SMOOTH IRREGULAR CURVES

BY PETER A. STEEVES,
B.Sc., M.Sc., P. Eng., Ph.D.

PREFACE

I have been developing geodetic software for the last 14 years; some of the programs that I developed or co-developed are: least squares adjustment of horizontal control, leveling, satellite doppler and photogrammetric data. I also taught the theory for some of these applications at the University of Toronto. Two years ago Mr. Gary Irwin, a former student of mine and an upcoming Ontario Land Surveyor, and I began writing plane survey software for the APPLE II and HP 9816 microcomputers. We have given several demonstrations of our software and the response has been very good. Several of our clients have expressed that they would like to see us present some of our ideas on paper to the Association of Ontario Land Surveyors, so that members might begin to write some of their own survey software and/or be better able to assess several survey computer programs on the market.

We are prepared to offer a series of articles on the development of an interactive coordinate geometry package, plotting, contouring and geodetic applications if there is in fact a demand. We will include subroutines for each subject covered. We will begin with subroutines for intersections, resections, curves and areas. After that we will continue on demand. If you are interested please send your requests to me or the editor of this magazine.

The first article is an algorithm for plotting contour lines. The algorithm assumes that the x and y planar coordinates for a number of points that lie on contour line are already known.

Abstract

AN APPROXIMATION procedure for defining a smooth curve to pass through a set of n arbitrary real data points,

$$[x_1,y_1;\ x_2,y_2;\ .\ .\ .\ .\ .;\ x_n,y_n],$$

assumed to lie sequentially along some unknown planar curve, specifically a contour line, is presented.

Introduction

The most accurate contour line is not a curve but a series of straight line segments drawn between consecutive points having equal elevation; however, contour lines drawn this way are unpleasing to the eye. That which is desired is a smooth curve which deviates little from the straight line segments.

The classical one dimensional interpolation procedures for defining $y = f(x)$ are not applicable to the contouring problem as usually neither x or y increase monotonically. However, the equation $y = f(x)$ can be expressed as $x = g(t)$ and $y = h(t)$, a set of parametric equations. The values of the parameter t will lie in some domain of real numbers. The locus of this pair of equations will be the set of points in the $x-y$ plane which results when t takes on all values in its domain.

Example: If $x = g(t) = t^2 + 2t$, $y = h(t) = t - 2$ eliminate the parameter t.

$$t = y + 2$$
$$x = (y + 2)^2 + 2(y + 2)$$
$$x = y^2 + 4y + 4 + 2y + 4$$
$$x = y^2 + 6y + 8$$

Equations $x = t^2 + 2t$, $y = t - 2$ and $x = y^2 + 6y + 8$ represent the same curve.

The Method

A series of curves are interpolated through points 1, 2 and 3; 2, 3 and 4; 3, 4 and 5; . . . .; $n-2$, $n-1$ and n. Segments of two overlapping curves are blended together as shown in Figure 1. The radius of curvature of the blended curve jk is equal to the radius of curvature of curve ijk at j and is equal to the radius of curvature of curve jkl at k.

Newton's divided-difference interpolating polynomial is used to define $x = g(t)$ and $y = h(t)$ for each three point curve segment; the general equations are:

$$x = x_i + (t - t_i)(x_j - x_i)/(t_j - t_i) + (t - t_i)(t - t_j)((x_k - x_j) - (x_j - x_i))/(t_k - t_i)$$
$$y = y_i + (t - t_i)(y_j - y_i)/(t_j - t_i) + (t - t_i)(t - t_j)((y_k - y_j) - (y_j - y_i))/(t_k - t_i)$$

The parameter t starts at t_i and goes through t_j to t_k. The parameters t_i, t_j and t_k can have any convenient value so long as the relative order $t_i < t_j < t_k$ is maintained. In order to lessen the quantity of arithmetic let t_0, ..., t_i, t_j, t_j, ..., t[illegible] be a set of consecutive integers; then,

$$x = x_i + (t - t_i)(x_j - x_i) + (t - t_i)(t - t_j)((x_k - x_j) - (x_j - x_i))/2 \quad \ldots 1$$
$$y = y_i + (t - t_i)(y_j - y_i) + (t - t_i)(t - t_j)((y_k - y_j) - (y_j - y_i))/2 \quad \ldots 2$$

