Research Notes and Comments

A Procedure for Identifying and Storing a Thiessen Diagram within a Convex Boundary *by Robert G. Cromley and Daniel Grogan*

Introduction

An important geographical problem, given a distribution of central points, has been the delineation of a locus of points that is closer to one center than to any other. This problem, illuminated by central place theory, is also an important element of applied geographical work in market analysis and in evaluating alternative districting plans. In cartography, this problem is known as the delineation of Thiessen polygons. Algorithms for analytically delineating Thiessen polygons have been developed (Rhynsburger 1973; Shamos 1977; Shamos and Hoey 1975; Shamos and Bentley 1978; Brassel and Reif 1979).

A Thiessen diagram has been used to solve closest point problems (Fowler 1978). The identification of the elements of a Thiessen diagram has also become an important step in performing a Delauney triangulation of the initial distribution of control points (McCullagh and Ross 1980). Elfick (1979) has presented a procedure for constructing contour maps based on a triangulation of the control points. This paper presents an algorithm to analytically define Thiessen polygons within a convex boundary. More importantly, the algorithm is organized to construct a cartographic data structure for storing the basic elements of a Thiessen diagram as they are calculated. This structure permits either a proximal or an isarithmic map to be constructed from an initial distribution of control points and facilitates the calculation of many analytical functions for a spatial distribution.

Basic Definitions and Concepts

The following set of definitions is modified from the discussion by Brassel and Reif (1979, pp. 291–92). A set **C** of n points is given in a plane bounded by a convex polygon which is given; these points are called Thiessen centroids (Fig. 1). The convex bounding polygon P is defined by a set **D** of m edges that separate all points enclosed by P from those of its unbounded compliment P'. Each edge is a line segment that connects two adjacent boundary vertices; thus P is alternatively referenced by a set **B** of m bounding vertices. Assume also that the bounding vertices are sequentially numbered in a clockwise direction around the polygon (Fig. 1).

The problem is to partition the bounded plane into a set **T** of n polygons such that one and only one Thiessen centroid is contained within each polygon and all other points within each polygon are closer to that polygon's centroid than to any other centroid; each of these polygons is called a Thiessen polygon. This problem is

Robert G. Cromley is associate professor of geography, University of Connecticut. Daniel Grogan is a graduate student in geography, Pennsylvania State University.

Geographical Analysis, Vol. 17, No. 2 (April 1985) © 1985 Ohio State University Press

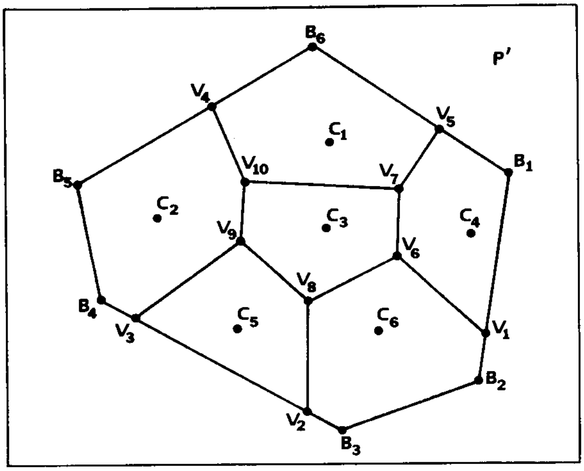
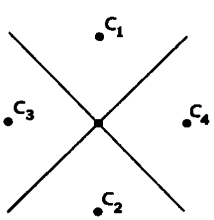


FIG. 1. An Example Thiessen Diagram

equivalent to finding a set **V** of p points that are equidistant and closest to three centroids; these points are called Thiessen vertices (Fig. 1). Each Thiessen vertex that lies on the convex bounding polygon is called a boundary Thiessen vertex; all others are known as interior Thiessen vertices. A Thiessen edge is defined as the locus of all points equidistant and closest to two centroids.

By definition, there is an omnipresent centroid C_{n+1} located in P' such that each bounding edge is equidistant between C_{n+1} and the respective interior centroids. Each portion of the convex boundary that is connected by two boundary Thiessen vertices is called a boundary Thiessen edge. Thus, each Thiessen vertex is directly connected to three other Thiessen vertices by either a boundary or interior Thiessen edge. Alternatively, each Thiessen vertex is formed by the intersection of three Thiessen edges. Thiessen edges can only intersect at a Thiessen vertex. Each Thiessen vertex on the boundary lies on a bounding edge that connects two sequentially numbered bounding vertices.

