Quick computation of the distance between a point and an ellipse

L. MAISONOBE*

September 2003, revised May 2005, minor revision February 2006

Contents

*luc@spaceroots.org

1 Introduction

We consider the following 2D problem: given a test point A on a plane and an ellipse $\mathcal{E}$, find the point of the ellipse which is the closest to the test point.

This problem occurs in several contexts. The first one is to geopositioning. Planet bodies are often roughly modeled as biaxial ellipsoids, i.e. ellipsoids having a rotational symetry axis (the polar axis). For such shapes, the meridian planes are ellipses. Given a point position in 3D cartesian coordinates, we want to compute the longitude, geodesic latitude and altitude. The last two data are obtained by solving this kind of problem. Another context was encountered while computing approximations of graphical representations of 3D circles on a 2D display. In order to build an error model of a Bézier-based approximation of the ellipse, the distance of thousands of approximated curves had to be computed. This later problem is described in another technical note which is available on the same place as this one.

The latest version of this document is always available in the SpaceRoots site downloads page at `http://www.spaceroots.org/downloads.html`. It can be browsed on-line or retrieved as a PDF, compressed PostScript or LaTeX source file.

2 Problem description

We will use the traditional notations used in geodesy to describe the problem. We assume the case of longitude λ has already been solved using $\lambda = \text{atan2}(y,x)$ and that the remaining problem has been restricted to the 2D meridian plane using the coordinates $r = \sqrt{x^2+y^2}$ and z.

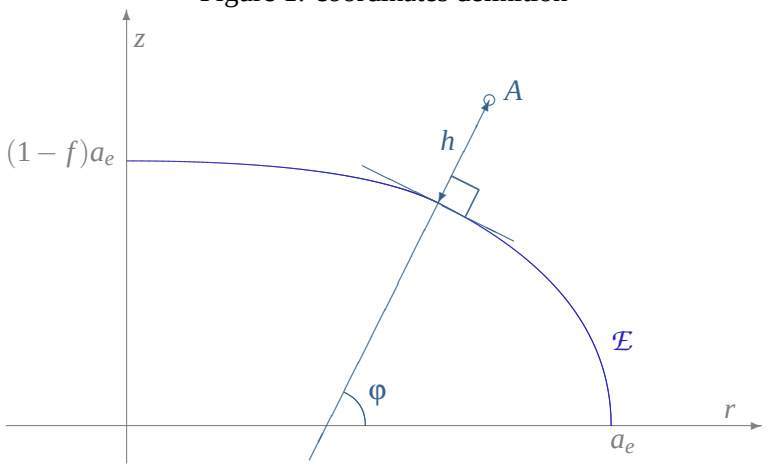
We consider a test point A and an ellipse $\mathcal{E}$. After a suitable coordinates change, we can consider the coordinates of the test point in the canonical reference frame of the ellipse. This frame is centered on the ellipse center, has its absissae axis along the major axis and has its ordinates axis along the minor axis. In this reference frame, the coordinates of A are (r,z). Using suitable axes orientations, we can arrange to have $r >= 0$. This assumption *is* used in the following equations.

The ellipse is defined by its semi-major axis is a_e, and either its flattening f ($0 \leq f < 1$) or its semi-minor axis $a_p = a_e(1-f)$.

The signed distance between the point and the ellipse h and the inclination φ of the projection of the point on the ellipse are related to the cartesian coordinates:

$$\left\{ \begin{array}{l} r = \left(\dfrac{a_e}{\sqrt{1-f(2-f)\sin^2\varphi}} + h \right) \cos\varphi \\ z = \left(\dfrac{(1-f)^2 a_e}{\sqrt{1-f(2-f)\sin^2\varphi}} + h \right) \sin\varphi \end{array} \right. \tag{1}$$

Figure 1: coordinates definition



In the geopositioning domain, a_e is the equatorial radius of the body, a_p is the polar radius, h is the altitude above the ellipsoid (negative when the point is below the surface of the ellipsoid) and φ is the geodesic latitude.

