two- and three- Variable Linear Programs," *SIAM J. Comp.*, vol. 13, pp. 31-45, 1984.
7. H. Edelsbrunner, "Finding Transversals for Sets of Simple Geometric Figures," *Theoretical Computer Science*, vol. 35, pp. 55-69.

8. H. Edelsbrunner, H. A. Maurer, F. P. Preparata, A.L. Rosenberg, E. Welzl, and D. Wood, ‘‘Stabbing Line Segments,’’ *BIT*, vol. 22, pp. 274-281, 1982.

9. H. Edelsbrunner, J. O’Rourke, and R. Seidel, ‘‘Constructing Arrangements of Lines and Hyperplanes with Applications,’’ *SIAM J. Computer Science*, pp. 341-363, 1986.

10. H. Edelsbrunner, *Algorithms in Combinatorial Geometry,* Springer-Verlag, 1987.

11. H. Edelsbrunner, L. Guibas, and M. Sharir, ‘‘The Upper Envelope of Piecewise Linear Functions: Algorithms and Applications,’’ *Technical Report UIUCDCS-R-87-1390*, Universtiy of Illinois at Urbana-Champaign, November 1987.

12. H. Edelsbrunner and M. Sharir, ‘‘The Maximum Number of Ways to Stab n Convex Non-Intersecting Objects in the Plane is 2n-2,’’ *Discrete and Computational Geometry*, 1988(to appear).

13. H. ElGindy and G. T. Toussaint, ‘‘Efficient Algorithms for Inserting and Deleting Edges from Triangulations,’’ in *Proc. Int. Conference on Foundations of Data Organization*, Kyoto, May 22-24, 1985.

14. J. E. Goodman and R. Pollack, ‘‘Hadwiger’s Transversal Theorem in Higher Dimensions,’’ *Journal of Am. Math. Soc.*, vol. 1, to appear.

15. H. Hadwiger, H. Debrunner, and V. Klee, *Combinatorial Geometry in the Plane,* Holt, Rinehart and Winston, 1964.

16. M. Katchalski, T. Lewis, and J. Zaks, ‘‘Geometric Permutations for Convex Sets,’’ *Discrete Mathematics*, vol. 54, pp. 271-284.

17. M. Katchalski, ‘‘Thin Sets and Common Transversals,’’ *J. of Geometry*, vol. 14, pp. 103-107, 1980.

18. M. Katchalski, T. Lewis, and A. Liu, ‘‘Geometric Permutations and Common Transversals,’’ *Discrete and Computational Geometry*, vol. 1, pp. 371-377, 1986.

19. M. Katchalski, ‘‘On a Conjecture of Grünbaum on Common Transversals,’’ *Mathematica Scandinavica*, vol. 59, pp. 192-198, 1986.

20. D. Kirkpatrick, ‘‘Optimal Search in Planar Subdivisions,’’ *SIAM J. Computing*, vol. 12, pp. 28-35, 1983.

21. N. Megiddo, ‘‘Linear Programming in Linear Time When the Dimension is Fixed,’’ *J. ACM*, vol. 31, pp. 114-127, 1984.

22. R. Seidel, ‘‘A Convex Hull Algorithm Optimal for Point Sets in Even Dimensions,’’ *Report 81-14*, University of British Columbia, Dept. of Computer Science, 1981.

23. J. Stoer and C. Witzgall, *Convexity and Optimization in Finite Dimensions I,* Springer-Verlag, 1970.

24. R. Wenger, ''Upper Bounds on Geometric Permutations,'' *Discrete and Computational Geometry*, to appear.

25. R. O. Winder, ''Partitions of N Space by Hyperplanes,'' *SIAM Journal of Applied Math.*, vol. 14, pp. 811-818, 1966.