IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Computing Equilibria Via Simplicial Sandwiches: Convergence Analysis And Applications In Economics

[1]Nazir Ahmad Parray, [2]Dr. R. S. Patel

[1]Researcher , [2]Professor & Head of the Department of Mathematics Govt. P.G. College, Satna (M.P.)

Abstract: This paper provides a thorough introduction to the sandwich approach, a potent method that uses simplicial subdivision to compute Brouwer fixed points and solve related issues, such as determining broad economic equilibria. We give a self-contained introduction to the basic ideas of simplicial subdivision and sandwiching. We explore the nature of the algorithm's approximation capabilities and demonstrate the convergence features that are built into the method through thorough analysis. We also provide a brief overview of the algorithm's mechanics and showcase a small number of computational outcomes to highlight its usefulness.

Key words: Sandwich technique, Simplicial subdivision, Brouwer fixed points, Stochastic equilibrium, Properties of convergence, Approximation.

I. INTRODUCTION

The sandwich method stands out as a versatile technique employed for computing Brouwer fixed points and addressing associated challenges, notably including the determination of general economic equilibria. Belonging to a class of algorithms known as pivotal methods, it shares a striking combinatorial resemblance with the pivot step of the simplex method utilized in solving linear programs. This paper explores the foundational principles and practical implications of the sandwich method, shedding light on its effectiveness in navigating complex mathematical landscapes and facilitating solutions to diverse problems in economics and beyond.

In 1964, Lemke and Howson [2] uncovered an algorithm applying pivotal methods to bimatrix games. Shortly after, Scarf [4,6] introduced algorithms for approximating Brouwer fixed points and economic equilibria in 1967. Following suit, Kuhn [7] proposed a similar algorithm leveraging simplicial subdivision in 1968. While these early algorithms provided quick but crude approximate solutions, achieving high accuracy at reasonable computational expense remained elusive.

Two different approaches evolved to deal with this problem. First, the homotopy technique was discovered in 1971 by Eaves [8], and it was later improved upon by Eaves and Saigal [9]. Second, Merrill [10,11] invented a different strategy that year known as sandwiching or restart approaches. MacKinnon independently rediscovered this method in 1972, and it was later determined that the sandwich method was this method applied to Kuhn's 1968 algorithm. Notably, the same idea was later used by Fisher, Gould, and Tolle [12] in their research. The sandwich method is versatile, applicable to various problem types. In the

version detailed here, these problems are situated on a unit $(n-1)$ dimensional simplex $S^{(n-1)}$, comprising all real vectors $x=(x_1,\dots,x_n)$ with nonnegative components summing to one. In the first class of applications, a continuous function f is provided, which maps the simplex into itself. Brouwer's fixed-point theorem guarantees the existence of at least one point $\bar{x}$ such that $f(\bar{x})=\bar{x}$. The sandwich method is employed to find approximate solutions to this equation, making it applicable to any problem formulable as a Brouwer fixed-point problem.

A second class of applications pertains to the classical general equilibrium model of an exchange economy, extensively discussed by Scarf [6,13] and Arrow and Hahn [14]. Here, $x_1,\dots,x_n$ represent the prices of n goods exchanged in the economy. The demands and supplies of each good are determined by these prices, and the disparity between demand and supply for good i is termed the excess demand, denoted by $g_i(x)$. These excess demand functions are assumed to be continuous and homogeneous of degree zero for nonnegative prices that are not all zero. Additionally, they are assumed to adhere to Walras' law, which stipulates that total expenditures must equal total revenues, expressed as $\sum_i x_i g_i(x)=0$.

An equilibrium is defined as a set of prices $\bar{x}$ where the excess demand for every good is non positive, i.e., $g_i(\bar{x})\le 0$ for all i. Walras' law further implies that $\bar{x}_i=0$ whenever $g_i(\bar{x})<0$, ensuring $\bar{x}_i g_i(\bar{x})=0$ for all i. Price vectors satisfying these conditions can be obtained using the sandwich method in two different ways. They can be transformed into Brouwer fixed-point problems, as demonstrated in Scarf [6], or solved directly, as outlined in MacKinnon [15].

Basic Definitions:

Definition of an (n-1)-dimensional simplex: It's defined as a set of points expressible as convex combinations of n affinely independent vectors. In the case of the unit simplex, its vertices are the unit vectors in n-space.

Faces of a simplex: A simplex has n faces, each formed by dropping one of its n vertices. These faces are $(n-2)-$ simplices.

Subdivision of a simplex: Any simplex can be subdivided into smaller simplices such that each part is itself a simplex. This subdivision ensures that if any two subsimplices share a common boundary of dimension $(n-2)$, that boundary is a face of both.

Regular subdivision method: Described as a method of subdividing the unit simplex. In this method, every vertex of a sub simplex in the subdivision can be expressed as $(\frac{X_1}{D},\frac{X_2}{D},\dots,\frac{X_n}{D})$, where D is the degree of the subdivision, and $X_1,X_2,\dots,X_n$ are nonnegative integers summing to D.

