SIAM J. Appl. Math.
Vol. 15, No. 5, September, 1967
Printed in U.S.A.

THE APPROXIMATION OF FIXED POINTS OF A CONTINUOUS MAPPING*

HERBERT SCARF†

1. Introduction. Brouwer's fixed-point theorem states that a continuous mapping of a simplex into itself has at least one fixed point. This paper describes a numerical algorithm for approximating, in a sense to be explained below, a fixed point of such a mapping.

Let S be the simplex $\{\pi \mid \sum_{i=1}^{n} \pi_i = 1, \pi_i \geqq 0\}$. A continuous mapping of the simplex into itself is given by a collection of n functions $f_1(\pi), \cdots, f_n(\pi)$, continuous for all $\pi \in S$, and having the properties: $\sum_{i=1}^{n} f_i(\pi) = 1$, and $f_i(\pi) \geqq 0$. Brouwer's theorem states that there exists a $\hat{\pi} \in S$ such that $f(\hat{\pi}) = \hat{\pi}$.

The theorem may be demonstrated by means of a combinatorial result known as Sperner's lemma [1], which it will be useful to review. Let $\pi^1, \cdots, \pi^k$ be a sequence of distinct points selected arbitrarily on the simplex S. By connecting π^1 to each of the n vertices of S we partition S into n subsimplices (see Fig. 1). We then connect π^2 to the n vertices of each subsimplex to which it belongs, and continue the successive refinement with $\pi^3, \cdots, \pi^k$. The result is a particular type of partition of S into a number of subsimplices, whose maximum diameter can be made arbitrarily small by a suitable selection of the sequence $\pi^1, \cdots, \pi^k$.

We associate with each vertex π an index i such that $\pi_i > 0$ and $f_i(\pi) \leqq \pi_i$. There clearly will be at least one such index for each vertex and if there are several we make an arbitrary choice among them. Sperner's lemma then states that at least one subsimplex of the partition has all of its vertices indexed differently. In other words, a subsimplex may be found so that at each of the n vertices a different coordinate is *not* increased by means of the mapping f.

As vertices are added the partitions become more refined, and the vertices may be selected in such a way that the maximum diameter of the subsimplices appearing in the partitions tends to zero. Each partition contains a subsimplex all of whose vertices are labeled differently, and a subsequence may be found whose vertices converge to a single point $\hat{\pi}$. Since the mapping is continuous, $f_i(\hat{\pi}) \leqq \hat{\pi}_i$ for all i, and therefore $\hat{\pi}$ is a fixed point of the mapping.

We can think of approximating $\hat{\pi}$ numerically in two distinct ways. The first is to attempt to determine a region of small diameter in which $\hat{\pi}$ must

* Received by the editors March 1, 1967.

† Department of Economics, Yale University, New Haven, Connecticut 06520.
This research was supported by a grant from the National Science Foundation.

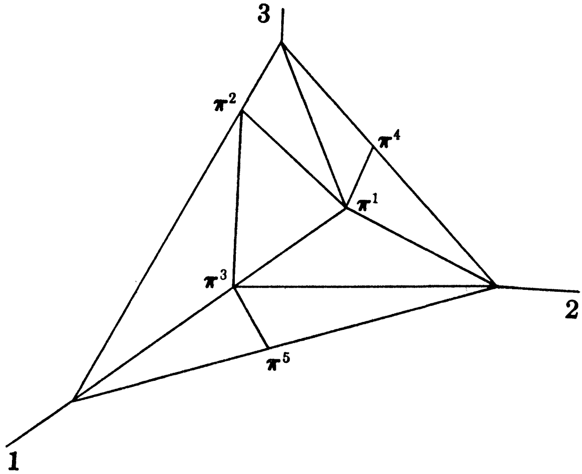

FIG. 1

necessarily lie. This approach requires us to anticipate the limit points of a sequence from a finite amount of data and is nonconstructive for general mappings.

An alternative approach is to determine, for arbitrary ϵ, a point π whose image is at a distance less than ϵ from itself. Sperner's lemma may be used to approximate a fixed point of f in this sense. Since f is continuous, for a given $\epsilon > 0$ there is a δ such that $|f(\pi') - f(\pi'')| \leqq \epsilon$ whenever $|\pi' - \pi''| < \delta$, where the norm $|x|$ is taken, to be specific, as $\max(|x_1|, \cdots, |x_n|)$. If the maximum diameter of the subsimplices in the partition is δ, then any point π in a subsimplex whose vertices are labeled differently will satisfy $|f(\pi) - \pi| \leqq (n-1)(\epsilon + \delta)$, and will therefore serve as an approximate fixed point in this sense.

There is a very serious practical difficulty, however, in this approach. The number of vertices required to determine a partition of small diameter is enormous even for moderate values of n. For example, if n is 7 and if the vertices are selected as the lattice points $(k_1/D, \cdots, k_n/D)$, with k_i nonnegative integers satisfying $\sum_1^n k_i = D$, then some 80 billion vertices are required for $D = 200$, and the number of subsimplices in the partition is of course larger. Moreover, Sperner's lemma suggests no procedure for the determination of an approximate fixed point other than an exhaustive search of all subsimplices until one is found with all vertices labeled differently. Clearly some substitute for an exhaustive search must be found if the problem is to be considered tractable, and the current proofs of Sperner's lemma offer no suggestion in this direction.

