BIT 7 (1967), 39–64

COMPUTER CARTOGRAPHY POINT-IN-POLYGON PROGRAMS

STIG NORDBECK and BENGT RYSTEDT

Abstract.

The official statistics and census reports give figures only for administrative units. The boundaries of these units are often changed and hence it is very difficult to compare statistics from two different periods. However, an administrative unit can always be approximated by a polygon. Real estate data are assigned to a central point for which the coordinates are known. A computer can determine whether a point belongs to a polygon or not by means of a special program. Data for all real estate central points belonging to the actual polygon are added. In this way it will be possible to compute data for arbitrary polygons, for instance administrative units which do not exist any longer, by assigning real estate data to the central points.

Key words: Cartography, computer, polygon.

Contents.

1. The Swedish real estate register and the coordinate method.

The Swedish "Fastighetsregisterutredningen", a committee appointed to perform a revision of the real estate registration in Sweden, delivered its report to the Swedish Government October 10, 1966. Among other reforms of the real estate register the committee proposes [2] that the location of every real estate property be given in the official real estate register by means of coordinates for some central point suggested to be the center of the house in case of a small estate with one house situated near the center of the largest circle inscribed in the real estate polygon [4, 5]. The coordinates of this latter point are given in two main cases, first if there is no house on the estate, and second as supplementary information to the house coordinates if the distance between the house and the central point is larger than 100 meters.

The Swedish land use map in scale 1:10000 and its parallel coordinate system covering the whole of Sweden are supposed to be used when the real estate coordinates are determined. Local maps in a larger scale (1:2000) available for towns and other larger agglomerations may also be used, but then the transformation between the coordinate systems must be very easy to perform.

A coordinate registration of real estates will determine the positions of many data, not only those included in the real estate register but also all data in other registers associated with it.

The techniques described here to locate areal data to central points, and to determine the coordinates of these points is in Sweden known as the coordinate method. The coordinates are used in two different ways. It is easy to find the map sheet that an estate belongs to by means of its coordinates. The real estate coordinates make it possible for a computer to deal with areal data according to their location which is the most important reason to establish the Swedish real estate coordinate register.

2. Computing data for squares by means of coordinates. The program NORK.

The ALGOL program NORK is the simplest of the programs dealing with the point-in-polygon problem. In this case the polygons are squares, all equal in size and organized in a regular quadratic grid net. They are not allowed to overlap and they must cover the whole area under consideration (figure 1). NORK is also the simplest of all mapping programs and the result is a common square net map (figure 1). More complicated mapping programs allow overlapping and require a triangular grid net. [1, 4].

procedure *NORK*(*x*0, *y*0, *l*, *w*, *h*, *map*);
integer *x*0, *y*0, *l*, *w*, *h*;

```
array map;
begin integer r, c;
  real x, y, data;
  for r := 0 step 1 until l do
  for c := 0 step 1 until w do map[r, c] := 0;
  New Point:
  x := read; if x > ₁₀7 then go to End;
  y := read; data := read;
  r := entier((y - y0)/h); c := entier((x - x0)/h);
  if 0 ≦ r ∧ r ≦ l ∧ 0 ≦ c ∧ c ≦ w then
  map[r, c] := map[r, c] + data;
  go to New Point;
  End:
end NORK;
```

The procedure NORK includes the following parameters:

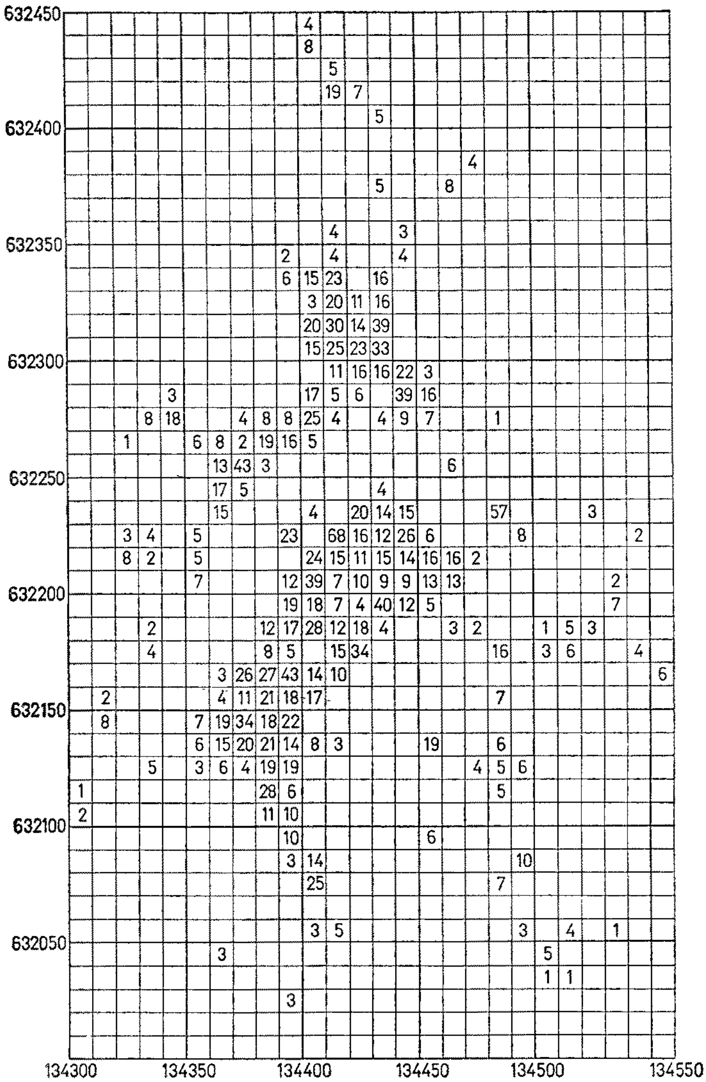
$(x0, y0)$ are the coordinates of the origin, which is the lower left corner of the map, and l is the length of the map. Hence, the number of rows is equal to $(l+1)$.

The width of the map is denoted by w and the number of columns is equal to $(w+1)$. Further, h is the size of the sides of the squares and also the distance between two consecutive grid points, since the squares do not overlap.

Input data are x, y and *data*, x, y being the coordinates of a real estate unit, and *data*, for instance, denoting population, information belonging to the real estate or information about persons living there. It can also be concerned with the real estate itself such as land use, size, value etc. The value of x must be less than 10^7, an exit from the procedure occurring otherwise.

The computer determines the row number (r) by taking the integer part of $(y-y0)/h$. The column number (c) is calculated in the same way from $(x-x0)/h$. The point (x, y) belongs to the map area if $0 \leqq r \leqq l$ and $0 \leqq c \leqq w$. Now *data* is added to cell number (r, c) in the array map. The computer then proceeds to the next point and deals with it in the same way and so on until the mapping is finished.

Figure 1 is a square net map over a small urban place, Hyltebruk, situated in the westernmost part of Småland, Sweden [3]. It shows the number of inhabitants per hectare in 1960. It may be observed how easy it is to get numerical information out of a square net map compared with a dot map, which is the most common kind of distribution map.



Arne Jakobsson 1966.

Figure 1. A population map (coordinate map) over an area around Hyltebruk, a small town in the westernmost part of Småland, Sweden. The map shows the number of inhabitants per hectare (2.471 acres) in 1960.

3. The procedure RECTANGLE.

A polygon is defined by its vertices with known coordinates. By determining a starting point (P_1) and a direction, in this case counter-clockwise, and by enumerating the vertices accordingly, the polygon can be defined as the area situated to the left of all connecting lines between two consecutive vertices. The vertices of the polygon will be designated $P_1, P_2, P_3, \ldots, P_{n-1}, P_n$ where P_n is equal to P_1. Thus the polygon has $n-1$ vertices.

