A Fortran subroutine for plotting the part of a conic that is inside a given triangle

S. Marlow and M. J. D. Powell
Computer Science and Systems Division
AERE Harwell, Oxfordshire
April 1976
A FORTRAN SUBROUTINE FOR PLOTTING THE PART OF A
CONIC THAT IS INSIDE A GIVEN TRIANGLE

by

S. Marlow and M.J.D. Powell

Abstract  The given Fortran subroutine approximates part of a conic by a sequence of straight line segments in order that the approximation can be drawn directly by an automatic graph plotter. The main purpose is to use few line segments subject to an accuracy parameter that is set by the user of the subroutine. All parts of the conic are drawn that are inside a given triangle. The conic is specified by function values at the vertices and at the mid-points of the sides of the triangle.

Computer Science and Systems Division,
Atomic Energy Research Establishment,
Harwell, Oxfordshire, England.

April, 1976

HL.76/1183 (C.13)
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>3</td>
</tr>
<tr>
<td>2. Instructions for using subroutine OBL3A</td>
<td>5</td>
</tr>
<tr>
<td>3. The method of calculating the piecewise linear approximation</td>
<td>9</td>
</tr>
<tr>
<td>4. Comments on the Fortran listing</td>
<td>35</td>
</tr>
<tr>
<td>5. Discussion</td>
<td>52</td>
</tr>
<tr>
<td>References</td>
<td>57</td>
</tr>
<tr>
<td>Figure 1</td>
<td>58</td>
</tr>
<tr>
<td>Figure 2</td>
<td>58</td>
</tr>
<tr>
<td>Appendix</td>
<td>59</td>
</tr>
</tbody>
</table>

ISBN 0-70-580476-3

-2-
1. **Introduction**

   It is expected that the main use of the given Fortran subroutine is to help the drawing of contours of the form

   \[ f(x,y) = H, \]  
   \hspace{1cm} (1.1)

   where \( f(x,y) \) is an underlying function and where \( H \) is a parameter, namely the contour height. A usual technique for plotting contours is to divide \((x,y)\)-space into small regions, to make a simple approximation to \( f(x,y) \) on each region, and to plot the contours of the approximation. Our subroutine treats the common case where each small region is a triangle and where each simple approximation to \( f(x,y) \) is a quadratic function

   \[ f(x,y) = ax^2 + 2hxy + by^2 + 2gx + 2fy + c. \]  
   \hspace{1cm} (1.2)

   Instead of the parameters \( a, h, b, g, f, \) and \( c \) we require the user to give the values of the function \( f(x,y) \) at the vertices and mid-points of the sides of the triangular region, because this method of specifying \( f(x,y) \) is good for ensuring continuity of the contour line where it passes from one triangle to another. Figure 1 shows a typical triangle and it provides a notation for the points at which the value of \( f(x,y) \) is given.

   The given subroutine treats only a single triangle. A contour plotting routine for a region that is made up of several triangles will be provided later. It will apply the method described by Powell (1974) and will make use of the present subroutine.

   Because automatic graph plotters are designed to draw straight lines, either directly by hardware or by software that is provided with the machine, the problem of plotting a conic section can be solved by making a sufficiently accurate approximation that is composed of straight
line segments. Therefore the purpose of the given subroutine is to calculate a sequence of points \((x_i,y_i)\) \((i=1,2,...,N)\) such that, if \((x_i,y_i)\) is joined to \((x_{i+1},y_{i+1})\) by a straight line for \(i=1,2,...,N-1\), then the resultant piecewise linear curve is sufficiently close to the required section of the conic \((1.1)\). The accuracy of the approximation is controlled by the user through a parameter called DELT. He can ensure that the Euclidean distance (i.e. the straight line distance) between the conic and its approximation does not exceed DELT.

Because much of the computation that is required by the graph plotting system at Harwell is given to processing straight line segments, our subroutine aims to make the number of segments small subject to the accuracy condition. Another advantage of this aim is to reduce the amount of storage space that is required if the output from our subroutine is buffered.

Besides the simple case when the required contour line is a single curve that runs through the triangle, there are some other cases to be considered also. For example each side of the triangle may be cut twice by the conic in which case the required contour is composed of three separate pieces. The contour line may degenerate into two straight lines. The contour may be a ring contour that is wholly inside the triangle. All of these possibilities were kept in mind when the subroutine was designed and it is believed that they are all treated correctly.

The calculated coordinates \(x_i\) and \(y_i\) \((i=1,2,...,N)\) are accumulated in two one-dimensional arrays that are called XARC(\(\cdot\)) and YARC(\(\cdot\)) in the Fortran listing. The length of these arrays is at the control of the user but usually they each have 100 elements. When the arrays are full or when they need to be emptied because the approximation of a piece of the conic is complete, then subroutine O8132Z is called to transmit the line segments to the graph plotter. More information about this subroutine is given in the next section.
A good policy is to call subroutine OB13Z only when necessary because writing to output streams is relatively expensive on the IBM 370 computer. Therefore, if a single contour is being followed through several triangles, then we may prefer to accumulate the entire piecewise linear approximation in XARC(+) and YARC(×), if there is room for it, instead of emptying the arrays after each triangle is processed. One of the subroutine parameters, namely NPT, provides this facility.

All of the parameters of the subroutine and other information that is relevant to the use of the subroutine are discussed in Section 2. In Section 3 the mathematics of the approximation method is given, which makes extensive use of areal coordinates. Comments on the Fortran listing are made in Section 4, which, with the listing itself, provide a complete description of the computer program. Section 5 gives two examples of approximations that are calculated by the subroutine. The Fortran instructions of the subroutine are listed in the appendix to this report.

2. **Instructions for using subroutine OB13A**

The name of the subroutine and its parameters are as follows

```fortran
SUBROUTINE OB13A(FD, XD, YD, ETA, DELT, NPT, XARC, YARC).
```

The purpose of each parameter is given below.

FD is a real array of length six in which the user must place the function values that define the parameters of expression (1.2). Specifically the values of \( f(x,y) \) at the six points A,B,C,P,Q and R, which are shown in Figure 1, must be given in the storage locations \( FD(i) \) \( \{i=1,2,...,6\} \).

XD and YD are arrays of length three. The coordinates of the points A,B and C in Figure 1 must be set in the locations \( \{XD(1),YD(1)\} \), \( \{XD(2),YD(2)\} \) and \( \{XD(3),YD(3)\} \) respectively.
ETA is a real variable that must be set to the contour height that is the right hand side of equation (1.1).

DELT is a real variable that must also be set by the user. It is the accuracy parameter that was mentioned in the third paragraph of Section 1. However it is not always used directly by the subroutine because it is wasteful to aim for too high precision. Therefore the subroutine ensures that the error of the piecewise linear approximation to the conic is at most the quantity

$$\Delta = \max(\text{DELT}, \overline{\Delta}),$$  \hspace{1cm} (2.1)

where \(\overline{\Delta}\) is a suitable default value that is defined later. Thus the user who does not wish to think about the accuracy may set \(\text{DELT}\) to zero. Instead it is better to set it to the value that corresponds to 0.002 inches on the graph paper.

NPT is an integer variable that gives the number of joins \((x_i, y_i)\) of the piecewise linear approximation that are waiting to be written to the graph plotter by subroutine OB13Z which was mentioned at the end of Section 1. Since at least two joins are required to define a part of the approximation we let the value NPT=1 have a conventional meaning and we also give a special meaning to negative values of NPT in the following way. If on entry to the subroutine NPT is in the range \(-1 \leq \text{NPT} \leq 1\) then it is assumed that any previous output is finished, but if \(|\text{NPT}| \geq 2\) then it is assumed that \(|\text{NPT}|\) joins of the approximation are awaiting output. If NPT\(\leq 0\) initially then the approximation calculated by OB13A is written to the graph plotter before the return from the subroutine and the final value of NPT is zero. However, if NPT is positive initially, then the contents of the arrays XARC(·) and YARC(·) are not passed to the graph plotter by calling OB13Z before the return from OB13A. In this case the final value
of NPT is positive and it gives the number of joins \((x_i, y_i)\) that are awaiting output at the end of the calculation. Thus, if subroutine OB13A is used once only, it is appropriate to set NPT=0 at the beginning of the calculation. However, if we wish to draw a contour line that passes through several triangles, then it is appropriate to set NPT=1 initially and to leave NPT alone after each triangle is processed, except that we reverse the sign of NPT immediately before treating the last triangle. Any discontinuities in the contour line are managed automatically by emptying the output buffer before a new branch of the contour line is approximated. Therefore there is no need for the user to give special attention to the value of NPT when the contour line may not intersect the perimeter of each triangle exactly twice.

XARC and YARC are one-dimensional arrays whose lengths must be set by the user to at least the value of IARC, which is a COMMON integer variable that is defined later and whose value is usually IARC=100. These arrays hold the calculated joins \((x_i, y_i)\). Because these joins are passed to the graph-plotter when the arrays are full, the value of IARC does not restrict the use of the subroutine.

The joins are transmitted to the graph-plotter by the statement

\[
\text{CALL OB13Z(XARC,YARC,NPT)}
\]

which occurs in line 0244 of subroutine OB13A. A subroutine OB13Z that is suitable for the graph plotting system in use at Harwell is present in the Harwell subroutine library, so Harwell computer users need not concern themselves with this auxiliary subroutine. However most other readers will have to provide a suitable subroutine. All it has to do is to instruct the graph-plotter to draw straight lines between 
\{XARC(i),YARC(i)\} and \{XARC(i+1),YARC(i+1)\} for \(i=1,2,\ldots,NPT-1\), where
the value of NPT is some integer in the range $2 \leq \text{NPT} \leq \text{IARC}$. Sometimes subroutine OBL3Z is called several times by OBL3A. When this is done for a continuous contour line then each new value of \{XARC(1),YARC(1)\} is equal to the previous value of \{XARC(NPT),YARC(NPT)\}. However, when these two points have to be different, then any freedom that is available to OBL3A in the choice of \{XARC(1),YARC(1)\} is used to minimize the pen movement from the previous value of \{XARC(NPT),YARC(NPT)\}.

The COMMON statement that has been referred to already is the declaration

\[
\text{COMMON/OBL3B/DD,IARC,}
\]

where DD is a real variable and where IARC is an integer variable. This use of BLOCK COMMON allows the user to change the values of DD and IARC, which are defined initially by the DATA statement

\[
\text{DATA DD/1000./, IARC/100/}.
\]

The integer IARC should be set to the length of the arrays XARC(·) and YARC(·), while DD controls the value of $\Delta$ that appears in equation (2.1). In fact $\Delta$ is made equal to the length of the longest side of triangle ABC divided by DD.

Because the only output from subroutine OBL3A is through subroutine OBL3Z, the user has to arrange for all other output that he requires, such as plotting and labelling the axes of the graph. Note that if the contour (1.1) is completely outside the triangle then there is no output from the subroutine, unless the initial value of NPT causes OBL3Z to be called to provide sections of contours of previous triangles.

Since the next two sections concern details of the numerical method and the computer program, those people who wish to use the subroutine without studying it are advised to turn to Section 5.
3. The method of calculating the piecewise linear approximation

Much of the calculation is done in homogeneous coordinates (Milne, 1924) with respect to the triangle ABC. These coordinates are distinguished from Cartesian coordinates by the fact that they have three components. Also we use capital letters for Cartesian coordinates and small letters for homogeneous coordinates. The "homogeneous" point \((x, y, z)\) has the Cartesian coordinates

\[
\begin{align*}
X &= \frac{xX_A + yX_B + zX_C}{x+y+z} \\
Y &= \frac{xY_A + yY_B + zY_C}{x+y+z}
\end{align*}
\]

(3.1)

where \((X_A, Y_A), (X_B, Y_B)\) and \((X_C, Y_C)\) are the Cartesian coordinates of the vertices of the triangle ABC. Thus the points \((x, y, z)\) and \((\theta x, \theta y, \theta z)\) are the same for all non-zero values of \(\theta\). Moreover the point \((x, y, z)\) is at infinity if the sum \(x+y+z\) is equal to zero. One of the advantages of homogeneous coordinates is that points at infinity have a finite representation.

The point \((x, y, z)\) lies on the required contour line if the equation

\[
a x^2 + b y^2 + c z^2 + 2f yz + 2g zx + 2h xy = 0
\]

(3.2)

is satisfied, where the coefficients of this quadratic have the values

\[
\begin{align*}
a &= f_A - H \\
b &= f_B - H \\
c &= f_C - H \\
f &= 2f_P - \frac{1}{2}(f_B + f_C) - H \\
g &= 2f_Q - \frac{1}{2}(f_C + f_A) - H \\
h &= 2f_R - \frac{1}{2}(f_A + f_B) - H
\end{align*}
\]

(3.3)
the quantities \( f_A, f_B, f_C, f_P, f_Q \) and \( f_R \) being the given function values at the points \( A, B, C, P, Q \) and \( R \) of Figure 1, and \( H \) being the given contour height. This notation is different from and replaces that of equation (1.2). The line of points \((x, y, z)\) that satisfy the condition

\[
\ell x + my + nz = 0
\]

is tangential to the quadratic (3.2) if and only if the equation

\[
r x^2 + sm^2 + tn^2 + 2umn + 2vnm + 2wxm = 0
\]

holds, where the coefficients have the values

\[
\begin{align*}
  r &= bc - f^2 \\
  s &= ca - g^2 \\
  t &= ab - h^2 \\
  u &= gh - af \\
  v &= hf - bg \\
  w &= fg - ch
\end{align*}
\]

Another use of these coefficients is that the centre of the conic is at the point

\[
(\xi, \eta, \zeta) = (r+v+w, s+w+u, t+u+v).
\]

It may be shown that the conic is an ellipse if \((\xi+\eta+\zeta)\) is positive, it is a parabola if \((\xi+\eta+\zeta)\) is zero and it is a hyperbola if \((\xi+\eta+\zeta)\) is negative.

Note that if the coefficients (3.7) are substituted in expression (3.2) we obtain \((\xi+\eta+\zeta)\) times the number
\[ \delta = abc + 2fgh - af^2 - bg^2 - ch^2 \]
\[ = ar + gv + hw \]
\[ = bs + hw + fu \]
\[ = ct + fu + gv. \]  
(3.8)

The conic passes through its centre if and only if \( \delta \) is zero. Thus the value of \( \delta \) indicates whether the conic degenerates into two straight lines. We find later that \( \delta \) is also useful in other ways.