Mesh of the subdivision: As D tends to infinity, the distance between any two points in a single sub simplex tends to zero. The mesh of the subdivision is not greater than $\frac{\sqrt{n}}{D}$, which approaches zero as D increases.

Proper labeling: An assignment of integer labels to the vertices of the subdivision is termed proper if a vertex x never gets label k if $x_k=0$. This ensures that each vertex receives a label between 1 and n.

1 Figure Explanation and Path Analysis

Figure 1 illustrates the 6-subdivision of S^2, where each vertex of the subdivision is assigned an integer label ranging from 1 to 3. More generally, for a subdivision of S^{n-1}, vertices are labeled with integers from 1 to *n*. A labeling is defined as a *proper labeling* if no vertex *x* is assigned the label *k* when $x_k = 0$. As can be verified, the labeling in **Figure 1** satisfies this condition and is thus a proper labeling.

This setup aligns with **Sperner's lemma**, which states that in every proper labeling of a subdivision, there exists an odd number of completely labeled subsimplices. In **Figure 1**, dotted lines connect all pairs of subsimplices that share a common face labeled with $1,2,\dots,n-1$. Such faces are termed *doors*. If two subsimplices can be connected by passing only through doors, they are considered *connected* or to lie along the same *path*.

There are four types of paths, all exemplified in **Figure 1**. Some paths may form endless loops, such as the one passing through *C*. Other paths must have distinct starting and ending points. Only specific types of subsimplices can serve as these endpoints:

- Subsimplices with labels $1, 2, \ldots, n-1$ on a boundary face of the simplex are referred to as **starts**.
- Subsimplices with all labels $1, 2, \ldots, n$ are referred to as **ends**. As a result, paths can be categorized as:

1. **Start-start paths**: For example, the path from *F* to *G*.
2. **End-end paths**: For example, the path from *A* to *B*.
3. **Start-end paths**: For example, the path from *E* to *D*.

It is critical to note that no paths can originate at a start or end and fail to terminate at another start or end. This property is fundamental to the operation of the sandwich method and ensures its structural integrity and correctness.

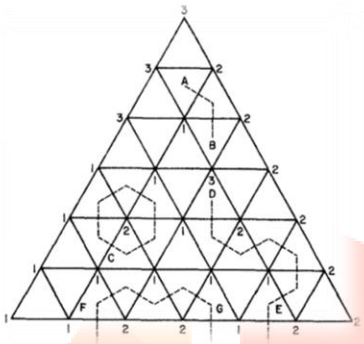

Figure 1

Proposition 2.1 :- **Any path that originates at a start or an end must terminate at a start or an end.**

Proof :- Let us consider a path traversing the simplicial subdivision of $S^{(n-1)}$. The proof proceeds as follows:

Two-Door Property: Any subsimplex on the path that is neither a start nor an end must possess exactly two doors:

- If the subsimplex has one face labeled $1, 2, \ldots, n-1$ and no vertex labeled *n*, then it must have exactly two such faces.
- These faces are not on the boundary of $S^{(n-1)}$, which ensures that the subsimplex is adjacent to exactly two other subsimplices on the path.

This ensures that the path proceeds uninterrupted through such subsimplices.

1. **Non-Reentrance:** A path cannot revisit the same subsimplex unless adjacent subsimplices on the path are revisited as well. If the path originates at a start or an end, at least one subsimplex exists that cannot be re-entered due to its unique structure or labeling. This guarantees that no subsimplex along the path is visited more than once.
2. **Finiteness of the Subdivision:** The simplicial subdivision of $S^{(n-1)}$ contains a finite number of subsimplices. Therefore, a path cannot continue indefinitely and must eventually terminate.
3. **Termination Criterion:** Since the path cannot terminate within a subsimplex that is neither a start nor an end (as it has two doors and continues), it must terminate at a start or an end.

By combining the two-door property, the non-reentrance condition, and the finiteness of the simplicial subdivision, it follows that any path starting at a start or an end must terminate at a start or an end.

Corollary 2.1 Restatement and Explanation

Corollary 2.1: If there is exactly one start, any path that originates from this start must necessarily terminate at an end.

This corollary forms the foundation of the sandwich method. The objective of the algorithm is to locate a completely labeled subsimplex, which represents an end. The algorithm achieves this by constructing a labeling scheme for a simplicial subdivision such that there is a unique start. Once identified, the path originating from this start is followed systematically until it reaches an end.

The significance of locating a completely labeled subsimplex lies in its ability to ap- proximate solutions to certain mathematical problems when *D*, the degree of subdivision, is sufficiently large. One notable application is the problem of finding a Brouwer fixed point. Specifically, the task is to determine an *n*-vector $\bar{x}$ within the unit simplex such that $\bar{x} = f(\bar{x})$, where $f(x)$ is a continuous mapping of the simplex into itself. In this context, every vertex of a regular subdivision of $S^{(n-1)}$ corresponds to a vector *x*. A labeling scheme can then be applied to these vertices, such as the modified rule proposed by Kuhn [16].

