THE COMPLEXITY OF THE SIMPLEX ALGORITHM

by

James D. Currie

A thesis submitted to the Faculty
of Graduate Studies and Research in
partial fulfillment of the requirements
for the degree of Master of Science

Department of Mathematics and Statistics
Carleton University
Ottawa, Ontario

August 1984

The Complexity of the Simplex Algorithm

James D. Currie, M.Sc. Thesis

Abstract

The thesis begins by giving background in linear programming and Simplex methods. Topics covered include the duality theorem, Lemke's algorithm, and the pathological linear programs of Klee-Minty.

Because of the bad behaviour of Klee-Minty programs, the behaviour of the Simplex algorithm is only good on average. To take such an average, certain assumptions on the distribution of linear programs are introduced and discussed.

A geometrical meaning is given for the number of steps Lemke's algorithm takes to solve a program. This gives rise to a formula bounding the average number of steps taken. The formula is heuristically justified in an original way.

The formula is combinatorially simplified, to get a bound on the complexity of Simplex.

TABLE OF CONTENTS

Preface

The topic of this thesis is "Why is Simplex so fast?" About half of this paper is devoted to expounding the work of Steve Smale [6,7]. The first half of the paper leads up to Smale's analysis by giving background in linear programming and Simplex methods. The paper should be self-contained. Here we will give a section breakdown.

Section 1: What is linear programming?

In this section, we define and motivate linear programming. Important concepts are defined. The section culminates in the duality theorem.

Section 2: What are Simplex methods?

Here we explain the general scheme of all Simplex methods.

Section 3: Lemke's algorithm.

The Simplex algorithm Smale analyzes is Lemke's algorithm. This algorithm is explained, and proved correct.

Section 4: Klee-Minty Examples.

Following Čvatal, we give the Klee-Minty examples of linear programs which take many steps to solve.

Section 5: Smale's Geometric Work.

Smale's analysis of Simplex breaks into two sections. In this section we explain how Smale developed a geometric meaning for the number of steps the Simplex algorithm takes.

Section 6: Smale's Combinatorial Work.

Smale works with his geometric construct combinatorially, and shows that Simplex will usually be fast.

Section 7: Summary.

Here we evaluate Smale's approach.

In expounding Smale's work, we have mostly followed [7], which is Smale's improvement and precis of [6]. We have filled in the gaps in [7], sometimes by modifying proofs of [6]. Unfortunately, [6] is very difficult to read because of typographical errors. Another problem which carries over to [7] is an apparent confusion between q_0 and another vector.

Because of this confusion, the geometric derivations in [6] and [7] are simply incorrect as they stand. Our contribution has been to give correct proofs for the assertions of [7], and especially to motivate the intricacies of both of Smale's papers. The way in which dominance can be used for motivation is our own discovery, spurred by Blair [1].

Thanks to Dr. Mortimer for his patience.

Also thanks to Dr. Cunningham for bringing [1] to my attention.

Linear Programming

A practical motivation can be given for linear programming. For example, consider the following problem.

Construct a diet using milk, bread and liver, subject to the condition that the minimum daily requirements of vitamins Q, R, S are provided by the diet. Construct the diet in such a way that cost is minimized.

Units of in	Q	R	S
liver	10	20	6
milk	2	15	3
bread	1	1	4
requirement	12	8	4

If we let the amounts of milk, bread and liver in our diet be x_1, x_2, x_3 respectively, with costs c_1, c_2, c_3, then the problem becomes

$$\text{minimize} \quad c_1x_1 + c_2x_2 + c_3x_3$$

$$\text{subject to} \quad 10x_1 + 2x_2 + x_3 \geq 12$$

$$20x_1 + 15x_2 + x_3 \geq 8$$

$$6x_1 + 3x_2 + 4x_3 \geq 4$$

$$\text{and} \quad x_1, x_2, x_3 \geq 0 .$$

This example motivates the <u>standard</u> form of a linear program (LP).

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0 \ ; \quad x \in R^n, \ b \in R^m, \ A \in R^{m \times n} . \end{aligned} \tag{1}$$

Geometry

A geometric interpretation of the problem may be helpful. Let $A^{(i)}$ be the ith row of the matrix A. In R^n, the solution set of $A^{(i)}x = b_i$ is a hyper-plane of dimension $n-1$. Thus $A^{(i)}x \geq b_i$ describes a half-space "above" that hyperplane. If in addition, $x \geq 0$, then x lies in a (possibly unbounded) polyhedron (see Fig. 1).

When all the row conditions are true and $x \geq 0$, then x lies in a polyhedron arising from the intersection of m half-spaces and the orthant $x \geq 0$. As $c^T x$ is a linear function, it is intuitively clear that any extrema will be at the "corners" of this polytope.

We take a very simple example:

$$\begin{aligned} \min \quad & x_1 + 2x_2 - 3x_3 \\ \text{s.t.} \quad & -x_1 \geq -1 \\ & -x_2 \geq -1 \\ & -x_3 \geq -1 \\ & x_1, x_2, x_3 \geq 0 . \end{aligned}$$

In 3-space, $x_1 = 1$ is a plane. In Figure 2, $-x_1 \geq -1$ would be the half-space to the left of this plane. When all 6 conditions of this LP are true, (x_1, x_2, x_3) must be inside a cube, as in Figure 3 .

The cost function we are trying to minimize is $x_1 + 2x_2 - 3x_3$. If we let $\vec{N}$ be a normal to the plane $x_1 + 2x_2 - 3x_3 = 0$, and call the direction of $\vec{N}$ "up", then solving the LP is finding the "highest" point of the cube.

In general, solving an LP requires finding the highest point in a certain n-dimensional polyhedron. Clearly, in our example, the highest point will be on the outside of the cube. However, without loss of generality, the highest point on a face will be at an edge, the highest point of an edge at a vertex. Therefore, solving LP is equivalent to the <u>finite</u> problem of finding which vertex of a polyhedron is highest.

Next, we introduce a notation to help distinguish vertices algebraically. In our example, each face of the cube corresponds to the points where a constraint holds with equality. Given any LP in standard form,

$$\min c^T x$$

$$Ax \geq b$$

$$x \geq 0$$

we introduce "slack" variables,

$$x_{n+i} = A^{(i)}x - b_i \text{ , for } i = 1 \text{ to } m.$$

In our example, we must introduce the variables

$$x_4 = -x_1 + 1$$

$$x_5 = -x_2 + 1$$

$$x_6 = -x_3 + 1 \quad .$$

When we are on the face where $x_1 = 1$, then $x_4 = 0$. In fact we have constructed the new variables so that whenever we are on the plane corresponding to constraint i, $x_{n+i} = 0$. For our cube, any time 3 variables are 0, we are at a vertex. In n dimensions, a vertex can be fixed by n faces intersecting there. Before continuing to the Simplex method, we give some algebraic background to what we have said in this section.

Solving a Linear Program

Definition: The vector $x \in R^n$ is a solution for (1) if $Ax \geq b$.

Definition: A solution x of (1) is feasible if $x \geq 0$.

Definition: A solution x of (1) is optimal if and only if it is feasible, and $c^T y \geq c^T x$ for all feasible y.

As we mentioned in our geometric interpretation of the linear programming problem, it is intuitive that the optimal solution of any LP should be in a corner of the corresponding polyhedron. In each

corner, several x_i are zero. Let this motivate the following discussion:

Augment $x \in R^n$ to a vector $\hat{x} \in R^{m+n}$ by adding slack variables as in the previous section. Augment A to the matrix $\hat{A} = [A \mid I_{m \times m}]$. The constraints of (1) are equivalent to

$$\hat{A}\hat{x} = [A \mid I_{m \times m}]\, \hat{x} = b \tag{2}$$

$$\hat{x} \geq 0 \tag{3}$$

$\hat{A}$ has rank m. Let B be a column basis for $\hat{A}$. Then one solution of $\hat{A}\hat{x} = b$ is given by setting $\hat{x}_i = 0$ if $\hat{A}^{(i)} \notin B$, and solving the system $\sum_{\hat{A}^{(k)} \in B} \hat{A}^{(k)} \hat{x}_k = b$. ($\hat{A}^{(k)}$ is the kth column of $\hat{A}$.)

Definition: A solution $\hat{x}$ of (2) corresponding to a column basis B of $\hat{A}$ is called a basic solution of (1).

Definition: A solution of (3) which is a basic solution of (1) is called a basic feasible solution of (1).

We remark that a basic feasible solution (b.f.s.) of (1) gives a feasible solution of (1).

Our intuition that $c^T x$ is minimized in a corner becomes a theorem:

Theorem: 1) If there is any feasible solution to (1), then there is a basic feasible solution;

2) If there is any optimal solution of (1) then there is a basic optimal solution.

Proof: 1) Let $\hat{x}$ correspond to a feasible solution of (1) with as many of the $\hat{x}_i$ zero as possible. If $\hat{x} = 0$, then $b = \hat{A}\hat{x} = 0$, and $\hat{x}$ corresponds to any basis B.

Otherwise, let the non-zero components of $\hat{x}$ be $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_k$, re-ordering columns if necessary. We will show that the columns $\hat{A}^{(1)}, \hat{A}^{(2)}, \ldots, \hat{A}^{(k)}$ are independent, and may thus be augmented to a basis. It will follow that $\hat{x}$ is basic.

Suppose these $\hat{A}^{(i)}$ are linearly dependent. Then there are $t_i \in R$, not all zero, such that

$$t_1\hat{A}^{(1)} + t_2\hat{A}^{(2)} + \ldots + t_k\hat{A}^{(k)} = 0 .$$

Let $t = (t_1, t_2, \ldots, t_k, 0, 0, 0, \ldots, 0)^T$. Then $A(\hat{x} - rt) = b$ for all $r \in R$. Then let $r = \min(\hat{x}_i / t_i \mid t_i \neq 0)$.

We get $\hat{x}_i - rt_i \geq 0$ for $i = 1$ to $m+n$, by our choice of r. But if $r = \hat{x}_j / t_j$, then $\hat{x}_j - rt_j = 0$, and the feasible solution $y = \hat{x} - rt$ has more zeroes than $\hat{x}$. This is a contradiction, and our result follows.

2) Let $\hat{x}$ give an optimal solution and require $\hat{x}$ to have the most zeroes possible.

Augment c to $\hat{c} \in R^{n+m}$ by adding m zeroes.

If $\hat{x} = 0$, then $\hat{x}$ is basic. Otherwise let the non-zero components of $\hat{x}$ be $\hat{x}_1, \ldots, \hat{x}_k$. Again, assume the corresponding

columns are dependent, and pick t as before.

$$\hat{A}(\hat{x} - rt) = b \quad \text{for all} \quad r \in \mathbb{R} \quad . \tag{4}$$

For small enough r, the solution corresponding to (4) will be feasible, and the new cost is

$$\hat{c}^T(\hat{x} - rt) = \hat{c}^T x - r\hat{c}^T t \quad .$$

However, $\hat{c}^T x$ is the optimal cost, so we need $\hat{c}^T t = 0$. (Hint: If $\hat{c}^T t > 0$ then pick small positive r).

Choosing r, y as before, we get a feasible solution y, with $\hat{c}^T y = \hat{c}^T x - r\hat{c}^T t = \hat{c}^T x$, and y is a basic optimal solution. □

By the above theorem, we can restrict our attention to basic feasible solutions. As $\hat{A}$ has at most $\binom{n+m}{m}$ column bases, we have moved from a continuous problem to a finite discrete problem.

Duality Theory

The standard linear program has as data an $m \times n$ matrix A, and two vectors, $c \in R^n$, $b \in R^m$. We then have the program

$$\begin{aligned} &\min c^T x \\ &Ax \geq b \\ &x \geq 0 \quad . \end{aligned} \tag{1}$$

However, in the interest of symmetry, we might ask, "Why not do it this way?"

$$\begin{aligned} &\max\ y^T b \\ &y^T A \le c^T \\ &y \ge 0 . \end{aligned} \tag{5}$$

The two problems are related in an interesting and important way.

Suppose x is a feasible solution of (1) and y is a feasible solution of (5). Then

$$Ax \ge b$$

$$\Rightarrow \quad y^T Ax \ge y^T b \tag{6}$$

also $\quad y^T A \le c^T$

$$\Rightarrow \quad y^T Ax \le c^T x \tag{7}$$

whence $\quad y^T b \le c^T x \ . \tag{8}$

We introduce some terminology:

Given a program of the form (1), the program (5) is called its <u>dual</u>. Alternatively, (1) is the <u>primal</u> of (5). The dual can be written as the standard LP

$$\begin{aligned} &\min\ (-b)^T y \\ &(-A^T) y \ge -c \\ &y \ge 0 . \end{aligned}$$

One sees that the dual of the dual is equivalent to the primal. Inequality (8) says, in words, that the cost of any solution of the

dual gives a lower bound on the cost of any solution for the primal. This yields immediately a sufficient condition for the optimality of a solution of (2).