In order to calculate a sequence of points on the conic it is convenient to express the conic parametrically. In general we can find coefficients \( x_0, x_1, x_2, y_0, y_1, y_2, z_0, z_1 \) and \( z_2 \) such that the points \( (x, y, z) \) that satisfy equation (3.2) are generated by the formula

\[
\begin{align*}
    x(\tau) &= x_0 + 2x_1\tau + x_2\tau^2 \\
y(\tau) &= y_0 + 2y_1\tau + y_2\tau^2 \\
z(\tau) &= z_0 + 2z_1\tau + z_2\tau^2
\end{align*}
\]  
(3.9)

as the parameter \( \tau \) ranges from \(-\infty\) to \( +\infty \). Next we consider the calculation of these coefficients.

The values \( \tau = 0 \) and \( \tau = \infty \) show that the parametric form (3.9) includes the points \( (x_0, y_0, z_0) \) and \( (x_2, y_2, z_2) \). In order to calculate the parametric form there is no loss of generality in supposing that these two points are any two fixed points on the conic (3.2). It is particularly convenient to let the two fixed points be on the line joining \( A, B \) or \( C \) to the centre (3.7) or to let them be on one of the sides of the triangle \( ABC \). We have to admit both possibilities to ensure that \( (x_0, y_0, z_0) \) and \( (x_2, y_2, z_2) \) are real and distinct.
It is appropriate to let these points be on the line joining A, B or C to the centre (3.7) when the conic is an ellipse whose centre is inside the triangle ABC. This case is characterized by the fact that $\xi, \eta$ and $\zeta$ are all positive, and the conic lies wholly or partly inside the triangle if the value of expression (3.2) at one of the vertices of the triangle is opposite in sign to the value of expression (3.2) at $(\xi, \eta, \zeta)$. Therefore we require at least one of the products \{a\delta, b\delta, c\delta\} to be negative, where $a, b, c$ and $\delta$ are defined by equations (3.3) and (3.8). Otherwise there is no contour to be drawn. When a contour is present we choose the vertex for which the product in the set \{a\delta, b\delta, c\delta\} is most negative and we let $(x_0, y_0, z_0)$ and $(x_2, y_2, z_2)$ be on the line joining this vertex to $(\xi, \eta, \zeta)$. For definiteness we suppose that this vertex is C.

Each point on the line joining C to the centre (3.7) can be written in the form $(\xi, \eta, \theta)$ for some value of the parameter $\theta$. This point is on the conic (3.2) if $\theta$ satisfies the quadratic equation

$$c\theta^2 + 2(f\eta + g\xi)\theta + (a\xi^2 + 2h\xi\eta + b\eta^2) = 0.$$  \hspace{1cm} (3.10)

It can be shown that one solution is the value

$$\theta_o = \zeta - \frac{\delta}{C} \left\{ 1 + \sqrt{1 - \frac{C}{\delta}} (\xi + \eta + \zeta) \right\}$$  \hspace{1cm} (3.11)

and there is no cancellation in evaluating this expression because $\xi, \eta$ and $\zeta$ are positive and $\delta/c$ is negative. It follows that the other solution has the value

$$\theta_1 = \frac{(a\xi^2 + 2h\xi\eta + b\eta^2)c\theta_o}{c\theta_o}.$$  \hspace{1cm} (3.12)

Thus we find six of the nine coefficients of the parametric form.
(3.9), except that we will multiply \( x_2, y_2 \) and \( z_2 \) by a scaling factor.

The algebraic relation between equations (3.2) and (3.9) is that when \( x, y \) and \( z \) are given the values \( x(\tau), y(\tau) \) and \( z(\tau) \) then expression (3.2), which is a quartic polynomial in \( \tau \), must be identically zero. Our choice of \( (x_0, y_0, z_0) \) and \( (x_2, y_2, z_2) \) ensures that the coefficients of \( \tau^0 \) and \( \tau^4 \) in the quartic polynomial are zero. The coefficients of \( \tau \) and \( \tau^3 \) are also zero provided that \( (x_1, y_1, z_1) \) is orthogonal to the two vectors

\[
\begin{align*}
\{ (ax_0+hy_0+gz_0, hx_0+by_0+fz_0, gx_0+fy_0+cz_0) \\
(ax_2+hy_2+gz_2, hx_2+by_2+fz_2, gx_2+fy_2+cz_2) \}
\end{align*}
\] (3.13)

so we deduce from the values of \( (x_0, y_0, z_0) \) and \( (x_2, y_2, z_2) \) that \( (x_1, y_1, z_1) \) is a multiple of the vector \( (f-c, c-g, g-f) \). Finally we have to ensure that the coefficient of \( \tau^2 \) in the quartic polynomial is zero which is the condition

\[
\begin{align*}
a(x_0^2+2x_1^2) + b(y_0^2+2y_1^2) + c(z_0^2+2z_1^2) \\
+ f(y_0z_2+4y_1z_1+y_2z_0) + g(z_0x_2+4z_1x_1+z_2x_0) \\
+ h(x_0y_2+4x_1y_1+x_2y_0) = 0.
\end{align*}
\] (3.14)

This condition can be satisfied by adjusting the scale factors that multiply \( (x_0, y_0, z_0), (x_1, y_1, z_1) \) and \( (x_2, y_2, z_2) \). After condition (3.14) is satisfied there is still some freedom in the scale factors because there is a free overall factor and also \( \tau \) may be multiplied by a constant in expression (3.9). We prefer to take up this freedom so that the coefficients of the parametric form have the values

-13-
\[ x_0 = \xi \]
\[ x_1 = (f-c)/\sqrt{1-(\xi+n+\zeta)c/\delta} \]
\[ x_2 = c\xi/(c(\xi+n+\zeta)-\delta) \]
\[ y_0 = n \]
\[ y_1 = (c-g)/\sqrt{1-(\xi+n+\zeta)c/\delta} \]
\[ y_2 = cn/(c(\xi+n+\zeta)-\delta) \]
\[ z_0 = \zeta - (\delta/c) \left\{ 1 + \sqrt{1 - (\xi+n+\zeta)c/\delta} \right\} \]
\[ z_1 = (g-f)/\sqrt{1 - (\xi+n+\zeta)c/\delta} \]
\[ z_2 = (a\xi^2+2h\xi n+bn^2)/z_0(c(\xi+n+\zeta)-\delta) \]  \[ (3.15) \]

One reason for our choice of scale is so that the equations

\[ x_0y_1z_2 + x_1y_2z_0 + x_2y_0z_1 - x_0y_2z_1 - x_1y_0z_2 - x_2y_1z_0 = 2\delta \]

and

\[ \xi + n + \zeta = \alpha \gamma - \beta^2 \]

(3.16)

(3.17)

hold, where \( \alpha, \beta \) and \( \gamma \) are the quantities

\[
\begin{align*}
\alpha &= x_0+y_0+z_0 \\
\beta &= x_1+y_1+z_1 \\
\gamma &= x_2+y_2+z_2
\end{align*}
\]

(3.18)

We stated that the parametric form (3.15) is not used unless all the components of the centre (3.7) are positive. When there is at least one non-positive component then a part of the contour line is inside triangle ABC only if the perimeter of the triangle is cut by the conic. In this case we let the vectors \((x_0, y_0, z_0)\) and \((x_2, y_2, z_2)\) of the parametric form (3.9) lie on one of the sides of the triangle, one reason being that the square root that occurs many times in expression (3.15) may be
imaginary. We choose a side of the triangle on which the quadratic function (3.2) takes both positive and negative values.

However, because of the possibility that the contour cuts the perimeter of the triangle only at two of the vertices, we extend the sides of the triangle beyond the vertices when considering the range of the quadratic function. For example Figure 2 shows the extension of side AB where SA=AR=RB=BT. We let m and M be the smallest and largest values of the function (3.2) on the line segment ST and we associate with the side AB the number

$$|m-M| - |m+M|, \quad (3.19)$$

which is positive if and only if the quadratic function takes positive and negative values on ST. Note that in the calculation of m and M the variables of expression (3.2) must satisfy the normalization condition

$$x+y+z = 1. \quad (3.20)$$

For each side of the triangle there is a number corresponding to expression (3.19) and we identify the side whose number is largest. If the largest number is not positive there is no contour to be drawn. Otherwise we let \((x_0, y_0, z_0)\) and \((x_2, y_2, z_2)\) lie on the side that gives the largest number. To derive formulae for the coefficients of the parametric form (3.9) we suppose that \((x_0, y_0, z_0)\) and \((x_2, y_2, z_2)\) lie on the side AB.

A general point on the side AB has the components \((\theta, 1, 0)\) where \(\theta\) is a parameter. Therefore, corresponding to equation (3.10), we now require to satisfy the quadratic equation

$$\alpha \theta^2 + 2\lambda \theta + b = 0, \quad (3.21)$$

which has the roots
\[ \theta_0 = -h(1 + \sqrt{1 - ab/h^2})/a \]  
and 
\[ \theta_1 = -b/h(1 + \sqrt{1 - ab/h^2}) \].  

Therefore we let \((x_0, y_0, z_0)\) be equal to \((\theta_1, 1, 0)\) and we let \((x_2, y_2, z_2)\) be a multiple of \((\theta_0, 1, 0)\). Again \((x_1, y_1, z_1)\) must be orthogonal to the vectors \((3.13)\) and now it follows that \((x_1, y_1, z_1)\) is proportional to the vector \((v, u, t)\), whose components are defined by equation \((3.6)\). We calculate the multipliers of \((x_0, y_0, z_0), (x_1, y_1, z_1)\) and \((x_2, y_2, z_2)\) so that equations \((3.14), (3.16)\) and \((3.17)\) are satisfied. Thus we obtain the coefficients

\[
\begin{align*}
    x_0 &= -b/h(1 + \sqrt{1 - ab/h^2}) \\
    x_1 &= -v/h \sqrt{1 - ab/h^2} \\
    x_2 &= \delta h(1 + \sqrt{1 - ab/h^2})/(h^2-ab) \\
    y_0 &= 1 \\
    y_1 &= -u/h \sqrt{1 - ab/h^2} \\
    y_2 &= -\delta a/(h^2-ab) \\
    z_0 &= 0 \\
    z_1 &= h \sqrt{1 - ab/h^2} \\
    z_2 &= 0
\end{align*}
\]  

\(3.24\)

An advantage of these coefficients is that it is feasible during their calculation to find the form of the contour in the degenerate case when \(\delta=0\). To explain this point we suppose first that \(|\delta|\) is very small. Then the factor \(\delta\) in the definition of \(x_2\) and \(y_2\) and the fact that \(z_2\) is zero imply that, for moderate values of \(\tau\), the second order terms of expression \((3.9)\) are so small that the locus \(\{x(\tau), y(\tau), z(\tau)\}\) is approximately the straight line through the points \((x_0, y_0, z_0)\) and \((x_1, y_1, z_1)\). However, \(\tau\) can be made arbitrarily large so, if the second
order terms are non-zero, they are of the same magnitude as the first order terms somewhere near the ends of the range \(-\infty < \tau < \infty\). In this case when \(\delta\) is small the zero order terms are relatively small. Thus large values of \(\tau\) give approximately the straight line through \((x_1, y_1, z_1)\) and \((x_2, y_2, z_2)\). By homogeneity the point \((x_2, y_2, z_2)\) of expression (3.24) has the coordinates

\[
(h_1 + \sqrt{1 - ab/h^2}, -a, 0).
\]  

Thus in the limiting case as \(\delta\) tends to zero the required contour line degenerates into a pair of straight lines. One line joins \((x_0, y_0, z_0)\) to \((x_1, y_1, z_1)\) and the other line joins \((x_1, y_1, z_1)\) to the point (3.25) where \((x_0, y_0, z_0)\) and \((x_1, y_1, z_1)\) are defined by formula (3.24). Thus the calculation of the coefficients (3.24) includes the calculation of the two straight lines that form the required contour when \(\delta = 0\).

It is encouraging to look at this calculation when one of the straight lines is a side of the triangle ABC, say the side BC. Then b, c and f and hence r, v and w are all equal to zero. Moreover for the side BC expression (3.19) is zero so the points \((x_0, y_0, z_0)\) and \((x_2, y_2, z_2)\) do not both lie on BC. We suppose that the parametric form (3.24) is used again. Then both \(x_0\) and \(x_1\) are zero so the line joining \((x_0, y_0, z_0)\) to \((x_1, y_1, z_1)\) is the side BC. Thus rounding errors do not prevent a side of the triangle from being recognised as part of the contour line.

In the usual case when \(\delta\) is non-zero the contour line has the parametric form (3.9) and we have to calculate the part that is inside triangle ABC. It is straightforward to obtain the values of \(\tau\) for which the contour cuts the sides of ABC. For example the side BC is cut at the roots of the equation

-17-
\[ x_0 + 2x_1\tau + x_2\tau^2 \neq 0. \] (3.26)

Thus the roots of \( x(\tau), y(\tau) \) and \( z(\tau) \) give the end points of the intervals in \( \tau \) for which the curve (3.9) has to be drawn. Usually some unwanted points occur in this list because we wish to exclude intersections with the triangle sides that are external to the triangle. In fact we require the ranges of \( \tau \) for which \( x(\tau), y(\tau) \) and \( z(\tau) \) are all non-negative and also the ranges of \( \tau \) for which they are all non-positive. The information necessary to find these ranges is contained in the coefficients \( x_0, x_1, x_2, y_0, y_1, y_2, z_0, z_1 \) and \( z_2 \) and a suitable method of calculation is given in the Fortran listing. It is slightly complicated by the fact that often one of the end points is at infinity and sometimes a range may have the point at infinity as an interior point because the curve (3.9) varies continuously if \( \tau \) is increased to \( +\infty \) and then is increased from \( -\infty \). Sometimes this calculation shows that the conic is wholly outside the triangle in which case there is a return from the subroutine.

One question of accuracy is particularly important to the method just described for finding the intersections of the contour line with the triangle sides. It is that if a single contour is drawn through a sequence of triangles by calling the present subroutine once for each triangle then we require continuity of the contour line across the triangle boundaries. This requirement is satisfied theoretically, but computer rounding errors may make it awkward to achieve when the contour line meets the triangle side at a very narrow angle. For instance this case is liable to occur when the three given function values on one of the sides of triangle ABC have much smaller moduli than the other three given function values. The difficulty due to rounding errors was kept in mind when deciding on the actual form of the coefficients (3.15) and (3.24).
Drastic cancellation is not caused by three small function values on a triangle side. Numerical results confirm that excellent continuity is obtained across triangle boundaries but a theoretical analysis of the effect of rounding errors on the continuity of the contour line has not been done.