Rule 1 (Modified Kuhn Rule)

Vertices are labeled according to the following rule:

$$L(x) = k \; if \; x_k > 0 \; and \; f_k(x) - x_k \leq f_i(x) - x_i \; \forall \; i \; with \; x_i > 0,$$
$$and \; if \; f_k(x) - x_k = f_i(x) - x_i, then \; k \leq i.$$

Explanation of Rule 1

- **Proper Labeling:** Rule 1 ensures proper labeling because every vertex satisfies $x_k > 0$ for at least one *k*, making it eligible for a label.
- **Label Properties:** When a vertex *x* receives label *k*, it implies$f_k(x) - x_k \leq 0$. If $f_k(x) - x_k$ were positive, it would follow that $f_i(x) - x_i > 0$ for all *i* with $x_i > 0$, violating the constraint:

$$\sum_i f_i(x) = \sum_i x_i = 1.$$

Thus, Rule 1 guarantees that $f_k(x) - x_k \leq 0$ whenever *x* is labeled *k*.

Implications for the Sandwich Method

The labeling properties defined by Rule 1 ensure the existence of completely labeled subsimplices for every *D*. Furthermore, these subsimplices approximate fixed points as *D* becomes sufficiently large.

Let $x^1(D), \ldots, x^n(D)$ denote the vertices of a completely labeled subsimplex, where each vertex $x^k(D)$ is labeled *k* for $k = 1, \ldots, n$. Any point ξ within this subsimplex serves as an approximation to the fixed point. This relationship between completely labeled subsimplices and fixed-point approximations will now be specified in greater detail. **Proposition 2.2:** For any $\epsilon > 0, |f(\hat{x}) - \hat{x}| < \epsilon$. for sufficiently large *D*. In other words, $\hat{x}$ is approximately a fixed point. **Proof:** By the uniform continuity of f, we can select D_0 such that:

$$\left|f_k(\hat{x}) - f_k\left(x^k(D)\right)\right| < \frac{\varepsilon}{2(n-1)\sqrt{n}} \quad \text{for } D \geq D_0.$$

Furthermore, since:

$$\left|\hat{x}_k - x_k^k(D)\right| \leq \frac{1}{D},$$

we can choose $D_1 \geq D_0$ such that:

$$\left|\hat{x}_k - x_k^k(D)\right| \leq \frac{\varepsilon}{2(n-1)\sqrt{n}}$$

for $D \geq D_1$.

Thus, for $D \geq D_1$, we have:

$$f_k(\hat{x}) \leq f_k\left(x^k(D)\right) + \frac{\varepsilon}{2(n-1)\sqrt{n}} \leq x_k^k(D) + \frac{\varepsilon}{2(n-1)\sqrt{n}},$$

where the second inequality holds because $x^k(D)$ is labelled k. Hence,

$$f_k(\hat{x}) \leq \hat{x}_k + \frac{\varepsilon}{2(n-1)\sqrt{n}}, \qquad for\ k = 1, \dots, n.$$

Additionally, since:

$$\sum_{k=1}^{n} (f_k(\hat{x}) - \hat{x}_k) = 0,$$

it follows that:

$$-(n-1)\frac{\varepsilon}{2(n-1)\sqrt{n}} \leq f_k(\hat{x}) - \hat{x}_k,$$

and thus:

$$|f_k(\hat{x}) - \hat{x}_k| < \frac{\varepsilon}{\sqrt{n}}, \qquad for\ k = 1, \dots, n.$$

From this, we conclude that:

$$|f(\hat{x}) - \hat{x}| < \epsilon.$$

This proves that $\hat{x}$ is approximately a fixed point.

Proposition 2.3: For any $\epsilon > 0$, there exists a fixed point $\bar{x}$ (which may depend on D) such that $|\hat{x} - \bar{x}| < \varepsilon$ for sufficiently large D. In other words, $\hat{x}$ is close to a fixed point.

Proof: Using Sperner's Lemma, select a point $\hat{x}(D)$ for each D. Assume, for the sake of contradiction, that $|\hat{x}(D) - \bar{x}| \geq \varepsilon$ for all fixed points $\bar{x}$, and for a sequence of D approaching infinity. By compactness, there exists a subsequence of $\{\hat{x}(D)\}$ that converges to a point x^*. From Proposition 2.2 and the continuity of f, x^* must satisfy $f(x^*) = x^*$, making it a fixed point. This contradicts the assumption that $|\hat{x}(D) - \bar{x}| \geq \varepsilon$ for all fixed points $\bar{x}$. Therefore, the proposition is proven.

Labelling rules similar to Rule 1 can also be applied to solve other types of problems. In fact, many such rules share the property that, for sufficiently large D, any point in a fully labelled subsimplex provides an approximate solution. For example, one could assign the label k if k is the first index where $f_k(x) \leq x_k$ and $x_k > 0$. Experimental comparisons of different labelling rules in the context of economic general equilibrium models are discussed in MacKinnon [15].