The x-coordinate for the extreme left (west) vertex of the polygon is denoted by W. In the same way and with obvious notations we define E, N, and S. As is evident from figure 2 the polygon in question is si-

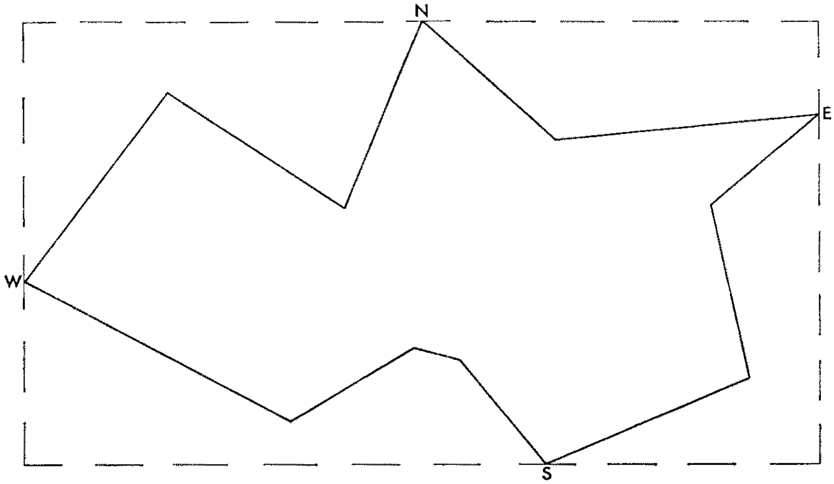


Figure 2. All points outside the rectangle (E, S), (E, N), (W, N), (W, S) are also outside the polygon. $E = x_{\max}$, $W = x_{\min}$, $N = y_{\max}$, and $S = y_{\min}$. (x, y) are the coordinates for the vertices of the polygon.

tuated wholly and entirely within the rectangle (W,S), (E,S), (E,N), and (W,N). This rectangle is one of the rectangles circumscribed round the polygon. A point Q with coordinates (x,y) does not belong to the polygon if $x < W$ or $x > E$ or if $y < S$ or $y > N$. By means of the procedure RECTANGLE the coordinates of the circumscribed rectangle round the polygon are computed and further it is determined whether the point Q belongs to this rectangle or not.

```
Boolean procedure RECTANGLE(x, y, n, P); value x, y, n;
real x, y; integer n; array P;
begin integer i;
  real W, S, E, N, px, py;
  W := E := P[1, 1]; N := S := P[1, 2];
  for i := 2 step 1 until n-1 do
  begin px := P[i, 1]; py := P[i, 2];
```

if $px < W$ **then** $W := px$ **else if** $px > E$ **then** $E := px$;
if $py < S$ **then** $S := py$ **else if** $py > N$ **then** $N := py$
end;
$RECTANGLE := W \leqq x \wedge x \leqq E \wedge S \leqq y \wedge y \leqq N$
end $RECTANGLE$;

The procedure RECTANGLE is also used when there is a whole set of polygons (P_ν), for instance blocks in a town, and the problem is to determine whether a point belongs to one of these polygons P_ν or not. The computer starts by calculating a rectangle circumscribed around all the polygons (see figure 3). In this case the vertices of all the polygons are treated as if they belonged to one and only one polygon. The x-coordinate of the westernmost vertex of the westernmost polygon is designated $Wmin$, and $Emax$, $Smin$, and $Nmax$ are defined in a similar way. Obviously, a point $Q(x,y)$ does not belong to any polygon P_ν if $x < Wmin$ or $x > Emax$ or if $y < Smin$ or $y > Nmax$. If the point Q belongs to the rectangle the individual polygons and their circumscribed rectangles are then treated one at a time.

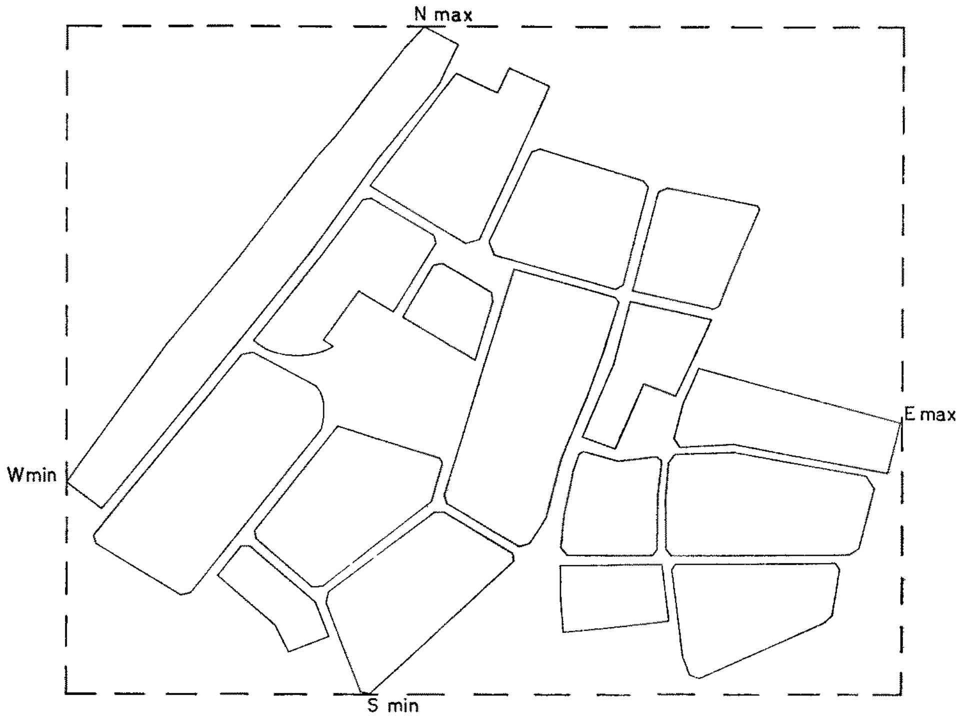


Figure 3. A rectangle is circumscribed a set of polygons. A point outside this rectangle does not belong to any one of the polygons.

4. The circumscribed circle.

Instead of using the circumscribed rectangle and the procedure RECTANGLE we may use the smallest circle circumscribing the polygon and the procedure CIRCUM CIRCLE. The circle is determined by the location of its center (CP) and the length of its radius (R). It is clear that the point Q lies outside the polygon P if the distance between Q and the center CP is greater than the length of the radius R.

The computer starts by calculating the length of the longest diagonal of the polygon which is then taken as diameter in a circle offering the first approximation of the circumscribed circle. This approximation is identical with the smallest circumscribed circle if all the vertices lie inside it. If this is not true the most remote vertex outside the circle is chosen to form a triangle, and a new circle passing through the three vertices of the triangle is constructed. This is the second approximation of the circumscribed circle. If it is not the right one the polygon corner lying farthest away from the circle is chosen and three new triangles are created. It is determined if any one of these triangles is the correct one. If not, the procedure is repeated, possibly by going back taking another of the first three triangles, but usually the iteration is finished well before that stage.

The procedure CIRCUM CIRCLE has some disadvantages compared with RECTANGLE. The determination of the circumscribed circle in most cases takes more time than the construction of the rectangle. The procedure RECTANGLE also generally excludes more points than CIRCUM CIRCLE. Of course, using both these procedures would exclude more points than using just one. However, the gain is very small and it can even be a loss since the use of the combination of the two procedures sometimes takes more time than it saves. It is therefore better to work with a point-in-polygon program directly after the procedure RECTANGLE has been passed.