The curve (3.9) has to be drawn for one, two or three intervals in \( \tau \) and it is approximated by a sequence of chords. The number of chords depends on the accuracy requested by the user of the subroutine. Although the calculation described already in this section has not used the Cartesian coordinates of \( A, B \) and \( C \), the number of chords does depend on these coordinates. In particular if the determinant

\[
D = \begin{vmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \end{vmatrix} \quad (3.27)
\]

is zero then \( A, B \) and \( C \) are collinear, so the triangle is too narrow to allow most deviations of the chord from the contour line. Therefore in this case a single chord is drawn for each interval in \( \tau \). However in the usual case when expression (3.27) is non-zero a division of each interval in \( \tau \) into suitable chords has to be made. Much of this calculation is independent of the actual range of \( \tau \) so it is done before the intervals in \( \tau \) are considered.

The piecewise linear approximation to the contour line comes from the following lemma which is interesting because it depends on the Cartesian coordinates of \( A, B \) and \( C \) only through the value of the determinant (3.27).
Lemma 1. Let \( \tau_k \) and \( \tau_{k+1} \) be related by the equation

\[
\tau_{k+1} = \frac{\tau_k + \mu (\alpha + \beta \tau_k)}{1 - \mu (\beta + \gamma \tau_k)},
\]

(3.28)

where \( \alpha, \beta \) and \( \gamma \) have the values (3.18) and where \( \mu \) is a parameter that satisfies the condition

\[
1 + \mu^2 (\xi + \eta + \zeta) > 0.
\]

(3.29)

Let \( P_k \) and \( P_{k+1} \) be the points \( \{x(\tau_k), y(\tau_k), z(\tau_k)\} \) and \( \{x(\tau_{k+1}), y(\tau_{k+1}), z(\tau_{k+1})\} \). Then the Euclidean distance from \( P_k \) to \( P_{k+1} \) times the Euclidean distance of the conic from the chord \( P_kP_{k+1} \) has the value

\[
\frac{2 \delta \mu^3 |D|}{[1 + \mu^2 (\xi + \eta + \zeta)][1 + \sqrt{1 + \mu^2 (\xi + \eta + \zeta)}]}.
\]

(3.30)

Proof of Lemma 1. Let \( (X_k, Y_k), (X_{k+1}, Y_{k+1}) \) and \( (X_{\tau}, Y_{\tau}) \) be the Cartesian coordinates of \( P_kP_{k+1} \) and \( \{(x(\tau), y(\tau), z(\tau))\} \). Then the Euclidean length of \( P_kP_{k+1} \) times the Euclidean distance of the conic from the chord \( P_kP_{k+1} \) has the value

\[
\max_{\tau \in [\tau_k, \tau_{k+1}]} |(X_{\tau} - X_k)(Y_{k+1} - Y_k) - (X_{k+1} - X_k)(Y_{\tau} - Y_k)|.
\]

(3.31)

By using equations (3.1), (3.9) and (3.18) to express the Cartesian coordinates in terms of \( \tau_k, \tau_{k+1} \) and \( \tau \) it follows that the modulus signs of expression (3.31) contain the quantity

\[
\frac{2(\tau - \tau_k)(\tau_{k+1} - \tau_k)(\tau_{k+1} - \tau)\phi}{(\alpha + 2\beta + \gamma \tau)(\alpha + 2\beta \tau_k + \gamma \tau_k)(\alpha + 2\beta \tau_{k+1} + \gamma \tau_{k+1})^2},
\]

(3.32)
where \( \phi \) is the number

\[
(X_A Y_B - X_B Y_A)\{\alpha (x_1 y_2 - x_2 y_1) + \beta (x_2 y_0 - x_0 y_2) + \gamma (x_0 y_1 - x_1 y_0)\}
+ (X_B Y_C - X_C Y_B)\{\alpha (y_1 z_2 - y_2 z_1) + \beta (y_2 z_0 - y_0 z_2) + \gamma (y_0 z_1 - y_1 z_0)\}
+ (X_C Y_A - X_A Y_C)\{\alpha (z_1 x_2 - z_2 x_1) + \beta (z_2 x_0 - z_0 x_2) + \gamma (z_0 x_1 - z_1 x_0)\}.
\]

By substituting for \( \alpha, \beta \) and \( \gamma \) we find that all three curly brackets of expression (3.33) are equal to the quantity (3.16), so from the definition (3.27) it follows that \( \phi \) has the value

\[
\phi = 2 \delta D.
\]

By choosing the value of \( \lambda \) appropriately we may express \( \tau \) in the form

\[
\tau = \frac{\tau_k + \lambda(\alpha + \beta \tau_k)}{1 - \lambda(\beta + \gamma \tau_k)}
\]

and also we make use of the identity (3.28) to deduce the equation

\[
\frac{(\tau - \tau_k)(\tau_{k+1} - \tau_k)(\tau_{k+1} - \tau)}{\lambda + 2\beta \tau_k + \gamma \tau_k^2 - 2\beta \tau_k + 2\beta \tau_{k+1} + \gamma \tau_{k+1}^2}
\]

\[
= \frac{\lambda \mu(\mu - \lambda)}{[1 + \lambda^2(\alpha \gamma - \beta^2)][1 + \mu^2(\alpha \gamma - \beta^2)]}
\]

It follows that expression (3.31) has the value

\[
\max_{\lambda \in [0, \mu]} \frac{4 \delta D \lambda \mu(\mu - \lambda)}{[1 + \lambda^2(\alpha \gamma - \beta^2)][1 + \mu^2(\alpha \gamma - \beta^2)]}
\]

This expression is stationary when \( \lambda \) satisfies the equation
\[
\mu \alpha^2 (\alpha \gamma - \beta^2) + 2 \lambda - \mu = 0,
\]

one root being the number

\[
\lambda = \frac{\mu}{1 + \sqrt{1 + \mu^2 (\alpha \gamma - \beta^2)}}.
\]

Equation (3.38) has two roots because there are two ways of following the conic from \(P_k\) to \(P_{k+1}\), one usually being much longer than the other because the two ways together give the whole locus of the conic. We require the value of \(\lambda\) that is approximately equal to \(i \mu\) when \(|\mu|\) is small so expression (3.39) is the appropriate root of the quadratic equation (3.38). By substituting the value (3.39) in expression (3.37) we obtain the quantity

\[
\frac{|2 \delta \mu^3 D|}{[1 + \mu^2 (\alpha \gamma - \beta^2)][1 + \sqrt{1 + \mu^2 (\alpha \gamma - \beta^2)}]}.
\]

Since equation (3.17) holds the proof of Lemma 1 is complete.

Condition (3.29) not only ensures that the square root of expression (3.30) is real but it also has a useful geometric interpretation. The condition can fail only when \((\xi + \eta + \zeta)\) is negative and we have noted already that in this case the conic is a hyperbola. The geometric interpretation is that the condition is sufficient for the parameters \(\tau_k\) and \(\tau_{k+1}\) of equation (3.28) to give points that are on the same branch of the hyperbola. To justify this statement we seek the value of \(\mu\) that causes \(P_{k+1}\) to be at infinity which is the condition

\[
\alpha + 2 \beta \tau_{k+1} + \gamma \tau_{k+1}^2 = 0.
\]

By substituting expression (3.28) in equation (3.41) we find that \(P_{k+1}\)
is at infinity if the equation

$$(\alpha + 2\beta \tau_k \gamma + \gamma^2)(1 + \mu^2(\alpha \gamma - \beta^2)) = 0$$ (3.42)

holds. Now the first term of expression (3.42) is non-zero if $P_k$ is a finite point, which holds in practice because $P_k$ is in the triangle ABC. Therefore, recalling equation (3.17), it follows that we must avoid the values of $\mu$ defined by the equation

$$1 + \mu^2(\xi + \eta + \zeta) = 0.$$ (3.43)

By continuity we deduce that when the conic is a hyperbola then expression (3.29) does force $P_k$ and $P_{k+1}$ to be on the same branch of the contour.

Lemma 1 is relevant to the piecewise linear approximation because it provides information about the distance of a line segment from the conic. We want the lengths of the line segments of our approximation to adapt to the shape of the conic so that relatively few line segments are used where the contour line curves less sharply. One method of achieving this would be to make the distance of each line segment from the conic equal to a constant that is at most the required accuracy of the approximation. Instead, however, it is more convenient to use equation (3.28) recursively for a fixed value of $\mu$ to generate a sequence \{\tau_0, \tau_1, \tau_2, \ldots\}. Thus we obtain a sequence of chords such that the chord length times the distance of each chord from the conic is constant. These chords become longer where the contour line curves less sharply and they are so easy to generate that we have chosen to use this method in the Fortran subroutine. Note that this method is analogous to using equal increments in $\theta$ to follow the locus.
\[ \begin{align*}
X &= p \cos \theta \\
Y &= q \sin \theta 
\end{align*} \]  
(3.44)

in Cartesian space, which gives an approximation to the ellipse.

\[ q^2x^2 + p^2y^2 = p^2q^2. \]  
(3.45)

Therefore, for each interval of the parameter \( \tau \) for which the contour is to be plotted, we require a value of \( \mu \) so that, if \( \tau_0 \) is the initial value of \( \tau \), and if subsequent values of \( \tau_{k+1} \) are defined by equation (3.28) for \( k \geq 1 \), until the other end of the interval of \( \tau \) is reached, then the piecewise linear function \( P_0P_1P_2\ldots \) is a sufficiently accurate approximation to the contour, where \( P_k \) is the point whose homogeneous coordinates are \( \{x(\tau_k), y(\tau_k), z(\tau_k)\} \).

To obtain a suitable value of \( \mu \) we therefore require the length of the chord \( P_kP_{k+1} \) when equation (3.28) is satisfied. A useful expression is given in Lemma 2.

**Lemma 2** Let \( g_0, g_1, g_2, h_0, h_1, h_2 \) be defined by the equations

\[
\begin{pmatrix}
g_0 \\
g_1 \\
g_2
\end{pmatrix}
= \Omega
\begin{pmatrix}
X_A - X_B \\
X_B - X_C \\
X_C - X_A
\end{pmatrix}
\]  
(3.46)

and

\[
\begin{pmatrix}
h_0 \\
h_1 \\
h_2
\end{pmatrix}
= \Omega
\begin{pmatrix}
Y_A - Y_B \\
Y_B - Y_C \\
Y_C - Y_A
\end{pmatrix}
\]  
(3.47)

where \( \Omega \) is the matrix.
\[ \Omega = \begin{pmatrix} x_1 y_2 - x_2 y_1 & y_1 z_2 - y_2 z_1 & z_1 x_2 - z_2 x_1 \\ x_2 y_0 - x_0 y_2 & y_2 z_0 - y_0 z_2 & z_2 x_0 - z_0 x_2 \\ x_0 y_1 - x_1 y_0 & y_0 z_1 - y_1 z_0 & z_0 x_1 - z_1 x_0 \end{pmatrix} \]  \tag{3.48}

Moreover let \( \tau \) and \( \lambda \) be defined by equations (3.35) and (3.39). Then the length of the chord \( P_k P_{k+1} \) has the value

\[
\frac{4\lambda \left[ (g_2 - g_1 \tau + g_0 \tau^2)^2 + (h_2 - h_1 \tau + h_0 \tau^2)^2 \right]^{1/2}}{[1 + \lambda^2 (\alpha \gamma - \beta^2)][\alpha + 2\beta \tau + \gamma \tau^2]}.
\]  \tag{3.49}

**Proof of Lemma 2** From expressions (3.1) and (3.9) we find that the length of the chord \( P_k P_{k+1} \) has the form

\[
\left[ (\sigma_A x_A + \sigma_B x_B + \sigma_C x_C)^2 + (\sigma_A y_A + \sigma_B y_B + \sigma_C y_C)^2 \right]^{1/2},
\]  \tag{3.50}

where the multipliers \( \sigma_A, \sigma_B \) and \( \sigma_C \) depend on \( \tau_k, \tau_{k+1} \) and the coefficients of the parametric form (3.9). For example \( \sigma_A \) is the number

\[
\sigma_A = \frac{x_0 + 2x_1 \tau_k + x_2 \tau_{k+1}^2}{\alpha + 2\beta \tau_k + \gamma \tau_{k+1}^2} - \frac{x_0 + 2x_1 \tau_k + x_2 \tau_{k+1}^2}{\alpha + 2\beta \tau_k + \gamma \tau_{k+1}^2}. \]  \tag{3.51}

Now the definitions of \( \lambda \) and \( \tau \) imply the values

\[
\tau_k = \frac{\tau - \lambda (\alpha + \beta \tau)}{1 + \lambda (\beta + \gamma \tau)} \]  \tag{3.52}

and

\[
\tau_{k+1} = \frac{\tau + \lambda (\alpha + \beta \tau)}{1 - \lambda (\beta + \gamma \tau)} \]  \tag{3.53}

which we substitute into expression (3.51). Thus we find the identity
\[ \sigma_A = \frac{4\lambda[-x_0(\beta+\gamma \tau) + x_1(\alpha-\gamma \tau^2) + x_2(\alpha \tau+\beta \tau^2)]}{[1+\lambda^2(\alpha \gamma-\beta^2)][\alpha+2\beta \tau+\gamma \tau^2]} \]  
(3.54)

Similar expressions can be obtained for \( \sigma_B \) and \( \sigma_C \). It follows that the equation

\[ \sigma_A x_A^+ + \sigma_B x_B^+ + \sigma_C x_C = \frac{4\lambda[\psi_0 + \psi_1 \tau + \psi_2 \tau^2]}{[1+\lambda^2(\alpha \gamma-\beta^2)][\alpha+2\beta \tau+\gamma \tau^2]} \]  
(3.55)

holds, where \( \psi_0, \psi_1 \) and \( \psi_2 \) are the multipliers

\[
\begin{align*}
\psi_0 &= (x_1\alpha-x_0\beta) x_A^+ + (y_1\alpha-y_0\beta) x_B + (z_1\alpha-z_0\beta) x_C \\
\psi_1 &= (x_2\alpha-x_0\gamma) x_A^+ + (y_2\alpha-y_0\gamma) x_B + (z_2\alpha-z_0\gamma) x_C \\
\psi_2 &= (x_2\beta-x_0\gamma) x_A^+ + (y_2\beta-y_0\gamma) x_B + (z_2\beta-z_0\gamma) x_C
\end{align*}
\]  
(3.56)

By substituting the values (3.18) and using the definitions (3.46) and (3.47), we find that \( \psi_0, \psi_1 \) and \( \psi_2 \) are equal to \(-g_2, +g_1 \) and \(-g_0 \) respectively. Similarly we obtain the equation

\[ \sigma_A y_A^+ + \sigma_B y_B^+ + \sigma_C y_C = \frac{4\lambda[-h_2+\gamma_1 \tau-h_0 \tau^2]}{[1+\lambda^2(\alpha \gamma-\beta^2)][\alpha+2\beta \tau+\gamma \tau^2]} \]  
(3.57)

Therefore expression (3.50) is equal to expression (3.49) which proves Lemma 2.