In this paper a new combinatorial theorem will be described which may also be used to demonstrate the Brouwer fixed-point theorem. This theorem involves, as does Sperner's lemma, the selection of a fine grid of points on the simplex S, but it differs from Sperner's lemma in that a systematic algorithm is used to determine the sequence of points to be examined. The algorithm has been applied to a number of examples and seems to work remarkably well. The computational experience, which is discussed in §§5 and 6, suggests that the algorithm is quite practical for the approximation of fixed points of certain mappings, when n is less than 15 or 20.

Section 7 discusses the generalization of this algorithm to continuous mappings of a closed, bounded, convex polyhedron into itself. It is somewhat more complex than the rest of the paper, and can be avoided by the casual reader.

Though it may not be apparent from the arguments of this paper, the algorithm is intimately related to the procedure described by Lemke [2] for the determination of Nash equilibrium points of two-person nonzero-sum games.

2. A combinatorial theorem. We consider a finite set P_k of vectors $\pi^1, \cdots, \pi^n, \cdots, \pi^k$ in n-dimensional space. The vectors $\pi^{n+1}, \cdots, \pi^k$ are selected arbitrarily on the simplex $S = \{\pi \mid \sum_i \pi_i = 1, \pi_i \geqq 0\}$. The first n vectors, which are not on the simplex, have the following specific form:

$$\begin{aligned}
\pi^1 &= (0, M_1, \cdots, M_1),\\
\pi^2 &= (M_2, 0, \cdots, M_2),\\
&\vdots\\
\pi^n &= (M_n, M_n, \cdots, 0),
\end{aligned}$$

with the M_i satisfying $M_1 > M_2 > \cdots > M_n > 1$.

DEFINITION. A set of n vectors $\pi^{j_1}, \cdots, \pi^{j_n}$ in P_k will be called a *primitive* set if there are no vectors π^j in P_k with

$$\begin{aligned}
\pi_1^j &> \min(\pi_1^{j_1}, \cdots, \pi_1^{j_n}),\\
&\vdots\\
\pi_n^j &> \min(\pi_n^{j_1}, \cdots, \pi_n^{j_n}).
\end{aligned}$$

There is a simple geometric interpretation of a primitive set.[1] Let $\pi^{j_1}, \cdots, \pi^{j_n}$ be a set of n vectors in P_k and consider the subsimplex of S defined by

$$\pi_i \geqq \min(\pi_i^{j_1}, \cdots, \pi_i^{j_n}) \quad \text{for} \quad i = 1, \cdots, n,$$

[1] In [3] the term "ordinal basis" was used for a primitive set of vectors, in order to suggest a connection with the use of "basis" in linear programming.

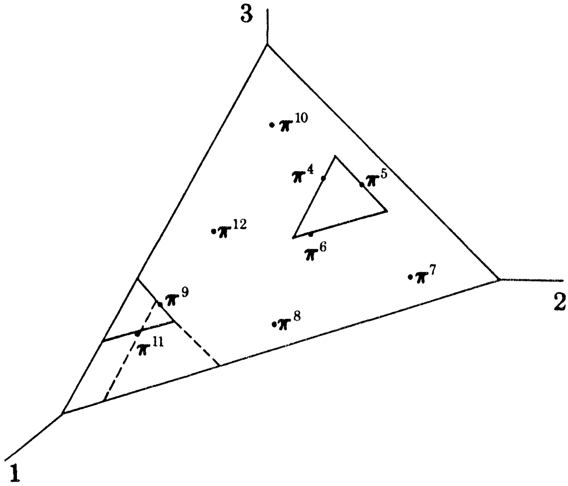

FIG. 2

and $\sum \pi_i = 1$. If the subsimplex contains no vectors of P_k in its *interior*, then the n vectors $\pi^{j_1}, \cdots, \pi^{j_n}$ form a primitive set. It will be useful to refer to such a subsimplex as a *primitive subsimplex*.

In Fig. 2 the vectors π^4, π^5 and π^6 form a primitive set, since no vector π^j in P_k is interior to the small subsimplex in the figure which contains π^4, π^5 and π^6.

As Fig. 2 illustrates, π^2, π^9 and π^{11} also form a primitive set, since no vector in P_k is interior to the subsimplex generated by π^9, π^{11}, and the edge of S in which the second coordinate is zero.

It will be convenient to make the following assumption which can easily be brought about by a perturbation of the vectors in P_k.

NONDEGENERACY ASSUMPTION. No two vectors in P_k have the same ith coordinate for any i.

With this assumption, a primitive subsimplex will have each of its n bounding faces parallel to one of the coordinate hyperplanes, and each face will contain precisely one vector in the primitive set, namely, that vector in which the corresponding coordinate is smallest. If the primitive set contains π^i with $i \leqq n$, then the primitive subsimplex contains that face of S with the ith coordinate equal to zero.

In our applications each vector in P_k will have associated with it an index selected from the integers between 1 and n. The index associated with a vector is arbitrary except for the first n vectors in the list. We shall require that π^1 have the index 1, π^2 the index 2, etc. The combinatorial theorem may now be stated.

THEOREM 1. *There exists a primitive set, all of whose vectors are indexed differently.*

When Theorem 1 is applied to Brouwer's theorem, each vector π^j, other than the first n vectors, is given an index i for which $f_i(\pi^j) \geqq \pi_i^j$. A primitive set of the type referred to in Theorem 1 will contain some vectors from the first n, say $\{\pi^j\}$ with j in an index set I, and some from the remaining vectors in P_k. The primitive subsimplex associated with this primitive set will be bounded by an edge $\pi_i = 0$ for each $i \in I$, and by an edge passing through each remaining vector in the primitive set. The latter vectors will have an index not in I, so that for every i there is some vector in this subsimplex for which $f_i(\pi) \geqq \pi_i$.