2 Sandwich Method

Consider the case where $X_1, X_2, \dots, X_n$ are nonnegative integers summing to D. These integers represent a vertex in the D-subdivision of an $(n-1)$-simplex. Now, suppose $X = (X_1, X_2, \dots, X_n, 1)$. This vector X corresponds to a vertex in the $(D+1)$-subdivision of an n-simplex. Therefore, the relevant subdivided $(n-1)$-simplex can be embedded within the subdivision of an n-simplex, which we denote as S^n. The subdivided $(n-1)$- simplices embedded in this structure are labeled as $S_0^{(n-1)}, S_1^{(n-1)}, S_2^{(n-1)}, \dots$, where the subscript refers to the last component of each vertex.

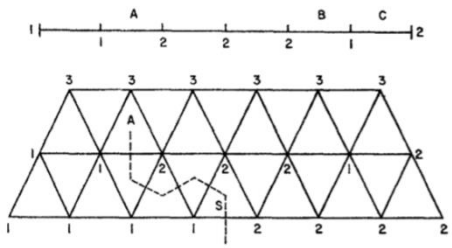

Figure 2

For clarity, consider the example of a 1-simplex, as shown in Figure 2. From this figure, it becomes apparent why the method is called the sandwich method, since the original simplex $S_1^{(n-1)}$ is sandwiched between $S_0^{(n-1)}$ and $S_2^{(n-1)}$. The additional layers of S^n, such as $S_3^{(n-1)}$ and beyond, do not play a role in the algorithm and are therefore omitted from the illustration.

In this case, the 1-simplex shown in Figure 2 has three "ends" located at points *A*, *B*, and *C*. Finding one of these ends is straightforward, but the sandwich method achieves this in a nontrivial manner. The labels of $S_0^{(n-1)}$ are selected so that there is a unique start. The labels for $S_2^{(n-1)}$ are all set to 3. Since only one start exists, by Corollary 2.1, the path that begins at this start must eventually terminate at a fully labeled 2-subsimplex. This subsimplex must include one vertex from $S_2^{(n-1)}$, as no other vertices have label 3, and two vertices from $S_1^{(n-1)}$. These two vertices form a completely labeled subsimplex of $S_1^{(n-1)}$. Thus, solving this auxiliary problem automatically solves the original problem. In practical applications, the sandwich method is typically used iteratively rather than only once. For instance, a solution for a given D_1 can serve as the starting point for $D_2 = 2D_1$ or $3D_1$. The solution for D_2 can then be used as the start for $D_3 = 2D_2$ or $3D_2$, and so on. For $n = 2$, this iterative process is illustrated in Figure 3.

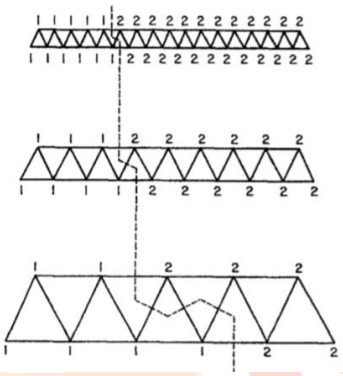

Figure 3

This method can be continued indefinitely, providing solutions with increasing accuracy at relatively small incremental costs compared to less accurate solutions. The only practical constraint is that *D* cannot exceed the largest integer that the computer can handle, which is $2^{31} - 1$ for IBM 360 and 370 machines.

The discussion above assumes that the sandwich method always terminates with a completely labeled subsimplex of $S_1^{(n-1)}$. We now provide a proof for this assertion.

2.1 Termination of the Sandwich Method

All vertices of S^n are denoted by integer $(n + 1)$-vectors that sum to $D + 1$. We assume that there is a proper labeling rule $L^1(X)$, such as Rule 1, that assigns integer labels between 1 and n to the vertices of $S_1^{(n-1)}$. Vertices of $S_0^{(n-1)}$ are labeled according to the following rule:

$$L^o(x) = k \; if \; (X_k - W_k) \geq (X_i - W_i) \; for \; i = 1,2, \dots, n \, ,$$

and

$$k \leq i \;\; if \; (X_k - W_k) \geq (X_i - W_i).$$

Here, W represents any vertex from $S_1^{(n-1)}$. Since the start will include W, it should be chosen as the best approximation of the expected location of the end.

Therefore, the labeling of vertices of S^n follows these rules:

$$-L(x) = L^0(x) \;\; if \;\; X_{(n+1)} = 0,$$
$$-L(x) = L^1(x) \;\; if \;\; X_{(n+1)} = 1,$$
$$-L(x) = n \; + \; 1 \text{ otherwise.}$$

This labeling strategy guarantees that the sandwich method will always terminate with a completely labeled subsimplex.

Proposition 3.1. The labeling rule $L(X)$ is proper.

Proof. By assumption, the labeling rule $L^1(x)$ is proper. The rule $L(X)$ assigns the label $n + 1$ to X only if $X_{n+1} \geq 2$. Consequently, $L(X)$ will be proper if $L^0(x)$ is also proper.

