POLYGONS: CALCULATING AREA AND IDENTIFYING INTERIOR POINTS

Richard O. Smith, Science Explorations
Arthur L. Carpenter, California Occidental Consultants

ABSTRACT

There are several programming problems encountered when dealing with polygons of unusual shape. The polygons are often digitized from published maps and reports and then associated with data from other sources. The resulting data are then summarized using SAS/GRAPH® (GMAP or GPLOT). Two of the more interesting of these problems deal with area estimation and the determination of interior points.

Polygons generated from digitized information are stored as a series of sequential x, y coordinate pairs. There is no information in the resulting data set which allows the programmer to determine if any given x, y coordinate pair is inside or outside of the polygon. The problem is, of course, compounded when the polygon has unusual shape or contains interior polygons that define empty space. There are no mathematical formulas for determining the area of such irregular shapes.

The motivating data for this study contained digitized x, y data of complex kelp bed borders. These borders were used to calculate the area of kelp forests and to identify data from related studies containing x, y coordinates that may or may not lie within the boundaries of the kelp forests.

Software was written using SAS® to calculate the areas of kelp beds, and to identify kelp density data points that lie within the borders of the kelp beds. This software calculates the total area of all kelp beds in the area, creates a variable identifying kelp density as inside or outside these kelp beds, and merges this data into a permanent database (Table 1). An example plot is provided as Appendix A that uses the database created in Table 1. This plot displays actual kelp forest borders with kelp density locations labeled differently inside and outside of the kelp beds. The total area of the kelp bed is output onto the plot using the ANNOTATE facility.

METHODOLOGY - Determining the Area

The area of the polygon is determined by calculating the area of a series of individual trapezoids. Each two successive pairs of coordinates defines a trapezoid based on the x axis. The area of the trapezoid is easily calculated and may include area that is either interior or exterior to the polygon or both. Examination of Figure 1 shows that for every trapezoid containing both interior and exterior areas, there is another trapezoid which contains the exterior area only. The difference between the two is the interior area.

Operationally, the polygon area is simply the summation of the areas of all kelp bed borders and to identify data points from other data sources that lie within the borders of the kelp beds. This software can be used on any polygon, the border of which is defined by a sequential series of x, y cartesian coordinate pairs.

INTRODUCTION

During a long term study of the life history of kelp forests in the San Onofre region of southern California, kelp bed borders and kelp density were determined using side-scan and down-looking sonar. The boundaries of the kelp beds were digitized as x, y coordinates. Kelp density was presented as density at specific locations identified by x, y coordinates. The kelp bed boundaries were often very complex, making it difficult to readily determine areas. Additionally, it was necessary to identify kelp density data that were in the interior of the kelp beds.

Software was written using SAS to calculate the areas of kelp beds, and to identify kelp density data points that lie within the borders of the kelp beds. This software calculates the total area of all kelp beds in the area, creates a variable identifying kelp density as inside or outside these kelp beds, and merges this data into a permanent database (Table 1). An example plot is provided as Appendix A that uses the database created in Table 1. This plot displays actual kelp forest borders with kelp density locations labeled differently inside and outside of the kelp beds. The total area of the kelp bed is output onto the plot using the ANNOTATE facility.
absolute value of the summation to determine the polygon area.

In the simplest case shown in Figure 1, where all $x$ and $y$ values are positive, we see that as we travel along the border clockwise (increasing values of $x$) each area is positive (trapezoid $A$) until direction changes toward the $y$ axis (trapezoid $B$). Each trapezoid of type $B$ along the bottom will have a negative area because $x(i)$ is larger than $x(i+1)$. When all areas are summed, the area below the bottom of the polygon will be subtracted from the total area below the top of the polygon.

This technique is also valid for cases where the border of the polygon passes below the $x$ axis or to the left of the $y$ axis. In Figure 2, we see that each area defined by points along the bottom of the polygon will be positive (trapezoid $C$), instead of negative as in Figure 1, because the average $y$ distance to the $x$ axis is negative. Therefore, in this case, areas inside the polygon below the $x$ axis will be added to the polygon areas above the $x$ axis. Passing to the left of the $y$ axis has no effect.

SAS code in the example program (Table 1) calculates the area of each kelp bed first and then sums the total area of all kelp beds. Polygons of empty space that lie inside the interior of larger kelp beds are identified by a location variable with a value of -1. All polygon areas are multiplied by the location variable before calculation of total area.

**METHODOLOGY - Determining Interior Points**

The sequential string of $x,y$ pairs that form the border of the polygon contain information only on the location of the border. Taken as a whole, however, it is possible to determine if a given point ($x,y$ pair), not on the border, is in the interior of the polygon. The method used is based on a series of determinations at constant values of the $y$ variable. If a polygon is cut by a line with constant $y$, the line will intersect the border at one or more locations. If the $x$ value of each data point in the related data set is compared at each $y$ to the $x$ values of the points of intersection from the polygon data set, it is possible to determine which are interior points.