An appropriate sequence of vectors may be selected, so that as k tends to infinity the maximum diameter of a primitive subsimplex tends to zero, since no vectors in P_k are interior to such a subsimplex. Therefore, a sequence of primitive subsimplices may be found which converge to a single vector $\hat{\pi}$. Using the continuity of f we see that $f_i(\hat{\pi}) \geqq \hat{\pi}_i$ for all i, so that $\hat{\pi}$ is a fixed point of the mapping.

3. A preliminary lemma. The following lemma is the main tool in our algorithm.

LEMMA 1. *Let $\pi^{j_1}, \cdots, \pi^{j_n}$ be a primitive set, and let π^{j_α} be a specific one of these vectors. Then, aside from one exceptional case, there is a unique vector $\pi^j \in P_k$, different from π^{j_α}, and such that $(\pi^{j_1}, \cdots, \pi^{j_{\alpha-1}}, \pi^j, \pi^{j_{\alpha+1}}, \cdots, \pi^{j_n})$ form a primitive set. The exceptional case occurs when the $n - 1$ vectors π^{j_i}, with $i \neq \alpha$, are all selected from the first n vectors of P_k, and in this case no replacement is possible.*

The lemma states that aside from the exceptional case, if an arbitrary vector is removed from a primitive set, there is a unique replacement so that the new set of vectors is a primitive set. The new vector π^j which replaces π^{j_α} may be found by a simple geometric construction. To illustrate this construction let us assume that $\pi_i^{j_i} = \min(\pi_i^{j_1}, \cdots, \pi_i^{j_n})$ so that π^{j_i} is on that face of the primitive subsimplex on which the ith coordinate is constant (see Fig. 3). Assume, moreover, that π^{j_1} is being removed.

Let $\pi^{j_{i^*}}$ be that vector in the primitive set with the *second* smallest value of its first coordinate. j_{i^*} will be greater than n unless the exceptional case arises. To find the vector to replace π^{j_1} we move the face containing $\pi^{j_{i^*}}$ parallel to itself, lowering the i^*th coordinate until we *first* intersect a vector π^j in P_k with

$$\pi_i^j > \pi_i^{j_i} \quad \text{for} \quad i \neq 1, i^*,$$

and

$$\pi_1^j > \pi_1^{j_{i^*}},$$

or the face of the simplex S in which $\pi_{i^*} = 0$.

The rule is applicable except when the vectors π^{j_i} with $i \neq 1$ are all selected from the first n vectors of P_k, and it clearly produces a new primitive set.

In order to finish the proof of Lemma 1 we must ask whether there is any vector π^l other than π^j which yields a primitive set when it replaces π^{j_1}.

Observation 1. If $(\pi^l, \pi^{j_2}, \cdots, \pi^{j_n})$ forms a primitive set, then for $i \neq 1, i^*$, we must have π^{j_i} on that bounding face of the new primitive subsimplex whose ith coordinate is constant.

If this were not the case for some such i, then π^{j_i} would be on *none* of the bounding faces of the new primitive subsimplex and this is impossible.

As a consequence of this observation we see that the new primitive set satisfies

$$\pi_i^{j_i} = \min (\pi_i^l, \pi_i^{j_2}, \cdots, \pi_i^{j_n}) \quad \text{for} \quad i \neq 1, i^*.$$

There are two alternatives to be considered for the remaining two coordinates. Either π^l is on that face with constant first coordinate and π^{j_i*} on that with constant i^*th coordinate, or vice versa.

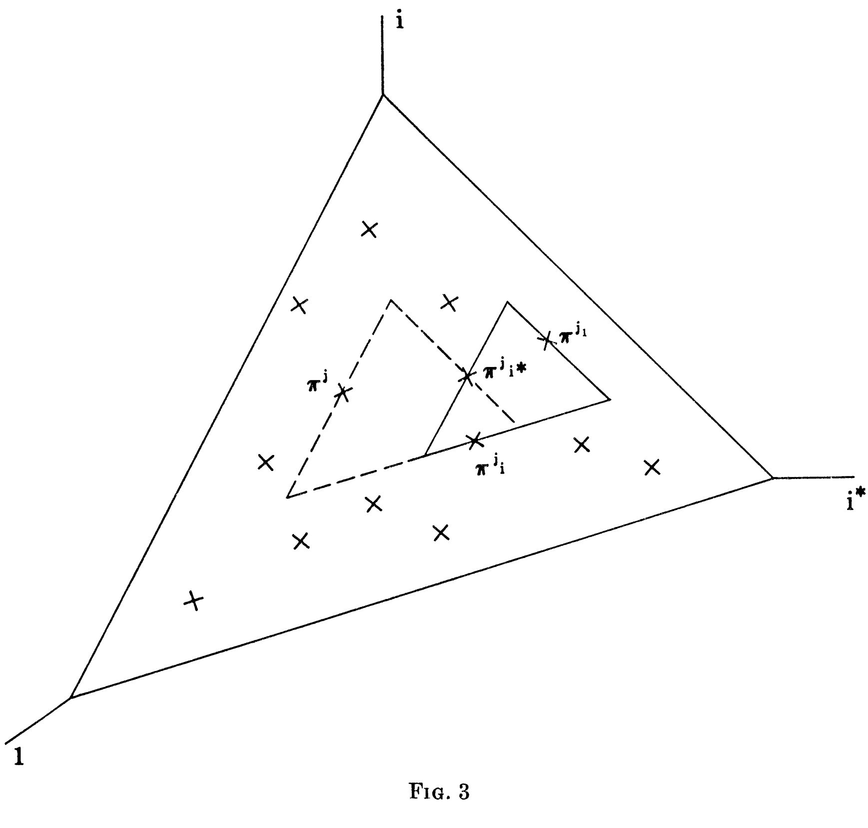

FIG. 3

Observation 2. If $(\pi^l, \pi^{j_2}, \cdots, \pi^{j_n})$ forms a primitive set and $\pi^l \neq \pi^{j_1}$, then π^l must be on the face of the new primitive subsimplex whose i^*th coordinate is constant, and $\pi^{j_{i^*}}$ on that face with constant first coordinate.