Let W be a vector in $S_1^{(n-1)}$ and X be a vector in $S_0^{(n-1)}$. By definition, the first n components of W sum to D, while the first n components of X sum to $D + 1$. Therefore, the difference $(X_i - W_i)$ must have a maximum value of at least 1 for some i.

Now, suppose that $X_j = 0$. Since $W_j \geq 0$, it follows that $(X_j - W_j) \leq 0$, which is strictly less than 1. Hence, if $X_j = 0$, $(X_j - W_j)$ cannot represent the maximum value of $(X_i - W_i)$ across all indices i. This implies that $L^0(x)$ cannot assign the label j to X.

As a result, the labeling rule $L^o(X)$ satisfies the condition of being proper. Combining this with the properness of $L^1(X)$, it follows that $L(X)$ is a proper labeling rule.

Proposition 3.2. If $X^1, X^2, \ldots, X^{(n+1)}$ are the vertices of a subsimplex of S^n, then for all i, j, and k, the components X_k^i and X_k^i differ by at most one.

Proof. Let Δ^k be a vector of $n + 1$ elements where the k-th coordinate is 1, the $(k - 1)$-th coordinate is -1 (with $k - 1 = n + 1$ if $k = 1$), and all other entries are 0. A subsimplex in a regular subdivision of an n-simplex can be fully characterized by a non-negative integer vector X whose components sum to $\overline{D}$ (where $\overline{D} = D + 1$ in the case of S^n) and a permutation P of the first $n + 1$ integers, as established by Kuhn [7].

The construction of the subsimplex proceeds as follows: the first vertex, X^1 , is given by X. Subsequent vertices are generated iteratively:

$$X^2 = X^1 + \Delta^{P(1)}, \qquad X^3 = X^2 + \Delta^{P(2)}, \ldots, X^{n+1} = X^n + \Delta^{P(n)}.$$

Key to this argument is the observation that only Δ^k has a k-th coordinate equal to 1, and only Δ^{k+1} has a k-th coordinate equal to -1. When constructing any vertex X^j from X^1, the vectors Δ^k and Δ^{k+1} can each contribute at most once to the $k - th$ coordinate. Consequently, the k-th component X_k^i differs from X_k^1 by at most one.

This argument is not limited to X_k^1 and X_k^i; it generalizes to any pair of vertices X^i and X^j within the subsimplex. By reordering the vertices and selecting a new permutation P such that any vertex occupies the first position, the same logic applies. Hence, X_k^i and X_k^j differ by at most one for all i, j, k.

Proposition 3.3

Let $X^1, X^2, \ldots, X^{(n+1)}$ be the vertices of a subsimplex of S^n, and suppose these vertices are assigned the labels $1, 2, \ldots, n + 1$, respectively. Then, the $(n + 1) - th$ component of X^i satisfies $X_{n+1}^i = 1$ for $i = 1, 2, \ldots, n$.

Proof: Since the vertex $X^{(n+1)}$ has the label $n + 1$, it follows that $X_{n+1}^{n+1} = 2$. None of the other vertices $X^1, X^2, \ldots, X^n$ are labeled $n + 1$, which implies they do not belong to $S_2^{(n-1)}$. By Proposition 3.2, these vertices must belong to $S_1^{(n-1)}$, where the $(n + 1)$-th component is equal to 1. Therefore, $X_{n+1}^i = 1 \; for \; i = 1,2, \ldots, n$.

Next, we establish the uniqueness of the starting point. Define the vectors $Z^1, Z^2, \ldots, Z^{(n+1)}$ as follows:

$$Z_j^i = W_j + \delta_{j,i} - \delta_{j,n+1} \, , for \; i = 1,2, \ldots, n+1,$$

where $\delta_{i,j}$ is the Kronecker delta, equal to 1 $if \; i = j \; and \; 0$ otherwise.

The vectors $Z^1, Z^2, \ldots, Z^{(n+1)}$ satisfy the conditions for being the vertices of a subsimplex of S^n, as defined in the proof of Proposition 3.2. Specifically, the vector $Z^{(n+1)}$ is equal to W , and the $(n + 1) - th$ component of Z^i satisfies $X_{n+1}^i = 0$ for $i = 1, 2, \ldots, n$.

Consequently, the first n vectors $Z^1, Z^2, \ldots, Z^n$ span a subsimplex of S_0^{n-1}.

Furthermore, the first n components of $Z^i - W$ are unit vectors with a 1 in the i-th position and zeros elsewhere. Thus, the labeling rule $L^o(Z^i) = i$ holds for $= 1, 2, \ldots, n$. It follows that $Z^1, Z^2, \ldots, Z^{n+1}$ uniquely determine the starting configuration for the algorithm.

Proposition 3.4

The vectors $Z^1, Z^2, \ldots, Z^n$ span the unique subsimplex of S_0^{n-1} with all labels $1, 2, \ldots, n$.