5. The intersection between the polygon *P* and a straight line through the point *Q*.

An arbitrary line through the point Q (cf. figure 4) intersects the sides of the polygon an odd number of times to the left (or to the right) of Q if this point lies inside the polygon. The coordinates of the point Q are designated $(x0, y0)$. As a rule, the polygon is intersected by the line $y = y0$ parallel to the x-axis. It must be observed that the coordinates of the point Q are here designated $(x0, y0)$ and not (x, y) as in section 3.

The intersection theorem is valid for all polygons, convex as well as

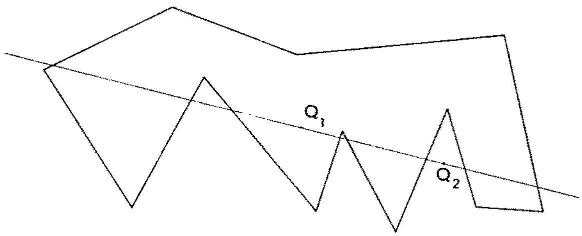


Figure 4. A point Q is situated inside a polygon if a straight line through it intersects the sides of the polygon an odd number of times to the left or to the right of the point. The point Q_1 belongs to the polygon (3 intersections to the left and 5 to the right) while Q_2 does not (2 intersections to the right).

concave. All inner angles of the vertices are less than 180° in a convex polygon. It follows then that there is at least one inner angle greater than 180° in a concave polygon as is shown in figure 4. It may be supposed that this theorem is very valuable since it is valid for all polygons. It was used in a procedure NORP1 published in 1962 [4, 5]. However, it was soon discovered that it had some great disadvantages. For example, no one of the sides of the polygon was allowed to be parallel to the x-axis, and the point Q could not lie on the sides or their extensions. Of course, it would be possible to take care of all these special cases but the program would become too complicated and too slow. It is more economical to use another point-in-polygon program than that based upon the intersection theorem.

6. The sign of the distance between the point Q and the sides $P_\nu P_{\nu+1}$ of the polygon.

As is evident from figure 5 the point Q lies inside the convex polygon P if for all sides $P_\nu P_{\nu+1}$ the distance between Q and the side $P_\nu P_{\nu+1}$ has

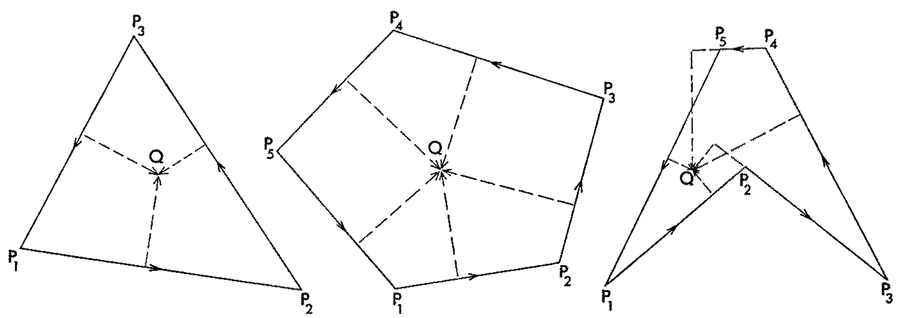


Figure 5. A point Q lies inside a convex polygon if the distance between Q and the side $P_\nu P_{\nu+1}$ has the same sign as the distance between Q and vertex $P_{\nu+2}$. $\nu = 1, 2, 3, \ldots, n-1$. This theorem is not valid for concave polygons.

the same sign as the distance between the vertex $P_{\nu+2}$ and the side. The signs of these distances are determined by inserting the coordinates of the point $Q(x,y)$ and of the vertex $P_{\nu+2}(x_{\nu+2}, y_{\nu+2})$ into the equation of the side written in normal form.

This theorem is not valid for concave polygons. The point Q does belong to the polygon to the right in figure 5 but the sign of the distance between the vertex P_4 and the side P_2P_3 is different from the sign of the distance between Q and P_2P_3.

7. The sum of the angles in the point Q in the triangles $P_\nu P_{\nu+1} Q$.

The polygon P has $n-1$ vertices and $n-1$ sides. It is divided into $n-1$ triangles by the connecting lines between Q and the vertices. Q belongs to the polygon if the sum of the $n-1$ angles in Q is equal to 360°. The point Q does not belong to the polygon if this sum is equal to 0°. (See figure 6). All angles in Q are counted with the proper sign.

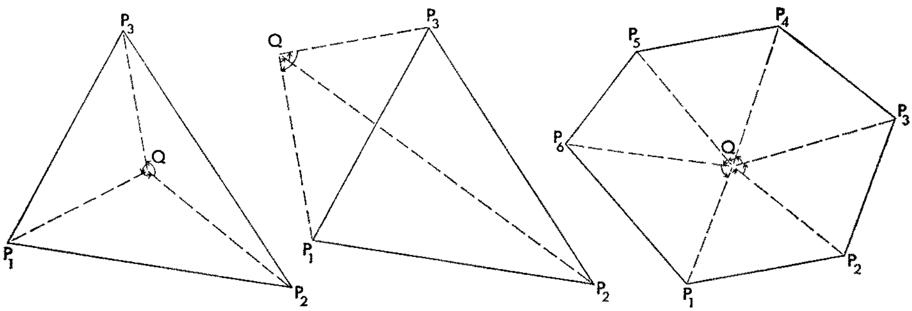


Figure 6. A polygon is divided into $n-1$ triangles by the connecting lines between the point Q and the vertices. Q belongs to the polygon if the sum of the angles is equal to 360°. The point also lies inside the convex polygon if the sum of the areas of the triangles is equal to the area of the polygon.

Considerable rounding errors can be accepted when the angles are determined except when the point Q lies near a side $P_\nu P_{\nu+1}$ or its extension.

A computer can determine whether an angle in Q is greater, equal to or less than 0° by investigating the sign of the area (the orientation) of the triangle under consideration. The point Q belongs to the side $P_\nu P_{\nu+1}$ or its extension if this area is equal to zero.

The greatest disadvantage with a program depending on this theorem is that it is quite slow due to the fact that the computation of the angles is very time-consuming. The program also includes a procedure which calculates the area and the orientation of every triangle, and this procedure can also be used as an independent point-in-polygon program (see the procedure NORPCONVEX).

8. The sum of the areas of the triangles $P_\nu P_{\nu+1} Q$.

The polygon P is divided into $n-1$ triangles by the connecting lines between the vertices of the polygon and the point Q. As is evident from figure 6 the point belongs to a convex polygon if the sum of the areas of the triangles is equal to the area of the polygon. The areas must be taken with their absolute values, the theorem otherwise being valid for all points and all polygons independent of the location of the point Q outside or inside of the polygon. It should be observed that the point-in-polygon theorem on the sum of the area of the triangles is not valid for concave polygons.

One advantage of this theorem is that it is very easy to calculate the areas of the polygon and the triangles by means of the determinant formula without any serious round-off errors. The area of the polygon is designated T_n where n is equal to the number of vertices of the polygon plus 1. The determinant formula is stated as follows:

$$2T_n = \{P_1, P_2, P_3, \ldots, P_{n-1}, P_n\} = \begin{Bmatrix} x_1 & x_2 & x_3 & \ldots & x_n \\ y_1 & y_2 & y_3 & \ldots & y_n \end{Bmatrix} =$$

$$= x_1 y_2 + x_2 y_3 + x_3 y_4 + \ldots + x_{n-1} y_n -$$

$$- (y_1 x_2 + y_2 x_3 + y_3 x_4 + \ldots + y_{n-1} x_n)$$

Thus, the area of a polygon is equal to the sum of $2(n-1)$ simple products between the x-coordinate and the y-coordinate of two consecutive vertices [5]. This formula is also used when the orientations of the polygon and the triangles are to be determined.