If this were not correct, then the new subsimplex would have π^{j_i} on that face on which the ith coordinate is constant for $i = 2, \cdots, n$. But then if $\pi_1^{j_1} < \pi_1^l$, the old subsimplex contains π^l in its interior, whereas if $\pi_1^{j_1} > \pi_1^l$ the new subsimplex contains π^{j_1} in its interior. It follows that $\pi_1^l = \pi_1^{j_1}$ and since no two different vectors have the same first coordinate we must have $l = j_1$, and we are back where we started.

Observation 3. If $(\pi^l, \pi^{j_2}, \cdots, \pi^{j_n})$ forms a primitive set and $\pi^l \neq \pi^{j_1}$, then π^l must be that vector π^j described above.

This follows since we have already shown that π^{j_i} is on that face of the new subsimplex with constant ith coordinate for $i \neq 1, i^*$ and that $\pi^{j_{i^*}}$ on the face with constant first coordinate.

The reader may easily finish the proof of Lemma 1, by demonstrating that no replacement is possible in the exceptional use.

4. The algorithm for Theorem 1. We recall that each vector in P_k has associated with it an index selected from the first n integers. In the application of Theorem 1 to Brouwer's theorem the indices depend on the particular mapping, but for the present the assignment of integers is arbitrary aside from the assumption that for $j = 1, \cdots, n$, π^j is associated with the index j.

Our purpose is to determine a primitive set all of whose members are indexed differently. The algorithm will begin with a primitive set whose members are indexed differently with the possible exception of one pair of vectors with the same index. Consider the set of vectors $(\pi^2, \cdots, \pi^n, \pi^{j^*})$ with π^{j^*} selected from those vectors beyond the first n so as to maximize the *first* coordinate. Clearly,

$$\min (\pi_i^{j^*}, \pi_i^2, \cdots, \pi_i^n)$$

is given by $\pi_i^{j^*}$ for $i = 1$, and zero for $i > 1$, and this set of vectors is primitive since no vector in P_k can have all of its coordinates strictly larger than those of $(\pi_1^{j^*}, 0, \cdots, 0)$.

If the vector π^{j^*} were associated with the index 1, then the problem would be over since all members of this primitive set would have a different index. Generally this will not be the case and π^{j^*} will share an index with one of the vectors $\pi^2, \cdots, \pi^n$. Our algorithm will always be involved with primitive sets of this type. In other words at *each* step of the algorithm we will have a primitive set whose indices have the following properties:

(i) The index 1 will not be associated with any vector.

(ii) All vectors in the primitive set will be indexed differently, except for one pair of vectors with the same index.

The algorithm proceeds by taking one of the two vectors with the same index and removing it from the primitive set, either obtaining another primitive set with the same properties or else terminating with a solution. If we are not at the initial primitive set *one* of the two vectors with a common index will have just been introduced in order to arrive at the current position. The algorithm proceeds by eliminating the other member of the pair.

In other words, at each stage of the algorithm after the first, there are two possible removals that will take us to a primitive set with the same properties. One of these steps has been taken to get to the current position. We therefore take the other step. There is only one vector which can be removed from the *initial* primitive set, namely, that vector π^j (with $2 \leqq j \leqq n$) with the same index as π^{j^*}. The other possibility, that of removing π^{j^*}, is the exceptional case referred to in Lemma 1.

The algorithm can only terminate when a primitive set is found, all of whose vectors are indexed differently. It should be clear that the algorithm can never return to a previous primitive set, for if the first return is made to a primitive set other than the initial one, then there would be *three*, rather than two, ways to emerge from that particular primitive set. On the other hand, if the first return is to the initial primitive set, there would be *two* ways of emerging from the initial set.

Since there are a finite number of primitive sets, the algorithm must terminate in a finite number of steps with all vectors indexed differently. This demonstrates Theorem 1.

5. Some computational techniques. The algorithm has been programmed on an IBM 7094, and several examples have been tried. Before describing the results of the computations, it might be useful to indicate a few of the special techniques that have been incorporated into the program.

The first problem encountered in programming the algorithm is that of selecting an appropriate set of vectors P_k. Each state of the algorithm involves a primitive set of n of these vectors. A specific one of these vectors is eliminated from the primitive set and its replacement found by calculating a vector a and a specific coordinate i^*, examining all vectors in P_k with $\pi_i^j > a_i$ for $i \neq i^*$ and selecting that vector with the largest value of $\pi_{i^*}^j$.

It is clearly quite useful to construct P_k so that the selection of the new vector can be done *without* an exhaustive search of all of the vectors in P_k. For example, if P_k consists (aside from its first n members) of all vectors $(k_1/D, \cdots, k_n/D)$ with k_i positive integers satisfying $k_1 + \cdots + k_n = D$, then each a_i will be an integer divided by D, and the new vector π^j will either be given by $\pi_i^j = a_i + (1/D)$ for $i \neq i^*$ and $\pi_{i^*}^j = 1 - \sum_{i \neq i^*} (a_i + (1/D))$, or else be one of the first n members of P_k.