Sufficient Condition for Optimality: Suppose x_0 is a feasible solution for (1), and y_0 is a feasible solution for (5), with

$$y_0^T b = c^T x_0 .$$

Then x_0 is an optimal solution of (1), and y_0 is an optimal solution of (5). □

The central result of duality theory for linear programs is the converse of this result.

Duality Theorem: If (1) and (5) have feasible solutions, then they have feasible solutions x_0, y_0 such that

$$c^T x_0 = y_0^T b .$$

This theorem is generally proved either by using the machinery of Dantzig's Simplex algorithm, or by using the following lemma.

Lemma: [Farkas] Let C be an $M \times N$ matrix, and let $z \in R^M$. Then either

(i) $Cp = z$ has a solution $p \in R^N$, $p \geq 0$, or (exclusive)

(ii) $q^T C \geq 0$, $q^T z < 0$ has a solution $q \in R^M$.

In geometrical terms, the set $\{x \in R^M \mid x = Cp \text{ for some } p \geq 0\}$ is the convex cone of the columns of C. The lemma then says that z

either lies in the cone of C, or else there is $q \in R^M$ such that q has non-negative projection on each column of C, but negative projection of z. See Figure 4 for an example in R^2. The theorem follows readily from basic theorems on convex sets and separating planes, a detailed development of which would be out of place here. An accessible treatment is found in [4] (p.44 ff.).

Proof of Duality Theorem: Consider the $(m+n+1) \times 2(m+n)$ matrix

$$C = \begin{bmatrix} I_{m+n} & -A & 0 \\ & 0 & A^T \\ 0 & -c^T & b^T \end{bmatrix}$$

together with the vector $z = (-b, c, 0)^T$.

If $Cp = z$ has a solution $p \geq 0$, then write $p = (u, v, x, y)$; $u, y \in R^m$; $v, x \in R^m$ so that

$$u - Ax = -b$$

$$v + A^T y = c$$

$$c^T x - b^T y = 0$$

and $$Ax = u + b \geq b$$

$$y^T A = (A^T y)^T = (c - v)^T \leq c^T$$

$$cx = b^T y = y^T b$$

so that our dual pair (1), (5) has a feasible solution x, y with $y^T b = c^T x$.

Otherwise we have the Farkas alternative.

There is a vector $q \in R^{m+n+1}$ with

$$q^T C \geq 0$$

$$q^T z < 0 \ .$$

Writing q as $(y, x, t)^T$ yields

$$-y^T A \geq -tc^T$$

$$x^T A^T \geq tb^T$$

$$x, y, t \geq 0$$

and $$-by + cx < 0 \ . \tag{10a}$$

Therefore

$$y^T A \leq tc^T \tag{9}$$

$$Ax \geq tb$$

$$(tc^T x) \leq (tb^T y) \ . \tag{10}$$

Case 1) If $t > 0$, then (8) applied to (9) becomes $tc^T x \geq ty^T b$, so that $tcx = tby$, and $t^{-1}x$, $t^{-1}y$ are feasible for (1), (5), so that our conclusion follows.

Case 2) If $t = 0$, then let x_0, y_0 be feasible solutions for (1), (5). Then (9), (10a) yield

$$y^T A \leq 0$$

$$Ax \geq 0$$

$$c^T x < b^T y \ .$$

But then $c^T x \geq (y_0^T A)x \geq 0 \geq (y^T A)x_0 \geq y^T b$. This contradiction shows that this case never occurs. □

Simplex Methods and Degeneracy

Recall our geometric exposition of the linear programming problem. We wish to examine vertices of a polyhedron. Each vertex is specified by the intersection of n linearly independent planes, and hence is given by <u>at least one</u> basic feasible solution. Obviously a point in R^n is fixed by its coordinates; by the words "at least one", we refer to the fact that the b.f.s. were labelled by <u>bases</u>. It is possible that two choices of bases give the same point as a basic feasible solution. For example,

$$\begin{aligned} x_1 + x_2 &= 0 \\ x_3 &= 0 \\ x_4 &= 0 \end{aligned}$$

has two column bases, $\{A^{(1)}, A^{(3)}, A^{(4)}\}$ and $\{A^{(2)}, A^{(3)}, A^{(4)}\}$, both giving as b.f.s. $(0, 0, 0, 0)$.

Having given this note of caution, let us turn to describing the Simplex methods.

In a Simplex method we attempt to examine vertices of our polyhedron by always moving from one vertex to another adjacent one, via some edge. This requires moving from one b.f.s. to another differing in exactly

one column of the basis. Usually we try to move to an adjacent vertex such that the cost never decreases. The implementation follows a scheme of the following sort:

1) Find a b.f.s.

2) If the cost is optimal stop.

3) Otherwise pick some basic column to leave the basis, and some non-basic column to enter the basis, and repeat 2).

This scheme does not always work. We have seen that looking either at all vertices, or looking at all b.f.s. of our LP guarantees that we will find the optimal vertex/b.f.s. However, there is nothing in this scheme to insure we visit every vertex. If one b.f.s. is represented by more than one basis, it is conceivable that we might spend all our time looking at that b.f.s., our changes of bases only exchanging one representation of the point for another. To clarify this idea, let us look at Figure 5 .

Figure 5 represents a square pyramid in 3 dimensions. Because 4 planes intersect at the top of the pyramid, there are 4 ways to address this top point as an intersection of 3 planes. Conceivably a Simplex algorithm might look at this top point many times under different aliases.

A polyhedron in n dimensions with more than n faces intersecting at a point is <u>degenerate</u>. An LP is degenerate if some b.f.s. has more than n of its variables equal to zero.

At best, degeneracy in an LP is an embarassment. The Simplex methods may look at one point many times. At worst, a phenomenon called "cycling" may occur: a Simplex algorithm may stick at one point, only exchanging one representation of the point for another.

In practical examples, degeneracy comes up all the time, as problems are often posed containing redundant information.

The ambiguities of degeneracy can be avoided by perturbation methods. Each of our b_i is adjusted by a small constant Σ_i, with

$$\Sigma_1 >> \Sigma_2 >> \Sigma_3 >> \ldots >> \Sigma_m, \quad \text{and}$$

each of these numbers much smaller than the numbers in a given LP. The effect of such a method on our pyramid of Figure 5 may be seen in Figure 6. Perturbation methods may be shown to work in general: We can always perturb an LP to remove degeneracy. This is not, however, desirable in general, because perturbation costs $0(m)$ additions. In practice, programmers will perturb only when an LP seems to be cycling. We will have to return to these remarks on the use of perturbation when we evaluate Smale's paper in our concluding remarks.

An Example of A Simplex Method

We will now give an example of a Simplex method: the method of least cost coefficient.

We begin at a b.f.s. of (1), and on each iteration, we add to the basis the non-basic column which is least expensive. We remove a column

from the basis in such a way that we add as much of the "entering" variable as possible.

Example 1 $\quad \min \; -x_1 - 2x_2 + 3x_3$

$$\begin{aligned} \text{s.t.} \quad -x_1 \qquad\qquad &\geq -1 \\ -x_2 \qquad &\geq -1 \\ -x_3 \quad &\geq -1 \\ x_1, x_2, x_3 \quad &\geq 0 \; . \end{aligned}$$

This is our cube. The progress of the algorithm can be seen in Figures 7-9.

First we introduce slack variables

$$\begin{aligned} -x_1 \qquad + x_4 \qquad\qquad &= -1 \\ -x_2 \qquad\quad - x_5 \qquad &= -1 \\ -x_3 \qquad\quad - x_6 \quad &= -1 \; . \end{aligned}$$

We are lucky, in this special case, to have the b.f.s. $x_1 = x_2 = x_3 = 0$ to start from: x_4, x_5, x_6 are a basis, and solving gives

$$\begin{aligned} x_4 &= 1 - x_1 \\ x_5 &= 1 - x_2 \\ x_6 &= 1 - x_3 \\ \text{cost} &= -x_1 - 2x_2 + 3x_3 \; . \end{aligned}$$

We will now add one of our non-basic variables to the basis. The variable x_2 has least cost efficient, and letting x_5 become zero allows us to increase x_2 to 1.

$$x_1 = x_5 = x_3 = 0$$

$$x_4 = 1 - x_1$$

$$x_2 = 1 - x_5$$

$$x_6 = 1 - x_3$$

$$\text{cost} = -2 - x_1 + 3x_3 + x_5$$

Now we increase x_1. This forces x_4 to leave the basis, as we wish to increase x_1 as much as possible.

$$x_4 = x_5 = x_3 = 0$$

$$x_1 = 1 - x_4$$

$$x_2 = 1 - x_5$$

$$x_6 = 1 - x_3$$

$$\text{cost} = -3 + x_4 + x_5 + 3x_3 .$$

We can no longer find a variable with cost coefficient negative, but this matters not. -3 is the optimal cost, as $x_4, x_5, x_3 \geq 0$.

Example 2 $\quad \min \; -x_1 + 2x_2 - 3x_3 + 4x_4$

$$-x_1 - x_2 - x_3 - x_4 \geq -1$$

$$-2x_1 - 2x_3 \geq -2$$

$$x_1 - x_2 - x_3 - x_4 \geq -1 .$$

We introduce slack variables

$$x_5 = 1 - x_1 - x_2 - x_3 - x_4$$

$$x_6 = 2 - 2x_1 - 2x_3$$

$$x_7 = 1 + x_1 - x_2 - x_3 - x_4 .$$

Start at the vertex

$$x_1 = x_2 = x_3 = x_4 = 0$$

$$\text{cost} = -x_1 + 2x_2 - 3x_3 + 4x_4 .$$

We increase x_3. x_3 can become 1 before x_5 becomes 0, so let our basis be x_3, x_6, x_7. Then

$$x_1 = x_2 = x_5 = x_4 = 0$$

and
$$x_3 = 1 - x_1 - x_2 - x_4 - x_5$$

$$x_6 = 0 + 2x_2 + 2x_4 + 2x_5$$

$$x_7 = 0 + 2x_1 - x_5$$

$$\text{cost} = 3 + 2x_1 + 5x_2 + 7x_4 + 3x_5$$

Our program is degenerate, yet nonetheless, we arrive at the optimal cost in only one iteration (!).

Early in the history of the Simplex algorithm it became apparent that the algorithm is usually very fast. Empirical data suggests that the algorithm is linear in m, and only logarithmic in n, even for degenerate programs. Why this should be so is what Smale has tried to show.

Lemke's Algorithm

We want to solve

$$\min c^T x$$

$$Ax \geq b \tag{1}$$

$$x \geq 0; \quad x \in R^n, \quad c \in R^n, \quad b \in R^m, \quad A \in R^{m \times n}$$

or equivalently

$$\max y^T b$$

$$y^T A \leq c^T \tag{5}$$

$$y \geq 0 \quad .$$

By the duality theorem, we know that if we have optimal solutions, then $c^T x = y^T b$. Let $u = Ax - b$, $v^T = c^T - y^T A$. Then $y^T u = y^T Ax - y^T$ and $v^T x = c^T x - y^T Ax$. If x and y are optimal, then we have

$$y^T u + v^T x = y^T b - c^T x = 0 \ .$$

This allows us to give an equivalent problem to (1).

Find $z, \omega \in R^{m+n}$ such that

$$\omega = \begin{bmatrix} c \\ -b \end{bmatrix} + \begin{bmatrix} 0 & -A^T \\ A & 0 \end{bmatrix} z \tag{11}$$

and $\omega, z \geq 0$

and $\omega^T z = 0$. (12)

Here we let $\omega = (v^T, u^T)^T$ and $z = (x^T, y^T)^T$ to show the equivalence.

With general $q \in R^{m+n}$, and $M \in R^{(m+n)\times(m+n)}$, the problem

Find ω, z

s.t. $\omega = q + Mz$

$\omega, z \geq 0$, $\omega^T z = 0$

is called the linear complementarity problem.

Lemke's algorithm attacks an LP by putting it in the form of this new problem. Let $N = m+n$. Note that since $\omega, z \geq 0$ and $\omega^T z = 0$, then for each $i \in \{1, \ldots, N\}$, $\omega_i = 0$ and/or $z_i = 0$.

Definition: If $\omega, z \in R^N$ for some N and $\omega^T z = 0$, we say that ω and z are complementary vectors in R^N.