9. The orientation theorem for convex polygons.

All point-in-polygon theorems described here have serious disadvantages. Most of them are not valid for concave polygons. It is not possible in any case to establish that a point does or does not belong to the polygon until the computer has dealt with all the sides. There are several special cases such as a point being situated on a side or its extensions. These disadvantages have the effect that a program using one of these theorems will be complicated and rather slow.

The connection lines between a point Q and the vertices divide a triangle into three other triangles as is seen in figure 7. The vertices of this triangle are given in a counter-clockwise order. The orientation of

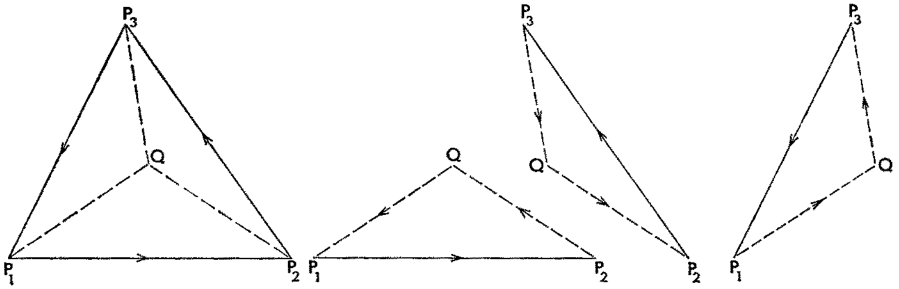


Figure 7. A point Q lies inside the convex polygon P if the triangles $P_\nu P_{\nu+1}Q$ all have the same orientation as the polygon.

the triangle is positive if the area is greater than zero when calculated by means of the determinant formula.

The lines QP_1, QP_2 and QP_3 divide the triangle $P_1P_2P_3$ into three other triangles P_1P_2Q, P_2P_3Q and P_3P_1Q (see figure 7). The connecting lines between a point Q in the same way divide the polygon $P_1P_2P_3 \ldots P_{n-1}P_n$ (where $P_n = P_1$) into $n-1$ triangles $P_\nu P_{\nu+1}Q$, $\nu = 1,2,3,\ldots,n-2, n-1$. The orientation theorem states that the point Q belongs to the convex polygon P if the triangles $P_\nu P_{\nu+1}Q$ all have the same orientation as the polygon. As is evident from figure 7 the orientations of the new triangles are determined by the orientation of the side $P_\nu P_{\nu+1}$.

The orientation point-in-polygon theorem also states that a point Q does not lie inside the convex polygon if one of the triangles has an opposite orientation compared with the polygon (see figure 8). The orientation of the triangle P_3P_1Q is negative while the given triangle is positively oriented. It follows then that Q is situated outside the triangle.

It could not be decided whether the point lies outside the triangle until the side (and the triangle) number 3 had been used in the calculation

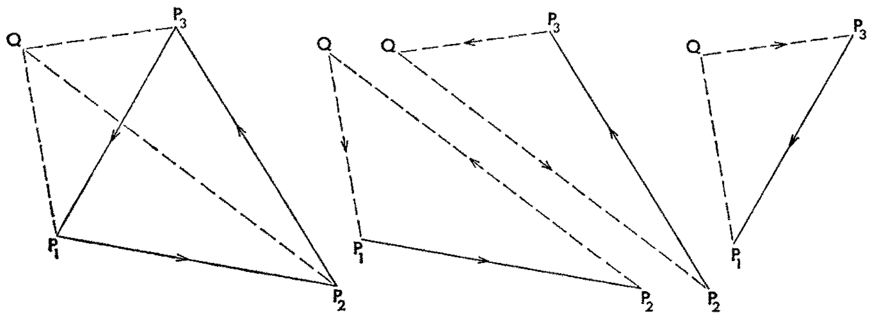


Figure 8. A point Q does not belong to the convex polygon P if one of the triangles $P_\nu P_{\nu+1}Q$ has an opposite orientation compared with the polygon.

in this example. However, a point Q situated outside the polygon in most cases is eliminated earlier. Let us assume that there are many points Q in the example in figure 8 and that they are uniformly distributed over the area around the triangle. The probability that a point Q lying outside the polygon will be eliminated by means of side number 1 (P_1P_2) is as high as 0.5. That means that 5 points out of 10 will be eliminated in this case. On the conditions which are valid in the example of figure 8, about 4 points out of 10 will be eliminated by side number 2. The probability that Q is situated as in figure 8 and that it is not eliminated until side number 3 is not more than 0.1.

The more sides there are in a polygon the less is the probability that a point Q lying outside the polygon will not be eliminated until we reach the last side. A very rough estimate shows that 50% of the points Q will be eliminated by side 1, 60–90% by sides 1 and 2 together, 70–95% by sides 1, 2 and 3, 80–100% by sides 1, 2, 3 and 4. The elimination percentages depend on the angles between side number 1 and the other sides. The larger the outer angle between side 1 and an other side, the more points Q are eliminated. If this outer angle is greater than 180°, almost 100% of the points will be eliminated. In this case the extension of the side number ν intersects the extension of side 1 to the left of vertex P_1. The direction left is explained by the fact that the vertex P_2 lies to the right of P_1.

The orientation theorem also takes care of the case when the point Q lies on a side or its extension. The sign of the triangle $P_\nu P_{\nu+1}Q$ will then be equal to zero. The point Q belongs to the extensions of the side if the sign of $x-x_\nu$ is equal to the sign of $x-x_{\nu+1}$. In this case it is also true that the point Q belongs to another triangle having opposite orientation compared with the polygon.

The orientation "point-in-polygon theorem" has many advantages compared with the other theorems presented in this paper. A procedure using this theorem automatically takes care of all special cases. It is not surprising that it is much faster than any other procedure which depends on some of the other theorems. The best way of determining the direction of the distance between the point Q and the sides $P_\nu P_{\nu+1}$ is to use the orientation theorem. The point Q does not belong to the polygon if one of the angles of the triangles surrounding Q is negative. This can be determined by means of the sign of the sine of the angle here designated A. This is best done by using the formula $\sin A = 2T/(r_\nu r_{\nu+1})$ where T is the area of the actual triangle, r_ν and $r_{\nu+1}$ are the positive distances from Q to P_ν and $P_{\nu+1}$ respectively. The sign of $\sin A$ obviously is identical with the sign of the area of the triangle T. The areas of the polygon and the triangles ought to be calculated by use of the determinant formula [4]. Hence, all these theorems depend upon the orientation theorem. Of course, this basic theorem ought to be used independently because the compound theorems are more complicated and less effective.