If P_k has this special structure, the selection of the new vector may therefore be done by a simple computation, rather than a search over an enormous number of vectors. On the other hand, this choice of P_k does not satisfy the assumption made in §3 that no two vectors in P_k have the same ith coordinate for any i, an assumption which is indispensable for the application of the rule given in Lemma 1. In order to avoid this difficulty some systematic procedure for resolving ties between two vectors must be used. The particular procedure used by this author is to construct at each step in the algorithm a matrix

$$\begin{bmatrix} 0 & \cdots & M_n & \pi_1^{n+1} & \cdots & \pi_1^{l} \\ M_1 & & & & & \\ \vdots & & \vdots & \vdots & & \vdots \\ M_1 & & 0 & \pi_n^{n+1} & & \pi_n^{l} \end{bmatrix}$$

consisting of the first n vectors of P_k and all other members of P_k which have previously been introduced into a primitive set, in the order in which they have been introduced. Then, if two columns in this matrix have identical elements in the ith row, the first is assumed to be larger, and if a vector *in* the matrix has an identical entry in the ith row with some vector *not* in the matrix, the former is assumed to be larger. It may be demonstrated that this procedure for resolving ties also leads to a finite algorithm.

In the determination of π^j a search is then made only over those vectors which have been used in some previous step; the remaining vectors in P_k are examined by a single algebraic calculation. The number of vectors to be examined explicitly can be no larger than the number of iterations plus n, and if the number of iterations is relatively small this search is quite manageable. There are, of course, other ways to resolve ties which surely involve even less computation, and which will be introduced in subsequent versions of the program.

The algorithm terminates with a primitive set all of whose vectors are differently indexed and any point in the geometric subsimplex of S corresponding to this primitive set will serve as an approximate fixed point. In order to select a unique point, it is assumed that the functions $f_i(\pi)$ are linear in a region around this subsimplex, and a point is selected which minimizes the maximum of $(f_1(\pi) - \pi_1, \cdots, f_n(\pi) - \pi_n)$, or some other measure of closeness. On the basis of computational experience, this seems to be a very useful way of terminating the algorithm.

6. An example from economics. The particular examples of Brouwer's theorem that will be described arise from an important problem in mathematical economics, that of determining equilibrium prices in a general economic model of exchange. Fixed-point theorems have been invoked by many authors to demonstrate the existence of equilibrium prices but have never been used for the purpose of explicit calculation.

Let n be the number of commodities in the economy and m the total number of economic agents. The lth agent is assumed to respond to a vector of nonnegative prices $\pi = (\pi_1, \cdots, \pi_n)$ by a vector of *excess* demands

$$g_1^l(\pi), \cdots, g_n^l(\pi) \quad \text{for the} \quad n \quad \text{commodities.}$$

More explicitly, the function $g_i^l(\pi)$ represents the net increase in commodity i desired by the lth agent at prices π. If $g_i^l(\pi) < 0$, the lth agent wishes to decrease his holdings of commodity i and to use the proceeds for the purchase of those commodities with positive excess demand. The following assumptions are customarily made about excess demand functions:

1. Each $g_i^l(\pi)$ is homogeneous of degree 0, an assumption implying that demands are determined by relative rather than absolute levels of prices. This permits us to restrict our attention to prices on the simplex $S = \{\pi \mid \sum \pi_i = 1, \pi_i \geqq 0\}$.

2. For each individual l we have $\pi_1 g_1^l(\pi) + \cdots + \pi_n g_n^l(\pi) \equiv 0$, or, in other words, purchases of commodities with positive excess demands are financed exclusively by the sale of commodities with negative excess demands.

3. Each excess demand function is continuous on the simplex S.

For each commodity i we define

$$g_i(\pi) = \sum_{l=1}^{m} g_i^l(\pi)$$

to be the market excess demand for that commodity.

A vector of prices is in equilibrium if at these prices the market excess demand for each commodity is less than or equal to zero, and actually equal to zero if the price associated with that commodity is strictly positive. It is a simple matter to demonstrate, by means of Brouwer's theorem, that an economic model satisfying the above assumptions does have at least one equilibrium price vector.

The mapping used in Brouwer's theorem is defined, for prices on the simplex S, by

$$f_i(\pi) = \frac{\pi_i + \lambda \max(0, g_i(\pi))}{1 + \lambda \sum_k \max(0, g_k(\pi))}$$

with λ a small positive constant. The mapping is clearly continuous and takes the simplex into itself, so that Brouwer's theorem is applicable. Let $\hat{\pi}$ be a fixed point. Suppose, first of all, that $\sum_k \max(0, g_k(\hat{\pi})) > 0$. Then $\hat{\pi}_i + \lambda \max(0, g_i(\hat{\pi})) = C\hat{\pi}_i$ with $C > 1$, and it follows that $g_i(\hat{\pi}) > 0$ for every i with $\hat{\pi}_i > 0$. Since this violates the assumption that $\hat{\pi}_1 g_1(\hat{\pi}) + \cdots + \hat{\pi}_n g_n(\hat{\pi}) = 0$, we may conclude that $\sum_k \max(0, g_k(\hat{\pi})) = 0$

and therefore $g_i(\hat{\pi}) \leqq 0$ for each i. Again appealing to $\hat{\pi}_1 g_1(\hat{\pi}) + \cdots = 0$, we conclude that $g_i(\hat{\pi}) = 0$ if $\hat{\pi}_i > 0$, so that a fixed point of this mapping does indeed yield an equilibrium price vector.