The equation (1) is easy to apply when h and φ are known and r and z are desired, but it is impossible to reverse in the general case. It can be reversed in the specific case where h is known to be null:

$$h = 0 \Rightarrow \varphi = \operatorname{atan2}\left(z, r(1-f)^2\right) \tag{2}$$

3 Preliminary results

3.1 Special case handling: center or the ellipse

A special case we will disgard in the algorithm is the ellipse center. This point is easily detected in a preliminary check using a test like $\sqrt{r^2+z^2} < \varepsilon_0$. The ellipse points closest to the center are both endpoints of the minor axis, we arbitrarily select the point having $\varphi > 0$: $\varphi = +\frac{\pi}{2}$, $h = -(1-f)a_e$.

3.2 Inside/outside check

Given a test point A, it is very simple to check if it lies inside the ellipse or outside of it. This check is done by computing the sign of expression:

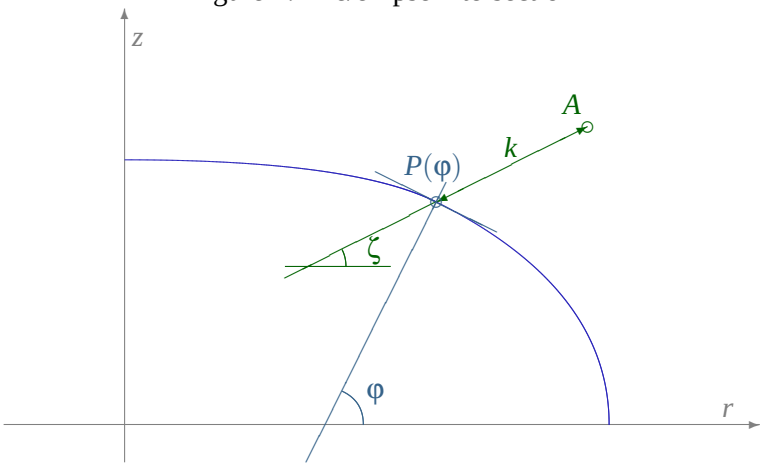
$$\left(\frac{r}{a_e}\right)^2 + \left(\frac{z}{(1-f)a_e}\right)^2 - 1 \tag{3}$$

the point is outside if the sign is positive, inside otherwise.

3.3 Intersection between a line and an ellipse

Lets consider a line having slope ζ and containing the test point A.

Figure 2: line/ellipse intersection



For well chosen ζ values, this line intersects the ellipse. Let (r',z') be the cartesian coordinates of the intersection point $P(\varphi)$:

$$\begin{cases} r' = r - k\cos\zeta \\ z' = z - k\sin\zeta \end{cases} \tag{4}$$

where k is the signed distance between the test point A and the intersection point $P(\varphi)$. k is positive if the test point is outside of the ellipse and negative otherwise. Note that the sign of k implies the choice of the line orientation, so depending on the test point location inside or outside of the ellipse, we have to choose either ζ or $\pi-\zeta$; the following equations take care of this choice. Since the intersection point $P(\varphi)$ belongs to the ellipse, its coordinates verify equation:

$$\left(\frac{r'}{a_e}\right)^2 + \left(\frac{z'}{(1-f)a_e}\right)^2 = 1$$

introducing the test point coordinates and the signed distance k

$$\begin{aligned} & \left(\frac{r-k\cos\zeta}{a_e}\right)^2 + \left(\frac{z-k\sin\zeta}{(1-f)a_e}\right)^2 = 1 \\ \Leftrightarrow\quad & (1-f)^2[(r-k\cos\zeta)^2 - a_e^2] + (z-k\sin\zeta)^2 = 0 \\ \Leftrightarrow\quad & ak^2 - 2bk + c = 0 \quad \text{where} \quad \begin{cases} a = (1-f)^2\cos^2\zeta + \sin^2\zeta \\ b = (1-f)^2 r\cos\zeta + z\sin\zeta \\ c = (1-f)^2(r^2 - a_e^2) + z^2 \end{cases} \end{aligned} \tag{5}$$

4 Iterative method

We intend to compute h and φ from r and z using a very quick iterative method based on a suite of lines containing the test point A and intersecting the ellipse on varying points $P(\varphi_n)$. The lines will converge to the projection line.