10. The point-in-polygon procedure NORPCONVEX.

The point-in-polygon procedure NORPCONVEX is based on the orientation theorem. It determines whether a point lies outside a polygon (-1) or whether it is situated inside the polygon ($+1$) or whether it belongs to one of the sides (0). The following parameters are included: n which as usual means that the polygon has $n-1$ vertices; P, a two-dimensional array which consists of the coordinates of the polygon vertices. Input data are x and y, the coordinates of the actual point Q. The area of the triangle $P_iP_{i+1}Q$ is denoted by a.

```
integer procedure NORPCONVEX(n, x, y, P); value n, x, y;
integer n; real x, y; array P;
begin integer i; real a;
  for i := 1 step 1 until n-1 do
  begin a := P[i, 1]×P[i+1, 2]+P[i+1, 1]×y+x×P[i, 2]-P[i+1, 1]
     ×P[i, 2]-x×P[i+1, 2]-P[i, 1]×y;
     if a < 0 then go to OUT else if a = 0 then
     begin if sign(x-P[i, 1]) = sign(x-P[i+1, 1]) then go to OUT
        else go to ON
     end
  end;
IN: NORPCONVEX := 1; go to END;
OUT: NORPCONVEX := -1; go to END;
ON: NORPCONVEX := 0;
END:
end NORPCONVEX;
```

11. The orientation theorem and concave polygons.

A concave polygon can always be divided into two or more convex polygons. It is true that if a point belongs to one of these convex polygons, it also belongs to the concave polygon. Hence, the procedure NORPCONVEX can be used even when it is to be determined whether or not a point lies inside a concave polygon. However, it can be quite complicated for the computer to divide the concave polygon into convex polygons.

As is evident from figure 9 the concave polygon can always be made convex by including the outer triangles of the concave vertices. The computer starts by determining if the polygon is concave or convex by aid of the procedure CONVEX.

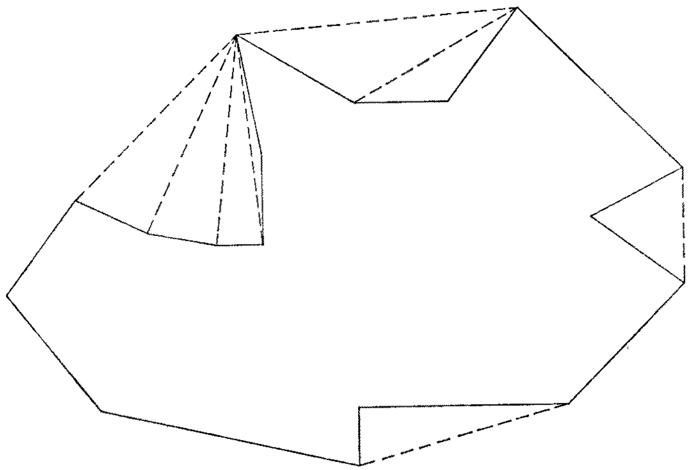


Figure 9. A concave polygon can always be made convex by including triangles around the concave vertices.

Boolean procedure *CONVEX*(n, P); **integer** n; **array** P;
begin integer i;
  for $i := 1$ **step** 1 **until** $n-1$ **do**
  if $P[i, 1] \times P[i+1, 2] + P[i+1, 1] \times P[i+2, 2] + P[i+2, 1] \times P[i, 2] -$
  $P[i+1, 1] \times P[i, 2] - P[i+2, 1] \times P[i+1, 2] - P[i, 1] \times P[i+2, 2] < 0$
  then begin *CONVEX* := **false**; **go to** *END* **end**;
  CONVEX := **true**;
END:
end *CONVEX*;

As before the polygon has $n-1$ vertices, P_n is equal to P_1 and P_{n+1} is equal to P_2. The orientation theorem is used when determining whether a vertex P_ν is concave or not, since the triangle $P_{\nu-1}P_\nu P_{\nu+1}$ is of opposite orientation compared with the polygon if P_ν is a concave vertex. The computer adds the triangle to the polygon and investigates if the polygon is now concave. The orientation of the new triangle $P_{\nu-1}P_{\nu+1}P_{\nu+2}$ is determined. This triangle is included in the polygon if the new vertex $P_{\nu+1}$ is concave. The computer continues with $P_{\nu+2}$ etc until it finds a new vertex, for instance $P_{\nu+k}$ which is convex. The old vertex $P_{\nu+k+1}$ is now included in the calculations and the computer continues with the old vertices $P_{\nu+k+2}$, $P_{\nu+k+3}$ etc. until it finds a new concave vertex which is then treated in the same way. All the new triangles added to the concave polygon are stored. It is now true that the point Q does not belong to the concave polygon if it lies outside the convex polygon or inside one of the new triangles. Obviously, the point Q belongs to the concave polygon if it belongs to the convex one but to none of the new triangles (also cf. figure 9).

The greatest disadvantage of the point-in-polygon theorems for concave polygons just described is in many cases that so many new triangles must be added that the procedure becomes very slow.

12. Concave polygons and the enlarged orientation theorem.

The enlarged orientation theorem consists of two different parts most easily explained by means of figure 10. The first part asserts that a point Q belongs to a concave polygon if it is situated closer to the nearest side than to the nearest vertex and if the triangle formed by this side and the point Q has the same orientation as the polygon. The first part of the theorem is valid for the point Q_1 in figure 10. This point is situated closer to its nearest side ($P_\nu P_{\nu+1}$) than to its nearest vertex, $P_{\nu+1}$ or maybe P_μ, since the distance $QP_{\nu+1}$ is very close to or equal to the distance QP_μ. The orientation of the triangle $P_\nu P_{\nu+1} Q_1$ is positive. Hence the point Q_1 lies inside the actual concave polygon.

The second part of the theorem states that a point Q belongs to a concave polygon if it is situated closer to its nearest vertex than to its nearest side if this vertex is concave. This second part of the theorem is

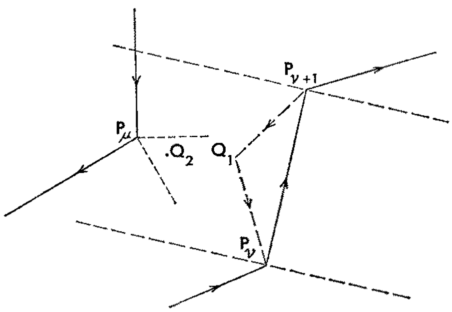


Figure 10. The orientation theorem for concave polygons consists of two different parts:
1. A point Q_1 belongs to a concave polygon if it lies closer to its nearest side $P_\nu P_{\nu+1}$ than to its nearest vertex and if the orientation of the triangle $P_\nu P_{\nu+1} Q$ is equal to the orientation of the polygon.
2. A point Q_2 lies inside a polygon if it is situated closer to its nearest vertex P_μ than to its nearest side and if this vertex P_μ is a concave one.

valid for the point Q_2 in figure 10. The distance between the concave vertex P_μ and the point Q_2 is much less than the distance between Q_2 and the side $P_\nu P_{\nu+1}$. It follows then that the point Q_2 belongs to the polygon.

The enlarged orientation theorem is valid for all polygons, convex as well as concave ones. However, it is very uneconomic to use it in the case of convex polygons because the simple orientation theorem is much better in this case, and further the procedure NORPCONVEX is very fast compared with the procedure based on the enlarged theorem. It is therefore very convenient to use the procedure CONVEX to determine whether the polygon is convex or not and use NORPCONVEX in all cases when the polygons are convex.