When we approximate the conic for a range of the parameter \( \tau \) by a sequence of chords we have to ensure that the distance of each chord from the conic is at most \( \Delta \), which is defined by equation (2.1). Let the range of \( \tau \) be \([\tau_0, \tau_f]\). Then it follows from Lemmas 1 and 2 that it is sufficient to choose the value of \( \mu \) so that the condition
\[ \frac{|2 \delta \mu^3 D|}{[1+\mu^2(\xi+n+\tau)][1+\sqrt{1+\mu^2(\xi+n+\tau)}]} \leq \frac{4\Delta \lambda[(g_2-g_1\tau+g_0\tau^2)^2 + (h_2-h_1\tau+h_0\tau^2)^2]^{\frac{1}{2}}}{[1+\lambda^2(\alpha\gamma-\beta^2)][\alpha+2\beta\tau+\gamma\tau^2]} \]  

(3.58)

holds for all \( \tau \) in \([\tau_0, \tau_f]\), where \( \lambda \) is related to \( \mu \) by equation (3.39).

Substituting the value of \( \lambda \) and using equation (3.17) we find that expression (3.58) is equivalent to the inequality

\[ \frac{|\delta \mu^2 D|}{\sqrt{1+\mu^2(\alpha\gamma-\beta^2)[1+\sqrt{1+\mu^2(\alpha\gamma-\beta^2)}]}} \leq \Delta \min_{\tau \in [\tau_0, \tau_f]} \frac{[(g_2-g_1\tau+g_0\tau^2)^2 + (h_2-h_1\tau+h_0\tau^2)^2]^{\frac{1}{2}}}{[\alpha+2\beta\tau+\gamma\tau^2]} . \]

(3.59)

which can be satisfied by letting \( \mu \) be sufficiently small. In order to help the calculation of the right hand side of this expression we study the function

\[ \rho(\tau) = \frac{(g_2-g_1\tau+g_0\tau^2)^2 + (h_2-h_1\tau+h_0\tau^2)^2}{(\alpha+2\beta\tau+\gamma\tau^2)^2} . \]

(3.60)

The required properties of \( \rho(\tau) \) are given in the following lemma.
Lemma 3 Let $l, m, n, \sigma$ and $\theta$ be the numbers

\[
\begin{align*}
\lambda &= h_1^2 - 4h_0h_2 \\
m &= 2g_0h_2 + 2g_2h_0 - g_1h_1 \\
n &= g_1^2 - 4g_0g_2
\end{align*}
\]

(3.61)

\[
\sigma = \frac{2}{n + l + \sqrt{(n - l)^2 + 4m^2}}
\]

(3.62)

and

\[
\theta = \frac{n - l + \sqrt{(n - l)^2 + 4m^2}}{2m}
\]

(3.63)

Then $\sigma$ is positive and the equation

\[
\tau^2 (h_0 - g_0 \theta) - \tau (h_1 - g_1 \theta) + (h_2 - g_2 \theta) = 0
\]

(3.64)

has real roots. If one or both of the roots is in the range $[\tau_0, \tau_f]$, then the least value of expression (3.60) for $\tau \in [\tau_0, \tau_f]$ is equal to $4D^2\delta^2\sigma$. Otherwise the least value is $\min[\rho(\tau_0), \rho(\tau_f)]$.

Proof of Lemma 3 The key to proving the lemma is the auxiliary function

\[
\psi(\omega) = \frac{4D^2\delta^2(1 + \omega^2)}{\lambda + 2m\omega + n\omega^2}
\]

(3.65)

because, if we let $\omega$ have the value

\[
\omega = \frac{h_2 - h_1 \tau + h_0 \tau^2}{g_2 - g_1 \tau + g_0 \tau^2}
\]

(3.66)

then $\psi(\omega)$ to equal to $\rho(\tau)$. This statement can be proved in three stages. First we deduce from expressions (3.16), (3.18), (3.27), (3.46) and (3.47) the
\[
\begin{align*}
\alpha &= (g_1 h_2 - g_2 h_1)/2D\delta \\
\beta &= (g_2 h_0 - g_0 h_2)/2D\delta \\
\gamma &= (g_0 h_1 - g_1 h_0)/2D\delta 
\end{align*}
\]

(3.67)

Secondly we make use of these values to obtain the identity

\[
4D^2\delta^2(\alpha + 2\beta\tau + \gamma\tau^2)^2 = \lambda (g_2 - g_1 \tau + g_0 \tau^2)^2 \\
+ 2m (g_2 - g_1 \tau + g_0 \tau^2)(h_2 - h_1 \tau + h_0 \tau^2) + n(h_2 - h_1 \tau + h_0 \tau^2)^2, 
\]

(3.68)

where \(\lambda, m\) and \(n\) are given in the definition (3.61). Thirdly we use equations (3.60), (3.66) and (3.68) to express \(\rho(\tau)\) in terms of \(\omega\). Thus we obtain the function (3.65). Note that, because \(\rho(\tau)\) is positive, one consequence of the fact that \(\psi(\omega)\) is equal to \(\rho(\tau)\) is that the inequality

\[
\lambda + 2m\omega + n\omega^2 \geq 0
\]

(3.69)

holds when \(\omega\) has the value (3.66) and \(\tau\) is real.

In fact inequality (3.69) is the necessary and sufficient condition on \(\omega\) for a real value of \(\tau\) to satisfy equation (3.66). We have proved the necessity already. To prove the sufficiency we note that we require the quadratic equation

\[
\tau^2(h_0 - g_0\omega) - \tau(h_1 - g_1\omega) + (h_2 - g_2\omega) = 0
\]

(3.70)

to have real roots, which is the condition

\[
(h_1 - g_1\omega)^2 \geq 4(h_0 - g_0\omega)(h_2 - g_2\omega).
\]

(3.71)

This expression is the same as inequality (3.69) because of the definitions (3.61).
We now turn to the assertions of the lemma and prove first that \( \sigma \) is positive. This result follows from the definition (3.62) when \( n>0 \) so we have only to consider the case when \( n\leq 0 \). We make use of the identity

\[
(g_2 h_0 - g_0 h_2)^2 (g_2^2 - 4g_0 g_2) = (g_0 g_1 h_2 - 2g_0 g_2 h_1 + g_1 g_2 h_0)^2 \\
+ g_0 g_2 \{ (h_1^2 - 4h_0 h_2) (g_1^2 - 4g_0 g_2) - (2g_0 h_2^2 + 2g_2 h_0 - g_1 h_1)^2 \} 
\]  

(3.72)

for with expression (3.61) it gives the condition

\[
n (g_2 h_0 - g_0 h_2)^2 \geq \frac{1}{4} (g_1^2 - n)(ln-m^2). 
\]  

(3.73)

It follows when \( n\leq 0 \) that \( (ln-m^2) \) is not positive so we have the inequality

\[
(n-l)^2 + 4m^2 \geq (n+l)^2. 
\]  

(3.74)

The two sides of expression (3.74) are equal only when \( (ln-m^2) \) is zero, but in this case it may be shown that the conic degenerates into two straight lines which has been accounted for already. Therefore expression (3.62) is positive which is the first result of the lemma.

We have noted that the equation (3.64) has real roots if the inequality

\[
l + 2m \theta + n \theta^2 \geq 0 
\]  

(3.75)

holds. Expression (3.63) and some manipulation provide the equation

\[
\frac{l+\theta^2}{l+2m \theta + n \theta^2} = \sigma 
\]  

(3.76)

and we have proved already that \( \sigma \) is positive. Therefore condition (3.75) is obtained which proves the second result of the lemma.
To prove the third result we show that the least value of expression (3.60) is obtained when \( \tau \) is a root of the equation (3.64). Because the functions (3.65) and (3.60) are equal when \( \omega \) has the value (3.66), because \( \omega = \theta \) when equation (3.64) holds, and because real values of \( \tau \) make \( \omega \) satisfy condition (3.69), it is equivalent to show that the minimum value of expression (3.65) subject to the condition (3.69) is obtained when \( \omega = \theta \). We make use of the equation

\[
(l+2m\theta+n\theta^2)(1+\omega^2) - (l+2m\omega+n\omega^2)(1+\theta^2)
= (\omega-\theta)(l-n)(\omega+\theta) + 2m\omega-2m
= (\omega-\theta)^2(l-n+2m\theta)
= (\omega-\theta)^2 \sqrt{(n-l)^2+4m^2}, \tag{3.77}
\]

whose third line is derived from the fact that the parameter (3.63) satisfies the quadratic equation

\[
m\theta^2 + (l-n)\theta - m = 0 \tag{3.78}
\]

and whose fourth line depends on the definition (3.63). Expressions (3.65), (3.69), (3.75) and (3.77) provide the inequality

\[
\psi(\omega) \geq \psi(\theta), \tag{3.79}
\]

which is the required result. Since equation (3.76) shows that \( \psi(\theta) \) is equal to \( 4D^2\sigma^2 \) the third assertion of the lemma is true.

Finally we must show that if the roots of the equation (3.64) are outside the range \([\tau_0, \tau_f]\) then the least value of \( \rho(\tau) \) for \( \tau \in [\tau_0, \tau_f] \) is \( \min[\rho(\tau_0), \rho(\tau_f)] \). Therefore we have to show that the only minima of \( \rho(\tau) \) occur when \( \tau \) satisfies equation (3.64). The turning points of \( \rho(\tau) \) occur when the derivative \( \rho'(\tau) \) is zero. From equations
(3.65) and (3.66) we find that this derivative has the value

$$\rho'(\tau) = \psi'(\omega) \omega'(\tau), \quad (3.80)$$

where $\psi'(\omega)$ and $\omega'(\tau)$ are the expressions

$$\psi'(\omega) = \frac{8D^2\delta^2(m\omega^2 + (\lambda-n)\omega-m)}{(\lambda+2m\omega+n\omega^2)^2}, \quad (3.81)$$

and

$$\omega'(\tau) = \frac{(g_1h_2-g_2h_1) + 2(g_2h_0-g_0h_2)\tau + (g_0h_1-g_1h_0)\tau^2}{(g_2-g_1\tau+g_0\tau^2)^2} \quad (3.82)$$

the last line being a consequence of expression (3.67). The sign of $(\alpha+2\beta\tau+\gamma\tau^2)$ is constant for the range of $\tau$ because the method of calculation of this range avoids both branches of a hyperbola. Therefore the turning points of $\rho(\tau)$ occur only when $\tau$ causes the quantity (3.66) to satisfy the equation

$$m\omega^2 + (\lambda-n)\omega-m = 0. \quad (3.83)$$

Already we have taken account of the root (3.63) of this equation so it remains to consider the other root

$$\bar{\theta} = \frac{n-n-\sqrt{(n-\lambda)^2+4m^2}}{2m} \quad (3.84)$$

We may replace $\theta$ by $\bar{\theta}$ in expression (3.77) and we find that the expression remains the same except that the sign of the last line must be altered to take account of the negative square root. It follows that
when \( \bar{\tau} \) yields a real value of \( \tau \) then the condition

\[
\psi(\omega) \leq \psi(\bar{\tau})
\]

(3.85)

is satisfied, so the corresponding stationary values of \( \rho(\tau) \) are maxima. Because we have now considered all the zeros of \( \rho'(\tau) \) the last part of the lemma is proved.

It is important to note that Lemma 3 can be extended to allow the range of \( \tau \) to include points at infinity. For example a single section of the required contour may be obtained by letting \( \tau \) increase from 3.0 to \( +\infty \) and then letting \( \tau \) increase from \( -\infty \) to 1.5. The extension is valid because, although the function (3.82) usually tends to zero as \( \tau \) tends to infinity, we find that the signs of \( \omega'(M) \) and \( \omega'(-M) \) are the same when \( M \) is large. Therefore the function \( \rho(\tau) \) remains monotonic through \( \rho=\infty \) unless there is a zero of expression (3.81), in which case the analysis in the proof of the lemma still holds.

The subroutine applies Lemma 3 to calculate the least value of expression (3.60) for each interval in \( \tau \) that gives a required section of the contour. To help this calculation the quantities (3.46),(3.47), (3.61),(3.62),(3.63) and one of the roots of equation (3.64) are evaluated and recorded before any sections of the contour are treated. For each section we let \( \chi \) be the positive number such that \( 4D^2 \delta^2 \chi^2 \) is the least value of expression (3.60).

It follows from inequality (3.59) that the value of \( \mu \) for a section of the contour must satisfy the condition

\[
\mu^2 \leq 2\Delta \chi \sqrt{1+\mu^2(\alpha\gamma-\beta^2)} \left[ 1+ \sqrt{1+\mu^2(\alpha\gamma-\beta^2)} \right],
\]

(3.86)

which can be written in the form
\[ \sqrt{1 + \mu^2 (\alpha y - \beta^2)} \left( \frac{1}{\alpha y - \beta^2} - 2\Delta x \right) \leq \frac{1}{\alpha y - \beta^2} \quad (3.87) \]

Because \( \Delta x \) is positive it follows that, if the inequality

\[ \Delta x (\alpha y - \beta^2) \geq \frac{1}{2} \quad (3.88) \]

holds, then the condition on \( \mu \) is always obtained, so we approximate the whole section of the contour by drawing the chord that joins its end points. Otherwise we satisfy the condition on \( \mu \) by making the two sides of expression (3.87) equal which gives the quantity

\[ \mu = \frac{\pm 2[\Delta x (1 - \Delta x (\alpha y - \beta^2))]^{\frac{1}{2}}}{1 - 2\Delta x (\alpha y - \beta^2)} \quad (3.89) \]

Note that this value always satisfies condition (3.29). The freedom of the \( \pm \) sign in expression (3.89) allows us to follow the contour in either direction. When we study the Fortran listing we will see that this freedom is used to make small the pen movement of the graph plotter between different sections of the contour.

Having chosen the starting point of the range of \( \tau \), say \( \tau_0 \), and having chosen the value of \( \mu \), we apply equation (3.28) for \( k = 0, 1, 2, \ldots \) to generate the sequence \( \{\tau_0, \tau_1, \tau_2, \ldots\} \) until a point of the sequence is outside the range of \( \tau \). This point is replaced by the final point of the range of \( \tau \). Thus we obtain a sequence of chords that is a sufficiently accurate approximation to the contour section.