Finally, one centroid is called a Thiessen neighbor of another centroid if the two centroids share a common Thiessen edge. Other authors have also used the concept of a Thiessen half-neighbor; centroids are half-neighbors if they share only a Thiessen vertex (Fig. 2a). However, if all Thiessen vertices are uniquely enumerated (although two or more may occupy the same geographical location), then by definition each vertex will be formed by the intersection of exactly three Thiessen edges. In Figure 2b, $V1$ and $V2$ are two unique vertices and $E1$ is the Thiessen edge connecting these two vertices (although in reality $E1$ has a length of zero).



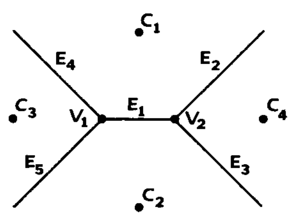


FIG. 2a (*left*). The Half-Neighbor Problem
FIG. 2b (*right*). A Resolution of the Half-Neighbor Problem

Centroids $C1$ and $C2$ are Thiessen neighbors because they share a Thiessen edge, whereas $C3$ and $C4$ are not neighbors because they do not share an edge. In the algorithm presented here, each Thiessen vertex is uniquely numbered and there are no half-neighbors.

Conceptual Background

It is important to enumerate each Thiessen vertex and edge because an exact number of them exist as a function of the number of centroids. Shamos and Bentley (1978) have proven that a Thiessen diagram on n centroids has $3(n - 2)$ Thiessen edges and $2(n - 2)$ Thiessen vertices. However, as discussed in the previous section, in a bounded polygon diagram there are $n + 1$ centroids resulting in $3(n - 1)$ edges and $2(n - 1)$ vertices. For the Thiessen diagram presented in Figure 1, there are $3(6 - 1)$ or 15 edges and $2(6 - 1)$ or 10 vertices.

This fact results from the well-known relationship between the Thiessen diagram and Delauney triangulation—one is the dual of the other. In a Delauney graph, a triangular mesh is formed by connecting each centroid to each of its Thiessen neighbors. Because Thiessen vertices are the circumcenters of Delauney triangles, one can construct a Thiessen data structure containing the topological relationships between vertices, centroids, and edges based on Elfick's (1979) triangulation structure for contour maps. As mentioned previously, each Thiessen vertex is directly connected to three other vertices by a Thiessen edge. A vertex reference file can be formed where each record of nine entries corresponds to the neighborhood information of each unique Thiessen vertex.

The first three entries of each record contain the reference values of the three adjoining Thiessen vertices recorded in a counterclockwise order. The next three entries are the reference values of the corresponding centroid that is the right-hand neighbor of the Thiessen edges connecting the current vertex to its neighbors; in Figure 3, C_1 is the right-hand centroid of the edge E, connecting V_1 and V_2. If the Thiessen vertex lies on the boundary, the next entry is the reference value of the lower number boundary vertex of the edge on which that Thiessen vertex rests; in Figure 4, B_1 is the reference value for V_1 and V_2 respectively. If the Thiessen vertex is an interior point, this entry has a value of zero. The final two entries in each record are the X and Y coordinates of the vertex. The vertex reference file for the Thiessen diagram presented in Figure 1 is presented in Table 1A.

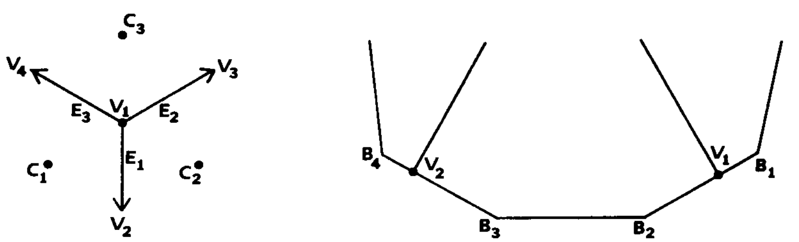


FIG. 3 (*left*). The Neighborhood Relationships for Each Interior Vertex
FIG. 4 (*right*). Boundary Vertices

The outlines of the individual Thiessen polygons can be retrieved by cross-referencing this file with a centroid pointer file that contains a starting vertex for each centroid. For example, in Table 1B, the starting vertex for $C1$ is $V5$; the next vertex in a clockwise direction is the adjoining vertex of $V5$ for which $C1$ is the corresponding right-hand centroid. As one "walks" around a Thiessen polygon in a clockwise direction, the generating centroid is always the right-hand centroid. In