Suppose the LP (1) is not degenerate. In particular, every b.f.s. of (1) has exactly n variables equal to zero, and we can solve for the basic variables in terms of the others. (In fact, in solving $Ax = b$ in linear algebra, we do just this, writing the non-basic variables as parameters.) Then if we increase a non-basic variable starting at a b.f.s., this induces a linear change in each of the basic variables. We introduce two cases.

(i) We increase the non-basic variable x_i until some basic variable x_j is forced to become zero.

(ii) We can increase a non-basic variable indefinitely without causing any variable to become zero. This produces an unbounded edge or ray of our polyhedron.

From the proof of part 1 of the theorem on b.f.s., we know that the basic solutions are characterized by the number of zeroes in such solutions. Then in a non-degenerate LP, each b.f.s. has exactly n zeroes, and this characterizes the set of b.f.s. Then in situation (1) above, the feasible solution resulting when we increase a non-basic variable from zero until a basic variable becomes zero, is a basic feasible solution.

This is because it has at least n zeroes. By non-degeneracy, it will then have exactly n zeroes.

Suppose the LP (1) and its dual (5) are both non-degenerate. Lemke's algorithm finds $w, z \in R^N$

$$\text{s.t. } w = q + Mz$$
$$w^T z = 0; \quad w, z \geq 0 \tag{13}$$

with $q = \begin{bmatrix} c \\ -b \end{bmatrix}$, $M = \begin{bmatrix} 0 & -A^T \\ A & 0 \end{bmatrix}$.

It finds w and z by looking at a sequence of 2-tuples in R^N, $w^1, z^1; \; w^2, z^2, \ldots; \; w^t, z^t$ where each w^s, z^s is a partial solution of (13).

Definition: A pair $w, z \in R^N$ is an almost complementary solution of (13) if

$$w = q + Mz$$

$w, z \geq 0$ and $w_i z_i = 0$ except for at most one coefficient i.

In Lemke's algorithm, we will begin each iteration at some w, z pair where $w = (v^T, u^T)^T$, $z = (x^T, y^T)^T$ correspond to the slack variables, ordinary variables of some b.f.s. $\hat{x}$ of (1) and some b.f.s. $\hat{y}$ of (5). Further, we will have w, z an almost-complementary (a-c) pair.

If w, z are also complementary, the algorithm will terminate, for we will have solved (13).

Let $\hat{x}$ be a b.f.s. of (1), and $\hat{y}$ be a b.f.s. of (5). Suppose the corresponding w, z pair

$$w = \begin{bmatrix} \hat{y}_{m+1} \\ \vdots \\ \hat{y}_{m+n} \\ \hat{x}_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix}, \qquad z = \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_n \\ \hat{y}_1 \\ \vdots \\ \hat{y}_m \end{bmatrix},$$

is almost complementary. Now by our non-degeneracy assumption $\hat{x}$, $\hat{y}$ have exactly $N = m+n$ zeroes between them. Then either N pairs w_i, z_i have $w_i z_i = 0$, and we have a solution to (13), or else $N-1$ pairs w_i, z_i have $w_i z_i = 0$, and there is exactly one coefficient ν such that w_ν, z_ν are both zero. Thus for each iteration of Lemke's algorithm, one coefficient ν is distinguished in a natural way.

One more notational matter before we describe the algorithm: in a given a-c solution w, z of (13), we say that w_i is non-basic

if $\omega_i = 0$, basic otherwise. Similarly z_i is basic if and only if $z_i \neq 0$ in the given solution.

Lemke's Algorithm:

1. The algorithm begins with an a-c pair ω, z and a unique coefficient ν such that $\omega_\nu = z_\nu = 0$. Increase the non-basic variable z_ν holding all other non-basic variables at zero, until

(i) a basic variable becomes zero;

(ii) an unbounded ray is produced.

Exactly one basic variable becomes zero in (i). Otherwise we could get a degenerate solution for one of our LP's (1) and (5).

2. In the general step of our algorithm, we have an a-c pair ω, z. If ω, z is complementary, then stop. Otherwise there will be a unique coefficient ν such that $\omega_\nu = z_\nu = 0$, and either ω_ν or z_ν became zero in the previous iteration. If ω_ν became zero last iteration, we increase z_ν. If z_ν became zero last iteration, we increase ω_ν.

Again we either arrive at an unbounded ray or a new a-c solution. □

The indices of R^N are always broken into three sets in our algorithm: one index β has $\omega_\beta, z_\beta \neq 0$. One index ν has $\omega_\nu = z_\nu = 0$. To fill out our a-c solution, the other indices give basic/non-basic pairs: if ω_i is basic, then z_i is not, and vice versa.

If we move from one b.f.s. to another in step 2, then a basic variable becomes zero. If this basic variable was one of w_β, z_β we are done. Otherwise we get a new a-c solution. One of the w_i, z_i from the basic/non-basic pairs becomes zero, and i takes the role of ν.

We can also describe Lemke's algorithm as a "dual Simplex" method. From w, z we can read off a pair of b.f.s. for the standard LP (1) and its dual (5). To begin with, we let x_ν enter the basis of (1), with x_i leaving. We then let y_i enter the basis of (5).

We go back and forth from primal to dual, selecting as entering variable the variable leaving the other program. Our arguments about ν, β etc. would become, in these terms, arguments that a certain column was indeed not yet in a basis.

Now that we have described Lemke's algorithm, we must prove its correctness. We answer three questions.

1. Will the algorithm terminate?
2. How will the algorithm start?
3. What happens if we end with an unbounded edge?

Answer 1: We show that the algorithm terminates.

Lemma 1: There are at most two edges followed by the algorithm in arriving at or leaving a given a-c solution w_0, z_0.

Proof: To leave the point ω_0, z_0 arrived at in the course of the algorithm, we either increase ω_ν or z_ν. It remains to show that the two edges generated in this way are the only edges entering ω_0, z_0. That is, if the algorithm goes from some point ω', z' to ω_0, z_0, then we arrive at ω', z' from ω_0, z_0 by increasing ω_ν or z_ν.

This follows by considering degrees of freedom. Suppose starting from ω', z' we increase the distinguished variable x to reduce ω_ν to zero and arrive at ω_0, z_0. Then ω', z' differs in its non-basic variables from ω_0, z_0 only in the substitution of x for ω_ν. Then ω', z' and ω_0, z_0 have $N-1$ non-basic variables in common, which are fixed at zero along the edges leaving ω', z' and ω_0, z_0.

But $\omega = q + Mz$ is an equation in R^N, so that fixing $N-1$ variables leaves only one degree of freedom. Changing any of the other variables must generate the same edge. □

Lemma 2: The first point to recur in Lemke's algorithm is the initial point.

Proof: Exactly two edges touch any a-c b.f.s. on the path of Lemke's algorithm. Thus the only way the algorithm revisits a point is by first revisiting the previous point. We can't repeat the second point of our path until the first is repeated. Induction finishes the lemma. (See figure 10). □

Corollary: If our algorithm commences at the endpoint of an unbounded ray, the algorithm terminates.

Proof: Since the first point never repeats, no point does. We visit each of finitely many b.f.s. pairs $\hat{x}, \hat{y}$ at most once. □

Answer 2: This last corollary tells us how we would like to start. We introduce dummy variables $\omega_0, z_0 \in R$.

$$\text{Let} \quad \omega = z_0 u_N + q + Mz \tag{14}$$

where $u_N = (1, 1, \ldots, 1)^T$.

Clearly for a large enough z_0 we can let $z = 0$, and still have $\omega \geq 0$.

We write

$$\begin{bmatrix} \omega_0 \\ \omega \end{bmatrix} = \begin{bmatrix} q \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & \ldots & 0 \\ 1 & & & & & \\ \vdots & & & M & & \\ 1 & & & & & \end{bmatrix} \begin{bmatrix} z_0 \\ z \end{bmatrix} \tag{15}$$

Here we have written (14) in the form of (13). We get an a-c ray by saying $z = 0$, and increasing z_0. To get to the endpoint of this a-c ray, we let z_0 decrease toward zero. If z_0 becomes zero with no ω_r becoming negative, then (14) reduces to

$$\omega = q + Mz$$

and we have a solution for (13).

Otherwise, one ω_r will become zero. We will then have an a-c solution for (15) with exactly $N+1$ zeroes, and can commence our

algorithm. If our algorithm reaches a complementary solution of (15), we will have $z_0 = 0$, and thus, a solution for (13). We must ask then, what happens if we end in an a-c ray?

Answer 3: If our algorithm terminates in an a-c ray of (15), then (13) has no solution. Suppose the endpoint of the a-c ray that the algorithm reaches is $(\omega^*; z_0^*; z)$.

Then let $(\omega^h; z_0^h, z^h)$ be another point on the ray. We have

$$\omega^h = z_0^h u_N + M z^h \tag{16}$$

$(\omega^h; z_0^h, z^h) \geq 0$.

Also the points on the ray are a-c. Thus for any $\lambda > 0$,

$$(\omega^* + \lambda\omega^h) = q + (z_0^* + \lambda z_0^h)^T u_N + M(z^* + \lambda z^h) \tag{17}$$

and $(\omega_i^* + \lambda\omega_i^h)(z_i^* + \lambda z_i^h) = 0$

$$\text{for } i = 1, 2, \ldots, N+1 \tag{18}$$

Case 1: $z^h = 0$.

Since $(\omega^h; z_0^h, z^h) \neq 0$, we must have $z_0^h \neq 0$, for $\omega^h = z_0^h u_N$.

But now $\omega^h \neq 0$, and by (18) $z^* + \lambda z^h = z^* = 0$.

But then the ray we have arrived at is the same one used to start the algorithm, which is impossible.

Case 2: $z^h \neq 0$.

Lemke defined a class of matrices as co-positive plus. Such matrices must satisfy

$$u^T M u \geq 0 \quad \text{for all} \quad u \geq 0 . \tag{19}$$

$$(M + M^T)u = 0 \quad \text{if} \quad u^T M u = 0 \quad \text{and} \quad u \geq 0 . \tag{20}$$

Our matrix M has the form $M = \begin{bmatrix} 0 & -A^T \\ A & 0 \end{bmatrix}$, and must satisfy both these conditions. (See [5], p. 102).

By (18),

$$\omega_i^* z_i^* = \omega_i^* z_i^h = \omega_i^h z_i^* = \omega_i^h z_i^h = 0 ,$$

since $\omega_i^*, \omega_i^h, z_i^*, z_i^h \geq 0$. Now, $0 = z^{h^T} \omega^h = z^{h^T} u_N z_0 + z^{h^T} M z^h$.

Both parts of this sum are non-negative, hence both are zero.

As $z^h \neq 0$, we must have $z_0^h = 0$.

Also by (20)

$$M z^h + M^T z^h = 0 .$$

Since $z_0^h = 0$, $M z^h = \omega \geq 0$. Therefore

$$M^T z^h \leq 0 , \quad \text{and} \quad z^{h^T} M \leq 0 \tag{21}$$

But $0 = z^{*^T} \omega^h = z^{*^T} M z^h = z^{*^T} (-M^T z^h) = -z^{h^T} M z^* .$

Now, $0 = z^{h^T} \omega^* = z^{h^T} q + z^{h^T} u_N z_0^* + z^{h^T} M z^*$

$= z^{h^T} q + z^{h^T} u_N z_0^* .$

We conclude that $z^{h^T} q < 0$, since $z^{h^T} u_N z_0^* > 0$.

But now we have

$$z^{h^T} M \leq 0 , \quad z^{h^T} q < 0 , \quad z^h \geq 0 , \quad \text{and}$$

(13) has no solution. For suppose

$$\omega = q + Mz \; ; \qquad \omega , z \geq 0 .$$

Then, multiplying by z^{h^T},

$$0 \leq z^{h^T} \omega = z^{h^T} q + z^{h^T} Mz < 0 .$$

This is a contradiction.

In conclusion, Lemke's algorithm always terminates, and gives a solution to (13), or proof no solution exists.

Klee-Minty Examples

We are almost ready to discuss the contents of Smale's paper. But Smale discusses the average speed of Simplex. Here we will use the examples of Klee-Minty to show that in some cases, the Simplex algorithm is very bad. Good average time is the best we can do.

As we know, the Simplex algorithm traces a path around the edges of a polyhedron. We will attempt to find a long path for a polyhedron given by $m+n$ planes. We will look at hyper-cubes. A line segment has a path of length, 1 edge. Thus as a square is generated by moving a line through a second dimension, we can find two paths of length 1 on opposite sides of a square, and connect these paths to form a path of length 3 edges. With a cube we can trace paths of length 3 on op-

posite faces and connect them for a path of length 7. See Figures 11-13. In general, by tracing a path of length $2^{n-1} - 1$ on opposite facets of a hypercube in n dimensions, we can find a path of length $2^n - 1$ on the hypercube.