4.1 Initialization

When the A point is not at the center of the ellipse, we initialize the algorithm using a first line containing both the test point A and the center of the ellipse. The cosine and sine of this line slope ζ_0 are defined by equation (6). Note that we do not compute ζ_0 itself, to avoid numerical problems when $r \ll z$ (i.e. when we are close to the minor axis), we also compute $t_0 = \tan\zeta_0/2 = \sin\zeta_0/(1-\cos\zeta_0)$ directly from $\cos\zeta_0$ and $\sin\zeta_0$ (using a stable formula when $\cos\zeta_0 >= 0$) as we will need it soon.

$$\begin{cases} \cos\zeta_0 = \dfrac{r}{\sqrt{r^2+z^2}} \\ \sin\zeta_0 = \dfrac{z}{\sqrt{r^2+z^2}} \\ t_0 = \dfrac{z}{r+\sqrt{r^2+z^2}} \end{cases} \tag{6}$$

Regardless of its slope, this line is known to have exactly two intersections with the ellipse. We choose the closest one to the A point, which corresponds to the smallest root of binomial equation (5) in absolute value.

Since $1-f > 0$, $\cos\zeta_0 > 0$ and $z\sin\zeta_0 \geq 0$, we can deduce $a > 0$ and $b > 0$. This implies that when the binomial has real roots, their sum is strictly positive, so the smallest root in absolute value is also the smallest root in algebric value.

The classical expression for the smallest root is:

$$k_0 = \frac{b - \sqrt{b^2 - ac}}{a}$$

However, this expression is not numerically accurate when $b^2 \gg ac$ because in involves computing something similar to $b-(b-\varepsilon)$, leading to a numerical cancellation when ε is too small. This case occurs when the test point A is very close to the ellipse $\mathcal{E}$. The limit case is for a test points lying exactly on the ellipse, for which $k = 0$ is a root so $c = 0$.

The cancellation can be avoided even at the limit case by using the dual expression of the root:

$$k_0 = \frac{c}{b + \sqrt{b^2 - ac}} \tag{7}$$

Since we know $b > 0$, we know this expression is more stable than the other one in all cases.

The first point $P(\varphi_0)$ is computed from ζ_0 and k_0 using equation (4) and the corresponding value φ_0 is computed from equation (2).

If the test point is very close to the ellipse, we have $|k_0| < \varepsilon\sqrt{r^2+z^2}$, for a given threshold ratio ε. In fact, in this case any slope would have given the same point since $P(\varphi_0) = A$. There is no need to perform any iteration and we can directly return the values $\varphi = \varphi_0$ and $h = k_0$.

4.2 Iterations

At the start of each iteration, we have an ellipse point $P(\varphi_n)$ and the corresponding inclination φ_n, the signed distance $k_n \neq 0$ between $P(\varphi_n)$ and the test point A and the value of $t_n = \tan\zeta_n/2$ where ζ_n is the slope of the line joining $P(\varphi_n)$ to the test point.

If the estimate $P(\varphi_n)$ were perfect, we would have $\varphi_n = \varphi$ and $k_n = h$. If $h \neq 0$ the circle centered on the test point A with radius $|k_n|$ would be tangent to the ellipse. In our iterative process, the estimate is not perfect and the circle intersects the ellipse at least at one other point $P(\tilde{\varphi}_n)$, $\tilde{\varphi}_n \neq \varphi_n$. Figures 3 and 4 show these circle/ellipse intersections in two different situations. As shown by these figures, there can be up to three intersection points in addition to the already known point $P(\varphi_n)$.

Figure 3: interior point

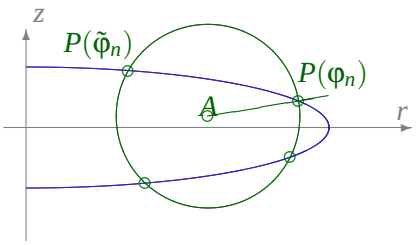
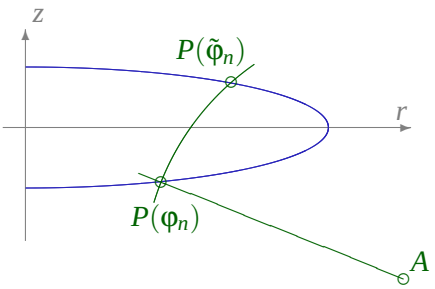


