Peridetic Transformations - Math Summary

Stellacore Corporation
Dave Knopp

Document ID: t_periMath_v1.0.1, 2021-03-21 10:10:02 -0600

***Abstract*—This technical note summarizes the mathematical formulae used within the Peridetic software module. The presented formulae provide a summary of the implementation within the Peridetic code, so that methodology and formulae can be readily compared with various other known techniques.**

***Index Terms*—Peridetic Software, Geodesy, Coordinate Transformation.**

I. Introduction

This technical note presents the mathematical structure used to perform geodetic transformations within the Peridetic software module [4]. The content presented herein is a summary extract from a more general treatment of ellipsoidal coordinate solutions [3].

The mathematical context of geodetic coordinate conversions is very well known and many algorithms are available with which to compute transformed coordinate values. Toms [5] includes a short algebraic summary of various techniques. Fok and Iz [1] provide detail on various approaches as well as a comparison of computation times between them. The approach presented here is similar in nature to the various "vector methods", but is formulated independent of any particular one of those methods.

Many of these algorithms involve approaches that are theoretically interesting, but in practice lead to unnecessarily complex, and expensive computation. Peridetic utilizes a simple numerical solution technique to provide highly precise and accurate results with fast computation. This approach is supported by results presented by Fok and Iz [1] that indicate, on the whole, numerical solutions are generally several times faster than evaluation of the various elaborate full analytical solutions.

Overall, the primary novelty of the Peridetic software lies almost entirely in its minimalist packaging. The required portion of the software package comprises only two simple header files and therefore extremely lightweight and free from external dependencies.

Although the Peridetic package has an extremely lightweight software footprint, it provides highly precise and accurate results based on a mathematical approach that is fully rigorous. This technical note describes the specific mathematical formulae utilized within the Peridetic software implementation.

Computation of the very simple "Cartesian from Geodetic" transformation is noted in section II-B (ref. expression 3). The considerably more involved "Geodetic from Cartesian" transformation is addressed in section II-C (ref. computational recipe summary in section III).

II. Methodology Summary

Let the (three dimensional) vector, x, indicate a point location of interest. The components, x_k, represent the Cartesian coordinates of x where the vector indices are assumed to span the space (i.e. $k = 1, 2, 3$). The corresponding geodetic component values comprise the scalar longitude angle, λ, and scalar parallel(latitude) angle, ϕ, and the scalar (ellipsoidal) altitude value, η.

For standard geodetic purposes, assume a conventional three-dimensional (3D) coordinate frame with directions indicated by index $k = 1, 2, 3$. This 3D space may be associated with a standard vector basis consisting of three mutually orthogonal unit vectors, e_k, each of which is unit length, $e_k^2 = 1$, and all of which are mutually orthogonal, $e_i \cdot e_j = 0\,; i \neq j$, and, when considered in cyclic order form a dextral (right-handed) system[1].

Assume the origin of the coordinate frame is at the center of an ellipsoid shape selected to represent a specific Earth model. The alignment of the basis is interpreted such that: e_3 points to the North pole, and e_1 points toward the specific prime meridian of interest[2].

In the notation that follows, all summation symbols are understood to apply over the range of all three spatial indices. E.g. interpret summation symbols, $\sum$, as $\sum_{\{i,j,k,l\}=1}^{3}$, where the individual dummy index lettering is varied as convenient.

A. Geodetic Angle/Direction Relationship

The conventional longitude, parallel (latitude) angles[3] are easily extracted from the geodetic "up" direction vector, u, or from any other vector parallel to it.

The geodetic angles are related to the local vertical direction via[4],

$$\lambda = \arctan\left(\frac{u_2}{u_1}\right) \tag{1}$$

$$\phi = \arctan\left(\frac{u_3}{\sqrt{u_1^2 + u_2^2}}\right) \tag{2}$$

Here, the four-quadrant arctangent function evaluation is implied.

The inverse relationships provides an expression for the local vertical "up" (unitary) direction vector, in terms of the

[1] I.e. $e_1 e_2 e_3 = e_2 e_3 e_1 = e_3 e_1 e_2 = \mathcal{I}$, where $\mathcal{I}$ is the unitary trivector (aka pseudo-scalar) for the 3D vector space.

[2] A somewhat more general definition of the coordinate frame alignment is that unitary bivector, $E_{12} \equiv e_1 e_2$, represents the Earth's equatorial plane of rotation, with the e_1 direction pointing toward the prime meridian of reference.

[3] The "LP" part of "LPA" coordinates in Peridetic parlance.