When the range of \( \tau \) includes infinity, for example we noted before that the range \( 3.0 \leq \tau \leq 1.5 \) is possible, then the method of generating the sequence of \( \tau \) values does not require modification. Equation (3.28) causes a suitable switch from large positive values of \( \tau \) to large negative values automatically. This situation is analogous to
letting \( \tau \) be \( \tan^{-1}\left(\frac{1}{3}\theta\right) \) for a range of values of \( \theta \) that includes \( \pi \) when the parametric form (3.44) is followed. When the value of \( \theta \) is varied through \( \pi \) then \( \tau \) switches from a large positive to a large negative value, and it is satisfactory to let the values of \( \tau \) be derived from equally spaced values of \( \theta \) that are sufficiently close.

However it may be necessary to represent infinite values of \( \tau \) in the computer. The method we use is to let \( \tau \) be the ratio \( TNUM/TDEN \) where \( TNUM \) and \( TDEN \) are separate Fortran variables. It does not matter if \( TDEN \) becomes zero because the division of \( TNUM \) by \( TDEN \) is unnecessary due to the fact that the homogeneity of the coordinates permits expression (3.9) to be written in the form

\[
\begin{align*}
    x(\tau) &= x_0 \frac{TDEN^2+2x_1 TNUM*TDEN+x_2 TNUM^2}{TDEN^2+2y_1 TNUM*TDEN+y_2 TNUM^2} \\
    y(\tau) &= y_0 \frac{TDEN^2+2y_1 TNUM*TDEN+y_2 TNUM^2}{TDEN^2+2y_1 TNUM*TDEN+y_2 TNUM^2} \\
    z(\tau) &= z_0 \frac{TDEN^2+2z_1 TNUM*TDEN+z_2 TNUM^2}{TDEN^2+2z_1 TNUM*TDEN+z_2 TNUM^2}
\end{align*}
\]  

(3.90)

The number (3.63) and the calculated root of the quadratic equation (3.64) are also expressed as the ratios of separate Fortran variables because they too may be infinite. Details are given in the Fortran listing.

We have displayed and explained most of the mathematical formulae that are used by the subroutine. Note that the formulae that occur in the proofs of the lemmas are not used directly by the calculation, so the amount of work that is required to apply the method is substantially less than is indicated by the length of this section.

4. **Comments on the Fortran listing**

This section explains the details of the Fortran subroutine. We refer to the lines of the subroutine by the numbers that are given in the left hand column of the listing in the Appendix. We run through the Fortran instructions in order.
The subroutine arguments and the COMMON statement in line 0002 have been explained already in Section 2. The DIMENSION statement introduces several private arrays whose purpose will be given when they are used. The names of these arrays suggest their purpose. The integer variable NA, which is set to zero in line 0004, remains at zero until the separate sections of the conic are treated. The integer variable NP is set to the number of items in the arrays XARC and YARC that are waiting to be passed to subroutine O813Z. The value of ISW is greater than or equal to two if and only if it may be possible to join the next section of the contour in the present triangle continuously to the data that is in the arrays XARC and YARC.

The instructions of lines 0007 to 0018 set ABC(1),ABC(2),ABC(3), FGH(1),FGH(2),FGH(3),RST(1),RST(2),RST(3),UVW(1),UVW(2),UVW(3),ZEZ(1), ZEZ(2) and ZEZ(3) to the values of a,b,c,f,g,h,r,s,t,u,v,w,ξ,η and ζ respectively, which are defined by equations (3.3),(3.6) and (3.7). They set DELTA and SUMZEZ to expression (3.3) and (ξ+η+ζ). The integers IA(I) and IB(I) set in lines 0008 and 0009 are used to avoid many special cases. For example, in addition to the coefficients (3.15), which are obtained by letting \((x_0,y_0,z_0)\) and \((x_2,y_2,z_2)\) be on the line through \((ξ,η,ζ)\) and \(C\), there are two sets of corresponding coefficients for the cases when \((x_0,y_0,z_0)\) and \((x_2,y_2,z_2)\) are on the line joining \((ξ,η,ζ)\) to either A or B. The elements of IA and IB allow the computer instructions for these three cases to be combined. Consequently some array addresses have the form IA(I) and IB(I), for instance see line 0011, which is not standard Fortran. We prefer not to be restricted by the conditions on array subscripts that are imposed by standardization.

Instruction 0020 is obeyed after instruction 0019 if and only if the conic is an ellipse whose centre is strictly inside triangle ABC.
In this case the integer J is set in line 0023 to 1, 2 or 3 to indicate the largest member of the set \{aδ, bδ, cδ\}, except that if the largest member is not positive there is a branch from instruction 0026 to the end of the subroutine because there is nothing to be plotted. When the largest number is positive the value of J indicates which vertex of the triangle is to be joined to \((ξ, η, ζ)\) to help the calculation of the coefficients of the parametric form (3.9).

These coefficients are evaluated by instructions 0027 to 0040. For \(i=1, 2\) and 3 they set the array elements \(\text{PAR}(1,i), \text{PAR}(2,i)\) and \(\text{PAR}(3,i)\) to \(x_{i-1}, y_{i-1}\) and \(z_{i-1}\) respectively. The calculation takes account of the value of J so that equations (3.15) are applied faithfully only when \(J=3\). Otherwise the equations are rearranged to take account of the fact that the line joining \((x_0, y_0, z_0)\) to \((x_2, y_2, z_2)\) passes through the vertex A or B instead of through C. Then there is a jump to line 0073 to calculate where the conic intersects the sides of the triangle.

Line 0042 is reached when the conic is not an ellipse whose centre is inside the triangle ABC. In this case instructions 0042 to 0056 calculate expression (3.19) for each side of the triangle and they set the integer J to indicate the side of the triangle that gives the greatest value of this expression provided that the greatest value is positive. The test in instruction 0047 is satisfied only when the quadratic function has a turning point in the line segment that is considered, and the test is made because when the test is satisfied the expression (3.19) is evaluated from a formula that is different from the one that is used when the quadratic function is monotonic on the line segment. Specifically, when the line segment is the one shown in Figure 2 and when the quadratic function is monotonic, then expression

-37-
(3.19) has the value

\[ 2|a-b| - \left| \frac{5}{2}a + \frac{5}{2}b - 3h \right|, \]  

(4.1)

which is set by instruction 0048. Alternatively, when there is a turning point in this line segment, then \( m \) and \( M \) have the values

\[ \frac{5}{4}a + \frac{5}{4}b - \frac{3}{2}h + \text{sign}(a-b,a+b-2h) \]

and \( (ab-h^2)/(a+b-2h) \)

(4.2)

which are set by instructions 0050 and 0051. Again the use of the arrays IA and IB provides the corresponding formulae for the other two sides of the triangle. Instruction 0057 branches to the end of the subroutine when no positive value of expression (3.19) is found because there is nothing to be plotted. Note that this branch is taken when the quadratic function (3.2) is identically zero, which happens when all the data function values \( FD(I) \) \( (I=1,2,\ldots,6) \) are equal to the contour height.

The parameters (3.24) of the parametric form of the conic are calculated by instructions 0058 to 0072. Note that when \( \delta \) is zero the branch 0070 omits the factor \( \delta \) that occurs in the formulae for \( x_2 \) and \( y_2 \). Thus, as explained in Section 3, we obtain numbers that specify the contour in the case when the conic degenerates into two straight lines.

In the usual case, when \( \delta \) is non-zero, instructions 0073 to 0089 place in the array \( W \) the finite values of \( \tau \) for which the parametric form (3.9) cuts the lines \( BC, CA \) and \( AB \). The number of these values is equal to the final value of the integer \( II \). They are found by calculating the roots of each of the functions \( x(\tau), y(\tau) \) and \( z(\tau) \). Often some coefficients of these quadratics are zero, in particular when \( z_0 \) and \( z_2 \) are defined by equation (3.24). Therefore the branch 0079 allows for the possibility that a coefficient of \( \tau^2 \) or of both \( \tau \) and \( \tau^2 \)
may be zero. When a coefficient of $\tau^2$ is zero then the conic cuts a triangle side at $\tau=\infty$, but this case is not included in the list of $\tau$ values that is assembled in the array W. Instead the variable C1 indicates whether the conic is inside the triangle when $\tau$ is large and positive. The method used depends on the signs of the dominant coefficients of $x(\infty), y(\infty)$ and $z(\infty)$. The final value of C1, calculated by instruction 0089, has modulus six if the signs of $x(\infty), y(\infty)$ and $z(\infty)$ are all the same. It has modulus four if the sign of $x(\infty)$ is opposite to the signs of $y(\infty)$ and $z(\infty)$. It has modulus two if just $x(\infty)$ and $z(\infty)$ have the same sign and it is zero if just $x(\infty)$ and $y(\infty)$ have the same sign. This information is useful later. Note also that an entry is made in the array IW for each entry in W. It has the value 1, 2 or 3 to show whether the corresponding value of $\tau$ is a root of $x(\tau), y(\tau)$ or $z(\tau)$. The information obtained by this block of the subroutine is sufficient to calculate the intervals in $\tau$ for which the conic is inside triangle ABC.

The required ranges of $\tau$ are obtained by instructions 0090 to 0123. They calculate which triangle sides the contour crosses as $\tau$ decreases continuously and monotonically from $+\infty$ to $-\infty$. The integer variable NW becomes the number of end points of contour sections that have to be plotted. The values of $\tau$ at these end points are assembled in the array TEND, which is two dimensional because we let the $i^{th}$ value of $\tau$ be $\text{TEND}(i,1)/\text{TEND}(i,2)$ in order that points at infinity can be represented. The roots of the functions (3.9) divide the range of $\tau$ into intervals. As $\tau$ is decreased the integer variable ISIDE is modified for each interval so that it gives information about the signs of $x(\tau), y(\tau)$ and $z(\tau)$ in the following way. ISIDE is zero when the three signs are the same and it is 1, 2 or 3 when $x(\tau), y(\tau)$ or $z(\tau)$ respectively
have the odd sign. Therefore the initial value of ISIDE, set by instruction 0090, is appropriate when \( \tau \) is large and positive. The integer I2, set by instruction 0091, saves the initial value of ISIDE because it is needed later. When ISIDE is zero initially line 0094 is reached because \( \tau=+\infty \) can be one of the required end points. The value \( +\infty \) is set at the beginning of the array TEND by instructions 0095 and 0096, the sign of TEND(NW,1) distinguishing \( +\infty \) from \( -\infty \). Instruction 0097 branches back to line 0094 only when the required contour is a ring contour and NW=1. In this case instructions 0095 and 0096 set the next entry in TEND to \( -\infty \) in order that the range of \( \tau \) is \( +\infty > \tau > -\infty \). Instruction 0098 branches to line 0124 when Il is zero because then all values of \( \tau \) that can be end points of contour sections have been treated. However, at line 0099, the value of Il is positive and is equal to the number of roots of the functions (3.9) that are less than the current interval in \( \tau \), these roots being the numbers \( W(1), W(2), \ldots, W(Il) \). Lines 0099 to 0104 set Cl to the largest of these roots because this is the next value of \( \tau \) that gives a point on BC, CA or AB as \( \tau \) decreases. \( IW(J) \) is the corresponding element of the array IW. Therefore, because of the purpose of ISIDE given above, the point on the conic at \( \tau=Cl \) is external to the triangle ABC if and only if ISIDE is non-zero and is different from \( IW(J) \). In this case the value of ISIDE for the next interval in \( \tau \) is \{6-ISIDE-IW(J)\} and it is set by instruction 0107 after which there is a branch to line 0119. However, if ISIDE is zero at instruction 0105, then line 0113 is reached because the contour leaves the triangle at \( \tau=Cl \). Line 0113 is also reached when ISIDE=IW(J) and when \((Il+I2)\geq2\) because in this case the contour enters the triangle at \( \tau=Cl \). Lines 0113 to 0116 store the value \( \tau=Cl \) in the array TEND and they change ISIDE to \{IW(J)-ISIDE\} for the next interval in \( \tau \).
case $(11+I2)\leq 1$ and $ISIDE=IW(J)$ is special although it is still known
that the contour enters the triangle at $\tau=Cl$. In this case $11$ is one
and $I2$ is zero. The value $11=1$ indicates that the next range of $\tau$
extends to $-\infty$ and the value $I2=0$ indicates that large positive values of
$\tau$ also give contour points that are inside the triangle. Thus a single
section of the contour inside the triangle is obtained from the range of
$\tau$ that decreases from $Cl$ to $-\infty$, then jumps to $+\infty$ and then decreases
to the value of $\tau$ that is present in the second entry of the array TEND.
Therefore instructions 0110 and 0111 place the value of $Cl$ at the
beginning of the array TEND, overwriting the value $+\infty$ that was set
earlier. In this case it is convenient to set $TEND(1,2)$ to ten to
indicate that the range of $\tau$ includes points at infinity. Then
instruction 0112 branches to line 0124 because the required entries in
TEND are complete. The purpose of line 0118 is to exclude double roots
from TEND, which may occur for example if the contour line passes
through a vertex of the triangle ABC without going into the interior of
the triangle. At line 0119 the value $\tau=Cl=W(J)$ has been treated so it
is overwritten by $W(11)$ in the list of numbers $W(1),W(2),...,W(II)$ and
then $11$ is reduced by one. If $11$ is still positive instruction 0122
branches back to line 0099 because some more roots of the functions (3.8)
will be found when $\tau$ is reduced further. Otherwise line 0123 is reached
and the current interval in $\tau$ extends to $-\infty$. If $ISIDE$ is zero this
interval gives a section of the contour that is inside the triangle ABC
so instruction 0123 branches to line 0094 to add the point $-\infty$ to the
entries in the array TEND. In all cases, when this rather long section
of the subroutine is finished, instruction 0124 is reached.
Line 0124 tests the value of NW and branches to the end of the subroutine if it is zero because there is nothing to be plotted. For example this case can occur if the contour line cuts the line segment ST of Figure 2 but does not cut the segment AB. Instructions 0125 to 0135 calculate the Cartesian coordinates of the end points of the contour sections that are to be plotted. In fact these coordinates are obtained by the branch from line 0129 to line 0233, where equations (3.90) and (3.1) are applied. The program flow comes back to line 0130 because NA was set to zero in line 0004. The calculated Cartesian coordinates are stored in the arrays XEND and YEND by instructions 0130 and 0131, and an integer is stored in the array IW by instruction 0132. The elements of IW will be used as pointers to avoid moving information in arrays when the elements of TEND are condensed after each section of the contour is plotted. The branch 0133 is taken only in the degenerate case when the contour is composed of two straight lines and we see later why it is present. Instruction 0134 stores the sum of the three expressions (3.90) in the array ZEND because it is required later.