In the application of our algorithm a vector π will be labeled with an index i for which $f_i(\pi) \geqq \pi_i$, or

$$\max(0, g_i(\pi)) \geqq \pi_i \sum_k \max(0, g_k(\pi)).$$

It will clearly be sufficient to select an index i which maximizes $g_i(\pi)/\pi_i$.

In order to proceed with the algorithm we need to specify the individual excess demand functions $g_i^l(\pi)$. The following will be selected from the many that have been described in the economic literature. Let $W = (w_{lk})$ and $A = (a_{lk})$ be two strictly positive matrices with m rows (one for each agent) and n columns (one for each commodity). Also let $b_1, \cdots, b_m$ be a strictly positive vector. We define $g_i^l(\pi)$ as

$$\frac{a_{li} \sum_k w_{lk} \pi_k}{\pi_i^{b_l} \sum_k a_{lk} \pi_k^{1-b_l}} - w_{li}.$$

Aside from a possible discontinuity on the boundary of the simplex, the assumptions previously made are satisfied for these excess demand functions, and the algorithm may be applied. For those readers who are curious about economics, these excess demands arise from a model of exchange in which the lth individual initially owns w_{lk} units of the kth commodity, and has a utility function given by

$$u_l(x) = \left(\sum_k (a_{lk})^{1-a_l} x_k^{a_l}\right)^{1/a_l}$$

with $b_l = 1/(1 - a_l)$. Other readers may find it sufficient that we are studying a class of continuous mappings which are highly nonlinear, and to which simple gradient methods do not apply [4]. Let us consider the following examples.

Example 1. In this example the number of commodities is five and the number of economic agents is three. The parameters of the excess demand functions are given by

$$W = \begin{bmatrix} 1. & 3. & 10. & 1. & 2. \\ .1 & 2. & 20. & 5. & 6. \\ 1.5 & 5. & 15. & 5. & 10.8 \end{bmatrix},$$

$$A = \begin{bmatrix} 2. & 1. & .8 & 1.5 & 1. \\ 3. & .5 & 1.2 & 1.6 & 1.8 \\ .9 & .8 & 2. & 1. & 1.8 \end{bmatrix}, \quad b = \begin{bmatrix} .9 \\ 1.3 \\ .8 \end{bmatrix}.$$

The set P_k aside from its first five members consists of all vectors $(k_1/160, \cdots, k_5/160)$ with k_i positive integers summing to 160. There are some $.26 \times 10^8$ such vectors. The algorithm terminated after only 158 iterations, with the following primitive set:

$$\begin{array}{ccccc} \pi^{j_1} & \pi^{j_2} & \pi^{j_3} & \pi^{j_4} & \pi^{j_5} \end{array}$$
$$\begin{bmatrix} 101 & 102 & 103 & 102 & 103 \\ 13 & 12 & 13 & 13 & 12 \\ 6 & 6 & 6 & 6 & 6 \\ 25 & 25 & 25 & 24 & 25 \\ 15 & 15 & 14 & 15 & 14 \end{bmatrix},$$

where the components of each vector add to 160 rather than 1.

When these five vectors are averaged according to a linear programming problem which treats the excess demands as locally linear, the following price vector is obtained:

$$\hat{\pi} = (104.9, \quad 12.3, \quad 5.2, \quad 23.6, \quad 14.1),$$

and the market excess demands are given by

$$(g_i(\hat{\pi})) = (.02, \quad -.02, \quad -.27, \quad -.01, \quad -.00).$$

The image of $\hat{\pi}$ under the mapping is given by

$$\pi_i' = \frac{\hat{\pi}_i + \lambda \max(0, g_i(\hat{\pi}))}{1 + \lambda \sum_k \max(0, g_k(\hat{\pi}))},$$

after the prices have been divided by 160. The degree of approximation of the mapping depends on the choice of λ, but the excess demands are a very small fraction of total supply (the column sums of W), and this is the relevant consideration.

Example 2.

$$W = \begin{bmatrix} 3. & 1. & .1 & .1 & 5. & .1 & .1 & 6. \\ .1 & 10. & .1 & .1 & 5. & .1 & .1 & .1 \\ .1 & 9. & 10. & .1 & 4. & .1 & 7. & .1 \\ .1 & .1 & .1 & 10. & .1 & 3. & .1 & .1 \\ .1 & .1 & .1 & .1 & .1 & .1 & .1 & 11. \end{bmatrix},$$

$$A = \begin{bmatrix} 1. & 1. & 1. & 1. & 1. & 1. & 1. & 1. \\ 2. & .8 & 1. & .5 & 1. & 1. & 1. & 1. \\ 1. & 1.2 & .8 & 1.2 & 1.6 & 2. & .6 & .1 \\ 2. & .1 & .6 & 2. & 1. & 1. & 1. & 2. \\ 1.2 & 1.2 & .8 & 1. & 1.2 & .1 & 3. & 4. \end{bmatrix},$$

$$b = \begin{bmatrix} .5 \\ 1.2 \\ .8 \\ 2. \\ 1.5 \end{bmatrix}.$$

Here there are eight commodities, and the vectors were selected as $(k_1/200, \cdots, k_8/200)$, with $\sum k_i = 200$. There are some $.22 \times 10^{13}$ such vectors and the algorithm terminated in 640 iterations with a primitive set given by

$$\begin{bmatrix} 51 & 53 & 53 & 52 & 53 & 52 & 53 & 53 \\ 7 & 6 & 7 & 7 & 7 & 7 & 7 & 7 \\ 14 & 13 & 13 & 14 & 14 & 14 & 14 & 13 \\ 20 & 20 & 20 & 19 & 20 & 20 & 19 & 20 \\ 15 & 15 & 14 & 15 & 14 & 15 & 15 & 15 \\ 58 & 58 & 58 & 58 & 58 & 57 & 58 & 58 \\ 23 & 23 & 23 & 23 & 22 & 23 & 22 & 23 \\ 12 & 12 & 12 & 12 & 12 & 12 & 12 & 11 \end{bmatrix}.$$