TABLE 1A
Vertex Reference File

	Counterclockwise Vertex Neighbors			Corresponding Right-Hand Centroids			Previous Boundary Vertex	X	Y
$V1$	$V2$	$V5$	$V6$	$C6$	$C7$	$C4$	$B1$	X_1	Y_1
$V2$	$V3$	$V1$	$V8$	$C5$	$C7$	$C6$	$B3$	X_2	Y_2
$V3$	$V4$	$V2$	$V9$	$C2$	$C7$	$C5$	$B3$	X_3	Y_3
$V4$	$V5$	$V3$	$V10$	$C1$	$C7$	$C2$	$B5$	X_4	Y_4
$V5$	$V1$	$V4$	$V7$	$C4$	$C7$	$C1$	$B6$	X_5	Y_5
$V6$	$V1$	$V1$	$V8$	$C6$	$C4$	$C3$	0	X_6	Y_6
$V7$	$V6$	$V5$	$V10$	$C3$	$C4$	$C1$	0	X_7	Y_7
$V8$	$V2$	$V6$	$V9$	$C5$	$C6$	$C3$	0	X_8	Y_8
$V9$	$V3$	$V8$	$V10$	$C2$	$C5$	$C3$	0	X_9	Y_9
$V10$	$V4$	$V9$	$V7$	$C1$	$C2$	$C3$	0	X_{10}	Y_{10}

this case $V7$ is the appropriate adjoining vertex. The next vertex is found by scanning the record corresponding to $V7$ to find the adjoining vertex for which $C1$ is again the right-hand centroid. The entire string of vertices has been enumerated when the next vertex is the original starting vertex.

If any edge of the Thiessen polygon is a boundary Thiessen edge, then the appropriate boundary vertices must also be inserted in the outline before it can be

TABLE 1B
Centroid Pointer File

$C1$	$V5$
$C2$	$V4$
$C3$	$V6$
$C4$	$V3$
$C5$	$V2$
$C6$	$V1$

drawn or shaded. Because the boundary vertices are sequentially numbered in ascending order and the lower-valued boundary vertex corresponding to each boundary Thiessen vertex is recorded in the vertex file, the number of boundary vertices to be included is the difference between the boundary vertex reference numbers of the second clockwise Thiessen vertex and the first Thiessen vertex. For example, the Thiessen polygon centered on $C2$ has a Thiessen boundary edge whose clockwise vertices are $V3$ and $V4$; there are two boundary vertices on this Thiessen edge $(5 - 3)$. Because B_3 is the boundary reference of the first clockwise Thiessen vertex, then B_4 and B_5 are the boundary vertices that are inserted in the outline string of this Thiessen polygon.

The problem then is to develop a strategy for constructing the vertex and centroid pointer files. The following procedure completes this task by first enumerating all Thiessen boundary vertices; this is done by walking around the perimeter of the bounding edges in a clockwise manner. This first stage is complete when the first edge is reencountered. Once these Thiessen vertices are found, the interior vertices are calculated by walking around the edges of the individual Thiessen polygons in a clockwise manner. An individual polygon is complete when its terminal centroid is encountered. The terminal centroid is the left-hand centroid of the adjoining vertex that is counterclockwise to the starting vertex of the polygon. For example, if $V6$ was the starting vertex for the polygon around $C3$, then $C6$ is the terminal centroid for that polygon. The entire procedure is finished when the vertex file has $2(n - 1)$ completed records.

Algorithm Design and Implementation

First, the set of centroids is sorted along the X-axis and the resulting information is stored in ascending and descending linked-list arrays; these arrays facilitate later computations. The enumeration of boundary Thiessen vertices begins by finding the centroid that is closest to the first bounding vertex B_i (initially $i = 1$). By definition, B_i is a point within the Thiessen polygon associated with this centroid C_k. A Thiessen vertex exists on the bounding edge between B_i and B_{i+1}, E_i, whenever a neighbor centroid of C_k, C_{k+1}, is closer to the next clockwise boundary vertex, B_{i+1}, than C_k. Elements of Brassel and Reif's circle test strategy (1979, pp. 293–94) are employed to calculate the coordinates of the next Thiessen boundary vertex. Initially, the next potential Thiessen vertex V_j is set equal to B_{i+1}. A circle around this vertex is formed with a radius equal to the distance between V_j and C_k. If no other centroid lies within the circumference of this circle, then no Thiessen vertex exists on E_i; the boundary vertex index is updated and the process is repeated until all boundary vertices have been examined. If another centroid is within the initial radius, then V_j does exist on E_i. In this case, V_j is recomputed as the intersection of the perpendicular bisector between C_k and C_{k+1} and the edge E_i. A new circle centered on this vertex is formed whose radius is again the distance between V_j and C_k (Fig. 5). A new centroid is selected from the linked lists until no centroid lies within the radius of the latest circle.