The accuracy parameter (2.1) and the determinant D, defined by equation (3.27), are calculated by lines 0136 to 0141. Instruction 0139 sets the variable ACC to the accuracy parameter. This quantity is required, not only when line 0136 is reached from line 0135, but also when the conic degenerates into two straight lines. In this case instruction 0136 is reached from line 0070 of the Fortran listing and when ACC has been calculated instruction 0140 branches to line 0167. Otherwise instruction 0141 multiplies DELTA by 2D. This is done instead of storing the value of the determinant because in all later calculations DELTA and D do not appear separately but the product (2D*DELTA) does occur, for example see inequality (3.58). We noted in Section 3 that the
determinant (3.27) is zero when the vertices of the triangle ABC are
collinear. In this case instruction 0142 branches to line 0186
because the curvature of the contour in the space of homogeneous co-
ordinates is not relevant.

Lines 0143 to 0166 of the subroutine calculate the quantities
that are usually needed to take account of the curvature of the conic.
Instruction 0144 applies equation (3.18) to set ABG(1), ABG(2) and
ABG(3) to \( \alpha, \beta \) and \( \gamma \) respectively, instruction 0148 calculates each element
of the matrix (3.48) as \( I \) and \( J \) range over the integers \( \{1,2,3\} \), and
instructions 0149 and 0150 set \( G(I) \) and \( H(I) \) to the components of the
vectors (3.46) and (3.47). Note that a little work is saved by using
the differences that are calculated by instructions 0137 and 0138 of the
previous section of the listing. Then lines 0151 to 0153 set \( AL, AM \)
and \( AN \) to \( \lambda, m \) and \( n \) respectively, which are defined by the equations
(3.61). At line 0160 the ratio \( THNUM/THDEN \) is equal to expression
(3.63). We use a ratio because one or both of \( THNUM \) and \( THDEN \) may be
zero. They are both zero when exact arithmetic is used and when the
required contour is a circle. This case is treated correctly by the
subroutine. Otherwise we normalize \( THNUM \) and \( THDEN \) so that the larger
modulus is one because this helps to prevent overflow or underflow in
extreme cases. Line 0161 gives the variable SIGMA the value (3.62).
Lines 0163 to 0165 set the variables \( TAUNUM \) and \( TAUDEN \) so that the ratio
\( TAUNUM/TAUDEN \) is a root of the quadratic equation (3.64). Then there is
a branch to line 0186 of the subroutine because we are now ready to
approximate the separate sections of the contour.

The instructions 0167 to 0185 calculate the end points of the
sections of the contour in the degenerate case when the conic is composed
of two straight lines. The Cartesian coordinates of these end points

-43-
are accumulated in the arrays XEND and YEND while NEND becomes the number of end points, which is zero, two or four. Thus the purpose of NEND, XEND and YEND is the same as in instructions 0124 to 0135. We recall that the coefficients set in lines 0061 to 0069 are such that in homogeneous coordinates one line passes through \( \{\text{PAR}(1,1),\text{PAR}(2,1), \text{PAR}(3,1)\} \) and \( \{\text{PAR}(1,2),\text{PAR}(2,2),\text{PAR}(3,2)\} \); and the other line passes through \( \{\text{PAR}(1,2),\text{PAR}(2,2),\text{PAR}(3,2)\} \) and \( \{\text{PAR}(1,3),\text{PAR}(2,3),\text{PAR}(3,3)\} \). The two lines are treated by the two cycles of the loop from instruction 0168 to 0184. Instruction 0170 sets \( W(1), W(2) \) and \( W(3) \) so that the point \( (x,y,z) \) is on the line under consideration if and only if the equation

\[
x \cdot W(1) + y \cdot W(2) + z \cdot W(3) = 0 \tag{4.3}
\]

holds. This line passes through a vertex of the triangle if line 0173 is reached from line 0172 in which case the Cartesian coordinates of the vertex are set by instructions 0174 and 0175. Moreover line 0178 is reached from line 0177 if the line \( (4.3) \) intersects a side of the triangle. In this case instructions 0179 to 0181 set the elements \( \{XYZ(1),XYZ(2),XYZ(3)\} \) to the homogeneous coordinates of the point of intersection. Then line 0182 branches to line 0235 in order that the required Cartesian coordinates are calculated by instructions 0236 and 0237. Next line 0238 branches back to instruction 0130 where the coordinates are placed in the arrays XEND and YEND, after which line 0133 branches forward to instruction 0183. Line 0183 is the last statement of the loop that is begun by instruction 0171. The three cycles of the loop causes the three vertices and the three sides of the triangle ABC to be inspected for a point on the line \( (4.3) \). Usually the correct coordinates are set in XEND and YEND at the end of the loop, including
the case when the line is a side of the triangle, but there is one case that requires correction. It is that the line (4,3) may cut just one vertex of the triangle and no sides between vertices. Then the part of the line inside the triangle reduces to a single point so should not be plotted. In this case the value of NEND is odd at instruction 0184 but in all other cases the value of NEND is even. Therefore instruction 0184 reduces NEND by one when it is odd, which annuls the unwanted coordinates in XEND and YEND. Finally instruction 0185 branches to the end of the subroutine if it finds that NEND is zero, because in this case the conic does not cut the perimeter of triangle ABC so there is nothing to be plotted.

Line 0186 is an important point in the subroutine because it is reached in all cases when a section of the conic is to be plotted. It is here that we begin to treat the sections of the conic separately. At this stage there may be one, two or three sections still to be plotted and the number is equal to \( \frac{1}{2} \text{NEND} \). The Cartesian coordinates of the end points of the sections have the components \( \{ \text{XEND}(\text{IW}(I)), \text{YEND}(\text{IW}(I)) \} \) for \( I = 1, 2, \ldots, \text{NEND} \). If the integer \( \text{NP} \), set initially in line 0005, is at most one, then nothing is known about the previous sections of the contour. However, if it is greater than one, then the point \( \{ \text{XARC}(\text{NP}), \text{YARC}(\text{NP}) \} \) will be plotted immediately before the next section of the conic. Instructions 0186 to 0196 select one of the points \( \{ \text{XEND}(\text{IW}(I)), \text{YEND}(\text{IW}(I)) \} \) (\( I = 1, 2, \ldots, \text{NEND} \)) to plot next. They set \( \text{NA} \) to the chosen value of \( I \) and \( \text{NB} \) to \( \text{IW}(\text{NA}) \). For definiteness they define \( \text{NA}=1 \) when \( \text{NP} \leq 1 \). Otherwise the value of \( \text{NA} \) is calculated to minimize the distance from \( \{ \text{XARC}(\text{NP}), \text{YARC}(\text{NP}) \} \) to the next section of the contour.
Instructions 0197 to 0207 cause the points \{XARC(I),YARC(I)\} (I=1,2,...,NP) to be transmitted to subroutine OB13Z if necessary and they set some variables. When NP≤1 there is no useful information in XARC and YARC so instruction 0197 branches to line 0203. Otherwise subroutine OB13Z is called unless it is appropriate to join the next section of the contour continuously to the previous one. A continuous join is made only when the tests of instructions 0199 and 0201 do not hold. It is a consequence of the value of ISW set in lines 0006 and 0200 and of instruction 0197 that the test of instruction 0199 is satisfied when the previous section of the contour was obtained by the present call of subroutine OB13A. In this case the method of our algorithm ensures that the next section of the contour is away from the previous one, so subroutine OB13Z is called to avoid a continuous join. The test 0201 is satisfied if the distance from the last point of the previous contour section to the first point of the next one is greater than 0.1*ACC. We call subroutine OB13Z in this case because it is almost certain that the calculated gap between the sections of the contour should really be obtained. The test makes use of the fact that the length of the gap was set by instruction 0192. The call of subroutine OB13Z is obtained by a jump to line 0244. Then a branch back to line 0203 is made because NA is positive and line 0198 has set NC to zero. Instructions 0203 to 0205 begin the piecewise linear approximation to the next section of the conic by placing the coordinates of its first point in \{XARC(1),YARC(1)\}. Instruction 0206 sets ND to one if NB is odd and ND is minus one if NB is even. It follows from the value of NB set in line 0196 that the far end of the next section of the conic is \{XEND(NC),YEND(NC)\}, where NC is set by instruction 0207. In addition ND has another important property, which is that it
indicates whether the parameter $\tau$ of equation (3.9) must be increased or decreased to follow the contour section in the required direction. Because the end points of the contour sections were found by letting $\tau$ range from $\pm \infty$ to $-\infty$, we see that $\tau$ must be decreased if ND is positive and it must be increased if ND is negative. This remark is true even when the range of $\tau$ includes points at infinity, except that the jump from $-\infty$ to $-\infty$ or from $-\infty$ to $\infty$ that occurs in the range of $\tau$ is in the direction that is opposite to the main direction of $\tau$.

Instruction 0208 branches to line 0240 if DELTA is zero because in this case the contour section is a single straight line. Otherwise instructions 0209 to 0227 calculate the value of $\mu$ by the method described in Section 3 for use in equation (3.28). The loop from instruction 0209 to 0217 sets $W(1)$ and $W(2)$ to $\rho(\tau_o)/\text{DELTA}**2$ and $\rho(\tau_f)/\text{DELTA}**2$, where $\tau_o$ and $\tau_f$ are the values of $\tau$ at the ends of the section of the contour and where $\rho(\tau)$ is expression (3.60). The loop also assigns values to $W(3), W(4), W(5)$ and $W(6)$ through instructions 0216 and 0217. Note that line 0215 makes use of the fact that instruction 0134 sets the elements of ZEND to values that correspond to $(\alpha + 2\beta \tau + \gamma \tau^2)$. Line 0218 sets DELTT to expression (3.62). Due to instructions 0211, 0212, 0219 and 0220 the values of $\tau_o$ and $\tau_f$ are TNUM/TDEN and TNFIN/TDFIN respectively. Only the sign of CI occurring in instruction 0221 is important. Because of the values of TEND(NW,2) set in lines 0096, 0111 and 0116, the sign of CI is positive unless the range of $\tau$ includes a jump from one end of the real line to the other end. In other words CI is positive if and only if the values of $\tau$ between $\tau_o$ and $\tau_f$ give the required contour section. Instructions 0222 and 0223 apply Lemma 3 by leaving DELTT unchanged if equation (3.64) has a root in the range of $\tau$. Otherwise they increase DELTT to the minimum of $W(1)$ and $W(2)$, in order that the least value of expression (3.60) for
the contour section is $\text{DELT}^*\text{DELT}**2$, where $\text{DELT}$ is set to the value $2DS$ by line 0141. One case when the left hand side of expression (3.64) has a root in the range of $\tau$ is when its sign at $\tau_0$ is opposite to its sign at $\tau_f$. In this case instruction 0222 branches to line 0224 in order that $\text{DELT}$ is not increased. Otherwise there is a root in the range of $\tau$ if and only if there are two roots in the range of $\tau$, so it is sufficient to test if TAUNUM/TAUDEN is in this range, where the values of TAUNUM and TAUDEN are assigned in lines 0164 and 0165. The root TAUNUM/TAUDEN is between $\tau_0$ and $\tau_f$ if and only if the product $W(5)*W(6)$ is negative. It follows from the value of C1 set in line 0221 that $\text{DELT}$ must be increased when $C1\text{*}W(5)*W(6)$ is positive, which is done in line 0223. Instruction 0224 sets C2 to the value of $\chi$ that occurs in inequality (3.86) and instruction 0225 sets C3 to the difference between the right and left hand sides of expression (3.88).

In accordance with the method of Section 3 when C3 is not positive, instruction 0226 branches to line 0240 to approximate the whole contour section by a single chord. However, if C3 is positive, then instruction 0227 sets WMU equal to the required value of $\mu$, which is defined by equation (3.89) except that the choice of sign depends on whether we require $\tau$ to increase or decrease through its range. Therefore this sign depends partly on the direction of $\tau$ when equation (3.28) is applied recursively for a fixed value of $\mu$. To investigate this direction we obtain from equation (3.28) the identity

$$\tau_{k+1} = \frac{\mu(\alpha+2\beta\tau_k+\gamma\tau_k^2)}{1-\mu(\beta+\gamma\tau_k^2)}$$

(4.4)

and we are concerned with the sign of the right hand side. We have noted already that the sign of $(\alpha+2\beta\tau+\gamma\tau^2)$ is constant for each range of $\tau$ that
gives a finite section of the contour, so the sign of \((\alpha+2\beta\tau_k+\gamma\tau_k^2)\) is the sign of the variable ZEND(J) which occurs in line 0215. The sign of \([1-\mu(\beta+\gamma\tau_k)]\) is usually positive, especially when \(\mu\) is small. Further analysis shows that the denominator of expression (4.4) is negative only when the change from \(\tau_k\) to \(\tau_{k+1}\) corresponds to the range of \(\tau\) jumping from one end of the real line to the other. Therefore the direction of \(\tau\) is mainly increasing when formula (3.28) is applied if and only if \(\mu*ZEND(J)\) is positive. At the end of the last paragraph we noted that we require \(\tau\) to increase when the integer variable ND is negative. Therefore instruction 0227 assigns the correct sign to WMU.