[4] Note that (the concept of) longitude angle value is fundamentally undefined for all locations on the polar axis.

two geodetic scalar angles (and coordinate frame basis vectors) as

$$u = \cos\phi\left(\cos\lambda e_1 + \sin\lambda e_2\right) + \sin\phi e_3$$

B. Cartesian from Geodetic

The transformation direction for computing the Cartesian coordinate expression, corresponds with the Peridetic software transform function, peri::xyzForLpa().

The Cartesian coordinates, x_k, can be computed easily from components of the geodetic up direction vector, u, and the (ellipsoidal) altitude value, η.

Results may be obtained by direct evaluation of a simple formula for each of the three, $k = 1, 2, 3$, components,

$$x_k = u_k\left(\frac{\mu_k^2}{\sqrt{\sum\mu_j^2u_j^2}} + \eta\right) \tag{3}$$

The μ_i are the three ellipsoid semi-axis magnitudes.

For ellipsoids representing the Figure of Earth, the first two semi-axis magnitudes are both equal to the equatorial radius, $\mu_{1,2}$, (often denoted as "a") while the polar radius, μ_3, (often denoted as "b") is different (typically about .3% smaller)[5].

Note that the denominator term is the same for all three components and need be evaluated only once. The squared ellipsoid shape factors, μ_i^2, are constant properties of the ellipsoid and may be pre-computed for any specific ellipsoid shape of interest.

C. Geodetic from Cartesian

The transformation direction for computing the Geodetic parameter values, corresponds with the Peridetic software function, peri::lpaForXyz().

Determining the geodetic position description, λ, ϕ, and η, from Cartesian coordinate values, x_k, is considerably more involved than the simple direct evaluation in the other direction.

The fundamental challenge in obtaining geodetic parameter values is associated with recovering the vertical direction vector, u, and the scalar altitude value, η, given the three Cartesian coordinate components, x_k. Once the vertical direction, u, is obtained, the two geodetic angle representations are easily computed with the two arctangent function evaluations in expressions 1 and 2 above.

The approach described here is similar (likely mathematically homomorphic) to a formulation of Lin and Wang as it is reported and summarized[6] by Fok and Iz [1].

The formulation herein utilizes combinations of parameter terms and factors that provide good numeric stability assuming that the formulae are applied to appropriately normalized data values (e.g. the article by Lakshmanan [2] provides a simple overview of general data normalization considerations). The Peridetic software implementation performs appropriate data value normalization and restoration operations before and after utilizing the various formulae[7].

1) General Approach: Desired Geodetic coordinate values may be computed with the following sequence of steps.

- Determine the pseudo-altitude value, $\breve{\sigma}$, which exactly satisfies this (challenging nonlinear) geodetic solution equation

$$f\left(\breve{\sigma}\right) = \sum\frac{\mu_k^2x_k^2}{\left(\mu_k^2 + \breve{\sigma}\right)^2} - 1 = 0 \tag{4}$$

  Details for computing the solution value, $\breve{\sigma}$, are presented further below. Interpretation and an outline derivation are presented in Appendix A.
- Use the pseudo-altitude solution value, $\breve{\sigma}$, to compute the (non-unitary) local gradient vector, $\breve{g}$, via

$$\breve{g} = 2\sum\frac{x_k}{\mu_k^2 + \breve{\sigma}}e_k$$

  Note that the leading factor of 2 is unimportant if $\breve{g}$ is used only for evaluation of ratios (such as in computing u, λ, and ϕ below). The altitude, η, can be computed without using $\breve{g}$ at all.
- If desired, compute the projection point on ellipsoid, $\breve{p}$, by evaluating its components,

$$\breve{p}_k = \frac{x_k}{\mu_k^2 + \breve{\sigma}}\mu_k^2$$

- Since expressions 1 and 2 involve only ratios of components, the geodetic angles for longitude, λ, and parallel, ϕ, may be computed directly from the components $\breve{g}_k$ without needing to compute the normalized "up" direction, $\breve{u} = \frac{\breve{g}}{\sqrt{\breve{g}^2}}$. In fact, only the three fractional terms, $\frac{x_k}{\mu_k^2+\breve{\sigma}}$, are needed to compute the two angles.
- The ellipsoidal altitude, $\breve{\eta}$, may be computed as

$$\breve{\eta} = \breve{\sigma}\sqrt{\sum\left(\frac{x_k}{\mu_k^2 + \breve{\sigma}}\right)^2}$$

  Note that the individual fractions in the summation are identical to those used in evaluation of, $\breve{g}_k$ and $\breve{p}_k$.

2) Equation Solution: There are many ways (cf. [1], [3]) to solve the nonlinear geodetic metric function 4. The Peridetic code uses a simple, fast, effective and precise, iterative algorithm that determines the solution value, $\breve{\sigma}$, to any desired level of precision.

a) Interpretation: Derivation for the function, $f\left(\sigma\right)$, in expression 4 is outlined in Appendix A.