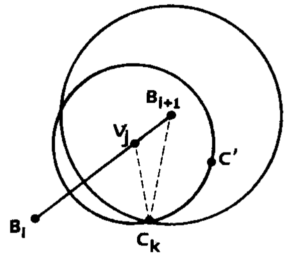


FIG. 5. The Circle Test for a Boundary Vertex

At this stage the coordinates of V_j have been found and the vertex reference centroid pointer and terminal centroid files are updated. The two known adjoining Thiessen vertices are entered in the record corresponding to V_j; these values are V_{j+1} and V_{j-1}. As the forward vertex, V_{j+1} is entered in the first position and V_{j-1} is entered in the second. The third adjoining vertex is an interior vertex and is not calculated until the next stage of the algorithm. However, all three right-hand centroid neighbors are known (C_k, C_{k+1}, and C_{N+1}) and are entered in the record for V_j. As the right-hand centroid of V_{j+1}, C_{k+1} is entered in the first position; C_{N+1} is entered in the second and C_k is recorded in the third position. The value of B_i is inserted in the record as the lower-valued boundary index for E_i. Finally, C_k is the terminal centroid for C_{k+1} and V_j is the corresponding terminal vertex. Table 2 presents the partial vertex reference file for the sample diagram that is completed at the end of the walk-around stage. The initial stage is flowcharted in Figure 6.

In the second stage, the interior Thiessen vertices for each Thiessen polygon are computed in a clockwise manner. The generating centroid for the first polygon is retrieved by examining the vertex reference file to find the first vertex V_g with an unknown adjoining interior vertex V_p. The corresponding right-hand centroid of this vertex is the first generating centroid C_g. Additionally, another known centroid

TABLE 2
Partial Vertex Reference File

	Vertex Neighbors			Corresponding Right-Hand Centroid			Previous Boundary Point
*V*1	*V*2	*V*5	—	*C*6	*C*7	*C*4	*B*1
*V*2	*V*3	*V*1	—	*C*5	*C*7	*C*6	*B*3
*V*3	*V*4	*V*2	—	*C*2	*C*7	*C*5	*B*3
*V*4	*V*5	*V*3	—	*C*1	*C*7	*C*2	*B*5
*V*5	*V*1	*V*4	—	*C*4	*C*7	*C*1	*B*6