Proof : Let us assume, for contradiction, that there exists another subsimplex of S_0^{n-1} with vertices $V^1, V^2, \ldots, V^n$, where these vertices are assigned the labels 1, 2, . . . , n respectively. We aim to show that this assumption can only hold under specific conditions that imply $V^i = Z^i$ for all i.

Consider the difference $V^i - W$, where W represents some reference point. We hypothesize that the vector $V^i - W$ must be the vector (1, 0, . . . , 0) for each i. To substantiate this hypothesis, we analyze the following cases:

- **Case 1**: If $V_1^1 - W_1 < 1$, it follows that vertex V^1could not be assigned the label 1.
- **Case 2**: If $V_1^1 - W_1 > 1$ then the difference $(V_i^1 - W_i)$ must be negative for at least one i. However, by Proposition 3.2, this implies a contradiction, as the vector $(V_i^i - W_i)$ cannot simultaneously be both positive and negative. Therefore, V^i cannot receive the label i.

From these cases, we conclude that for each j, the difference $(V_j^1 - W_j)$ must satisfy $(V_j^1 - W_j) = 0$. If this were not the case, further contradictions would arise

- If $(V_j^1 - W_j) < 0, then\ (V_j^i - W_j)$, could not be positive, and hence vertex V could not receive label j
- If $(V_j^i - W_j) > 0$, it would follow that $(V_k^1 - W_k) < 0\ for\ some\ k$, leading to a similar contradiction.

  Consequently, we deduce that for each i, the difference $(V^i - W)$ must be a vector that has a 1 in the i-th position and zeros in all other positions. This implies that $V^i = Z^i$ for all i, thereby establishing that the vertices $Z^1, Z^2, \ldots, Z^n$ are the only vertices satisfying the required label assignments.

Therefore, the vectors $Z^1, Z^2, \ldots, Z^n$ span the unique subsimplex of $S_0^{(n-1)}$ with all labels $1, 2, \ldots, n$, completing the proof.

Proposition 3.5

If the algorithm starts at $Z^1, Z^2, \ldots, Z^n$ and follows the path that begins at these vertices, it must eventually terminate with a subsimplex that includes a completely labelled subsimplex of $S_1^{(n-1)}$.

Proof : By **Proposition 3.4**, there is only one starting point on $S_0^{(n-1)}$. Additionally, by **Proposition 3.1**, there are no starting points on any of the other faces of S^n. Therefore, by **Corollary 2.1**, the path that starts at $Z^1, Z^2, \ldots, Z^{n+1}$ must eventually reach an endpoint. Furthermore, by **Proposition 3.3**, we conclude that any such endpoint must include a completely labelled subsimplex of $S^{(n-1)}$. Thus, the algorithm, when following the path from $Z^1, Z^2, \ldots, Z^{n+1}$, must eventually terminate at a subsimplex that contains a fully labelled subsimplex of $S_1^{(n-1)}$, completing the proof.

3 Concise Description of the Algorithm

The following description provides a detailed outline of the algorithm, which could serve as the basis for its implementation in a computational setting. Let X be an $(n + 1) \times (n + 1)$ matrix, where each row of X represents the vertices of the subsimplex currently occupied by the algorithm. Denote the i-th row of X as X^i, which corresponds to the i-th vertex of the subsimplex. For each i, let X_j^i denote the j-th coordinate of the i-th vertex, scaled by $D + 1$. All elements of X are integers, and each row sums to $D + 1$.

Let W be an integer vector that sums to $D + 1$, where the first n elements of W represent the best available integer approximation to the location of the solution. Let L be an integer vector with $n + 1$ elements, where L_i is the label associated with X^i Let D_o be the degree of the initial subdivision, and D represent the degree of the current subdivision. Define $L^o(Z)$ as the artificial labelling rule, as discussed earlier, and let $L^1(Z)$ represent any proper labelling rule, such as Rule 1, which guarantees that completely labelled subsimplices approximate the solution.

The algorithm proceeds as follows:

1. **Initialization:** Set $D = D_o$. If W is not specified, initialize W as an integer approximation to
$$\left(\frac{D}{n}, \frac{D}{n}, \ldots, \frac{D}{n}, 1\right).$$

2. **Vertex Update:**
$$X_j^i = W_j + \delta_{j,i} - \delta_{j,n+1}, for\ i = 1,2, \ldots, n+1.$$
Here, $L_i = i$ for $i = 1, \ldots, n$, and $E = n + 1$ denotes that X^{n+1} is the entering vertex.

3. **Labelling Step:**
$$L_E = \begin{cases} L^0(X) & if\ X_{n+1}^E = 0, \\ L^1(X) & if\ X_{n+1}^E = 0. \end{cases}$$

4. Pivoting and Update: Set $E = k$ if $L_E = L_k$. Then, calculate the new vertex as:
$$Z = \mathrm{X}^{E-1} + \mathrm{X}^{E+1} - \mathrm{X}^{E},$$
with boundary conditions $\mathrm{X}^{E-1} = \mathrm{X}^{n+1}$ when $E = 1$, and $\mathrm{X}^{E+1} = \mathrm{X}^{1}$ when $E = n + 1$. This pivoting rule is taken from Kuhn [7]. Set $\mathrm{X}^{\mathrm{E}} = \mathrm{Z}$. If $\mathrm{X}_{\mathrm{n+1}}^{\mathrm{E}} < 2$, return to step (3).