This is well and good, but a Simplex method using least coefficient on non-degenerate polyhedra always decreases the cost function on each iteration. We need to deform our cubes so that the cost function decreases everywhere along our path. For example, in 3-dimensions the rule of least cost coefficient takes 3 iterations on any cube, not 7. (Think about Figures 7-9 to see why this is so.)

We can build a "squashed" hypercube on which some cost always decreases along our path in the following inductive way: Given a "squashed" hypercube of dimension $n-1$, generate the "squashed" cube of dimension n by first moving the $(n-1)$-dimension cube through the nth dimension, and then drawing our long path. Then shrink the path edge connecting the two $(n-1)$-dimensional components. (See Figures 14-16 for "squashed" objects in 1, 2, 3 dimensions, and cost directions along which the path is always decreasing.)

Finally, we must force the Simplex algorithm to always take a long path. We do this by remarking that the method of least cost coefficient is very sensitive to scale. If we replace x_q in an LP by $\frac{1}{100} t_q$, then in the new program the magnitude of t_q's cost coefficient

will be $100\, c_q$. We now construct the entire Klee-Minty example for $n = 3$. Let us take the cube of Example 1 of our geometric exposition.

$$\begin{aligned} x_1 &\leq 1 \\ x_2 &\leq 1 \\ x_3 &\leq 1 \\ x_1, x_2, x_3 &\geq 0; \end{aligned}$$

Then

$$\begin{aligned} x_1 &\leq 1 \\ 0.2x_1 + x_2 &\leq 1 \\ 0.02x_1 + 0.2x_2 + x_3 &\leq 1 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

is the squashed cube in Figure 17. If we let our objective function be $10000x_1 + 1000x_2 + 100x_3$, then this function always decreases along the indicated path. We will now use scaling to make this path attractive to the Simplex method.

We will change our scale so that the cost coefficients of the top and bottom faces are least attractive, those of the front and back faces second, and those of the left and right faces, most attractive. This will force the Simplex algorithm to follow our path. The pairs of faces are given by

right/left: x_1 and x_4 = $1 - x_1$

front/back: x_2 and x_5 = $1 - 0.2x_1 - x_2$

top/bottom: x_3 and x_6 = $1 - 0.02x_1 - 0.2x_2 - x_3$.

Letting $t_1 = x_1$, $t_2 = \frac{1}{10000} x_2$, $t_3 = \frac{1}{(10000)^2} x_3$ will certainly be adequate.

This gives the program

$$\text{minimize } 10000\, t_1 + 10000000\, t_2 + 10000000000\, t_3$$

subject to

$$\begin{aligned} t_1 &\leq 1 \\ 2t_1 + 100000\, t_2 &\leq 10 \\ 2t_1 + 200000\, t_2 + 10000000000\, t_3 &\leq 100 \\ t_1, t_2, t_3 &\geq 0 . \end{aligned}$$

We now know why the Simplex algorithm takes 7 iterations on this program. Also, following the steps of our explanation shows how to prove the following lemma by induction.

Lemma: The following program takes $2^n - 1$ iterations to solve by the least cost coefficient Simplex method:

$$\text{minimize } \sum_{i=1}^{n} 10^{2+3i} t_i$$

subject to

$$\left(2 \sum_{j=1}^{i-1} (100000)^{i-j} t_j\right) + (100000)^{i-1} t_i \leq 10^{i-1}$$

for $i = 1$ to n

$$t_i \geq 0 .$$

Thus the Simplex method can take exponentially long. □

Smale's Geometrical Approach to the Simplex Method

Smale analyzes Lemke's algorithm to find the expected number of steps the algorithm takes. To speak of an expected number of steps, we must give a distribution of input. Our input space for the Simplex algorithm is the set of 3-tuples (A, b, c) where $A \in R^{mn}$, $b \in R^m$, and $c \in R^n$. Originally Smale projected his 3-tuples onto a sphere and gave the sphere a uniform distribution. Later he realized that he only used certain properties of his particular measure. We will list the properties we wish our measure on $R^{mn} \times R^m \times R^n$ to have, with reasons.

We assume a measure $\mu_{mn,m,n}$ (probability) on $R^{mn} \times R^m \times R^n$, with the following three properties:

(1) Continuity. The measure $\mu_{mn,m,n}$ is absolutely continuous with respect to Lebesgue measure. In particular a subset of $R^{mn} \times R^m \times R^n$ of Lebesgue measure zero occurs as input with probability zero.

This assumption allows Smale to reduce his expected complexity for Simplex. Later, in a certain sum, property (1) will be invoked to throw away half the sum. The terms discarded have measure zero. This will be crucial to proving a small expected number of steps.

(2) Independence. The measure $\mu_{mn,m,n}$ is a product of three measures $\mu_{R^{mn}}$, μ_{R^m}, μ_{R^n} on R^{mn}, R^m, R^n respectively, and in the natural way. Also μ_{mn} is a product of measures $\mu_n \times \mu_n \times \ldots \times \mu_n$ on the rows of $m \times n$ matrices.

These independence properties allow us to break our complexity problem into sub-problems. For example, in later estimates we work in $R^N = R^n \times R^m$. Because of property (2), a measure on R^N is available to us.

(3) Symmetry. Permuting coordinates of R^N induces permutations on R^{mn}, and on $R^n \times R^m$. We will want to make estimates later based on the number of columns having a certain property. In order that we don't have to worry <u>which</u> columns have property P, we want our measure to be invariant under the above permutations. In particular, switching columns in A or renaming variables should have no effect.

This is the end of our discussion of measure.

Suppose we are given the LP (1).

$$\begin{aligned} &\min\ c^T x \\ \text{s.t.}\ &Ax \geq b \\ &x \geq 0 \quad ; \quad x \in R^n,\ b \in R^m,\ A \in R^{mn}. \end{aligned} \tag{1}$$

To <u>solve</u> (1) means to exhibit a solution, or to decide that no solution exists.

Smale's Main Theorem: Let $\rho(m,n)$ be the average number of steps to solve an LP of form (1) under our measure. Then for m fixed, and $\epsilon > 0$, $\rho(m,n)$ grows more slowly than Kn^{ϵ}, for some constant K.

If we take $\epsilon = 1$, we find that the Simplex algorithm takes less than linear time in n.

Define $\rho_A = \int_{R^m \times R^n} \rho_{A,b,c} \, \mu_m \, \mu_n$ where $\rho_{A,b,c}$ is the number of steps to solve (1) using Lemke's algorithm.

Then $\rho(m,n) = \int_{R^{mn}} \rho_A \, \mu_{mn}$.

To prove Smale's main theorem, we derive a geometric meaning for ρ_A. By manipulating this geometric expression for ρ_A, we prove the theorem.

To develop Smale's geometric formula for ρ_A, we use something called a loop invariant. In complexity theory, when we wish to count iterations of an algorithm, it is useful to label each iteration by some well-behaved function of the quantities used by the algorithm. This label is called a loop invariant.

At each step of Lemke's algorithm we have

$$\omega = q + Mz + z_0 u_N \tag{25}$$

where $u_N = (1, 1, \ldots, 1)^T$.

Recall that when we commence the algorithm we let $\omega = q_0 = q + z_0^+ u_N$ for some z_0^+, and $z = (0, 0, \ldots, 0)^T$. Also z_0 never increases, but

rather decreases to zero. Although the coordinates of ω and z move up and down, z_0 is well-behaved. We can label each iteration by z_0. We now wish to recast formula (25) into a functional notation (pun not intended).

Let $x = (a_1, a_2, \ldots, a_N)$

where

$$a_i = \begin{cases} -z_i & \text{if } z_i > 0 \\ w_i & \text{otherwise} \end{cases}$$

we use here the fact that $\omega_i z_i = 0$.

Then if $x^+ = (\max(0,x_1), \ldots, \max(0,x_N))$ and $x^- = (\min(0,x_1), \min(0,x_2), \ldots, \min(0,x_N))$, we have $x^+ = \omega$ and $-x^- = z$. Therefore, (25) becomes a simple equation.

$$x^+ + Mx^- = q + z_0 u_n \quad . \tag{26}$$

In fact, if we find any x for which (26) holds, we can let $\omega = -x^-$ and $z = x^+$, and we will have

$$\omega = q + Mz + z_0 u_n$$

and $\omega^T z = 0$.

Let $x^+ + Mx^- = \varphi_M(x)$. Then solving L.C.P. consists in finding $x \in R^N$ such that $\varphi_M(x) = q$. $\varphi_M(x)$ will be our loop invariant. At the beginning of our algorithm, as we remarked earlier,

$$\omega = q + z_0^+ u_N = \varphi_M(x) = q_0 \, .$$

During the algorithm, z_0 decreases monotonically towards zero. Thus, during the execution of Lemke's algorithm, $\varphi_M(x)$ varies linearly from q_0 to q. However, we have not yet given a method of counting iterations by looking at $\varphi_M(x)$.

In each iteration of Lemke's algorith, some z_i (or w_i) increases from zero, and some w_j (or z_j) decreases to zero. These two changes cause x to move from one orthant of R^N to another: The ith and jth coordinates of x change sign. Thus an iteration corresponds to the change of x from one orthant to another. We can describe the image of an orthant under φ_M easily. This will allow us to bound the number of orthants x visits by looking at $\varphi_M(x)$.

Recall, $\varphi_M(x) = x^+ + M(x^-)$. An orthant of R^N corresponds to a subset S of $\{1, 2, \ldots, N\}$ in the following obvious way.

$$S \leftrightarrow Q_s = \{x \in R^N \mid x_i \leq 0 \text{ if } i \in S$$
$$x_j \geq 0 \text{ if } j \notin S\} .$$

Then $\varphi_M(Q_s) = \{ - \sum_{i \in S} \lambda_i m_i + \sum_{j \notin S} \lambda_j e_j \mid \lambda_k \geq 0\}$, where e_i is the ith component vector in R^N, and $M = [m_1, m_2, \ldots, m_N]$, m_j the jth column of M.

Define the complementary cone of M, $K_{S,A}$, by

$$K_{S,A} = \varphi_M(Q_s) .$$

We are now almost ready to give Smale's "First Main Formula".

The number of steps Lemke's algorithm will take is equal to the number of orthants through which x varies. But if $x \in Q_s$, then $\varphi_M(x) \in K_{S,A}$. But, $\varphi_M(x)$ varies from q_0 to q, thus

$$\rho_{A,b,c} = \#(\text{orthants } x \text{ moves through}) - 1$$
$$\leq \#(\text{cones } K_{S,A} \text{ touching } q_0q) - 1$$

where q_0q is the path in R^N from q_0 to q. We subtract 1 because x must begin in some orthant.

Lemma: The segment q_0q meets $K_{S,A}$ if and only if q lies in the cone generated by $-u_N$ and $K_{S,A}$. □

The lemma can be proved by trivial algebra. This allows us to state a new lemma.

Lemma:

$$\rho_{A,b,c} \leq -1 + \#\{S \mid q \in K(-u_N, K_{S,A})\}$$

where $K(-u_N, K_{S,A})$ is the cone generated by $K_{S,A}$ and $-u_N$. □

We will now prove Smale's "First Main Formula."

Formula: $$\rho_A \leq -1 + \sum_{S \subset \{1,\ldots,N\}} VK(-u_N, K_{S,A})$$

where VP means the volume of P, in this case with respect to $\mu_{m \times n}$.

Proof: From the previous lemma,

$$\rho_{A,b,c} \leq -1 + \#\{S \mid q \in K(-u_N, K_{S,A})\}.$$

$$\text{Now } \rho_A = \int_{R^m \times R^n} \rho_{A,b,c}\, \mu_{m\times n}$$

$$\leq \int_{R^m \times R^n} (-1)\, \mu_{R^m \times R^n} + \int_{R^m \times R^n} \#\{S \mid q \in K(-u_N, K_{S,A})\}\, \mu_{m\times n}$$

$$= -1 + \sum_{S \subset \{1,\ldots,N\}} \int_{q \in K(-u_N, K_{S,A})} 1 \cdot \mu_{R^m \times R^n}$$

$$= -1 + \sum_{S \subset \{1,\ldots,N\}} VK(-u_N, K_{S,A}). \qquad \square$$

This lemma is the most important result of the geometric portion of Smale's paper. (He, in fact, by more careful attention to cases where no solution exists, proves equality.)