The parameter σ, can be interpreted as an abstract parameter that has units of area (linear dimension squared) with magnitude that is related to geodetic altitude (cf. Appendix A). Its definition in terms of physical altitude, η, and local gradient vector, g, is

$$\sigma \equiv \frac{\eta}{\frac{1}{2}\left|g\right|}$$

[5]E.g. For rotation symmetric Earth ellipsoids, the denominator factor could be expressed as $\sqrt{\sum\mu_j^2u_j^2} = \sqrt{\mu_{1,2}^2\left(u_1^2 + u_2^2\right) + \mu_3^2u_3^2}$.

[6]The Lin and Wang paper itself is distributed only in a commercial publication unavailable to the free and open Peridetic software effort.

[7]If a consuming application is already using normalized data values, then, in principal, the normalization/restoration overhead could be removed from the Peridetic transformation code. However, that level of optimization is outside the scope of Peridetic's "easy to use" requirement.

The geodetic solution metric function, $f(\sigma)$, is a scalar valued function that represents an abstract "misclosure" metric associated with point, x, relative to an ellipsoid with semi-axis magnitudes, μ_k. When evaluated at $\sigma = 0$, the function value, $f(0)$, is exactly the algebraic misclosure of the ellipsoid constraint in association with point location, x. For point locations, not on the ellipsoid, a nonzero value of σ is required to satisfy the metric function. Overall, the free parameter, σ, is a mathematical surrogate for representing a point's altitude.

For a specific location, x, there are several σ solution values[8] that exactly satisfy the metric function such that $f(\sigma) = 0$. Of those, mathematical solutions, let $\breve{\sigma}$, denote the one that is geodetically relevant.

Note: everything in this present work involves only the one geodetically relevant solution. The other solution cases[9] are avoided automatically when employing the specific formulae and algorithms presented herein.

The following paragraphs describe an iterative process with specific steps and equations that determine the value for $\breve{\sigma}$. The solution process utilizes a classic Newton-Raphson root-finding method.

b) Initial Value Estimation: As described in Appendix B a useful initial value, σ_0, for the free parameter, σ, can be computed with the formula,

$$\sigma_0 = \frac{\left(\sum x_j^2\right) - \sqrt{\mu_{1,2}\mu_3}\sqrt{\sum x_j^2}}{\sqrt{\sum \frac{\mu_{1,2}\mu_3}{\mu_k^4} x_k^2}} \tag{5}$$

Note that the factors involving the μ_k, are constants associated with the ellipsoid shape. The second term in the numerator requires evaluating the square root of the first. The denominator is simply a weighted version of the same summation structure in the numerator terms.

This estimation formula 5 is sufficient to initialize root finding iterations that provide (very) fast iterative convergence while also avoiding the physically irrelevant additional roots associated with the nonlinear metric function 4.

c) Newton-Raphson Iteration: Any (sufficiently reasonable) "current" estimate for the value of the pseudo-altitude parameter, σ, may be improved via an iteration based on the linear approximation to function 4.

For the n-th iterative step, evaluation of equation 4 provides a scalar function value, f_n, in terms of the n-th approximate solution value, σ_n, as

$$f_n = \sum \frac{\mu_k^2 x_k^2}{\left(\mu_k^2 + \sigma_n\right)^2} - 1$$

The corresponding scalar derivative function, $\dot{f}_n \equiv \frac{\partial}{\partial\sigma} f_n$, also evaluated at σ_n, is

$$\dot{f}_n = \sum \frac{-2}{\left(\mu_k^2 + \sigma_n\right)} \frac{\mu_k^2 x_k^2}{\left(\mu_k^2 + \sigma_n\right)^2}$$

The improved root value, σ_{n+1}, is given as a linear adjustment to the current root value, σ_n, via

$$\sigma_{n+1} = \sigma_n - \frac{f_n}{\dot{f}_n}$$

Computations can exploit the common and recurring factors and fraction values involved in these expressions.

d) Evaluate Iterative Update, σ_{n+1}: An initial value for σ_0 can be computed directly from formula 5. Subsequent iteration values may be computed efficiently in terms of temporary values,

$$\xi_{kn} \equiv \frac{\mu_k}{\mu_k^2 + \sigma_n}$$

Evaluate a second trio of temporary values, by squaring these and multiplying with the x_k^2 (already available from computing σ_0)

$$\psi_{kn} = \xi_{kn}^2 x_k^2$$

Use these to evaluate the function and derivatives used to update the estimated solution via,

$$\sigma_{n+1} = \sigma_n + \frac{1}{2} \frac{\sum \psi_{kn} - 1}{\sum \xi_{kn} \psi_{kn}}$$

III. Computation Recipe

This section recapitulates the preceding content in the format of a direct step-by-step computational recipe.