5. **Solution Approximation:** Set $\mathrm{X}^{\mathrm{E}} = 0$ and compute:
$$\bar{x}_{\mathrm{j}} = \frac{1}{n\mathrm{D}} \sum_{i=1}^{n+1} \mathrm{X}_j^i \quad \text{for } j = 1, \ldots, n.$$

The vector $\bar{x}$ is then an approximation to the solution. If $\bar{x}$ is a sufficiently accurate approximation, terminate the algorithm. Otherwise, increase $D\ to\ rD,$, where r is a positive integer greater than 1, and update W to the integer approximation of
$$(\bar{x}_1 D, \bar{x}_2 D, \ldots, \bar{x}_n D, 1).$$
Return to step (2).

3.1 Notes on Parameters:

The choice of D_0 and r depends on the specific problem being addressed. Both theoretical considerations and empirical experience suggest that D_0 should be reasonably small (e.g., $4n$) unless prior information is available. The best choice for r is often 3, though values of 2, 4, and 5 may also yield good results. It is not strictly necessary for W to be chosen as an approximation to the barycenter for D/r. In some problems, the solution tends to move systematically as D increases, making it useful to extrapolate from previous moves. These considerations are discussed in MacKinnon [15].

4 Computational Experience

The sandwich method has been applied to a wide variety of problems, primarily involving economic general equilibrium models. This experience, discussed in detail in MacKinnon[15], demonstrates that the cost of solving these problems varies significantly depending on the specific problem. The number of iterations required to achieve a desired accuracy level appears to scale approximately with $(n-1)^2$. Since the cost per labelling operation typically scales as $n-1$ or $(n-1)^2$, the total computational cost grows as $(n-1)^3$ or $(n-1)^4$. This implies that while the sandwich method may be less effective for very large problems, it is often highly efficient for smaller ones.

One notable advantage of the sandwich method is its reliance on integer arithmetic to store the current subsimplex as a matrix. This approach avoids the numerical issues associated with floating-point arithmetic. Provided that the solution is not required to exceed six-digit accuracy, the labelling routine can be implemented in single-precision arithmetic. For many computational environments, this can yield significant cost savings compared to the double-precision arithmetic typically required by gradient-based methods.

To illustrate the method's effectiveness, it was applied to three theoretical economic general equilibrium models originally solved by Scarf [6]. These problems are highly nonlinear but relatively well-behaved compared to other cases. The labelling rule used was:

$$L^1(x) = k \ \ if\ x_k > 0\ and\ \frac{x_k g_k(x)}{s_k(x)} \geq \frac{x_i g_i(x)}{s_i(x)} \quad \forall\, i \text{ with } x_i > 0,$$

and

$$k \leq i\ if\ \ \frac{x_k g_k(x)}{s_k(x)} \geq \frac{x_i g_i(x)}{s_i(x)}.$$

Here, $g_i(x)$ represents the excess demand for good $i, and\ s_i(x)$ denotes the supply of good i. This rule is a proper labelling rule and ensures an approximate equilibrium. Both theoretical arguments and experimental results suggest that it performs better than many other rules for problems of this type (see MacKinnon [15]).

4.1 Results Summary

The results for the three problems are summarized in Tables 1–3. The accuracy of each solution is evaluated by comparing demand and supply. All computations were performed on an IBM 360/91 using a program written in FORTRAN and compiled with the H compiler. The reported time represents the CPU time for executing the compiled program. The majority of computational resources are utilized in evaluating $L^1(x)$, as the opera- tions associated with artificial labellings and pivoting are computationally inexpensive in comparison.

Table 1: Computational Results for $Problem\ 1\ (n = 5, D = 20 \times 3^{10} = 1,180,980)$

Good	Price	Demand	Supply	Excess Demand
1	0.674543	2.6000	2.6000	0.000002
2	0.079105	10.0000	10.0000	-0.000029
3	0.033422	45.0000	45.0000	-0.000031
4	0.122178	11.0000	11.0000	-0.000006
5	0.090752	18.7999	18.8000	-0.000063

Additional Metrics: Number of genuine labellings: 161 Number of artificial labellings: 140, Time: 0.18 seconds

Table 2: Computational Results for Problem 2 $(n = 8, D = 32 \times 3^{10} = 1,889,568)$

Good	Price	Demand	Supply	Excess Demand
1	0.271236	3.4000	3.4000	0.000001
2	0.029566	20.1998	20.2000	-0.000161
3	0.062938	10.4000	10.4000	0.000009
4	0.093090	10.4000	10.4000	-0.000010
5	0.067234	14.2001	14.2000	0.000059
6	0.305901	3.4000	3.4000	-0.000003
7	0.104365	7.4000	7.4000	-0.000015
8	0.065672	17.2999	17.3000	-0.000064