A number of properties of $\varphi_M(x)$, to be used later, also fit in this geometric section.

Recall $\varphi_M(x) = x^+ + M(x^-)$. Suppose $x \in Q_s$, $S = \{1, 2, 3, \ldots, s_1, n+1, n+2, \ldots, n+s_2\}$. We partition the matrix A:

$$\left[\begin{array}{c|c} \overbrace{A_1}^{s_1} & A_3 \\ \hline A_2 & A_4 \end{array}\right] \Big\} s_2$$

Now for $x \in Q_s$, $\varphi_m(x) = x^+ + M(x^-)$

$$= \begin{bmatrix} 0 & 0 & -A_1^T & 0 \\ 0 & I_{n-s_1} & -A_2^T & 0 \\ A_1 & 0 & 0 & 0 \\ A_3 & 0 & 0 & I_{m-s_2} \end{bmatrix} [x] \quad .$$

From this it follows that $\varphi_M(x)$ is linear on Q_s, the identity on R^{N+}, and continuous on R^N. □

Smale's "First Main Estimate"

In terms of concepts, Smale's paper breaks into two parts. In the first part, he develops his "Main Formula"; Using a loop invariant, he gives a pleasing geometrical interpretation of the expected number of steps for the Simplex method, as the sum of volumes of cones in R^{m+n}.

However, it seems difficult to motivate the combinatorial section of Smale's paper in terms of these geometric ideas. To motivate the second section of Smale's paper, we turn to the idea of dominance, which Smale himself does not mention explicitly.

This writer became aware of dominance through a paper of Blair [1]. Blair seems hostile to Smale's geometric approach, and attempts to prove

Smale's result for a broad class of Simplex methods, using only dominance. He succeeds in proving a weaker result, but only at the expense of assuming $m << n$.

Consider the following linear program:

$$\begin{aligned} \min \quad & -x_1 - 2x_2 - x_3 + x_4 \\ & -x_1 + 15x_2 + x_3 + x_4 \geq 1 \\ & 3x_1 + 4x_2 - 2x_3 - 2x_4 \geq 2 \\ & -2x_1 - 2x_2 + x_3 - x_4 \geq 6 \\ & x \geq 0 . \end{aligned}$$

A moment's reflection tells us that no optimal solution contains column 1, because column 2 has lower cost, and the constraint coefficients of column 2 are all greater than those of column 1. If a solution contains x_1 , we can still satisfy our constraints by replacing x_1 with an equal amount of x_2 , and the cost is reduced.

We say column 1 is <u>dominated</u> by column 2 . Similarly, column 4 is dominated by column 3 . In general, if $A^{(i)} \leq A^{(j)}$ and $c_i \geq c_j$ in a linear program (1), we say $A^{(i)}$ (or sometimes x_i) is dominated by $A^{(j)}$ (sometimes x_j) .

Blair observes that some Simplex methods, in particular Lemke's algorithm, but others also, never allow dominated columns to enter the basis. Then if matrix A has U undominated columns, we only choose

bases from our starting set of m basis columns, and the U undominated columns. We thus look only at $\binom{U+m}{m}$ bases, at most.

So far we have followed Blair, but now we turn from Blair, to attempt to use dominance to give Smale's "First Main Estimate".

Let $P(m, n, s)$ be the probability under our measure $\mu_{m,n}$ that an $m \times n$ matrix A has as undominated columns precisely its first s columns. Because we have assumed in condition 3 on our measure that permuting columns does not matter, this is the probability that <u>any</u> specified set of s columns of A is undominated. There are $\binom{n}{s}$ such sets.

Surprisingly, we can now almost prove Smale's "First Main Estimate". We have,

$$\rho_{m,n} \leq \sum_{s=0}^{n} (\text{Probability of } s \text{ undominated columns}) \times \binom{s+m}{m}$$

$$= \sum_{s=0}^{n} \binom{n}{s} P(m,n,s) \binom{s+m}{s} \quad .$$

Now, suppose we can show that $P(m,n,s) = 0$ for $s > m+1$. Then

$$\rho_{m,n} \leq \sum_{s=0}^{m+1} \binom{n}{s} P(m,n,s) \binom{s+m}{s}$$

$$\leq \max_{s=0 \text{ to } m+1} \binom{s+m}{s} \sum_{s=0}^{m+1} \binom{n}{s} P(m,n,s)$$

$$= C_m \sum_{s=0}^{m+1} \binom{n}{s} P(m,n,s) \quad .$$

To make our supposition plausible, we note that Smale introduces a lemma to reduce his largest index in the summation from n to $m+1$.

We have not been able to deduce completely Smale's estimate from dominance, but the above remarks serve to motivate Smale's estimate, and particularly the definition of $\sigma(m,n,s)$. We now turn to Smale's work.

Let E be a partial order on a set S of n elements, $S = \{1,2,\ldots,n\}$. For example, an $m \times n$ matrix A induces a partial order $E(A)$ by dominance:

$$p \geq_{E(A)} q \quad \text{iff} \quad a_{ip} \geq a_{iq}, \quad i = 1 \text{ to } m.$$

We define an invariant $\gamma(E)$ to be the number of total orders on S compatible with the partial order E, divided by $n!$. As there are $n!$ total orders on S, we can interpret $\gamma(E)$ to be the (uniform) probability that if we pick a total order at random, it is compatible with E.

Define another partial order on S as follows. Let $S_1 \subset S$. Then $E_{S_1}(A)$ is given by

$$p \geq_{E_{S_1}(A)} q \quad \text{iff} \quad p \geq_{E(A)} q \quad \text{and} \quad q \in S_1.$$

Define $\sigma(m,n,s)$ by

$$\sigma(m,n,s) = \int_{R^{mn}} \gamma(E_{S_1}(A))\, \mu_{m,n}$$

where $|S_1| = s$. Again, by property 3 of the measure, $\sigma(m,n,s)$ depends only on s, not on the actual elements of S_1. Clearly, $\sigma(m,n,s)$ has the form of a probability.

Smale's First Main Estimate:

$$\rho_{m,n} \leq C_m \sum_{s=0}^{m+1} \binom{n}{s} \sigma(m,n,s)$$

where C_m is a constant depending on m.

We see that Smale's estimate is of the same form as our previous attempt. In fact, his constant will have the same form. Although $\sigma(m,n,s)$ is clearly the probability that a matrix has some sort of ordering property, it is not clear to this writer what type of property this is.

Before giving Smale's proof of this estimate, we recall the main formula, with some of its accompanying definitions.

Recall: $\rho_A = -1 + \sum_{S \subset \{1,2,\ldots,N\}} VK(-u_N, K_{S,A})$

Recall: $VK(-u_N, K_{S,A})$ is the volume of the cone generated by $-u_N$ and $K_{S,A}$.

Recall: $K_{S,A}$ is the image under φ_M of the orthant Q_S. In particular,

$$K_{S,A} = \{-\sum_{i \in S} \lambda_i m_i + \sum_{j \notin S} \lambda_j e_j \mid \lambda_K \geq 0\}.$$

Recall: $M = \left[\begin{array}{c|c} 0 & -A^T \\ \hline A & 0 \end{array}\right] = [m_1, \ldots, m_N]$.

Note that the first n columns of M correspond to columns of A, with zeroes added. The last m columns correspond to rows of A with zeroes added. If S is a subset of $\{1, 2, \ldots, N\}$ then let $S_1 = S \cap \{1, 2, \ldots, n\}$, that is, the elements of S corresponding to columns of A.

To prove Smale's estimate, we start with the following lemma.

Lemma 1: For E a partial order on $S = \{1, 2, \ldots, n\}$, let $X_E = \{x \in R^n \mid x_p \geq x_q \text{ if } p \geq_E q\}$.

Then under our measure, $VX_E = \gamma(E)$.

Proof: The measure referred to is the measure on R^n assured us by property 2 of our measure.

We divide R^n into $n!$ parts, one for each total order $(i_1, \ldots, i_n)$, by

$$X_{(i_1, \ldots, i_n)} = \{x \in R^n \mid X_{i_1} \geq X_{i_2} \geq \ldots \geq X_{i_n}\}.$$

If $(i_1, \ldots, i_n) \neq (j_1, \ldots, j_n)$, then we find $V(X_{(i_1, \ldots, i_n)} \cap X_{(j_1, \ldots, j_n)}) = 0$. For example, in R^3, $R_{(1,2,3)} \cap R_{(1,3,2)} = \{x \in R^3 \mid x_1 \geq x_2 = x_3\}$, a subspace of dimension 2 in R^3. By condition 1 of the measure, such a subspace has measure 0.

But $R^n = \bigcup X_{(i_1,\ldots,i_n)}$, and by condition 3 (symmetry) of our measure, each $X_{(i_1,\ldots,i_n)}$ has equal measure $\frac{1}{n!}$. Since $X_{(i_1,\ldots,i_n)} \subset X_E$ exactly when the order $(i_1,\ldots,i_n)$ is compatible with E, the result follows.

Lemma 2: Let $\pi: R^N \to R^n$ be the projection onto the first half of $R^N = R^n \times R^m$. Then $\pi(K(-u_N, K_{S,A})) \subset X_{E_{S_1}(A)}$.

Proof: Let $x = (x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_N)$. Suppose $x \in K(-u_N, K_{S,A})$, so that $(x_1, \ldots, x_n) \in \pi(K(-u_N, K_{S,A}))$. As we recall, $x_K = -\lambda_0 + \lambda_K + \sum_{i \in S} \lambda_i a_{iK}$ if $K \notin S_1$

and $x_K = -\lambda_0 + \sum_{i \in S} \lambda_i a_{iK}$ if $K \in S_1$,

for $K = 1$ to n.

We must show that $(x_1, \ldots, x_n) \in X_{E_{S_1}(A)}$. Therefore, suppose $p \geq_{E_{S_1}(A)} q$. This means that $q \in S_1$, and $a_{jp} \geq a_{jq}$ for all j. From our above formulae for x_K, we see $x_p \geq x_q$. Q.E.D. □

It is interesting to remark that in the previous two lemmas we have been working in R^n rather than in R^N. In fact in lemma 2, for no obvious reason, we throw away our second factor of $R^N = R^n \times R^m$. This is a serious step; by condition 3, our measure on R^N is a product of probability measures. This means that $VK(-u_N, K_{S,A}) \leq V_\pi(K(-u_N, K_{S,A}))$ under the measures in R^N, R^n respectively.

Lemma 3: $\rho_A \leq \sum_{\substack{S_1 \subset \{1,\ldots,n\} \\ S_2 \subset \{n+1,\ldots,N\}}} {}^{VX}E_{S_1}(A)$

Proof: The main formula gives $\rho_A \leq \sum_{S \subset \{1,..,N\}} {}^{VK}(-u_N, K_{S,A})$. We divide S into two sets, $S_1 = S \cap \{1,2,\ldots,n\}$ and $S_2 = S \cap \{n+1,\ldots,N\}$. We know that ${}^{VK}(-u_N, K_{S,A}) \leq V(\pi(K(-u_N, K_{S,A}))) \leq {}^{VX}E_{S_1}(A)$. However, each time a given S_1 appears, there are $\binom{m}{r}$ possibilities for S_2 where r runs from 0 to m, which gives us the double summation.

Lemma 4: $V(K(-u_N, K_{S,A})) = 0$ unless $|S_1| = |S_2|$ or $|S_1| = |S_2| + 1$ or $|S_1| = |S_2| - 1$.

Proof: Let S_1 correspond to the first $|S_1|$ columns of A, S_2 to the first $|S_2|$ rows. Then write A as

$$A = \left[\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array}\right] \begin{array}{l} \} |S_2| \\ \} m - |S_2| \end{array}$$

$$\underbrace{}_{|S_1|} \quad \underbrace{}_{n - |S_1|}$$

Thus M has the form

$$M = \begin{bmatrix} 0 & 0 & -A_1^T & -A_3^T \\ 0 & 0 & -A_2^T & -A_4^T \\ A_1 & A_2 & 0 & 0 \\ A_3 & A_4 & 0 & 0 \end{bmatrix}$$

We recall that φ_M is linear on the orthant Q_s, and thus has the form

$$\varphi_M|_{Q_s} = \begin{bmatrix} 0 & 0 & -A_1^T & 0 \\ 0 & I_{n-|S_1|} & -A_2^T & 0 \\ A_1 & 0 & 0 & 0 \\ A_3 & 0 & 0 & I_{m-|S_2|} \end{bmatrix}.$$

Thus the image of $\varphi_M|_{Q_s}$ is generated by the columns of the matrix

$$B = \begin{bmatrix} 0 & 0 & A_1^T & 0 \\ 0 & I_{n-|S_1|} & A_2^T & 0 \\ -A_1 & 0 & 0 & 0 \\ -A_3 & 0 & 0 & I_{m-|S_2|} \end{bmatrix}.$$

Now $K_{S,A} = \varphi_M(Q_s)$ and thus $\dim K_{S,A} = \text{rank } B$. By property 1 of the measure, $VK(-u_N, K_{S,A}) = 0$ if

$$\dim < -u_N, K_{S,A} > \neq N.$$

We thus require $\dim K_{S,A} \geq N-1 \Leftrightarrow \text{rank } B \geq N-1$, if $VK(-U_N, K_{S,A}) \neq 0$.