Lines 0228 to 0232 apply equation (3.28) recursively to calculate the sequence \(\tau_k(k=0,1,2,\ldots)\) until the end of the range of \(\tau\) is reached. We consider the problem of recognising the end of the range because it requires some care. We have to take account of the value of \(\tau_f\), the direction of change of \(\tau\) and whether the range of \(\tau\) includes a jump from one end of the real line to the other. It is convenient to use the expression

\[TNUM*TDFIN - TDEN*TNFIN,\]  

(4.5)

where, because of instructions 0219 and 0220, \(\tau_f\) is equal to TNFIN/TDFIN, and where TNUM and TDEN are altered each time equation (3.28) is applied, the alteration being given by the formula

\[
\begin{align*}
TNUM &= TNUM + \mu(\alpha*TDEN + \beta*TNUM) \\
TDEN &= TDEN - \mu(\beta*TDEN + \gamma*TNUM)
\end{align*}
\]  

(4.6)

Since expression (4.5) is zero if and only if TNUM/TDEN is equal to \(\tau_f\), we find that the formula (4.6) causes the sign of expression (4.5) to change when \(\tau_k\) is inside but \(\tau_{k+1}\) is beyond the required range of \(\tau\).
Alternatively, when a sign change does not occur and when $\tau_k$ is inside the range of $\tau$, then $\tau_{k+1}$ is also inside the range. Thus the sign of expression (4.5) show when to terminate the sequence $\tau_k (k=0,1,2,\ldots)$. However we must be careful when setting the initial sign of expression (4.5) because, if the contour section is a ring contour and if TNUM/TDEN is equal to $\tau_0$, then expression (4.5) is zero. To allow for this case we calculate what the initial sign would be if we moved $\tau_0$ a very small distance into the range of $\tau$ that gives the contour section. Because the quantities set in lines 0111 and 0116 are positive and because the choice of sign made by instruction 0095 allows us to regard the sign of TEND(NW,2), set in line 0096, as positive, the value of TDFIN and the initial value of TDEN are essentially positive. Therefore the initial sign of expression (4.5) is the sign of $(\tau-\tau_f)$ when $\tau$ is near the beginning of its range. This sign can be obtained from two pieces of information, namely whether $\tau$ increases or decreases as we proceed from $\tau_0$ to $\tau_f$ and whether the range of $\tau$ includes a jump from one end of the real line to the other. It follows that initially expression (4.5) should have the sign of the product (C1*ND), which is set in line 0228 of the Fortran listing. Instructions 0229 to 0231 apply formula (4.6). Then line 0232 branches to line 0240 if and only if our test for the end of the range of $\tau$ shows that the piecewise linear approximation can be completed by drawing the chord from $\tau_k$ to $\tau_f$. Otherwise instruction 0233 is reached.

Instructions 0233 to 0237 calculate the Cartesian coordinates of a point on the contour section by applying equations (3.90) and (3.1) for the current values of TNUM and TDEN. The branch from instruction 0238 has been considered already. It is not taken when the sequence $\tau_k (k=0,1,2,\ldots)$ is being formed. Instead instruction 0239 branches to line 0243.
Instructions 0240 to 0242 set \((X, Y)\) to the Cartesian coordinates of the final point of the contour section, using the fact that these coordinates are already present in the arrays XEND and YEND. Line 0240 sets NB to zero to indicate that the piecewise linear approximation to the contour section does not require the coordinates of any more points.

In all cases when line 0243 is reached we require to draw the chord from \(\{XARC(NP),YARC(NP)\}\) to \((X, Y)\), which is a sufficiently accurate approximation to the conic between these two points. Because of the purpose of the arrays XARC and YARC it is sufficient to add \(X\) and \(Y\) to these arrays if there is room for them. There is room if \(NP\) is less than IARC, which is tested by line 0243. Otherwise subroutine OB13Z is called by instruction 0244 to empty the arrays. It passes the contents of XARC and YARC to the graph plotter. This instruction is also obeyed in two other circumstances, one of which has been discussed already and which are recognised by lines 0245 and 0246. Otherwise, in the case we are considering now, lines 0247 and 0248 are reached. They move the coordinates that were at the end of \(\{XARC, YARC\}\) to the beginning of these arrays and \(NP\) is set to one to correspond. Now there is always room to add the new point \((X, Y)\), which is done by instructions 0250 to 0252. Line 0253 branches back to line 0229 if the approximation of the contour section is not yet complete.

When the approximation is complete we must find whether there are any more contour sections to be plotted. If there are some more sections then again we will make use of the elements of the integer array IW as subscripts that point to the remaining end points of the contour sections. For example the subscript IW(I) occurs twice in line 0189. In case there are some more lines to be drawn, instructions 0254 to 0256 overwrite the elements of IW for the finished contour section by the
two elements that are at the end of the list $\text{IW(I) \{I=1,2,...,NEND\}}$. We may then reduce NEND by two. Its new value gives the number of end points of contour sections that have not yet been drawn, so instruction 0258 branches back to line 0186 if NEND is positive.

When all the sections of the contour have been approximated it remains to take account of the initial value of NPT, following the rules given in Section 2. Instructions 0259 and 0261 set NA to the required final value of NPT. Line 0262 is reached only if we have to pass to the graph plotter any chords that are held in the arrays XARC and YARC. There are some chords if NP≥2 so in this case line 0262 branches to the instruction that calls subroutine OB13Z. The flow comes back to line 0263 from instruction 0245 because line 0261 has set NA to zero. Line 0263 copies the value of NA into NPT and then the return from subroutine OB13A is made.

5. Discussion

One reaction to this report may be that the amount of work seems to be too great for such a basic calculation. We were surprised at the length of the final subroutine so wish to make two remarks on this point. The first is that, in addition to the given subroutine, we programmed a version that uses Cartesian coordinates throughout and it applies the parametric form

\[
\begin{align*}
X(t) &= \left( a_0 + a_1 t + a_2 t^2 \right) / \left( c_0 + c_1 t + c_2 t^2 \right) \\
Y(t) &= \left( b_0 + b_1 t + b_2 t^2 \right) / \left( c_0 + c_1 t + c_2 t^2 \right)
\end{align*}
\]

(5.1)

which traces a conic as $t$ varies. We found that the alternative subroutine requires more special cases to take account of degenerate situations so the computer code is longer. The amount of computation is greater also on the test problems we tried but the times are
sufficiently close for this factor to be unimportant. The other remark on the length of the subroutine is that it is possible to do the calculation described in Section 3 by hand in about one hour. Therefore the execution time on an automatic computer is quite short, mainly because most of the instructions are obeyed only once. For instance the approximation to a ring contour was obtained in 1.6 milliseconds. The details of this example are given later.

A different method of obtaining a piecewise linear approximation to the conic in the triangle ABC is to subdivide the triangle into smaller triangles whose sides are not too large. For instance we may bound the length of each side by ten times the required accuracy. The value of the quadratic function (3.2) is calculated at every small triangle vertex and then points on the contour are estimated by linear interpolation on each triangle side. These points may be joined by straight line segments or by a more sophisticated method, for example the one proposed by McConalogue (1970). A computer subroutine for this method would be shorter than OB13A but it would have the following disadvantages. A satisfactory procedure for obtaining the lengths of the sides of the small triangles does not suggest itself. The method is not suitable for providing the sharp corners that occur when a hyperbola almost degerates into two straight lines. Ring contours can be missed. The number of segments in the piecewise linear approximation to the conic may be much larger than necessary. Therefore we prefer to follow the parametric form (3.9) or (5.1).

At many computer installations the number of line segments obtained by the subroutine is an important factor. Therefore we have aimed to keep the number small by using a method that takes account of the curvature of the conic. For example at Harwell the line segments are buffered
by the GHOST system (Prior and Jones, 1972). They are processed separately for the graph plotter after they are taken from the buffer. Therefore, to reduce the work at this stage, we wish to make small the total number of line segments. We feel, therefore, that the rather elaborate calculation that is done by our subroutine to obtain the value of \( \mu \) for equation (3.28) is worthwhile.

We considered making two further refinements to subroutine OBL3A but decided that the extra work was not justified. Since the decision was a close one in both cases we mention these refinements because the reader may wish to include them. One is to let the value of \( \mu \) in equation (3.28) depend on \( k \) so that the distance of each chord from the conic is close to the required accuracy instead of this property being obtained only where the chords are shortest. We decided against this refinement because it complicates the inner loop of the algorithm, where the sequence \( \tau_k (k=0,1,2,...) \) is generated, and it hardly ever makes a substantial difference to the total number of line sections. The other refinement is to ensure that the last chord of each piecewise linear approximation is not very short. At present, when formula (3.28) gives a value of \( \tau_{k+1} \) that is outside the range of \( \tau \), the difference \( (\tau_f-\tau_k) \) may be very small so there is a tendency for the last chord to be shorter than the others. Instead we could extend the calculation of \( \mu \) so that the number of chords remains the same and the final value of \( \tau_{k+1} \) is equal to \( \tau_f \). Thus the lengths of the chords would be commensurate and also the accuracy of the approximation could improve a little. The refinement requires the calculation of arctangents when the conic is an ellipse and of hyperbolic arctangents when the conic is a hyperbola and it does make the output from the subroutine nicer. However again we judged that the extra calculation is not justified,
because we expect the user to choose the accuracy parameter so that the separate line segments of the approximation are not apparent when the contour is drawn, and then the disadvantage of our decision is not noticed.

The appendix includes a short test program that provides piecewise linear approximations to two conics. In both cases the coordinates of A, B and C are (2,0), (2,1) and (0,2) and the six data function values are \( f_A=1, \ f_B=4, \ f_C=9, \ f_P=\frac{1}{3}, \ f_Q=1 \) and \( f_R=\frac{1}{4} \), where the subscripts refer to the notation of Figure 1. The two conics are obtained by letting the contour height have the values \( H=-\frac{1}{4} \) and \( H=+\frac{1}{4} \). The first contour is a complete ellipse but the second contour cuts each triangle side twice so it is composed of three sections. Because the accuracy parameter is set to zero the subroutine uses its default value, which is the length of the longest side of the triangle divided by DD. The variable DD is set to one thousand in a BLOCK DATA statement. Instead of subroutine OBL3Z calling a graph plotting package it is programmed to print out the end points of the sections of the piecewise linear approximations. This output is also given in the appendix, the left hand columns being obtained from the ring contour and the right hand columns being obtained when \( H=\frac{1}{4} \). The last two lines of the right hand columns show a disadvantage that was discussed in the previous paragraph. It is that the last chord of the approximation is much shorter than the rest. Otherwise the calculated numbers are entirely satisfactory.

The calculations were done on an IBM 370/168 computer using the G compiler. We timed the approximation of the ring contour by deleting the printing and running the calculation one hundred times. The total computation time was 0.16 seconds so the time to process the ring contour is about 1.6 milliseconds. On real problems, where a contour
is being followed through a sequence of triangles, the average calculation
time per triangle is less because it is unusual for the contour to turn
through 360 degrees in a triangle.

One result of the rather small amount of computation per triangle is
that by comparison the time that is needed when there is nothing to be
plotted is significant - it is about 0.3 milliseconds. Therefore, if
a region is divided into a large number of small triangles and if the
present subroutine is applied to each triangle, then often more than
half of the computation time is spent on empty triangles. We recommend
therefore that, if the total amount of computation is expensive, then
the calling routine should include some tests to avoid using OB13A
when there is nothing to be plotted. It is planned to provide an outer
subroutine of this type for the algorithm described by Powell (1974),
which fits a piecewise quadratic function of two variables to function
values on a regular rectangular grid. A preliminary subroutine for
this algorithm has been in use at Harwell for about two years and it has
been applied to many real problems. The contour maps that it provides
are excellent so it is very worthwhile to develop tests that will save
computer time. This work will be reported later.
References


Figure 1
The triangular region

Figure 2
The extension of AB
Appendix

The Fortran listing of subroutine OB13A

and

a test program with output
SUBROUTINE OB13A (FC, XD, YD, ETA, DELT, NPT, XARC, YARC)
COMMON /OB13B/ DD, IAARC
DIMENSION FD(6)*XD(3), YC(3)*XARC(IAARC), YARC(IAARC), IA(3), TB(3),
1 W(6), ABC(3)*FGH(3)*RST(3)*UWV(3), ZEZ(3), ABG(3)*PAR(3)*G(3),
2H(3), XEY(3)*XEND(6), YEND(6), ZEEND(6), TEND(6,2)*W(6), WW(3)
NA=0
NP=IAABS(NPT)
ISw=NP

C
C CALCIULATE THE COEFFICIENTS OF THE CONIC AND ITS CENTRE
C
DC IC I=1,3
IA(I)=1+MCD(I,3)
IB(I)=1+MCD(I+1,3)
ARC(I)=FD(I)-FTA
FGH(I)=2*FD(I+3)-ETA-0.5*(FC(IA(I)) FC(IB(I)))
DO 20 I=1,3
RST(I)=ABC(I AIA(I))*ABC(IB(I))-FGH(I)**2
UWV(I)=FGH(I*FGH(IB(I))-ABC(I)*FGH(I)
DO 30 I=1,3
ZER(I)=RST(I)+UWV(I AIA(I))+UWV(I E(I))
DELTA=ABC(I)*RST(I)+FGH(I)*UWV(I)+FGH(3)*UWV(3)
SUMZEZ=ZEZ(1)+ZEZ(2)+ZEZ(3)
C
C TEST IF WE REQUIRE THE PARAMETRIC FORM FOR AN INSIDE ELLIPSE
C
IF (AWIN1(ZER(I), ZER(2), ZER(3)) .LE. 0.) GO TO 50
C1=0.
DO 40 I=1,3
IF (DELTA*ABC(I) .GE. C1) GO TO 40
J=I
C1=DELTA*ABC(I)
40 CONTINUE
IF (C1 .GE. 0.) GO TO 520
C
C CALCULATE THE PARAMETRIC FORM FOR AN INSIDE ELLIPSE
C
C1=ABC(J)*SUMZEZ-DELTA
C2=SYNC(ABS(C1/DFLTA))
C3=ABC(J)/C1
I1=IA(J)
I2=IB(J)
PARI(J+1)=ZEZ(J)-DELA*(1+C2)/ARC(J)
PARI(J+2)=(FGH(I2)-FGH(I1))/C2
PARI(J+3)=(ABC(I1)*ZEZ(I1)**2+2.*FGH(J)*ZEZ(I1)*ZEZ(I2)+
1ARC(I2)*ZEZ(I2)**2)/(C1+P8R(J,1))
PARI(I1,1)=ZEZ(I1)
PARI(I1,2)=(FGH(I1)-ARC(J))/C2
PARI(I1,3)=C3*ZEZ(I1)
PARI(I2,1)=ZEZ(I2)
PARI(I2,2)=(ABC(J)-FGH(I2))/C2
PARI(I2,3)=C3*ZEZ(I2)
GO TO 90

FIND THE SIDE OF THE TRIANGLE CN WHICH THE QUADRATIC VARIES MOST

C
50 C1=C.
DO 80 I=1,3
C2=ABS(ABC(IA(I))-ABC(IB(I)))
C3=ABC(IA(I))+ABC(IB(I))-2.*FGH(I)
C4=1.25*C3*FGH(I)
C5=IF (ABS(C3),GT,0.5*C2) GO TO 60
C6=C2=CASE(C2-ABS(C4))
GO TO 70
C7=CASE(C2)
60 CASE(C2+C3)
C3=CASE(C3)
C2=CASE(C2-C3)-ABS(C2+C3)
70 IF (C2.LE.C1) GO TO 80
80 CONTINUE
IF (C1.LE.0.) GO TO 520