After averaging, the following price vector and excess demands were obtained:

$$\hat{\pi} = (56.4, \quad 6.3, \quad 12.7, \quad 18.5, \quad 13.6, \quad 60.0, \quad 21.5, \quad 11.1),$$

$$(g_i(\hat{\pi})) = (-.1, \quad -.2, \quad .05, \quad .05, \quad .03, \quad .07, \quad .05, \quad -.04).$$

The answer here seems not to be as close a fit as the answer to the first problem, but the impression of the author is that this can be remedied by either an extension of the terminal linear programming problem, or the imposition of a finer grid for the first two commodities.

Example 3. This final problem terminates quite rapidly with a remarkably good fit, even though it is a larger problem than the previous ones, involving 10 commodities. We have

$$W = \begin{bmatrix} .6 & .2 & .2 & 20. & .1 & 2. & 9. & 5. & 5. & 15. \\ .2 & 11. & 12. & 13. & 14. & 15. & 16. & 5. & 5. & 9. \\ .4 & 9. & 8. & 7. & 6. & 5. & 4. & 5. & 7. & 12. \\ 1. & 5. & 5. & 5. & 5. & 5. & 5. & 8. & 3. & 17. \\ 8. & 1. & 22. & 10. & .3 & .9 & 5.1 & .1 & 6.2 & 11. \end{bmatrix},$$

$$A = \begin{bmatrix} 1. & 1. & 3. & .1 & .1 & 1.2 & 2. & 1. & 1. & .7 \\ 1. & 1. & 1. & 1. & 1. & 1. & 1. & 1. & 1. & 1. \\ 9.9 & .1 & 5. & .2 & 6. & .2 & 8. & 1. & 1. & .2 \\ 1. & 2. & 3. & 4. & 5. & 6. & 7. & 8. & 9. & 10. \\ 1. & 13. & 11. & 9. & 4. & .9 & 8. & 1. & 2. & 10. \end{bmatrix},$$

$$b = \begin{bmatrix} 2. \\ 1.3 \\ 3. \\ .2 \\ .6 \end{bmatrix}.$$

The prices were selected by $\sum_1^{10} k_i = 250$. There are some $.87 \times 10^{16}$ such vectors and the algorithm terminated with 468 iterations. After averaging the ten vectors in the primitive set, the following prices and excess demands were obtained:

$$\hat{\pi} = (47.0 \;\; 28.5 \;\; 24.0 \;\; 10.0 \;\; 26.7 \;\; 19.3 \;\; 29.4 \;\; 25.7 \;\; 24.8 \;\; 12.6),$$
$$(g_i(\hat{k})) = (-.07 \;\; .04 \;\; .03 \;\; .00 \;\; .02 \;\; .00 \;\; .02 \;\; .02 \;\; .02 \;\; -.07).$$

The excess demands for this last example are very close to zero, when compared with the total supply. What is even more surprising is that the total time on the 7094 required to do all *three* problems was 1 minute 36 seconds. This suggests that with improvements in the algorithm and its programming, the approximation of fixed points of mappings involving 15 to 20 dimensions might very well be feasible.

7. Some extensions of Theorem 1. The argument that has been given for Brouwer's theorem may be extended to a more general problem. As before, let $\pi^{n+1}, \cdots, \pi^k$ be a sequence of vectors on the simplex S, and let $\pi^1, \cdots, \pi^n$ have the special form previously described. Consider also the system of equations

$$Ax = b,$$

with A an $n \times k$ matrix of the form

$$A = \begin{bmatrix} 1 & \cdots & 0 & a_{1,n+1} & \cdots & a_{1,k} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & & 1 & a_{n,n+1} & \cdots & a_{n,k} \end{bmatrix}$$

and b a strictly positive vector. A feasible basis for this system of equations (in the sense used in linear programming) is a collection of n columns $j_1, \cdots, j_n$, which are linearly independent and such that the equations

$$\sum_\alpha a_{i,j_\alpha} x_{j_\alpha} = b_i$$

have a nonnegative solution.

As shown in [3] the arguments of this paper may be extended to demonstrate the following theorem.

THEOREM 2. *If the set of nonnegative solutions of $Ax = b$ form a bounded set, then there exists a primitive set $\pi^{j_1}, \cdots, \pi^{j_n}$ such that $(j_1, \cdots, j_n)$ is a feasible basis.*

In [3], Theorem 2 was used to provide general sufficient conditions for the core of an n-person game to be nonempty. It may also be used to demonstrate Brouwer's theorem for a mapping of a bounded polyhedral convex set, other than the simplex, into itself. To do this, we proceed by means of an intermediary theorem which has some interest in itself.