Discounting the identity matrices, this gives rank $\begin{bmatrix} 0 & A_1^T \\ -A_1 & 0 \end{bmatrix} \geq |S_1| + |S_2| - 1$

But rank $\begin{bmatrix} 0 & A_1^T \\ -A_1 & 0 \end{bmatrix}$ = rank $(-A_1)$ + rank (A_1^T) = 2 rank $A_1 \leq 2 \min(|S_1|, |S_2|)$

If $|S_1| < |S_2| - 1$, then $2(\min(|S_1|, |S_2|)) \leq 2|S_1| < |S_1| + |S_2| - 1$.

Then if $VK(-U_N, K_{S,A}) \neq 0$, we require $|S_1| \geq |S_2| - 1$, $|S_2| \geq |S_1| - 1$, and the lemma follows.

Thus lemma 3 is strengthened to say

$$\rho_A \leq \sum_{\substack{S_1 \subset \{1,\ldots,n\} \\ S_2 \subset \{n+1,\ldots,N\} \\ |S_1| = |S_2| \\ \text{or } |S_1| = |S_2| +/- 1}} VX_{E_{S_1}}(A)$$

$$= \sum_{\substack{S_1 \subset \{1,\ldots,n\} \\ S_2 \subset \{n+1,\ldots,N\} \\ |S_1| = |S_2| \\ \text{or } |S_1| = |S_2| +/- 1}} \gamma(E_{S_1}(A))$$

In particular, $|S_1| \leq m+1$. Thus by working with cones, Smale has made a crucial step.

Now,

$$\rho_{m,n} = \int_{R^{mn}} \rho_A \mu_{m,n} \leq \int_{R^{mn}} \sum_{\substack{S_1 \subset \{1,\ldots,n\} \\ S_2 \subset \{n+1,\ldots,N\} \\ |S_1| = |S_2| \\ \text{or } |S_1| = |S_2| +/- 1}} \gamma(E_{S_1}(A)) \mu_{m,n}$$

$$= \int \sum_{S_2 \subset \{n+1,\ldots,N\},\ |S_2| = |S_1|,\ |S_1|+/-1} \ \sum_{S_1 \subset \{1,\ldots,n\}} \gamma(E_{S_1}(A))\, \mu_{m,n}$$

$$= \sum_{S_2 \subset \{n+1,\ldots,N\},\ |S_2| = |S_1|,\ |S_1|+/-1} \ \sum_{S_1 \subset \{1,\ldots,n\}} \int_{R^{mn}} \gamma(E_{S_1}(A))\, \mu_{m,n}$$

$$= \sum_{S_2 \subset \{n+1,\ldots,N\},\ |S_2| = |S_1|,\ |S_1|+/-1} \ \sum_{s=0}^{m+1} \binom{n}{s} \sigma(m,n,s)$$

$$\leq \max_{r=0 \text{ to } m} \binom{m}{r} \sum_{s=0}^{m+1} \binom{n}{s} \sigma(m,n,s)$$

$$= C_m \sum_{s=0}^{m+1} \binom{n}{s} \sigma(m,n,s) \ .$$

In a strict analogy to our earlier motivation, the $\binom{n}{s}$ appears because $\sigma(m,n,s)$ is dependent only on $s = |S_1|$. Smale has essentially followed our outline in arriving at this combinatorial estimate, but by working with his geometrical formula, he has replaced n by $m+1$. □

Smale's "Second Main Estimate"

In the previous section, we showed, following Smale, that

$$\rho(m,n) \leq C_m \sum_{s=0}^{m+1} \binom{n}{s} \sigma(m,n,s)$$

where C_m is a constant depending on m, and

$$\sigma(m,n,s) = \int_{A \in R^{mn}} \gamma(E_{S_1}(A)) \mu_{m,n} .$$

In this section, we will introduce a partial order $\tilde{J}_s(A)$ which is weaker than $E_{S_1}(A)$. It will follow that

$$\rho(m,n) \leq C_m \sum_{s=0}^{m+1} \binom{n}{s} \sigma_0(m,n,s)$$

where

$$\sigma_0(m,n,s) = \int_{A \in R^{mn}} \gamma(\tilde{J}_s(A)) \mu_{m,n} .$$

The combinatorial definition of $\tilde{J}_s(A)$ will then allow us to prove the main theorem.

We motivated the previous section by looking at undominated columns. As we saw, this was natural in terms of the Simplex method. However, now we wish to use counting arguments, and it is easier to look at dominated columns. This is because "undominated" is a negative property. For example, if a column is undominated, there is little we can say about the other columns. But if column A is dominated, it must be

dominated by some undominated column B: An undominated column may dominate something or not, but a dominated column always induces some grouping of columns, which facilitates counting.

Suppose the $m \times n$ matrix A has s undominated columns. As the order of columns is immaterial, assume that it is the first s columns of A that are undominated. As a first step toward grouping the dominated columns with undominated columns, we divide the $n - s$ dominated columns as evenly as possible into s sets:

Label the n columns of A as $1, 2, 3, \ldots, n$. We give a partition of these columns having the form $(S_s, H_1, H_2, \ldots, H_s)$, where the partition is fixed for each $s > 0$. The partition is given on the labels by $S_s = \{1, 2, \ldots, s\}$, and if $h_i = |H_i|$,

(a) $[(n - s)/s] \leq h_i \leq [(n - s)/s] + 1$,

(b) the labels of H_i are smaller than those of H_{i+1}, $i=1,\ldots,s-1$,

(c) the h_i are non-decreasing.

Example: If $s = 5$ and $n = 14$, the partition would give:

$S_s = \{1, 2, 3, 4, 5\}$; $H_1 = \{6\}$; $H_2 = \{7, 8\}$; $H_3 = \{9, 10\}$;

$H_4 = \{11, 12\}$; $H_5 = \{13, 14\}$. □

To arrive at a partial order, we will look at the columns of H_k which are dominated by the undominated column k.

Let $\mathcal{J}_s$ be the set of all s-tuples $(J_1, \ldots, J_s)$ where $J_i \subseteq H_i$. A map $\varphi_s: R^{mn} \to \mathcal{J}_s$ is defined by

$$\varphi_s(A) = (J_1, \ldots, J_s)$$

where $J_k = \{ j \in H_k \mid a_{ij} \geq a_{ik}, \forall i \}$.

Example: If $s = 5$ and $n = 14$, then

$$\varphi_s \left(\left[\begin{array}{cccccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 2 & 4 & 6 & 8 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 8 & 5 & 2 & -1 & -4 & -7 & -8 & 11 & 12 & 1 & 3 & 6 & 6 & 6 \end{array} \right] \right)$$

$$\begin{array}{ccccc} \underbrace{\qquad\qquad\qquad\qquad\qquad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} \\ S_s & H_1 & H_2 & H_3 & H_4 \end{array} \qquad \underbrace{\quad}_{H_5}$$

$= (\{\ \}, \{8\}, \{9\}, \{11\}, \{\ \})$, because no column in H_1 is greater than column 1, column 8 in H_2 is greater than column 2, etc. □

By means of $J = \varphi_s(A)$, A induces a partial order $\tilde{J}_s$ of columns:

$$p \underset{\tilde{J}_s}{\geq} q \quad \text{if} \quad q \in S_s \quad \text{and} \quad p \in J_q .$$

Clearly $\tilde{J}_s$ is a weaker partial order than $E_{S_1}(A)$ with $S_1 = S_s$. Therefore

$$\gamma(\tilde{J}_s(A)) \geq \gamma(E_{S_1}(A)),$$

by the definition of γ. This gives Lemma 1.

Lemma 1: Let $\sigma_0(m,n,s) = \int_{R^{mn}} \gamma(\tilde{J}_s(A))\, \mu_{m,n}$. Then

$\sigma_0(m,n,s) \geq \sigma(m,n,s)$. □

We will no longer need to talk about undominated and dominated columns. In fact the discerning reader will note that we have not

really used dominance to define σ_0. However, only thinking in terms of dominance makes the partition used, and the partial order introduced, have any natural meaning. On consideration, the reader should see that the motivating use of dominance here is exactly analogous to that in the previous section.

The usefulness of the new partial order becomes immediately evident in Lemma 2.

Lemma 2: $\gamma(\tilde{J}_s(A)) = \prod_{i=1}^{s} \frac{1}{(1+|J_i|)}$.

Proof: By definition, $\gamma(\tilde{J})$ is the number of partial orders on n elements compatible with $\tilde{J}_s$, divided by $n!$. Thus $\gamma(\tilde{J})$ is the (uniform) probability that if we pick an order on n elements, it agrees with $\tilde{J}$. But the only order given by $\tilde{J}$ is on sets of the form $\{q, H_q\}$. Any order compatible with $\tilde{J}_s$ on each of these sets is compatible with $\tilde{J}_s$. What is the probability that some order is compatible with $\tilde{J}_s$ on $\{q, H_q\}$?

$\tilde{J}_s$ only specifies $q \leq j$ for each $j \in J_q$. Thus an order is compatible with $\tilde{J}_s$ on $\{q, H_q\}$ if and only if q is before the $|J_q|$ elements of J_q in the order. This happens with probability $\frac{1}{|\{q, J_q\}|} = \frac{1}{1+|J_q|}$. Thus $\gamma(\tilde{J}_s(A)) = \prod_{i=1}^{s} \frac{1}{1+|J_i|}$. □

Now that γ has been given a simple form, we change the integral in the definition of σ_0 to a sum, so that we can simplify further.

Lemma 3: Let $P_J = V(\varphi_s^{-1}(J))$. Then $\sigma_0(m,n,s) = \sum_{J \in \mathcal{J}_s} \gamma(\tilde{J})\, P_J$.

Proof: Simple change of variable.

Since γ has already been simplified, we next attack P_J.

Lemma 4: For $J, J' \in \mathcal{J}_s$ we define $|J' - J| = \sum_{i=1}^{s} |J'_i - J_i|$, where $|J'_i - J_i|$ is the number of elements in $J'_i - J_i$. Then

$$P_J = \sum_{J' \subset J} (-1)^{|J' - J|}\, \gamma(\tilde{J}')^m .$$

Proof: First we define a region L_J which contains $\varphi_S^{-1}(J)$:

$$L_J = \{A \in R^{mn} \mid a_{\ell j} \geq a_{\ell i} \;\; \forall\, j \in J_i ,\;\; \forall\, i ,\;\; \ell = 1,2,\ldots,m\}$$

Thus $\varphi(L_J)$ gives an order compatible with $\tilde{J}$, but possibly stricter.

Claim: $V(L_J) = \gamma(\tilde{J})^m$.

Proof of claim: Each row of any matrix in L_J must be compatible with $\tilde{J}$. Each matrix has m rows. The probability that one row gives an order compatible to $\tilde{J}$ is $\gamma(\tilde{J})$. As the rows are independent, we get $V(L_J) = \gamma(\tilde{J})^m$. Compare the proof that $V(X_E) = \gamma(E)$ in the previous section. □

Notation: Given $J, J' \in \mathcal{J}_s$, let union, intersection, containment be understood component-wise, e.g. $J \cup J' = (J_1 \cup J'_1, \ldots, J_s \cup J'_s)$.

Clearly, $L_{J'} \subset L_J$ if $J' \supset J$, and $L_{J*} \cap L_J = L_{J* \cup J}$.

Write $\mathcal{B}_J = \{J' \in \mathcal{J}_s \mid J' \supset J \text{ and } |J' - J| = 1\}$.

We are preparing to use an inclusion-exclusion argument, and $\mathcal{B}$ is the set of J' differing from J only in one component $J'_{i_0} = J_{i_0} \cup \{k\}$ for some k in $H_{i_0} - J_{i_0}$ and some i_0.

Now $\varphi_s^{-1}(J) = L_J - \bigcup_{J' \in \mathcal{B}} L_{J'}$.