Figure 1. Blending two Curves into one Smooth Curve

The plane coordinate values x and y can then be solved for any value of t. Since we want to plot a smooth contour line, small increments of t have to be used. The increment can be computed as follows:

$$n = \text{INT}(\text{scale}*((x_k - x_j)^2 + (y_k - y_j)^2)^{1/2}/\text{plotting inc.})$$
$$\text{delta } t = 1/n.$$

The plane coordinate values x and y are computed for both curves in the overlapping area and then the two curves are blended together as follows:

$$x_b = x_1*(\text{INT}(t + 1.0) - t) + x_2*(t - \text{INT}(t)) \quad \ldots 3$$
$$y_b = y_1*(\text{INT}(t + 1.0) - t) + y_2*(t - \text{INT}(t)) \quad \ldots 4$$

Conclusion

The method produces smooth contour lines that deviate little from the straight line segments joining consecutive points of equal elevation. Smoother curves can be obtained by using the accumulated chord distance along the contour line as the basis for t rather than the integer basis used in this paper. Further straightening can be forced by having the computer place additional base points along long straight segments at certain intervals (e.g., at the 1/3 and 2/3 positions). The algorithm coded in BASIC will compute contour lines to be plotted with a 1mm increment at the rate of 8 cm/sec. on the HP 9816 and at the rate of 1.5 cm/sec. on the APPLE II (compiled BASIC).

Reference

Rudeen, K. M., Curve Design (Blending Parabolas on the TRS-80), Creative Computing, February 1984.