13. A mathematical proof of the enlarged orientation theorem.

A concave polygon P and a point Q are given according to figures 11 and 12. A new figure is constructed in the following way. The sides of the polygon are shifted inwards or outwards the same distance, the directions of the sides being kept unchanged, until the point Q is situated on the boundary of the new figure. A side is never allowed to become larger than it was before, and if necessary the vertices are replaced

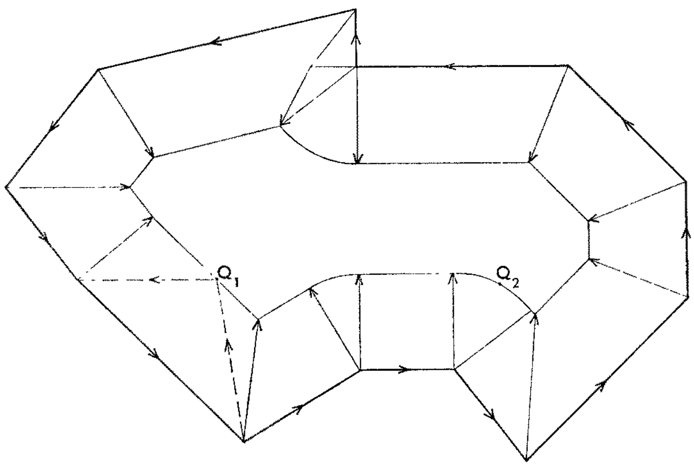


Figure 11. The sides of a polygon are shifted by the same amount until the boundary of the new figure contains the point Q.

The shift is always perpendicular to the original side. If necessary, the vertices are replaced by circular arcs. The shift parameter is $=\min(h, d)$ where h is the distance from Q to its nearest side (Q_1) and d the distance from Q to its nearest vertex (Q_2). The point Q belongs to the polygon if the sides had to be shifted inwards.

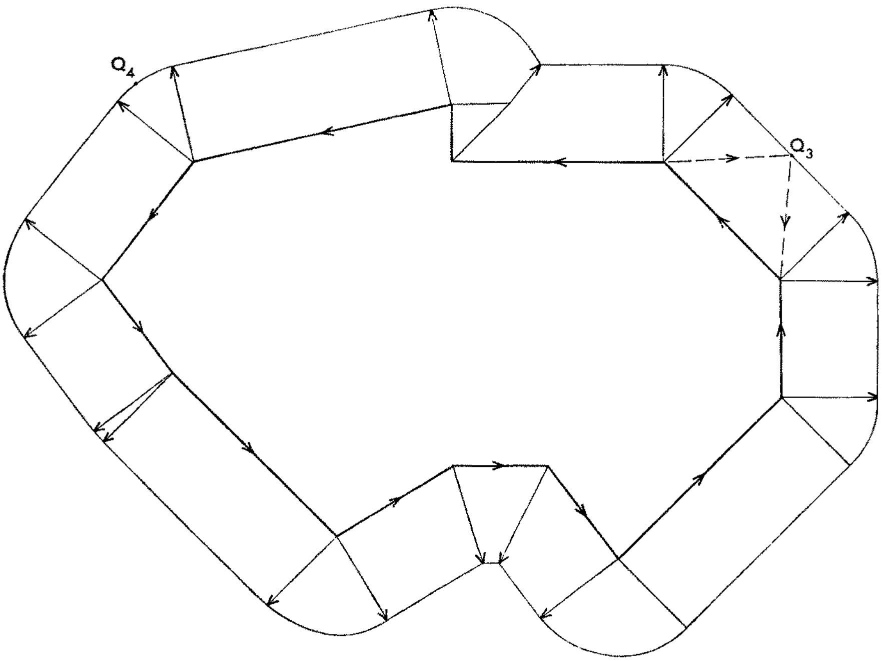


Figure 12. A point Q does not lie inside the polygon if the sides have to be shifted outwards in order to reach the point Q.

by circular arcs with radius equal to the perpendicular displacement. Hence, the constructed figure consists of not more than $n-1$ straight lines corresponding to the sides of the polygon and a number of circular arcs corresponding to the concave vertices if the shift has been made inwards, and the convex vertices otherwise. (see figures 11 and 12). The point Q lies inside the polygon if the sides must be shifted inwards in order to include the point Q. It is also obvious that Q does not belong to the polygon if the sides had to be shifted outwards in order to reach the point Q and that Q belongs to one of the sides if no shift at all was necessary.

The length of the parallel displacement is defined as the distance h from the side $(P_\nu P_{\nu+1})$ closest to the point Q if h is less than the distance d from Q to the nearest vertex P_μ. If this distance d is the smaller one it is used as displacement parameter. In the first case the direction of the shift is determined by means of the orientation theorem for the triangle formed by the nearest side and the point Q. In the second case $(d<h)$ the shift is directed inwards if the vertex P_μ is concave and outwards if P_μ is convex. As is evident from figure 11 and 12 the distance from Q

to a convex vertex is always greater than the distance to the corresponding sides if Q lies inside the polygon. The same theorem is valid for a concave vertex and a point outside the polygon.

The distance from the point Q to the vertex P_ν is determined by means of the Pythagorean theorem. The distance h from Q to the side $P_\nu P_{\nu+1}$ is calculated by aid of the formula $h = 2T/r_{\nu,\nu+1}$ where T is the area of the triangle $P_\nu P_{\nu+1}$ determined for instance by the determinant formula, and $r_{\nu,\nu+1}$ is the length of the side $P_\nu P_{\nu+1}$. It must also be observed that the angles $QP_\nu P_{\nu+1}$ and $QP_{\nu+1}P_\nu$ both must be less than 90° (or greater than 270°) if the distance d is to be determined (see figures 10, 11 and 12).

14. The procedure NORP.

The parameters and input data of the procedure NORP are equal to those of NORPCONVEX. As before d is the distance from the point Q with the coordinates (x,y) to a vertex of the polygon and h the distance between Q and a side of the polygon.

The procedure NORP contains a real procedure DET which determines the double area ($2T$) of a triangle with the vertices $(X1, Y1)$, $(X2, Y2)$ and $(X3, Y3)$ by means of the determinant formula.

The computer determines the square of the smallest d-value (*dmin*), i.e. the distance from the point Q to its nearest vertex. The number of this corner (j) is also stored.

```
integer procedure NORP(n, x, y, P);
value n, x, y; integer n; real x, y; array P;
begin integer i, j, k; real dmin, d, hmin, h;
  real procedure DET(X1, Y1, X2, Y2, X3, Y3);
  value X1, Y1, X2, Y2, X3, Y3;
  real X1, Y1, X2, Y2, X3, Y3;
  begin DET := X1 × Y2 + X2 × Y3 + X3 × Y1 − X2 × Y1 − X3 × Y2 −
     X1 × Y3
  end DET;
  dmin := hmin := ₁₀6;
  for i := 1 step 1 until n − 1 do
  begin d := (P[i, 1] − x) ↑ 2 + (P[i, 2] − y) ↑ 2;
     if d = 0 then go to ON;
     if d < dmin then
     begin dmin := d; j := i
     end
  end;
```

```
    for i := 1 step 1 until n-1 do
    begin if (x-P[i, 1])×(P[i+1, 1]-P[i, 1])+(y-P[i, 2])×
      (P[i+1, 2]-P[i, 2]) ≧ 0 ∧ (P[i, 1]-P[i+1, 1])×(x-P[i+1, 1])+
      (P[i, 2]-P[i+1, 2])×(y-P[i+1, 2]) ≧ 0 then
      begin h := abs(DET(P[i, 1], P[i, 2], P[i+1, 1], P[i+1, 2], x, y))/
        sqrt((P[i+1, 1]-P[i, 1])↑2+(P[i+1, 2]-P[i, 2])↑2);
        if h = 0 then go to ON;
        if h < hmin then begin hmin := h; k := i end
      end
    end;
    if hmin×hmin < dmin then
    begin if DET(P[k, 1], P[k, 2], P[k+1, 1], P[k+1, 2], x, y) < 0
      then go to OUT else go to IN
    end;
    if j = 1 then begin P[0, 1] := P[n-1, 1]; P[0, 2] := P[n-1, 2] end;
    if DET(P[j-1, 1], P[j-1, 2], P[j, 1], P[j, 2], P[j+1, 1],
      P[j+1, 2]) < 0
    then go to IN;
  OUT: NORP := -1; go to END;
  ON: NORP := 0; go to END;
  IN: NORP := 1;
END:
end NORP;
```

By computation of the cosines of the angles $QP_\nu P_{\nu+1}$ and $QP_{\nu+1}P_\nu$ it is determined whether the point belongs to the strip limited by the two parallel straight lines perpendicular to $P_\nu P_{\nu+1}$ through the vertices P_ν and $P_{\nu+1}$, this being the case if both cosines are larger than zero. The value of h is calculated by aid of the formula $h=2T/r_{\nu,\nu+1}$. The smallest h-value ($hmin$) is determined and the number of the corresponding side (k) is stored.