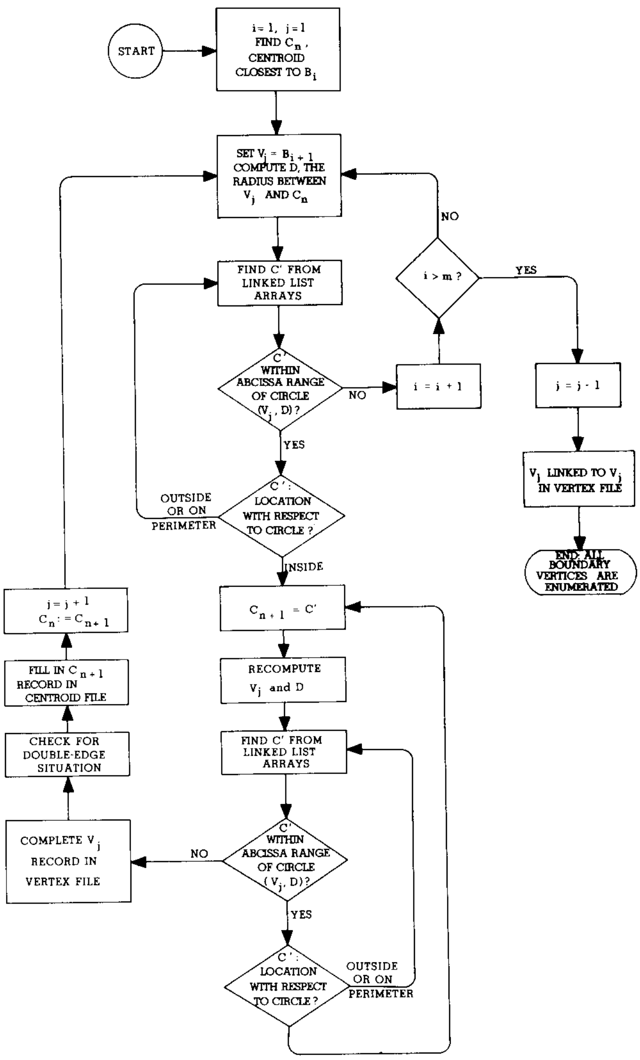


FIG. 6. Flowchart of the Walk-Around Stage

neighbor of V_p, C_N, is known from the vertex record of V_g. The problem is to find the third centroid neighbor of V_p and in the process the coordinates of V_p. This is accomplished by again using the circle test strategy of Brassel and Reif. If C_N is also the terminal centroid for C_g the polygon is complete and processing advances to the next uncompleted polygon that corresponds to the next open adjoining vertex in the vertex reference file. Once C_p is found it becomes the terminal centroid for the current C_N. C_N is updated and the process repeated until C_N is the terminal centroid for C_g. The vertex reference file is checked for open adjoining vertices until all $2(N - 1)$ vertices have been examined. This portion of the procedure is diagrammed in Figure 7.

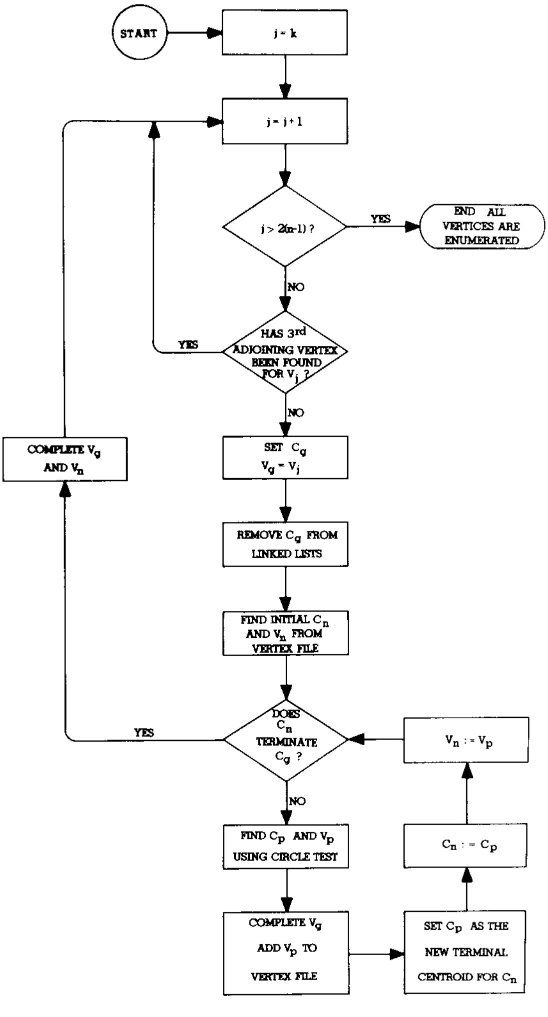


FIG. 7. Flowchart of Stage Two

The entire procedure has been encoded in FORTRAN IV and implemented on an IBM 370. Although the focus of the algorithm presented here has been to identify the elements of a Thiessen diagram and to store them in a flexible cartographic data structure, the efficiency of the procedure is comparable to other approaches. The Thiessen diagram presented in Figure 8 was generated for 1,000 centroids and computed in 10.763 CPU seconds. It should be noted that although a rectangle was used to bound the diagram, any convex polygon could have been used (e.g., an ellipse). In the future, this approach will be extended to concave polygons as these shapes are useful for display purposes.