THEOREM 3. *Let $C_1, \cdots, C_k$ be closed sets on the simplex S, whose union is the entire simplex. Assume that $C_i \supset \{\pi \in S \mid \pi_i = 0\}$ for $i = 1, \cdots, n$. Then, if the set of nonnegative solutions to $Ax = b$ form a bounded set, there is a feasible basis $(j_1, \cdots, j_n)$ such that the intersection $\bigcap_{\alpha=1}^{n} C_{j_\alpha}$ is not empty.*

To prove this theorem we take a finite set of vectors $\pi^{n+1}, \cdots, \pi^l$ on the simplex, which, as l tends to infinity, will become everywhere dense on S. The vectors $\pi^1, \cdots, \pi^n$ are constructed as before. We define an $n \times l$ matrix $\tilde{A}$ to which Theorem 2 will be applied, as follows. The first n columns of $\tilde{A}$ form a unit matrix. To determine the entries in column r, with $r > n$, we select one of the sets C_j which contains π^r, and enter

$$\begin{pmatrix} a_{1j} \\ \vdots \\ a_{nj} \end{pmatrix}$$

in the rth column of $\tilde{A}$. As we see, $\tilde{A}$ is composed of some of the columns of A suitably repeated.

The hypotheses of Theorem 2 are clearly satisfied by $\tilde{A}$, and we may therefore find a primitive set of π's which correspond to a feasible basis for the equations $\tilde{A}x = b$. But since the columns of a basis are necessarily linearly independent, no two such columns can be identical, and a basis for $\tilde{A}x = b$ will also be a basis for $Ax = b$. If the columns of the basis are denoted by $j_1, \cdots, j_n$, the primitive set described in Theorem 2 will consist of a single vector from each of the sets $C_{j_1}, \cdots, C_{j_n}$.

If we let l tend to infinity in such a way that the vectors $\pi^{n+1}, \cdots, \pi^l$ become everywhere dense on the simplex, we may select a subsequence of l's so that the bases for $Ax = b$ do not change and such that the vectors forming the primitive set converge. But these vectors must all converge to the same point π. If some of the first n vectors are used in forming the primitive set, then the corresponding coordinates of π are equal to zero. π is therefore contained in $\bigcap_\alpha C_{j_\alpha}$ and Theorem 3 is demonstrated. It should be realized that the vector π may be approximated by an algorithm quite similar to that used in approximating a fixed point of a continuous mapping.

Now let C be a convex polyhedral subset of the simplex S defined by $C = \{\pi \mid \pi \in S, \sum \pi_i a_{ij} \geqq 0 \text{ for } j = n + 1, \cdots, k\}$, and $f(\pi) = (f_1(\pi), \cdots, f_n(\pi))$ a continuous mapping of C into itself. We assume, as before, that the set of nonnegative solutions to $Ax = 1$ is bounded, where

$$A = \begin{bmatrix} 1 & \cdots & 0 & a_{1,n+1} & \cdots & a_{1,k} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & & 1 & a_{n,n+1} & \cdots & a_{n,k} \end{bmatrix}.$$

It will also be useful to assume that the equations $\sum a_{ij_\alpha} x_{j_\alpha} = 1$ have a strictly positive solution if $j_1, \cdots, j_n$ is a feasible basis. This is a non-degeneracy assumption quite familiar in linear programming.

Define the sets $C_1, \cdots, C_k$ as follows. C_j contains all vectors in S with $\sum \pi_i a_{ij} \leqq 0$. Moreover, if $\pi \in C$, then $\pi \in C_j$ if $\sum \pi_i a_{ij} \leqq \sum f_i(\pi) a_{ij}$. Clearly, $\bigcup C_j = S$.

If Theorem 3 is applied, we obtain a feasible basis $j_1, \cdots, j_n$ for the equations $Ax = 1$, and a vector $\pi \in \bigcap_\alpha C_{j_\alpha}$. The author claims that $\pi \in C$, for if it is not, then $\sum \pi_i a_{ij_\alpha} \leqq 0$ for all α. But if x_{j_α} is the positive solution to the equations $\sum a_{ij_\alpha} x_{j_\alpha} = 1$, we obtain

$$0 \geqq \sum_\alpha \sum_i \pi_i a_{ij_\alpha} x_{j_\alpha} = \sum \pi_i = 1.$$

Since $\pi \in C$, we have $\sum \pi_i a_{ij_\alpha} \leqq \sum f_i(\pi) a_{ij_\alpha}$ for all α. But then

$$1 = \sum \pi_i = \sum_\alpha \sum_i \pi_i a_{ij_\alpha} x_{j_\alpha} \leqq \sum_\alpha \sum_i f_i(\pi) a_{ij_\alpha} x_{j_\alpha} = \sum f_i(\pi) = 1,$$

and since $x_{j_\alpha} > 0$ we see that $\sum (\pi_i - f_i(\pi)) a_{ij_\alpha} = 0$ for all α. But the columns of a basis are linearly independent, and therefore, $\pi_i = f_i(\pi)$. We therefore have a proof of Brouwer's theorem for continuous mappings of C into itself, and an algorithm for the approximation of a fixed point.

REFERENCES

[1] L. M. Graves, *The Theory of Functions of Real Variables*, 2nd ed., McGraw-Hill, New York, 1956.

[2] C. E. Lemke, *Bimatrix equilibrium points and mathematical programming*, Management Sci., 11 (1965), pp. 681–689.

[3] H. Scarf, *The core of an n-person game*, Econometrica, 35 (1967), to appear.

[4] ———, *Some examples of global instability of the competitive equilibrium*, International Economic Review, 1 (1960), pp. 157–172.