The point lies closer to its nearest side than to its nearest vertex if $(hmin)^2 < dmin$. Using the orientation theorem we see that the point Q belongs to the polygon if the area of the triangle $P_kP_{k+1}Q$ is greater than zero. It also belongs to the polygon if $dmin < (hmin)^2$ and if the nearest vertex is concave.

15. Circles and the procedure NORI.

The procedure NORK did not allow overlapping of the reference squares and it used a quadratic grid net. The procedure NORI works with reference circles which are allowed to overlap and a modified regular triangular grid net. The distance between two consecutive grid points in the same row is denoted by h which is also the distance between two consecutive rows. The length of the connecting line between two consecutive grid points situated in two different rows is $h\sqrt{5}/2$ or approximately $1.1h$. This means that the odd rows are shifted $0.5h$ to the right. The overlapping constant s is equal to the radius of the reference circle given in the transformed coordinate system with unit length h and with the origin in the map origin. The coordinates of a point (x,y) in the untransformed system has the coordinates (a,b) in the new coordinate system (see figure 13). The other parameters required by NORI are the same as those in NORK (see section 2 above).

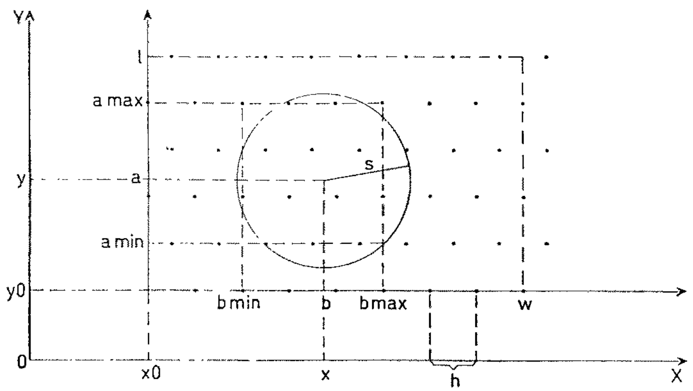


Figure 13. The parameters required by the procedure NORI.

```
procedure NORI(x0, y0, l, w, h, s, map);
value x0, y0, l, w, h, s;
integer x0, y0, l, w; real h, s; array map;
begin integer r, c, amin, amax, bmin, bmax, one;
  real eps, ss, x, y, data, a, b, hf, aq;
  Boolean aeven;
  for r := 0 step 1 until l do
  for c := 0 step 1 until w do map[r, c] := 0;
  eps := ₁₀-6; ss := s × s;
 New Point:
  x := read; if x < 0 then go to End;
  y := read; data := read;
  a := (y-y0)/h; b := (x-x0)/h;
  amin := entier(a-s-eps+1); amax := entier(a+s);
```

```
    if amin < 0 then amin := 0; if amax > 1 then amax := 1;
    if amax < amin then go to New Point;
    bmin := entier(b-s-eps+1); bmax := entier(b+s);
    if bmin < 0 then bmin := 0; if bmax > w then bmax := w;
    if bmax < bmin then go to New Point;
    aeven := amin = amin÷2×2;
    one := if bmin = 0 ∨ abs(b-bmin-s) ≦ 0.5 then 0 else 1;
    for r := amin step 1 until amax do
    begin hf := if aeven then 0 else 0.5; aq := (a-r)×(a-r);
      for c := if aeven then bmin else bmin-one step 1 until bmax do
      if aq+(b-c-hf)↑2 ≦ ss then map[r,c] := map[r,c]+data;
      aeven := ¬aeven
    end;
    go to New Point;
  End:
end NORI;
```

The computer calculates the value of a function $f(x,y)$ at the grid points (x,y) arranged in a triangular net convenient for construction of isarithms and isarithmic maps [1, 4]. The function $f(x,y)$, for instance, can be the number of real estates within the distance hs from the points (x,y) or the number of people living in that circle.

16. Applications.

The point-in-polygon procedures are used when the coordinates of points with assigned data are available and when it is desirable to obtain information and summarized data for arbitrary non-administrative areas such as blocks or combinations of blocks or no-longer existing administrative units. Of course, there are many other cases when the coordinate method and the point-in-polygon procedure can be used. Such an example will be given here.

In most countries urbanized areas are defined by means of population densities. Thus an urbanized area in USA was defined in 1960 as an area with at least 50000 inhabitants. All enumeration districts around it having less than 2500 inhabitants per square mile are excluded. Exclaves situated near the urbanized area are also included if they have typical urban functions.

In the Scandinavian countries urbanized areas are defined with the aid of dwelling house density. Thus, the Swedish "tätort" (urbanized area, built-up area) is defined as an agglomeration of houses. All houses situated within a distance of 200 meters from a house belonging to the agglomeration are included in the "tätort". Areas having typical urban functions such as parks, industrial areas, traffic areas etc. are also included. The "tätort" must have at least 200 inhabitants.

The Swedish "tätort" is defined in the following way. "Tätort" is a built-up area having at least 200 inhabitants and consisting of all points for which it holds true that there is at least one house within a distance of 100 meters. This definition is equivalent to the following one: A "tätort" is the inner area bounded by the 1-isarithm on an isarithmic map showing the distribution of houses. The reference area is a circle with the radius 100 meters. As an approximation of this house density the population density may be used under the following conditions: same reference circle and a high degree of overlapping.

As is shown in figure 1 Hyltebruk is an agglomeration of houses with a total of more than 200 inhabitants. The map was constructed by aid of the procedure NORK. The side of the reference square is 100 meters but it ought to be 200 meters according to the definition of "tätort". The populations are summarized for all connected 200 meters squares. If this sum is greater than 200 inhabitants, the computer constructs a new map by means of NORI. This must be done with such a high degree of overlapping that the errors from the location of the isarithms are negligible. The program NORIP (published in BIT in 1964 [1]) is then used and the 1-isarithms of the map are constructed. All these isarithms are given by the computer as a polygon. The computer starts working with this polygon by aid of the point-in-polygon procedures and determines the total population of the "tätort". This population is then divided into different classes etc. Further the total area of the "tätort" is computed by aid of the determinant formula for polygons. Hence the delimitation of the boundaries of the urban place and the processing of its data has been made automatically by the computer.

Figure 14 is a map over the same area as figure 1. It shows the 1-isarithm of the population in Hyltebruk (the solid line). The boundaries of the official delimitation of the urban place Hyltebruk is drawn with a dashed line. It must be observed that the 1-isarithm divides Hyltebruk into two built-up areas. The reason for this is that the river Nissan passes through it and that there is an industrial area with a paper mill located close to the river.