Figure 4: exterior point



The additional intersection points can be computed by introducing $\tau = \tan\zeta/2$

in equation (5) an multiplying by $(1+\tau^2)^2$. We get:

$$\begin{aligned}
& k^2[(1+\tau^2)^2 - f(2-f)(1-\tau^2)^2] \\
& -2k[(1-f)^2 r(1+\tau^2)(1-\tau^2) + 2z\tau(1+\tau^2)] \\
& +[(1-f)^2(r^2-a_e^2)+z^2](1+\tau^2)^2 = 0 \\
\Leftrightarrow \quad & \alpha\tau^4 + \beta\tau^3 + \gamma\tau^2 + \delta\tau + \varepsilon = 0
\end{aligned}$$

$$\text{where} \quad \begin{cases} \alpha = (1-f)^2((r+k)^2 - a_e^2) + z^2 \\ \beta = -4kz \\ \gamma = 2[(1-f)^2(r^2-k^2-a_e^2) + 2k^2 + z^2] \\ \delta = -4kz \\ \varepsilon = (1-f)^2((r-k)^2 - a_e^2) + z^2 \end{cases}$$

This is a quartic equation for which we already know one root: $t_n = \tan\zeta_n/2$. We can therefore reduce this equation to a normalized cubic polynomial by a simple coefficients identification (the value of the constant term ε of the quartic is of course not needed for this degree reduction):

$$\tau^3 + a_1\tau^2 + a_2\tau + a_3 = 0 \quad \text{where} \quad \begin{cases} a_1 = \dfrac{\beta}{\alpha} + t_n \\ a_2 = \dfrac{\gamma}{\alpha} + a_1 t_n \\ a_3 = \dfrac{\delta}{\alpha} + a_2 t_n \end{cases} \tag{8}$$

The other circle/ellipse intersections are given by the real roots of equation (8). In order to compute them, we compute first the discriminant D:

$$Q = \frac{3a_2 - a_1^2}{9} \qquad R = \frac{9a_1a_2 - 27a_3 - 2a_1^3}{54} \qquad D = Q^3 + R^2$$

If D is positive, the following expressions compute the two real numbers S et T and allow to deduce the unique real root $\tilde{t}_{n,1}$. We do not need the two complex roots $\tilde{t}_{n,2}$ and $\tilde{t}_{n,3}$. This corresponds to the case depicted in figure 4. It also occurs for interior points when we are close to the solution.

$$S = \sqrt[3]{R + \sqrt{Q^3+R^2}} \qquad T = \sqrt[3]{R - \sqrt{Q^3+R^2}}$$

$$\begin{cases} \tilde{t}_{n,1} = S + T - \dfrac{a_1}{3} \\ \tilde{t}_{n,2} = -\dfrac{S+T}{2} - \dfrac{a_1}{3} + i\dfrac{\sqrt{3}(S-T)}{2} \\ \tilde{t}_{n,3} = -\dfrac{S+T}{2} - \dfrac{a_1}{3} - i\dfrac{\sqrt{3}(S-T)}{2} \end{cases} \tag{9}$$

If D is negative, the three roots are all real ones. This corresponds to the case depicted in figure 3. In this case, the trigonometric expressions of the root are simpler to use:

$$(10)\qquad \begin{cases} \tilde{t}_{n,1} = 2\sqrt{-Q}\cos\dfrac{\theta}{3} - \dfrac{a_1}{3} \\ \tilde{t}_{n,2} = 2\sqrt{-Q}\cos\dfrac{\theta+2\pi}{3} - \dfrac{a_1}{3} \\ \tilde{t}_{n,3} = 2\sqrt{-Q}\cos\dfrac{\theta+4\pi}{3} - \dfrac{a_1}{3} \end{cases} \quad \text{with} \quad \cos\theta = \frac{R}{\sqrt{-Q^3}}$$

For each $\tilde{t}_{n,i}$ representing a circle/ellipse intersection, we compute $\tilde{\zeta}_{n,i}$ and $P(\tilde{\varphi}_{n,i})$ using equation (4). Since this point belongs to the ellipse we deduce $\tilde{\varphi}_{n,i}$ using equation (2).

When there are three roots, only one of them[1] leads to a point on the same half plane ($z < 0$ or $z > 0$) as the test point A and the already known intersection point $P(\varphi_n)$. We select this root and disgard the two other ones. This selection is done by comparing the signs of φ_n and $\tilde{\varphi}_{n,i}$.