By inclusion-exclusion we get

$$P_J = V(L_J) + \sum_{r=1}^{|\mathcal{B}_J|} (-1)^r \sum_{\substack{J^{(i)} \in \mathcal{B} \\ J^{(i)} \text{ distinct}}} V(L_{J^{(1)}} \cap \ldots \cap L_{J^{(r)}})$$

$$= V(L_J) + \sum_{r=1}^{|\mathcal{B}_J|} (-1)^r \sum_{\substack{J^{(i)} \in \mathcal{B}_J \\ J^{(i)} \text{ distinct}}} V(L_{\bigcup_{i=1}^{r} j(i)})$$

Therefore,

$$P_J = \sum_{r=0}^{n-s-|J|} (-1)^r \sum_{\substack{J' \supset J \\ J' \in \mathcal{J}_s \\ |J'-J| = r}} V(L_{J'})$$

By our claim,

$$P_J = \sum_{J' \supset J} (-1)^{|J'-J|} \gamma(\tilde{J}')^m . \qquad \square$$

Lemma 5: We have

$$\sigma_0(m,n,s) = \sum_{\substack{J \in \mathcal{J}_s \\ J' \in J,\ J' \in \mathcal{J}_s}} \gamma(\tilde{J})^m \gamma(\tilde{J}')(-1)^{|J'-J|}$$

Proof: This follows from

$$\sigma_0(m,n,s) = \sum_{J \in \mathcal{J}_s} \gamma(\tilde{J})\, P_J$$

and the previous lemma, by interchanging J, J'. □

We are almost ready to prove the main theorem. We now need two lemmas. The first deals with binomial coefficients:

Lemma 6: $$\sum_{\ell=0}^{j} \binom{j}{\ell} \frac{(-1)^\ell}{\ell+1} = \frac{1}{j+1} .$$

Proof: $$\sum_{\ell=0}^{j} \binom{j}{\ell} \frac{(-1)^\ell}{\ell+1} = \frac{1}{j+1} \sum_{\ell=0}^{j} \binom{j+1}{\ell+1} (-1)^\ell$$

$$= \frac{-1}{j+1} \sum_{k=1}^{j+1} \binom{j+1}{k} (-1)^k = \frac{-1}{j+1} \sum_{j=0}^{j+1} \binom{j+1}{k} (-1)^k + \frac{1}{j+1}$$

$$= \frac{1}{j+1} . \qquad \square$$

In the next lemma, we show how to introduce our slowly growing function.

Lemma 7: For $m \geq 0$, $N > 0$ define $G(m,N) = -\sum_{q=1}^{N} (-1)^q \binom{N}{q} \frac{1}{q^m}$.

Then $0 < G(m,N) \leq (1 + \log N)^m$.

Proof: First of all, let us show

$$G(m,N) = -\int_0^1 \cdots \int_0^1 \frac{[(1 - t_1 t_2 \cdots t_m)^N - 1]}{t_1 \cdots t_m} \, dt_1 \cdots dt_m .$$

Expanding the right hand side by the binomial theorem gives

$$-\sum_{q=1}^{N} \binom{N}{q} (-1)^q \int_0^1 \cdots \int_0^1 (t_1 \cdots t_m)^{q-1} \, dt_1 \cdots dt_m ,$$

and integration gives

$$-\sum_{q=1}^{N} \binom{N}{q} \frac{(-1)^q}{q^m} . \quad \text{Call this claim 1 .}$$

Claim 1: $$-\sum_{q=1}^{N} \binom{N}{q} \frac{(-1)^q}{q^m} = -\int_0^1 \cdots \int_0^1 \frac{[(1 - t_1 \cdots t_m)^N - 1]}{t_1 \cdots t_m} \, dt_1 \cdots dt_m .$$

Claim 2: $G(0,N) = 1 \quad \forall N$, and $G(m,N) = \sum_{k=1}^{N} \frac{1}{k} G(m-1, k)$.

Proof of claim 2: Part 1 is clear. Using $x = 1 - t_1 \cdots t_m$ in claim 1,

write $$G(m,N) = \int_0^1 \cdots \int_0^1 \frac{1 - X^N}{1 - X} \, dt_1 \cdots dt_m = \sum_{j=0}^{N-1} \int_0^1 \cdots \int_0^1 (t_1 \cdots t_m)^j \, dt_1 \cdots dt_m .$$

Integrating w.r.t. t_m gives

$$\sum_{j=0}^{N-1} \int_0^1 \cdots \int_0^1 \frac{(t_1 \cdots t_{m-1})^j}{j+1} \, dt_1 \cdots dt_{m-1} ,$$

and letting $k = j+1$ establishes our claim. □

From claim 2 ,

$$G(1,N) = 1 + \tfrac{1}{2} + \ldots + \frac{1}{N} < 1 + \log N$$

$$G(2,N) = \sum_{k=1}^{N} \frac{1}{k} \sum_{\ell=1}^{k} \frac{1}{\ell} \leq (1 + \log N)^2$$

$$G(3,N) = \sum_{k=1}^{N} \frac{1}{k} \sum_{\ell=1}^{k} \frac{1}{\ell} \sum_{j=1}^{\ell} \frac{1}{j} \leq (1 + \log N)^3 \ldots$$

proving our lemma.

We now can prove the main theorem.

Proof of Main Theorem: By lemma 5 we know that

$$\sigma_0(m,n,s) = \sum_{\substack{J \in \mathcal{J}_s \\ J' \subset J,\ J' \in \mathcal{J}_s}} \gamma(\tilde{J})^{\bullet m}\, \gamma(\tilde{J}')(-1)^{|J-J'|}$$

Write $J = (J_1, \ldots, J_s)$ and $J' = (J_1', \ldots, J_s')$.

Let $\ell_i = |J_i|$; $\ell_i = 0, 1, \ldots, h_i = |H_i|$. Let $\ell_i' = |J_i'|$; $\ell_i' = 0, 1, \ldots, \ell_i$ (As $J_i' \subset J_i \subseteq H_i$).

Then we can write

$$\sigma_0(m,n,s) = \left(\sum_{\ell_1=0}^{h_1} \binom{h_1}{\ell_1} \sum_{\ell_2=0}^{h_2} \binom{h_2}{\ell_2} \cdots \sum_{\ell_s=0}^{h_s} \binom{h_s}{\ell_s} \right) \left((-1)^{\ell_1+\ell_2+\ldots\ell_s-\ell_1'-\ell_2'-\ldots\ell_s'} \right)$$

$$\left(\prod_{i=1}^{s} \frac{1}{1+\ell_i} \right)^m \left(\prod_{i=1}^{s} \frac{1}{1+\ell_i'} \right) \cdot \left(\sum_{\ell_1'=0}^{\ell_1} \binom{\ell_1}{\ell_1'} \cdots \sum_{\ell_5'=0}^{\ell_s} \binom{\ell_s}{\ell_s'} \right)$$

by lemma 2. This can be rewritten as

$$\sum_{\ell_1=0}^{h_1} \binom{h_1}{\ell_2} \frac{(-1)^{\ell_1}}{(1+\ell_1)^m} \cdots \sum_{\ell_s=0}^{h_s} \binom{h_s}{\ell_s} \frac{(-1)^{\ell_s}}{(1+\ell_s)^m}$$

$$\sum_{\ell_1'=0}^{\ell} \binom{\ell_1}{\ell_1'} \frac{(-1)^{\ell_1'}}{(1+\ell_1')} \cdots \sum_{\ell_s'=0}^{\ell_s} \binom{\ell_s}{\ell_s'} \frac{(-1)^{\ell_s'}}{(1+\ell_s')} .$$

If we now use lemma 6, this simplifies to

$$\sigma_0(m,n,s) = \prod_{i=1}^{s} \sum_{\ell_i=0}^{h_i} \binom{h_i}{\ell_i} \frac{(-1)^{\ell_1}}{(1+\ell_i)^{m+1}} ,$$

which gives, letting $k_i = \ell_i + 1$,

$$\sigma_\sigma(m,n,s) = \prod_{i=1}^{s} \frac{-1}{(h_i+1)} \sum_{k_i=1}^{h_i+1} (-1)^{k_i} \binom{h_i+1}{k_i} \frac{1}{k_i^m}$$

which recognize from lemma 6 as

$$\prod_{i=1}^{s} \frac{G(m,\, h_i+1)}{h_i+1} \quad . \tag{1}$$

From claim 1 of lemma 6, $G(m,N)$ increases with N, and as $h_i \leq (n-s)/s+1$, we get

$$G(m,\, h_i+1) \leq G(m,\, (n-s)/s+2) \quad . \tag{2}$$

This is an abuse of notation as $(n-s)/s$ may not be an integer.

Also, $h_i+1 \geq (n-s)/s$, so that

$$\frac{1}{h_i+1} \leq \frac{s}{n-s} \tag{3}$$

Combining (1), (2) and (3), and using lemma 6, we find that

$$\sigma_0(m,n,s) \leq \prod_{i=1}^{s} \left(\frac{s}{n-s}\right) \left(1+\log\left(\frac{n-s}{s}+2\right)\right)^m .$$

$$= \left(\frac{s}{n-s}\right)^s \left(1+\log\left(\frac{n-s}{s}+2\right)\right)^{ms} .$$

It remains to substitute this in the first main estimate and simplify:

$$\rho(m,n) \leq C_m \sum_{s=0}^{m+1} \binom{n}{s} \sigma(m,n,s)$$

$$= C_m \left(1 + \sum_{s=1}^{m+1} \binom{n}{s} \sigma(m,n,s)\right)$$

$$\leq C_m \left(1 + \sum_{s=1}^{m+1} \binom{n}{s} \sigma_0(m,n,s)\right)$$

$$\leq C_m \left(1 + \sum_{s=1}^{m+1} \binom{n}{s} \left(\frac{s}{n-s}\right)^s \left(1 + \log\left(\frac{n-s}{s} + 2\right)\right)^{ms}\right)$$

$$\leq C_m \left(1 + (1+\log(n+1))^{m(m+1)} \sum_{s=1}^{m+1} \binom{n}{s} \left(\frac{s}{n-s}\right)^s\right) .$$

Now $\max_{s=1,\ldots,m+1} \frac{s^s}{s!}$ is a constant depending on m. Thus

$$\sum_{s=1}^{m+1} \frac{n!}{s!(n-s)!} \cdot \frac{s^s}{(n-s)^s} \leq K_m \sum_{s=1}^{m+1} \frac{n(n-1)\ldots(n-s+1)}{(n-s)(n-s)\ldots(n-s)}$$

$$\leq K_m \sum_{s=1}^{m+1} \left(\frac{n}{n-s}\right)^s , \quad \text{where } K_m \text{ is a constant depending on } m .$$

Suppose $n > 2m$. Then $\frac{n}{n-s} \leq 2$, and $K_m \sum_{s=1}^{m+1} \left(\frac{n}{n-s}\right)^s \leq K_m \sum_{s=1}^{m+1} 2^s$, a constant depending on m. We have thus shown that

$$\rho(m,n) \leq C_m + K_m (1 + \log(n+1))^{m(m+1)} .$$

Thus $\rho(m,n)$ grows with n more slowly than any positive power of n.

Concluding Remarks

Having understood Smale's proof, we must now ask "What has Smale really proved?" Lemke's algorithm assumes non-degeneracy of (1) and (5), so that no vertex is visited twice, and this is crucial to Smale's proof. Otherwise the main formula does not follow. We cannot merely claim that the proof is general by perturbing (1), (5): This would take $O(n+m)$ operations, dominating the complexity we wish to show! One may claim that degenerate cases are rare, have measure zero, and thus do not effect our average, but then we do not show what we have set out to show: the Simplex method is empirically seen to solve LP's, most of which are in practice degenerate, in time logarithmic in n, linear in m. We still do not know why this is so. Smale shows that for "random" programs (which will be non-degenerate usually), the complexity seen will be expected.

At least one attempt to "patch up" Smale has been made by Blair, who seems hostile to Smale's geometric approach. By using dominance, Blair gets a result close to Smale's. However, Blair assumes $n \gg m$. That is, Blair really shows a "big oh" complexity. Smale only needs $n > 2m$. In effect $\rho(m,n)$ converges to Smale's estimate most immediately, while Blair takes limits. Also, Blair's estimate is weaker than Smale's. As Blair admits, the main importance of his paper is to bring dominance to a central role.

Smale's exact formula for the number of pivots of Lemke's algorithm is very satisfying: the sum of a volume of cones. According to Smale, these volumes are difficult to find.