Additional Metrics: Number of genuine labellings: 461 Number of artificial labellings: 308, Time: 1.17 seconds

Table 3: Computational Results for Problem *3* ($n = 10,\ D = 40 \times 3^{10} = 2,361,960$)

Good	Price	Demand	Supply	Excess Demand
1	0.187262	10.2000	10.2000	0.000034
2	0.109379	26.2000	26.2000	0.000039
3	0.098896	47.1998	47.2000	0.000133
4	0.043191	55.0000	55.0000	0.000043
5	0.116867	25.4000	25.4000	0.000001
6	0.076974	27.9000	27.9000	0.000004
7	0.116966	39.1000	39.1000	0.000007
8	0.102381	23.0999	23.1000	0.000066
9	0.098691	26.2000	26.2000	0.000003
10	0.049393	63.9997	64.0000	0.000304

Additional Metrics: Number of genuine labellings: 733 Number of artificial labellings: 524 ,Time: 2.33 seconds

5 Conclusion

The sandwich method represents a straightforward and efficient algorithm for identifying Brouwer fixed points and addressing related computational problems. The method is mathematically guaranteed to terminate, ensuring robustness. Empirical results demonstrate its practical effectiveness, particularly for problems with low to moderate dimensionality, where convergence is achieved in a reasonable timeframe.

The algorithm is characterized by its simplicity in both implementation and application. Once implemented, it provides a versatile framework for solving a diverse array of problems. Moreover, solutions can be computed to virtually any desired degree of accuracy, subject only to the inherent limitations of the computational hardware. This flexibility, coupled with its ease of use, underscores the utility of the sandwich method as a reliable computational tool for fixed-point problems.

References

1. Patel, R.S., Rusia, R., and Patel, P. (2013). *Computation of Fixed Point by Using Efficient Algorithm*. Journal of Computational and Mathematical Sciences, Vol. 4, Issue 4, August 31, 2013, pp. 202–321.

2. Lemke, C. M., & Howson, J. T. (1964). Equilibrium Points of Bimatrix Games. *Journal of the Society for Industrial and Applied Mathematics*, 12, 413–423.

3. Patel, R.S., Agarwal, A., and Bhahel, P.S. (2011). *Comparison of A.D.D. of Triangulations*. Journal of Nanabha, Vol. 41, 2011.

4. Scarf, H. E. (1967). The Approximation of Fixed Points of a Continuous Mapping. *SIAM Journal on Applied Mathematics*, 15, 1328–1343.

5. Patel, R. S.,& Pandey, S.K., (2016). *Accelerarion technique for kakutani fixed point algorithm*. Vindhya Research Journal Vol. 2, No. 1, 61-66.

6. Scarf, H. E. (1967). On the Computation of Equilibrium Prices. In W. Fellner et al. (Eds.), *Ten Economic Studies in the Tradition of Irving Fisher* (pp. 299–317). New York, NY: John Wiley & Sons.

7. Kuhn, H. W. (1968). Simplicial Approximation of Fixed Points. *Proceedings of the National Academy of Sciences, USA*, 61, 1238–1242.

8. Eaves, B. C. (1972). Homotopies for Computation of Fixed Points. *Mathematical Programming*, 3, 1–22.

9. Eaves, B. C., & Saigal, R. (1972). Homotopies for Computation of Fixed Points on Unbounded Regions. *Mathematical Programming*, 3, 225–237.

10. Merrill, O. H. (1971). Applications and Extensions of an Algorithm that Computes Fixed Points of Certain Non-Empty Convex Upper Semi-Continuous Point-to-Set Mappings. Technical Report No. 71-7, Department of Industrial Engineering, Uni- versity of Michigan.

11. Merrill, O. H. (1972). Applications and Extensions of an Algorithm that Computes Fixed Points of Certain Upper Semi-Continuous Point-to-Set Mappings. *Ph.D. Thesis*, University of Michigan.

12. Fisher, M. L., Gould, F. J., & Tolle, J. W. (1975). A New Simplicial Approximation Algorithm with Restarts: Relations Between Convergence and Labelling. *Proceedings of the Conference on Computing Fixed Points with Applications*. Clemson University, Clemson, South Carolina.

13. Scarf, H. E., with Hansen, T. (1973). *The Computation of Economic Equilibria*. New Haven, CT: Yale University Press.

14. Arrow, K. J., & Hahn, F. H. (1971). *General Competitive Analysis*. San Francisco, CA: Holden-Day.

15. MacKinnon, J. G. (1975). Solving Economic General Equilibrium Models by the Sandwich Method. *Proceedings of the Conference on Computing Fixed Points with Applications*. Clemson University, Clemson, South Carolina.

16. Kuhn, H. W. (1960). Some Combinatorial Lemmas in Topology. *IBM Journal of Research and Development*, 4, 508–524.