The delimitation of a built-up area by aid of the population density ought to be completed with some rules in order to exclude errors like that in figure 14. Such a rule is given by Rikkinen [6]. Exclaves are incorporated into the urban place even if this breaks the 200 meters rule. He suggests that an exclave with H houses be included if the distance (d)

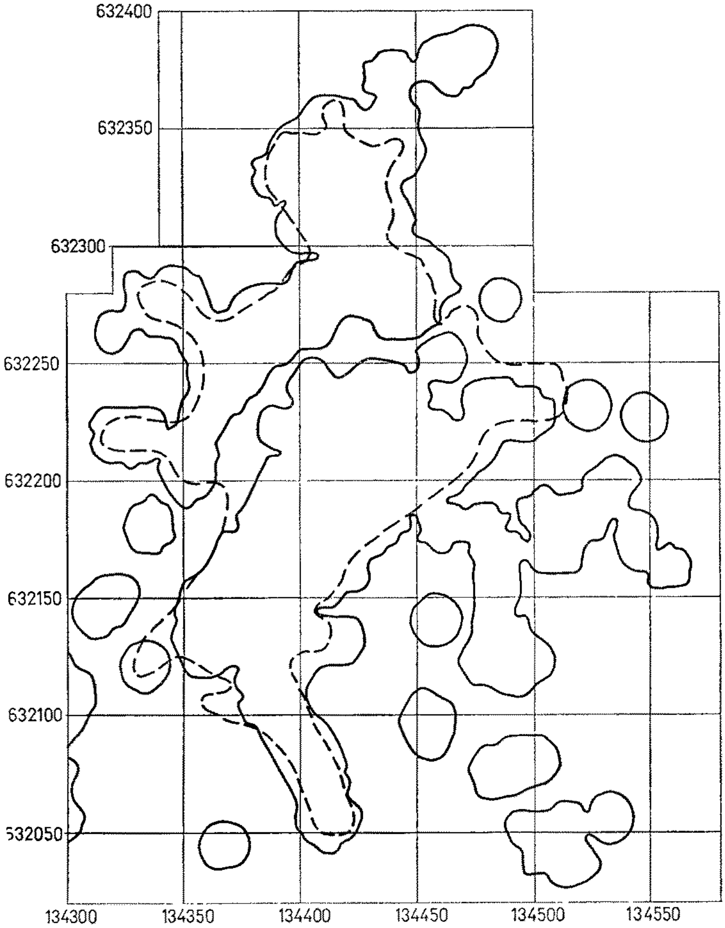


Figure 14. A map over the Hyltebruk area. The dashed line is the official delimitation line of the Hyltebruk built-up area. The solid line is the 1-isarithm of the population. It was calculated by use of the program NORI. The reference area was a circle with radius 100 meters. Note how closely the isarithm and the official boundary follow each other.

between the exclave and the urban place is less than $100\sqrt{H}$ meters. It must be observed that he is working with a 100 meters rule. It seems reasonable to change his formula to $200\sqrt{H}$ according to the Swedish definition of "tätort" and to include the population instead of the number of houses. One urban house is assumed to correspond to 8 persons and instead of using the square root of the population the cube root is used. This revision gives the following formula: $d < 100\sqrt[3]{P}$. This formula means that an exclave will be included into the urban place if the shortest distance between the two corresponding isarithms is less than $(100\sqrt[3]{P}-200)$ meters. The Nissan area in figure 14 is included into the "tätort" Hyltebruk which becomes one urban place if this rule is used. However, it is probable that including all industrial areas into the built-up area is both preferable and easier.

It should be observed how close the 1-isarithm follows the official delimitation of the Hyltebruk built-up area except for the eastern part where quite a large area is neglected. The delimitation of a built-up area by means of computers has one great advantage over a manual one: no areas are neglected.

It is not surprising that the isarithm follows the official delimitation as closely as it does. They are both built on the same definition of "tätort". It follows, then, that an introduction of the isarithmic delimitation of urban areas will not make it impossible to compare now existing statistical data for built-up areas with those which will then become available.

17. Summary.

The Swedish Real Estate Register Committee proposes that the location of every Swedish estate shall be given in the official real estate register by the coordinates of central points belonging to each estate. The coordinates are used when a computer constructs a simple square net map using the procedure NORK. They are also used when the problem is to investigate whether a point Q belongs to a polygon or not.

The orientation theorem for convex polygons is the most outstanding of all "point-in-polygon theorems" discussed here. A triangle is constructed by means of the point Q and a side of the polygon. The point Q belongs to the polygon if it is true that every triangle constructed in the same way as this first one has the same orientation as the actual polygon. The procedure NORPCONVEX uses this theorem.

The orientation theorem has been revised to handle concave polygons as well. There are then two cases.

1. The point is situated closer to its nearest side $(P_\nu P_{\nu+1})$ than to its nearest vertex. It belongs to the polygon if the triangle $(P_\nu P_{\nu+1} Q)$ has the same orientation as the polygon.

2. The point is situated closer to its nearest vertex than to its nearest side. The point lies inside the polygon if this vertex is a concave one.

The procedure NORP uses this more general orientation theorem valid for all polygons. The orientation of the polygons and triangles are determined by means of the determinant polygon area formula. This formula is also used by the procedure CONVEX which determines whether a polygon is convex or concave.

The point-in-polygon programs are used in connection with data assigned to a point (x,y) to compute statistical data about arbitrary polygons such as blocks and no-longer existing administrative units. They can also be used in connection with the isarithmic procedures NORI and NORIP when the computer automatically constructs the boundaries of urban areas. Such an example shows the delimitation of Hyltebruk "tätort'. The 1-isarithm follows the official manual delimitation very closely. Its area is very easily computed by aid of the determinant formula for polygon areas. Data determined by means of the isarithmic delimitations of "tätort" are almost identical with now existing manually determined data. Hence, an introduction of the automatic delimitation of built-up areas will not make historical investigations of the urban areas impossible.

Acknowledgements.

The research for this paper was supported through a grant from The National Swedish Council for Building Research, Stockholm. The procedures published here were all tested on the SMIL computer at the Computer Department, Lund.

The figures were drawn by Miss Elisiv Herbertsson and Mr. Rezsö László, Department of Geography, Lund.

Dr Arne Jakobsson has very kindly allowed us to use his data with coordinates for the real estate properties in the Hyltebruk area and to publish his population coordinate map over this area.

Professor Carl-Erik Fröberg, Department of Computer Science, Lund, has read the manuscript and suggested several corrections and improvements.

To the above-mentioned institutions and individuals we wish to express our grateful appreciation.

REFERENCES

1. Bengtsson, Bengt-Erik and Nordbeck, Stig, *Construction of Isarithms and Isarithmic Maps by Computers*, BIT 1964:2.
2. *Fastighetsregistrering.* Betänkande av fastighetsregisterutredningen angående principiellt system, fastighetsbeteckningar, koordinatregistrering. (Report of the Swedish Real Estate Register Committee.) SOU 1966:63.
3. Jakobsson, Arne, *Omflyttningen i Sverige 1950–1960,* Monographs published by the National Central Bureau of Statistics. Forthcoming.

4. Nordbeck, Stig, *Framställning av kartor med hjälp av siffermaskiner*, Lund 1964.
5. Nordbeck, Stig, *Location of Areal Data for Computer Processing*, LSG. C2. Lund 1962.
6. Rikkinen, Kalevi, *Geographische Abgrenzung von städtischen Agglomerationen*, Annales Academiae Scientiarum Fennicae. Series A. III. Geologica-Geographica 88. Helsinki 1966.

DEPARTMENT OF HUMAN AND ECONOMIC GEOGRAPHY
UNIVERSITY OF LUND
LUND, SWEDEN.