CALCULATE THE ALTERNATIVE PARAMETRIC FORM

C1=SIGN(SQR(ABS(RST(J))).*FGH(J))
I1=IA(J)
I2=TR(J)
PARI(J,1)=0.
PARI(J,2)=C1
PARI(J,3)=C0
PARI(I1,1)=-ABC(I2)/(C1+FGH(J))
PAR(12,1) = -UVW(I2)/C1
PAR(12,2) = -UVW(I1)/C1
PAR(12,3) = ABC(I1)/RST(J)

IF (DELTA.EQ.0.) GC TO 240
PAR(11,3) = DELTA*PAR(11,3)
PAR(12,3) = DELTA*PAR(12,3)

C CALCULATE WHERE THE CONIC INTERSECTS THE TRIANGLE SIDES

90 C1=0.
II=0
DO 140 I=1,3
J=4
100 J=J-1
IF (PAR(I,J).EQ.0.) GC TO 100
GO TO (140,110,120)*J
110 W(II+1) = -0.5*PAR(I,1)/PAR(I,2)
GO TO 130
120 C2=PAR(I,2)**2-PAR(I,1)*PAR(I,3)
GO TO 140
130 II=II+J-1
140 C1=C1+SIGN(FLOAT(I),PAR(I,J))

C CALCULATE THE INTERVALS IN T FOR THE CONTOUR SECTIONS

ISIDE=IFIX(0.1+0.5*(6.-ABS(C1)))
I2=ISIDE
NW=0
IF (ISIDE.GT.0) GO TO 160
NW=NW+1
150 NW=NW+1
TENC(NW,1)=1.5-FLOAT(NW)
TEND(NW,2)=0.
IF (I1+NW.LE.1) GC TO 150
IF (I1.LE.0) GC TO 210
160 C1=W(II)
DO 170 I=1,II
  IF (CI.GT.W(I)) GO TO 170
  CI=W(I)
  J=I
C104
  170 CONTINUE
C105
  IF (ISIDE.EQ.0) GO TO 190
  IF (ISIDE.EQ.IW(J)) GO TO 180
  ISIDE=6-ISIDE-IW(J)
  GO TO 200
C106
  180 IF (II+I2.GT.1) GO TO 190
C107
  TEND(1,1)=10.*CI
  TEND(1,2)=10.*
C108
  GO TO 210
C109
  190 NW=NW+1
C110
  ISIDE=IW(J)-ISIDE
C111
  TEND(NW,1)=CI
  TEND(NW,2)=1.
C112
  IF (NW.LE.1) GO TO 200
C113
  IF (ABS(CI-TEND(NW-1,1))+ABS(1.-TEND(NW-1,2))).LE.0.) NW=NW-2
C114
  200 W(J)=W(I1)
C115
  IW(J)=IW(I1)
C116
  I1=I1-1
C117
  IF (I1.GT.0) GO TO 160
C118
  IF (ISIDE.EQ.0) GO TO 150
C119
C120
  CALCULATE THE X-Y COORDINATES OF THE END POINTS
C121
C122
  210 IF (NW.LE.0) GO TO 520
C123
  NEND=0
C124
  220 NEND=NEND+1
C125
  TAUW=TEND(NEND,1)
C126
  TDEW=TEND(NEND,2)
C127
  GO TO 450
C128
  230 XEND(NEND)=X
C129
  YEND(NEND)=Y
C130
  IW(NEND)=NEND
C131
  IF (DMLT.EQ.0.) GO TO 330
C132
  ZEND(NEND)=SUM
C133
  IF (NEND.LT.NW) GO TO 220
C134
C135
  CALCULATE THE REQUIRED ACCURACY OF THE CONTOUR LINE
C
24C DO 25C I=1,3
0137 W(I)=XD(I)-XD(IA(I))
0138 250 W(I+3)=YD(I)-YD(IA(I))
0139 ACC=A*X1(DELTA,SQRT(AMAX1(W(1)**2+W(4)**2,W(2)**2+W(5)**2,W(3)**2+W(6)**2)))/DD)
0140 IF (DELTA.EQ.0.) GC TO 300
0141 DELTA=2.*DELTA*(XD(I)**2)+XD(2)*YC(3)+XD(3)*YD(1)-
0142 1X(D(1)**2+YD(3)**2)*YD(2)
0143 IF (DELTA.EQ.0.) GC TO 350
C
C CALCULATE THE CONSTANTS THAT DETERMINE THE CHANGES IN T
C
0143 DO 270 I=1,3
0144 A(I)=PAR(1,I)+PAR(2,I)+PAR(3,I)
0145 I1=IA(I)
0146 I2=I2(I)
0147 DO 260 J=1,3
0148 260 WW(J)=PAR(J,1)*PAR(I,J)**2-PAR(J,1)*PAR(I,J)**2
0149 270 H(I)=WW(1)**2+WW(2)**2+WW(3)**2
0150 AL=H(2)**2+H(1)**2+H(3)**2
0151 AM=2.*G(1)**2+G(3)**2+G(1)**2+G(3)**2
0152 AN=G(2)**2+G(1)**2+G(3)**2
0153 C1=SQRT((AL-AN)**2+AM**2+AM**2)
0154 THNUM=AN-AL+C1
0155 THDEN=AM+AN
0156 C2=AMAX1(ABS(THNUM),ABS(THDEN))
0157 IF (C2.LE.0.) GO TO 280
0158 THNUM=THALM/C2
0159 THDEN=THDEN/C2
0160 SIGMA=2./(AL+AN+C1)
0161 280 DO 250 I=1,3
0162 250 W(I)=THNUM*SIGMA(I)-THDEN*H(I)
0163 290 TAUNUM=W(2)+SIGN(SIGMA(ABS(W(2)**2+W(1)**2+W(3)**2,),W(2))
0164 TAUNUM=W(2)**2+W(1)**2+W(3)**2
0165 GO TO 350
C
C FIND END POINTS OF STRAIGHT LINES IN THE DEGENERATE CASE
C
0167 GO TO 300 NEND=C
DO 340 I1=1,2
DO 310 J=1,3
C170 310 \( w(j) = p(a(j), i)*p(b(j), i+1) - p(a(j), i+1)*p(b(j), i) \)
C171 DD 330 J=1,3
C172 IF \((w(j) .LE. 0.)\) GO TO 320
C173 NEND=NEND+1
C174 XEND(NEFD)=XD(J)
C175 YEND(NEFD)=YD(J)
C176 IW(NEND)=NEND
C177 320 IF \((w(ia(j)) .GE. 0.)\) GC TO 330
C178 NEND=NEND+1
C179 XYZ(J)=C.
C180 XYZ(IA(J))=W(IR(J))
C181 XYZ(IR(J))=-W(IA(J))
C182 GO TO 470
C183 330 CONTINUE
C184 340 NFND=NEFD-MOD(NEND,2)
C185 IF \((NEFD.LE.0)\) GO TO 520
C
C FIND THE FIRST POINT OF THE NEXT SECTION OF THE CONIC
C
C350 NA=1
C351 IF \((NF.LE.1)\) GO TO 390
C352 I=1
C353 360 C2=(XARC(KP)-XEND(IW(I)))*2+(YARC(KP)-YEND(IW(I)))*2
C354 IF \((I.LE.1)\) GO TO 370
C355 IF \((C1.LE.C2)\) GO TO 380
C356 370 C1=C2
C357 NA=1
C358 I=I+1
C359 380 IF \((I.LE.NEND)\) GO TO 360
C360 NR=IW(NA)
C
C OUTPUT THE PREVIOUS PART OF THE CONIC IF NECESSARY
C
C397 IF \((NF.LE.1)\) GO TO 400
C398 NC=0
C399 IF \((ISW.LE.1)\) GO TO 500
C400 ISW=C
C401 IF \((C1.GE.0.01*ACC*ACC)\) GO TO 500
C402 GO TO 410
ACM NP=1
"ARC(NP)=XEND(NR)
YARC(NP)=YEND(NR)
410 ND=2*MOD(NR,2)-1
NC=NR+ND

C
CALCULATE THE VALUE OF WMU

C
IF (DELT.A.EQ.0.) GO TO 480
DO 420 I=1,2
J=AC+ND-I*ND
TNUM=TEND(J+1)
TDEN=TEND(J+2)
C1=TNUM*TNUM*G(1)-TNUM*TDEA*G(2)+TDEA*TDEN*G(3)
C2=TNUM*TNUM*H(1)-TNUM*TDEN*H(2)+TDEN*TDEN*H(3)
W(I)=(C1+C1+C2*C2)/(DELT*ZENC(J)**2
W(I+2)=THNUM*C1+THDEN*C2
W(I+4)=TNUM*TDEN-TAUSEN*TNUM

DELTT=SIGMA
TNFIN=TENC(NC,1)
TDFIN=TENC(NC,2)
C1=5.0-DELT-TDFIN
IF (W(3)*W(4), LE, 0.) GO TO 430
IF (C1*W(5), GT, 0.) DELTT=AMIN1(W(1), W(2))
420 C2=SQR(TDELTT)
C3=0.5-ACC*C2*SUMZE
226 IF (C3, LE, 0.) GO TO 480
WMU=SIGN(SQR(ACC*C2*(0.5+C3))/C3,FLOAT(-ND)*ZEND(J))

C
OBTAIN THE REQUIRED PIECEWISE LINPAR APPROXIMATION

C
C1=C1*FLOAT(ND)
440 C2=TNUM
TNUM=NUM+WMU*(TDEA*ARG(1)+TNUM*ARG(2))
TDEN=TDEN-WMU*(TDEA*ARG(2)+C2*ARG(3))
IF (C1*(TNUM*TDFIN-TDEA*TNFIN),LE,0.) GO TO 480

C
FIND THE COORDINATES OF THE NEW POINT

DO 460 I=1,3
450 XYZ(I)=TDEN*TDEN*PAR(I,1)+2.*TDEN*TNUM*PAR(I,2)+TNUM*TNUM*PAR(I,3)
C235  470  SUM=XYZ(1)+XYZ(2)+XYZ(3)
C236     X=(XD(1)*XYZ(1)+XD(2)*XYZ(2)+XC(3)*XYZ(3))/SUM
C237     Y=(YD(1)*XYZ(1)+YD(2)*XYZ(2)+YC(3)*XYZ(3))/SUM
C238     IF (NA.EQ.0) GO TO 230
C239
C    PROVIDE THE FINAL POINT OF THE PRESENT CONTOUR SECTION
C
C240  480  NB=0
C241     X=XFND(NC)
C242     Y=YFND(NC)
C
C    CALL SUBROUTINE OB13 Z WHEN NECESSARY
C
C243  490  IF (NP.LT.IARC) GO TO 510
C244   500  CALL OB132 (XARC,YARC,NP)
C245
C246     IF (NA.EQ.0) GO TO 530
C247     IF (NC.EQ.0) GO TO 400
C248     XARC(1)=XARC(NP)
C249     YARC(1)=YARC(NP)
C250     NP=1
C251  510  NP=NP+1
C252     XARC(NP)=X
C253     YARC(NP)=Y
C254     IF (NB.GT.0) GO TO 440
C
C    BRANCH BACK IF MORE SECTIONS REMAIN TO BE PLOTTED
C
C255  520  IF (ND.LT.0) NA=NA-1
C256     IW(NA)=IW(NFND-1)
C257     IW(NA+1)=IW(NFND)
C258     NEND=NFND-2
C259     IF (NEND.GT.0) GO TO 350
C
C    FOLLOW THE INSTRUCTIONS GIVEN BY THE VALUE OF NPT
C
C260  530  NA=NP
C261     IF (NPT.GT.0) GO TO 530
C262     NA=0
C263     IF (NP.GE.2) GO TO 500
C264
C265  530  NPT=NA
0264 RETURN
END

0001 BLOCK DATA
0002 COMMON /0B13B/ CD,IARC
0003 DATA CD/1000.//IARC/100/
0004 END

0001 SUBROUTINE 0B13Z (XARC,YARC,NPT)
0002 DIMENSION XARC(NPT),YARC(NPT)
0003 PRINT 10,(XARC(I),YARC(I),I=1,NPT)
0004 10 FORMAT (2E20.8)
0005 RETURN
0006 END

0001 DIMENSION FD(6),XD(3),YD(3),XARC(100),YARC(100)
0002 XD(1)=2.
0003 YD(1)=0.
0004 XD(2)=2.
0005 YD(2)=1.
0006 XD(3)=0.
0007 YD(3)=2.
0008 FD(1)=1.
0009 FD(2)=4.
0010 FD(3)=9.
0011 FD(4)=0.25
0012 FD(5)=1.
0013 FD(6)=0.25
0014 ETA=-0.25
0015 DELT=0.01
0016 1 CALL CR13A (FD,XD,YD,ETA,DELT,0,XARC,YARC)
0017 ETA=-ETA
0018 IF (ETA.GT.0.) GO TO 1
0019 STOP
0020 END
| 0.1916714E+01 | 0.5138285E+00 | C.1999999CE+01 | 0.5CCCC012E+00 |
| 0.1838519E+01 | 0.6748209E+00 | 0.18212585E+01 | 0.8176285E+00 |
| 0.17248411E+01 | 0.2552017E+00 | 0.1564192E+01 | 0.1136051E+01 |
| 0.1583620E+01 | 0.10325065E+01 | 0.1389556E+01 | 0.130002E+01 |
| 0.1432510E+01 | 0.1109202E+01 | 0.1000000E+00 | 0.94762874E+00 |
| 0.12891264E+01 | 0.12583371E+01 | 0.1266556E+01 | 0.12681336E+01 |
| 0.11717939E+01 | 0.13458421E+01 | 0.11436214E01 | 0.11663094E+01 |
| 0.1094604E+01 | 0.13401718E+01 | 0.98078066E+00 | 0.9728382E+00 |
| 0.1066594E+01 | 0.1266556E+01 | 0.79950512E+00 | 0.76621372E+00 |
| 0.1023548E+01 | 0.11436214E+01 | 0.6231184CF+00 | 0.7455964E+00 |
| 0.1167511E+01 | 0.98078066E+00 | 0.4724238E+00 | 0.1750000E+01 |
| 0.12834463E+01 | 0.79950512E+00 | 0.36515778E+00 | 0.2500000E+00 |
| 0.142588E+01 | 0.6231184CF+00 | 0.31553584E+00 | 0.18936753E+01 |
| 0.15772457E+01 | 0.4724238E+00 | 0.22524622E+00 | 0.1745652E+00 |
| 0.17192335E+01 | 0.36515778E+00 | 0.40461195E+00 | 0.15983554E+01 |
| 0.18344607E+01 | 0.2500000E+00 | 0.5138285E+00 | 0.2000000E+01 |
| 0.19326853E+01 | 0.15983554E+01 | 0.16666681E+00 | 0.16666681E+00 |