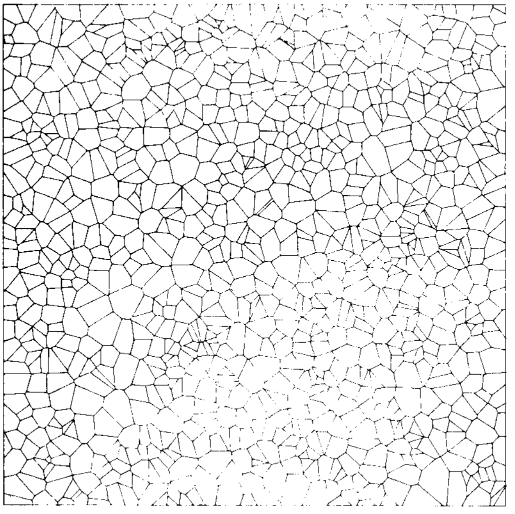


FIG. 8. A Generated Thiessen Diagram

Because each record is a fixed length and the number of vertices is a function of the number of centroids, this procedure does not require the dynamic allocation of additional storage. In the procedure presented here, all parameters are fixed and known in advance. In Brassel and Reif's approach, however, the vertex information was organized for the Thiessen neighbors of each centroid. Since the number of neighbors for each centroid cannot be known in advance, the list is expanded as new neighbors are found.

Summary

The Thiessen diagram is a flexible and useful tool in cartographic analysis. In the context of a geographic information system, the calculation of a Thiessen diagram is secondary to the utility of the topological information it contains. Besides providing direct information regarding the nearest areal neighborhood for central locations, the elements of a Thiessen diagram can also be used to construct an isarithmic or proximal map. The algorithm presented here identifies the basic components of a Thiessen diagram and stores them in a flexible data structure that retains all of the topological relationships among these elements. This method performs the same processes related to file structuring for proximal and isarithmic maps that automated digitizing systems perform for conformant zone maps. The Thiessen data structure outlined here can be used for a range of geographic display and analysis functions.

LITERATURE CITED

Brassel, K., and D. Reif (1979). "A Procedure to Generate Thiessen Polygons." *Geographical Analysis*, 11, 289–303.

Elfick, M. H. (1979). "Contouring by Use of a Triangular Mesh." *The Cartographic Journal*, 16, 24–29.

Fowler, R. J. (1978). "Approaches in Multi-Dimensional Searching." *Harvard Papers on Geographical Information Systems*, 6.

McCullagh, M., and C. Ross (1980). "Delauney Triangulation of a Random Data Set for Isarithmic Mapping." *The Cartographic Journal*, 17, 93–99.

Rhynsburger, D. (1973). "Analytic Delineation of Thiessen Polygons." *Geographical Analysis*, 5, 133–44.

Shamos, M. I. (1977). "Computational Geometry." Ph.D. dissertation, Yale University, 1977.

Shamos, M. I., and J. Bentley (1978). "Optimal Algorithms for Structuring Geographic Data." *Harvard Papers on Geographical Information Systems*, 7.

Shamos, M. I., and D. Hoey (1975). "Closest Point Problems." *Proceedings*, Sixteenth Annual Symposium on the Foundations of Computer Science, pp. 151–62.

Comment on The Optimal Length of a Marketing Week *by Alan Hay*

1. Introduction

In a recent paper Zemanian (1984) offers a study of the optimal length of a marketing week in a periodic market system. The starting point of his paper is to assume that "market clearing depresses the supply and demand of goods in the vicinity of that market" and that these supply and demand curves then recover. This claim is indisputable as is his remark that few papers "even allude to the recovery of supply and demand between market days" (p. 134). Zemanian then argues that the supply and demand curves recover in such a way that producer surplus plus consumer surplus are maximized if the market meeting is delayed for a time interval which can be derived from those same supply and demand curves. The object of this comment is to dispute the form of recovery implied by Zemanian's treatment of the problem and to show that his conclusions are not valid if the form of recovery is changed.

2. The Form of Recovery

Figure 1 in Zemanian's article uses conventional aggregate supply and demand curves which interact to yield a market clearing equilibrium price $P(w)$. Just after market clearance, he argues, there is still unsatisfied demand ($D(w+)$) and potential supply ($S(w+)$), but the demand is only effective at prices below $P(w)$ and supply is only effective at prices above $P(w)$. He draws these curves parallel to the original preclearing curves and then argues that with the passage of time the curves shift across the graph *in a parallel fashion* (Fig. 1). If the sum of producer and consumer surplus is defined as the area between and to the left of the two curves at the point of market clearing (this is the conventional definition), the social surplus increases with time.

There is some question as to whether the aggregate supply curve is the correct curve to use in such an analysis, but setting aside that question for the moment, how will demand and supply recover over time? Consider the simple case in Table 1. Suppose at a given price $p = 7$ the consumers consume 20 units in aggregate in one time period; at the same price, the demand (if it can be "stored") will increase

Alan Hay is reader in geography, University of Sheffield.