4.3 Selecting a new point for next iteration

We now have two points $P(\varphi_n)$ and $P(\tilde{\varphi}_n)$ known to be both at signed distance k_n from the test point A. The ellipse points between $P(\varphi_n)$ and $P(\tilde{\varphi}_n)$ are all closer to A than $P(\varphi_n)$ and $P(\tilde{\varphi}_n)$ (they are inside the circle). So selecting one point in this elliptical arc as the next point ensures $|k_{n+1}| < |k_n|$.

As can be noted from figure 4, when we are far from the solution, the elliptical arc is not restricted to the angular sector bounded by the two lines $(T,P(\varphi_n))$ and $(T,P(\tilde{\varphi}_n))$, so we cannot use a simple interpolation between ζ_n and $\tilde{\zeta}_n$. We can however interpolate between φ_n and $\tilde{\varphi}_n$.

Once we are close to the solution, the circle is almost tangent to the ellipse, so the elliptical arc between the two intersection point is small and has an almost constant radius of curvature: it is very close to a circle. In this case, $P(\tilde{\varphi}_n)$ is close to the symetrical of $P(\varphi_n)$ with respect to the true projection. We rely on this asymptotical behavior to set up the selection scheme for the next iteration inclination:

$$(11)\qquad \varphi_{n+1} = \frac{\varphi_n + \tilde{\varphi}_n}{2}$$

In order to start the new iteration, we also need to compute k_{n+1} and t_{n+1} from φ_{n+1}. The simplest way to do this is to compute first the vector joining the new

[1] a proper demonstration would be welcome here ...

intersection point $P(\varphi_{n+1})$ to the test point:

$$\begin{cases} \Delta r = r - \dfrac{a_e \cos\varphi_{n+1}}{\sqrt{1-f(2-f)\sin^2\varphi_{n+1}}} \\ \Delta z = z - \dfrac{(1-f)^2 a_e \sin\varphi_{n+1}}{\sqrt{1-f(2-f)\sin^2\varphi_{n+1}}} \end{cases} \tag{12}$$

and to deduce the signed distance and the tangent of the half slope:

$$\begin{cases} k_{n+1} = \pm\sqrt{\Delta r^2 + \Delta z^2} \\ t_{n+1} = \dfrac{\Delta z}{\Delta r + k_{n+1}} \end{cases} \tag{13}$$

The sign of k_{n+1} is chosen to be negative when the test point is inside the ellipse and positive otherwise, which is checked using equation (3). Here again, we do not compute ζ_{n+1} but directly t_{n+1} from the vector coordinates, again in order to avoid numerical problems for points close to the minor axis ($\zeta \approx \pm\pi/2$) where the arc tangent function accuracy is very bad in typical computers.

4.4 Convergence

Since at the end of each iteration we have an interval known to contain the solution, and since we choose the middle of this interval (φ-wise), we can use its half-size as a convergence criterion. We stop the iterations when:

$$\frac{|\varphi_n - \tilde{\varphi}_n|}{2} < \varepsilon_c \tag{14}$$

where ε_c is the convergence threshold.

When this threshold is met, we use $\varphi = \varphi_{n+1}$ and we compute h using the following expression which is true when φ has been found:

$$h = r\cos\varphi + z\sin\varphi - a_e\sqrt{1-f(2-f)\sin^2\varphi} \tag{15}$$

5 Order of the method

The numerical example in appendix shows that the error decreases very quickly. Some experiments have been performed with a large number of significant digits (100 digits). These experiments seem to show the method would be of order 2. This order is not very impressive. However, the main strength of the method is that its starts very quickly, even for very flat ellipses. The example shows that in a quite difficult case we already have 0.013° after only one iteration.

This order should be analyzed more precisely.

6 Robustness

The method also seems quite robust. The cancellation problem when computing k_0 is avoided by using the proper expression. Another cancellation proble occurs while computing $R - \sqrt{Q^3 + R^2}$ when $R \gg Q$. However, this has no consequence as it affects only the T term which in this case is much smaller than S and only $S+T$ is used to compute the solution. As the algorithm converges, all the coefficients also converge (k, α, β, γ, δ, a_1, a_2, a_3, Q, R, S, T). There is no numerical explosion anywhere.