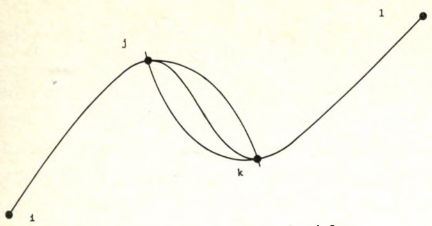

Figure 1. Blending two Curves into one Smooth Curve

```
1000  HOME
1010  HGR : POKE  - 16301,0
1020  DIM XA(200),YA(200),IX(2),I
     Y(2),X(2),Y(2)
1030  VTAB 23: PRINT "<A>=UP  <Z>
     =DOWN    <N>=LEFT  <M>=RIGHT
     "
1040  PRINT "<SPACE BAR> TO SET P
     T  <RETURN> TO STOP"
1050  GOSUB 1660
1060  VTAB 24: INPUT "OPEN (1) OR
      CLOSED (2) CURVE ? ";A
1070  VTAB 24: INPUT "PLOTTING IN
     CREMENT ? ";PL
1080  HOME
1090  FOR I = 1 TO NP
1100  HPLOT XA(I),YA(I)
1110  NEXT I
1120 NS = NP
1130 XA(0) = XA(NP)
1140 YA(0) = YA(NP)
1150 XA(NP + 1) = XA(1)
1160 YA(NP + 1) = YA(1)
1170 XA(NP + 2) = XA(2)
1180 YA(NP + 2) = YA(2)
1190  IF A = 1 THEN NS = NP - 1:X
     A(0) = XA(1) + 0.00001:YA(0)
      = YA(1) + 0.00001:XA(NP + 1
     ) = XA(NP) + 0.00001:YA(NP +
     1) = YA(NP) + 0.00001
1200  GOSUB 1270
1210  VTAB 24: PRINT "REPEAT CURV
     E Y/N ? ";
1220  GET A$: IF A$ = "" THEN 122
     0
1230  IF A$ = "N" THEN 1030
1240  IF A$ = "Y" THEN  HOME : HGR
     : GOTO 1060
1250  END
1260  REM  COMPUTE CURVE
1270 IX(1) = XA(1)
1280 IY(1) = YA(1)
1290  FOR S = 1 TO NS
1300 DIST =  SQR ((YA(S + 1) - YA
     (S)) ^ 2 + (XA(S + 1) - XA(S
     )) ^ 2)
1310 ST =  INT (DIST / PL) + 1
1320 T = S - 1 / ST + 0.00000001
1330  REM  COMPUTE POLYNOMIAL COE
     FFICIENTS
1340 T0 = S - 1:T1 = S:T2 = S + 1
     :T3 = S + 2
1350 C1 = XA(T1) - XA(T0)
1360 C2 = (XA(T2) - 2 * XA(T1) +
     XA(T0)) / 2
1370 C3 = YA(T1) - YA(T0)
1380 C4 = (YA(T2) - 2 * YA(T1) +
     YA(T0)) / 2
1390 C5 = XA(T2) - XA(T1)
1400 C6 = (XA(T3) - 2 * XA(T2) +
     XA(T1)) / 2
1410 C7 = YA(T2) - YA(T1)
1420 C8 = (YA(T3) - 2 * YA(T2) +
     YA(T1)) / 2
1430  REM  INTERPOLATE POINTS ON
     THE
1440  REM  TWO CURVE SEGMENTS
1450  FOR J = 1 TO ST + 1
1460 T = T + 1 / ST
1470  IF T >  = S + 1 THEN T = S +
     0.99999999
1480 A = T - T0:B = T - T1:C = T -
     T2:D = A * B:E = B * C
1490 X1 = XA(T0) + A * C1 + D * C
     2
1500 Y1 = YA(T0) + A * C3 + D * C
     4
1510 X2 = XA(T1) + B * C5 + E * C
     6
1520 Y2 = YA(T1) + B * C7 + E * C
     8
1530  REM  BLEND THE TWO CURVES
1540 LT = T -  INT (T)
1550 RT =  INT (T + 1) - T
1560 IX(2) = X2 * LT + X1 * RT
1570 IY(2) = Y2 * LT + Y1 * RT
1580  REM  PLOT STRAIGHT LINE SEG
     MENT
1590  HPLOT IX(1),IY(1) TO IX(2),
     IY(2)
1600 IX(1) = IX(2)
1610 IY(1) = IY(2)
1620  NEXT J
1630  NEXT S
1640  RETURN
1650  REM  INPUT POINTS DEFINING
     CURVE
1660 X = 140:Y = 75:NP = 0:P = 0:
     DX = 140:DY = 75
1670  HPLOT X,Y
1680  GET A$: IF A$ = "" THEN 168
     0
1690  IF A$ <  > "A" AND A$ <  >
     "Z" AND A$ <  > "N" AND A$ <
      > "M" AND  ASC (A$) <  > 8 AND
      ASC (A$) <  > 10 AND  ASC (
     A$) <  > 11 AND  ASC (A$) <
      > 13 AND  ASC (A$) <  > 21 AND
      ASC (A$) <  > 32 THEN 1680
1700  IF  ASC (A$) = 13 THEN 1910
1710  IF  ASC (A$) <  > 32 THEN 1
     760
1720 NP = NP + 1
1730 XA(NP) = X:YA(NP) = Y:P = 1
1740  GOTO 1680
1750  IF A$ = "M" THEN DX = X + 1

1760  IF  ASC (A$) = 8 THEN DX =
     X - 1
1770  IF  ASC (A$) = 10 THEN DY =
     Y + 1
1780  IF  ASC (A$) = 11 THEN DY =
     Y - 1
1790  IF  ASC (A$) = 21 THEN DX =
     X + 1
1800  IF A$ = "A" THEN DY = Y - 1

1810  IF A$ = "Z" THEN DY = Y + 1

1820  IF A$ = "N" THEN DX = X - 1

1830  IF A$ = "M" THEN DX = X + 1

1840  IF DX < 0 OR DX > 279 THEN
     1680
1850  IF DY < 0 OR DY > 159 THEN
     1680
1860  IF P = 0 THEN  HCOLOR= 0: HPLOT
     X,Y: HCOLOR= 3
1870 P = 0
1880 X = DX:Y = DY
1890  HPLOT X,Y
1900  GOTO 1680
1910  RETURN
2000 :
2010 :
2020  REM   LOAD THE PROGRAM
2030  REM   PRESS <RUN>
2040  REM   MOVE THE DOT TO WHERE

2050  REM   YOU WANT TO MARK A
2060  REM   POINT WITH THE KEYS
2070  REM   <A>=UP, <Z>=DOWN
2080  REM   <N>=LEFT, <M>=RIGHT
2090  REM   OR WITH THE ARROW
2100  REM   ON THE APPLE IIe.
2110  REM   PRESS <SPACE BAR> TO
2120  REM   SET A POINT.  PRESS
2130  REM   <RETURN> TO STOP
2140  REM   ENTERING POINTS.
2150  REM   THE COMPUTER WILL
2160  REM   ASK IF YOU WANT AN
2170  REM   OPEN OR CLOSED
2180  REM   CURVE.  THE COMPUTER
2190  REM   WILL THEN ASK FOR
2200  REM   THE INCREMENT IN
2210  REM   PIXELS.
2220  REM
2230  REM   HAVE FUN !!!!
```