In figure 3, point $A$ must be outside of the polygon because it has an $x$ value which is less than both of the boundary points. By the same token $C$ is also outside because its value of $x$ is larger than both boundary points. $B$ is the...
only interior point because its x lies between the two boundary points. This concept, interior points can be defined by pairs of boundary points, is called the 'Two Points Rule' (TPR) and is the cornerstone of this method.

The difficulty lies in the "what if ...." exceptions to the 'Two Points Rule'. Figure 4 shows a slightly more difficult polygon cut by two lines. The first (L1) cuts the polygon four times and has an imbedded exterior space. The second line (L2) intersects the border three times with the center point at a local minimum. Closer examination shows that L1 is really not an exception to the TPR if the points are taken in order as pairs. Point A is within the first pair and point B is outside of both pairs. L2 results in three intersection points which makes it difficult to form the pairs needed to satisfy the TPR. Pairs can be formed if all local maximum and minimum points are duplicated, this allows the line L2 to behave like L1 by creating an exterior zone of zero width and thus the TPR is satisfied.

![Figure 4](image)

Just as local maxima and minima must be identified, so must points which create borders of constant y values. The line in Figure 5 intersects the border at three successive points and is therefore a potential exception to the TPR. To solve this problem, every second data point along the constant y border is offset either above or below the y axis by 0.5 units of resolution (B') and any local maxima or minima created are doubled as above.

![Figure 5](image)

In the example software (Table 1), locations of kelp density points are identified in the following sequence: 1) It was determined that the data are accurate to +/- 3 meters. A resolution of 1 meter was chosen for the software, and y was rounded to the nearest 1 meter in both the kelp bed border data set and the kelp density data set. 2) The polygon is completed by duplicating the first starting data point at the end of each polygon data set. 3) Borders of constant y values are adjusted, and local maxima and minima are then doubled. 4) New data points are created by interpolating between sequential data points when the change in y is greater than the resolution chosen, thus filling in the border of the polygon. 5) An array of x intersect values is loaded for each value of y, and these values of x are compared against the values stored in the array using the TPR to determine if the point is interior or exterior to the polygon. Interior polygons are merely extensions of the TPR. A database is created containing the original kelp density data with a polygon location identifier for each datum. Total area of the kelp bed is then merged with this database and the results are plotted in Appendix A.

**SUMMARY**

The problems of dealing with digitized polygons are often difficult and mathematical models rarely apply. It is possible to develop tools which allow the programmer to solve some of these problems such as determination of interior points and the calculation of enclosed area.
Table 1. SAS code for area and location (in/out) determination.

* DEKdens SAS *
CALCULATES THE AREAS OF SUBSTRATE POLYGONS CREATED FROM SIDE-SCAN SONAR SURVEY DATA. IDENTIFIES KELP DENSITY DOWN-LOOKING SONAR DATA POINTS AS INSIDE OR OUTSIDE OF THE SUBSTRATE POLYGONS. CREATES A PERMANENT DATABASE. USED TO PLOT KELP BED BORDERS WITH SHOT-POINTS, SUBSTRATE AND KELP AREA.

FLOW CHART:
DBSONDEN.SURO??+DBANNO.SMKSUB?? --> DBKDENS SAS --> DBKDENS.SMKSUK11

OPTIONS DQUOTE MACROGEN SYMBOLGEN MPRINT NOTEXT82;

%MACRO
DOIT(SURVEY,DATE,DBSON,DBSUB,DBKDENS);
* KELP DENSITY DATA SET *
DATA DENS1;
SET DBSONDEN.&DBSON ;
X=XLOC;Y=YLOC;
* ROUNDS TO NEAREST INTEGER *
Y=ROUND(Y,1);
DB="&DBSUB
BED=SUBSTR(DB,1,3);
IF BED='SOK' THEN 00;
IF X>=-50D AND X<=2000 AND
Y<=-500 AND Y>=-2500
END;
IF BED='SMK' THEN 00;
IF X>=-5500 AND X<=-3500 AND
Y=O AND Y>=-1500
END;
KEEP Y X Kdens ;
PROC SORT;BY Y;

* CALCULATE AREA OF KELP BEDS *
DATA AREAl;
SET DBANNO.&DBSUB ;
BY POLYGON;
RETAIN XO YO OX OY ;
IF X=. OR Y=. THEN DELETE ;
IF FIRST. POLYGON THEN DO;
OX=X;OY=Y;XO=.;YO=~;
END:
AREA=((YO+OY)/2)*(X-OX);OUTPUT;
XO=X;YO=Y;
* COPIES FIRST OBS AS LAST OBS TO COMPLETE AN UNFINISHED POLYGON *
IF LAST. POLYGON THEN DO;
LD=OY-OY2
D=Y-OY 
TY=YiTX=X
IF LD=O AND D=. THEN DO:
Y=OY;X=OX;OUTPUT:
Y=TY;X=TX;
END;
IF ABS(D) > 1 THEN DO;
Y=OY + SIGN(D);X=OX;OUTPUT;
Y=TY+SIGN(D);X=TX;
END;
IF ABS(D) = 1 THEN DO;
Y=OY - SIGN(D);X=OX;OUTPUT;
Y=TY-SIGN(D);X=TX;
END:

* SUMS TOTAL AREA OF EACH POLYGON *
DATA AREA2;
SET AREAl ;
PROC MEANS NOPRINT;
VAR AREAl;
BY POLYGON ;ID LOC ;
OUTPUT OUT=AREAl SUM=AREA ;
END;
END;
IF LAST.POLYGON THEN OUTPUT;
OY2=OY;
OY=Y;OX=X;
KEEP POLYGON X Y ;
----------------------------------------;
* CREATE VALUE FOR EACH Y BY
INTERPOLATION *;
DATA POLY2;
SET POLYH;BY POLYGON;
RETAIN OLDY OLDX ;
IF FIRST.POLYGON THEN DO;

OLDY=.;OLDX=. ;
END;
* ELIMINATES EXACT DUPLICATES *;
IF X=OLDX AND Y=OLDY THEN DELETE;
DIFFY=Y-OLDY ;
DIFFX=X-OLDX ;
I=0;
IF ABS(DIFFY) >1 THEN DO
I=SIGN(DIFFY) TO DIFFY BY SIGN(DIFFY);
OLDY+I; OLDX+DIFFX/DIFFY*I;OUTPUT;
END;
ELSE OUTPUT;
OLDY=Y;OLDX=X;
KEEP POLYGON X Y ;
----------------------------------------;
* CREATE DUPLICATE VALUES OF Y
WHEN VERTICAL DIRECTION CHANGES * ;
DATA POLY3;
SET POLY2 ; BY POLYGON;
RETAIN LD OY OX OY2 TY TX FD N ;
IF FIRST.POLYGON THEN DO;
LD=-D=.;OY=.;OX=.;
OY2=.;TY=.;TX=.;FD=.;
N=O;
END;
N=N+1;
LD=SIGN(OY-OY2) ;
D=SIGN(Y-OY) ;
IF N=2 THEN FD=D;
TY=Y;TX=X;
IF (LD ^= D AND LD ^=.) THEN DO;
Y=OY;X=OX;OUTPUT;
TY;TX=TX;
END;
IF LAST.POLYGON AND FD = D THEN DELETE;
OUTPUT;
OY2=OY;
OY=Y;OX=X;
KEEP POLYGON X Y ;
PROC SORT DATA=POLY3;
BY Y X ;
----------------------------------------;
* CALCULATE THE NUMBER OF X PER Y * ;
PROC MEANS DATA=POLY3 NOPRINT;
BY Y ;
VAR X ;
----------------------------------------;
* MERGE TRANSPosed POLYGON DATA
AND MNXDATA WITH SUBJECT DATA ;
DATA KDENS ;
MERGE DENS1(IN=IN1)
MNXDATA
TPDATA(IN=IN2);
BY Y ;
* CREATE A DUMMY VARIABLE USED TO
DIMENSION ARRAY XX *;
EXTRAX = . ;
* SELECTS ONLY Y'S PRESENT IN
SUBJECT DATA *;
IF 'IN1 THEN DELETE ;
* LABEL DATA OUTSIDE OF THE POLYGON
DIFFERENTLY THAN DATA WITHIN *;
I=0 ;
GOOD=0 ;
IF NX ^= . THEN DO;
ARRAY XX (*) XI == EXTRAX;
DO I=2 TO NX BY 2 UNTIL (GOOD);
IF XX(I-1) <= X <= XX(I) THEN GOOD=1;
END;
END;
IF NX -. THEN GOOD= . ;
IF GOOD THEN LOCATION='IN ';
ELSE LOCATION='OUT ';
DROP EXTRAX I GOOD;
----------------------------------------;
* MERGE AREA OF KELP BED WITH
KDENS DATA *;
DATA DBKDENS.&DBKDENS ;
* DUPS AREA FOR EACH OBS OF KDENS *;
IF _N_=1 THEN SET AREA
SET KDENS ;
PROC SORT;BY LOCATION;
PROC PRINT;
TITLE 'DATA PRESENT IN DBKDENS.';
"$DBKDENS";
----------------------------------------;
MEND PLOTIT;
*** MACRO CALLS ***;
%DbIT(1~,'23JAN88'D
,SUROR11,SURSUB26,SURSUR11) RUN;
----------------------------------------;
TRADEMARK NOTICER
SAS and SAS/GRAPH are registered
trademarks of SAS Institute Inc., Cary,
NC, USA.

CONTACT AUTHORS
Richard O. Smith
Science Explorations
2270 Camino Vida Roble, Suite L
Carlsbad, CA 92009
(619) 438-1336

Arthur L. Carpenter
California Occidental Consultants
4239 Serena Avenue
Oceanside, CA 92056
(619) 724-8579
APPENDIX A

Plot of San Onofre kelped substrate borders with kelp density sonar shot-points displayed as inside (x) or outside (+). Kelped area, calculated in Table 1, is placed on the plot with ANNOTATE.