A Approximate formulas

The technical manual for *Geocentric Datum of Australia* provides direct transformation formulas[2] that we reproduce here, after having renamed some parameters to be consistent with our notation and to correct an ambiguity[3], and after having replaced an equality sign by an approximation sign:

$$\tan\varphi \approx \frac{z(1-f)+f(2-f)a_e sin^3 u}{(1-f)(r-f(2-f)a_e\cos^3 u)} \tag{16}$$

where

$$\tan u = \frac{z}{r}\left[(1-f)+f(2-f)\frac{a_e}{\sqrt{r^2+z^2}}\right] \tag{17}$$

Some numerical checks show that these expressions are very good approximation but that they can be applied only for almost spherical bodies. For the Earth, the maximal error are about 10^{-11} degrees in latitude and 2×10^{-9} metres in altitude!

However, when we consider very flat ellipses (for example $f = 0.9$), the errors become tremendous (more than 100 degrees on φ).

[2] `http://www.anzlic.org.au/icsm/gdatm/xyzcd.htm`, but this URL seems not valid anymore

[3] f was used in the document for both the flatness and the latitude

B Numerical example

The following example show the first iterations of the algorithm for a very flat ellipse:

$$\begin{cases} a_e = 100 \\ f = 0.9 \end{cases}$$

We consider a test point A having the following coordinates:

$$T \begin{cases} \varphi = 75^\circ \\ h = 0.1 \end{cases} \Rightarrow \begin{cases} r = 93.7139699 \\ z = \ \ 3.5930796 \end{cases}$$

We assume we know only r and z and want to compute the unknown φ and h.

Equations (6) give us the tangent of the half slope:

$$t_0 = 0.0191634$$

Applying equation (7), we get the signed distance between A and the intersection of the first line and the ellipse:

$$k_0 = 0.3419763 \Rightarrow \varphi_0 = 75.3818944^\circ$$

¿From this value of k, we deduce the first coefficients α, β, γ and δ of the quartic (we don't compute ε). We reduce it to a cubic using the already known root t_0:

$$\tau^3 + a_1\tau^2 + a_2\tau + a_3 = 0 \quad \text{where} \quad \begin{cases} a_1 = -3.5542558 \\ a_2 = \ \ 1.3365805 \\ a_3 = -3.5478057 \end{cases}$$

This polynomial has only one real root:

$$\tilde{t}_0 = 3.4640705 \Rightarrow \tilde{\zeta}_0 = 147.7954987^\circ \Rightarrow \tilde{\varphi}_0 = 74.5916410^\circ$$

The middle point between φ_0 and $\tilde{\varphi}_0$ is the best approximation we have at the end of this first iteration:

$$\varphi_1 = 74.9867677^\circ$$

We know an upper bound for the error of this approximation is the half-width of the interval, which is 0.3951267° (in fact the error is about 30 times smaller). As we consider this is too much, we compute the starting values for a second iteration:

$$k_1 = 0.1005980 \quad t_1 = 0.8577741$$

The steps of the second iteration are similar to the steps of the first one. They lead to the following result:

$$\tilde{\varphi}_1 = 75.0132026^\circ$$

So at the end of the second iteration, we have an estimate $\varphi_2 = 74.9999851^\circ$ and an upper bound of its error of 0.0132174° (the real error being almost 900 times smaller).

C Implementation

An implementation in the Java language of the algorithms explained in this paper is available in source form as a stand-alone file at `http://www.spaceroots.org/documents/distance/Ellipsoid.java`. It is included in a class modelling an ellipsoid.

The language-independant form of this implementation follows. In order to improve readability, some code optimizations have been removed (use of precomputed values like $(1-f)$, $(1-f)^2(r^2-a^2)+z^2$ and others). The control parameters of the algorithm are the maximal number of iterations and the two thresholds ε and ε_c.

if $(\sqrt{r^2+z^2} < \varepsilon a_e)$ **return** $\varphi = \pi/2, \quad h = -(1-f)a_e$

$\text{inside} \leftarrow (1-f)^2(r^2-a^2)+z^2 \leq 0$
$\cos\zeta \leftarrow r/\sqrt{r^2+z^2}$
$\sin\zeta \leftarrow z/\sqrt{r^2+z^2}$
$t \leftarrow z/\left(r+\sqrt{r^2+z^2}\right)$