A final question is "Is Smale's method applicable in other complexity problems?" Of course, "Smale's method" can refer to various parts of Smale's analysis. In an important lemma, the structure of M is used explicitly in finding a determinant, to prove the first main estimate. Since the derivation of the first main estimate uses the structure of M, we cannot immediately transfer Smale's analysis to the complexity of the linear complementarity problem: however, the First Main Formula does not use the form of M and <u>can</u> be directly used for bimatrix games or quadratic programming, where Lemke's algorithm is also used.

Perhaps, however, using our analysis of dominance more sharply, the First Main Estimate can be proved, bypassing the Main Formula. This would generalize Smale's result to a whole class of Simplex methods.

Finally, we can speak of Smale's method in a very broad sense: Solve $f(x) = a$ by following a path $f^{-1}(aa_0)$ where $f(x_0) = a_0$ is our first approximation. Smale has also used this approach to Newton's method in numerical analysis [8]. He suggests applying it to solving piece-wise linear equations on R^N.

In summary, Smale's paper gives an interesting and pleasing result about the Simplex methods. However, more insight into degeneracy will be required to explain the empirical speed of Simplex. Smale's method may generalize to other Simplex problems via dominance. His approach should certainly be considered when attacking the complexity of some path-finding algorithm for $f(x) = A$.

Bibliography

[1] C. Blair, "Random Linear Programs with Many Variables and Few Constraints", In proof.

[2] Cottle, R. and Dantzig, G. 1968, "Complementary Pivot Theory of Mathematical Programming", Math. of the Decision Sciences (G. Dantzig and A. Veinott, Jr. eds.), Amer. Math. Soc. Providence, R.I., pp. 115-136.

[3] V. Cvatal, 1983. Linear Programming, W.H. Freeman and Co., New York.

[4] J. Franklin, 1980. Methods of Mathematical Economics, Springer-Verlag, New York.

[5] C. Lemke, 1968. "On Complementary Pivot Theory", Math. of the Decision Sciences.

[6] S. Smale, 1982, "On the average number of steps of the Simplex method of linear programming", Mathematical Programming (27), North-Holland, New York.

[7] S. Smale, 1983, "The Problem of the Average Speed of the Simplex Method", Math. Programming - The State of the Art, Ed. Backen et al., Springer-Verlag, New York.

[8] S. Smale, 1976, "A Convergent Method of Price Adjustment and Global Newton Methods", J. Math. Economics, 3, pp. 107-120.

Figure 1

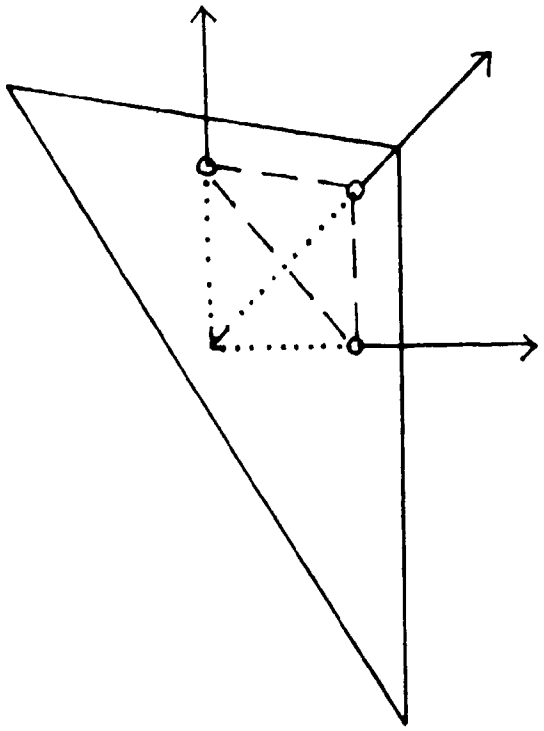


Figure 2

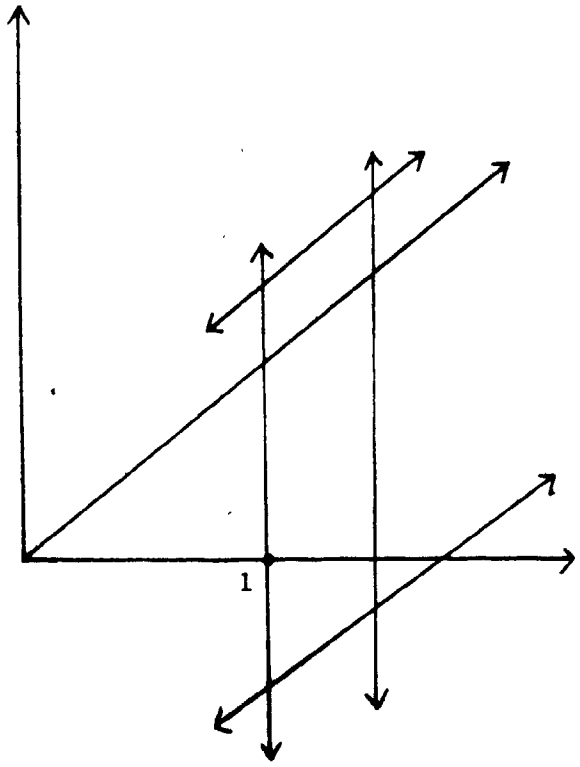


Figure 3

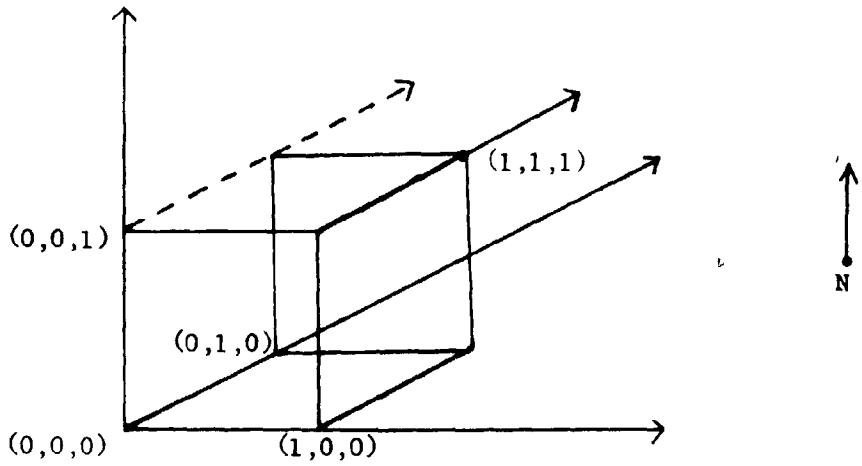


Figure 4

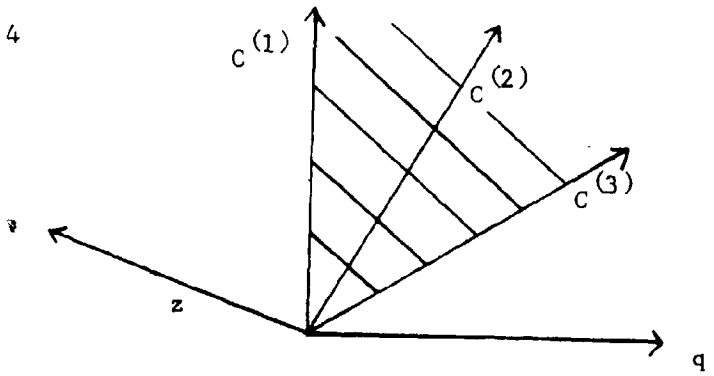


Figure 5

Figure 6

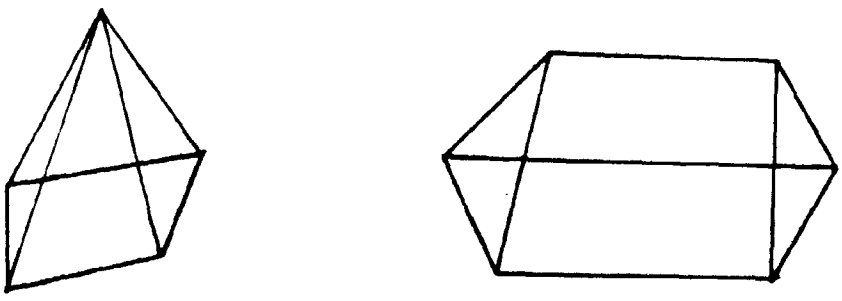


Figure 7

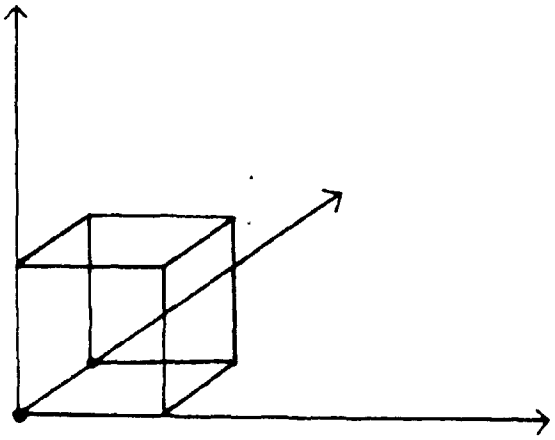


Figure 8

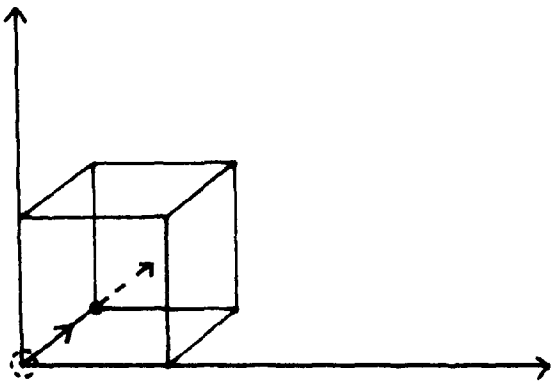


Figure 9

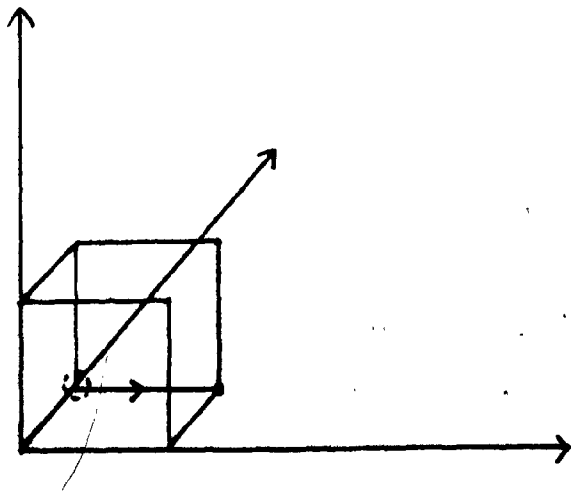
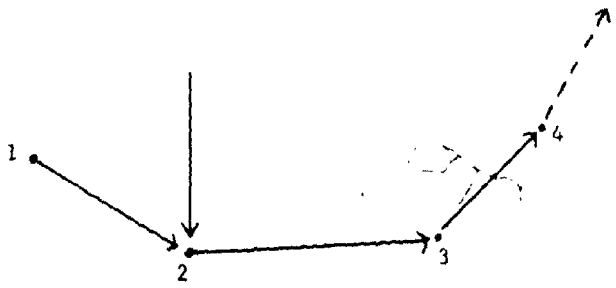


Figure 10



If 2 repeats before 1, then 2 has three a-c edges.

Figure 11

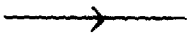


Figure 12

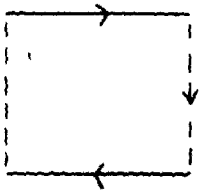


Figure 13

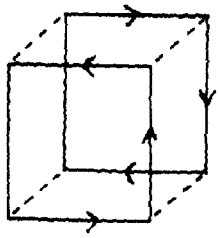


Figure 14

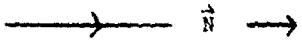


Figure 15

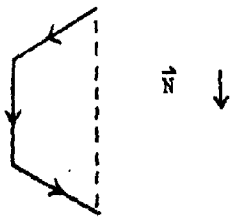


Figure 16

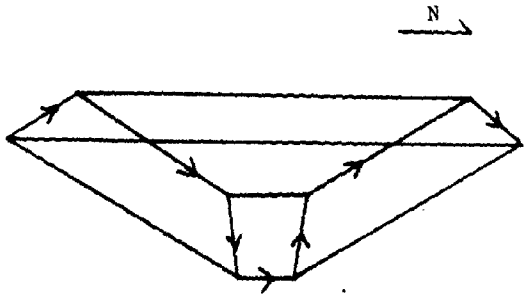


Figure 17