// distance to the ellipse along the current line as the root of a 2nd degree
// polynom closest to zero: $ak^2-2bk+c=0$
$a \leftarrow (1-f)^2\cos^2\zeta+\sin^2\zeta$
$b \leftarrow (1-f)^2 r\cos\zeta+z\sin\zeta$
$c \leftarrow (1-f)^2(r^2-a_e^2)+z^2$
$k \leftarrow c/(b+\sqrt{b^2-ac})$
$\varphi \leftarrow \text{atan2}(z-k\sin\zeta,(1-f)^2(r-k\cos\zeta))$

if $(|k| < \varepsilon\sqrt{r^2+z^2})$ **return** $\varphi, \quad h = k$

begin loop for at most N iterations

// 4^{th} *degree normalized polynom describing circle/ellipse intersections*
// $\tau^4+b\tau^3+c\tau^2+d\tau+e=0$ *(there is no need to compute e here)*
$a \leftarrow (1-f)^2((r+k)^2-a_e^2)+z^2$
$b \leftarrow -4kz/a$
$c \leftarrow 2\left[(1-f)^2(r^2-k^2-a_e^2)+2k^2+z^2\right]/a$
$d \leftarrow b$

// reduce the polynom to degree 3 by removing the already known real root
// $\tau^3+b\tau^2+c\tau+d=0$
$b \leftarrow b+t$
$c \leftarrow c+t$
$d \leftarrow d+t$

// find the other real root
$Q \leftarrow (3c - b^2)/9$
$R \leftarrow (b(9c - 2b^2) - 27d)/54$
$D \leftarrow Q^3 + R^2$
if $(D >= 0)$ **then**
// beware that some programming languages do not provide a cubic root function,
// and that the power function does not accept negative arguments
// (both $R + \sqrt{D}$ *and* $R - \sqrt{D}$ *can be negative)*
$\tilde{t} \leftarrow \sqrt[3]{R + \sqrt{D}} + \sqrt[3]{R - \sqrt{D}} - b/3$
$\tilde{\varphi} \leftarrow \text{atan2}\left(z(1+\tilde{t}^2) - 2k\tilde{t}, (1-f)^2[r(1+\tilde{t}^2) - k(1-\tilde{t}^2)]\right)$
else
$Q \leftarrow -Q$
$\theta \leftarrow \arccos R/(Q\sqrt{Q})$
$\tilde{t} \leftarrow 2\sqrt{Q}\cos(\theta/3) - b/3$
$\tilde{\varphi} \leftarrow \text{atan2}\left(z(1+\tilde{t}^2) - 2k\tilde{t}, (1-f)^2[r(1+\tilde{t}^2) - k(1-\tilde{t}^2)]\right)$
if $(\tilde{\varphi} \times \varphi < 0)$ **then**
$\tilde{t} \leftarrow 2\sqrt{Q}\cos((\theta + 2\pi)/3) - b/3$
$\tilde{\varphi} \leftarrow \text{atan2}\left(z(1+\tilde{t}^2) - 2k\tilde{t}, (1-f)^2[r(1+\tilde{t}^2) - k(1-\tilde{t}^2)]\right)$
if $(\tilde{\varphi} \times \varphi < 0)$ **then**
$\tilde{t} \leftarrow 2\sqrt{Q}\cos((\theta + 4\pi)/3) - b/3$
$\tilde{\varphi} \leftarrow \text{atan2}\left(z(1+\tilde{t}^2) - 2k\tilde{t}, (1-f)^2[r(1+\tilde{t}^2) - k(1-\tilde{t}^2)]\right)$
end if
end if
end if

// φ*-wise midpoint on the ellipse*
$\Delta\varphi \leftarrow |\tilde{\varphi} - \varphi|/2$
$\varphi \leftarrow (\varphi + \tilde{\varphi})/2$

if $(\Delta\varphi < \varepsilon_c)$ **return** φ, $\quad h = r\cos\varphi + z\sin\varphi - a_e\sqrt{1 - f(2-f)\sin^2\varphi}$

$\Delta_r \leftarrow r - \frac{a_e\cos\varphi}{\sqrt{1-f(2-f)\sin^2\varphi}}$
$\Delta_z \leftarrow z - \frac{a_e(1-f)^2\sin\varphi}{\sqrt{1-f(2-f)\sin^2\varphi}}$
$k \leftarrow \sqrt{\Delta_r^2 + \Delta_z^2}$
if (inside) $k \leftarrow -k$
$t \leftarrow \Delta z/(\Delta r + k)$

end loop

generate a convergence error