Assume it is desired to obtain geodetic components, λ, ϕ, and η, associated with an arbitrary point location specified by vector, x.

1) Cache values for ellipsoid derived constants (geometric mean radius ρ, adjusted semi-axis shape coefficients, υ_k^{-2}, and squared semi-axis magnitudes, μ_k^2),

$$\begin{aligned} \rho &\equiv \sqrt{\mu_{1,2}\mu_3} \\ \upsilon_k^{-2} &\equiv \frac{\mu_{1,2}\mu_3}{\mu_k^4} = \left(\frac{\mu_{1,2}\mu_3}{\mu_k^2}\right)\frac{1}{\mu_k^2} \\ \mu_k^2 &\equiv \left(\mu_k\right)^2 \end{aligned}$$

   Note that the formulae throughout require using normalized data consistent with an ellipsoid for which, $\rho \equiv 1$. This normalization condition provides an appropriate and well defined scale for tolerance testing, and ensures good numeric precision in expression evaluations.

2) Compute squared coordinate values and retain to use in several places,

$$x_k^2$$

3) Compute initial estimate for σ, utilizing the squared coordinate values and constant ellipsoid parameters, viz.

$$\sigma_0 = \frac{\left(\sum x_j^2\right) - \rho\sqrt{\left(\sum x_j^2\right)}}{\sqrt{\sum \upsilon_k^{-2} x_k^2}}$$

[8] In general, other than special cases (at ellipsoid center), there are four unique solution values for σ. At least two of these are always physically meaningful. For most point locations "near" to the ellipsoid surface, all four solutions are physically real. For "far away" locations and locations approaching the equator, two solutions become mathematically "imaginary". The potential ambiguities are resolved by selecting the one real mathematical solution that corresponds with a projection onto the ellipsoid surface that is closest to the point of interest.

[9] The alternative solution cases are considered and explored in [3].

4) Iterate, $n = 0, 1, \ldots$, to obtain refined values, σ_{n+1}, starting with $\sigma_{n=0}$. Each iterative step includes:
   a) Compute intermediate values

$$\xi_{kn} \equiv \frac{\mu_k}{\mu_k^2 + \sigma_n}$$
$$\psi_{kn} \equiv \xi_{kn}^2 x_k^2$$

   b) Update σ estimate via (summations over k, not n)

$$\sigma_{n+1} = \sigma_n + \frac{1}{2} \frac{\sum \psi_{kn} - 1}{\sum \xi_{kn} \psi_{kn}}$$

   c) Check for convergence against a desired fixed tolerance[10] value , ϵ,

$$|(1 + \sigma_{n+1}) - (1 + \sigma_n)| < \epsilon$$

   d) Let $\breve{\sigma} = \sigma_{n+1}$ after convergence[11].
5) Compute the geodetic (ellipsoidal) altitude offset,

$$\breve{\eta} = \breve{\sigma} \sqrt{\sum \left(\frac{x_k}{\mu_k^2 + \breve{\sigma}} \right)^2}$$

6) Compute the geodetic angles,
   a) Compute the local ellipsoid gradient

$$\breve{g} = 2 \sum \left(\frac{x_k}{\mu_k^2 + \breve{\sigma}} \right) e_k$$

   b) Compute the geodetic longitude and latitude angle values directly from the gradient vector components[12],

$$\lambda = \arctan \left(\frac{\breve{g}_2}{\breve{g}_1} \right)$$
$$\phi = \arctan \left(\frac{\breve{g}_3}{\sqrt{\breve{g}_1^2 + \breve{g}_2^2}} \right)$$

Note that the last two steps, (5 and 6) represent two independent computation paths.

If only the "vertical" constituent of geodetic position is needed by an application, the altitude value, η, can be computed at step 5 without needing to evaluate the gradient vector or geodetic angle values in the last step.

Similarly, if only the "horizontal" constituent of geodetic position are needed, the longitude angle, λ, and parallel (latitude) angle, ϕ, can be computed from the last step 6 alone without needing to evaluate the altitude value.

[10]For example, use a value of ϵ near to machine precision (In Peridetic, a value of $\epsilon \sim 10^{-15}$ is coded to support full 64-bit IEEE-754 floating point precision while allowing some room for residual computation noise, $\epsilon \simeq 10^{-15}$). Alternatively, an ϵ value could be used that is proportionally "small enough" relative to the application of interest (e.g. $\epsilon \simeq 10^{-7}$ for meter level quality on Earth and so on).

[11]For legitimate geodesy data and configurations, convergence occurs typically within 2 or 3 iterations.

[12]Note that the factors of 2 in the individual gradient components cancel within the ratios and therefore can be avoided during computation if the gradient vector itself is not needed elsewhere.

IV. Summary

This technical note presents the mathematical formulae used to perform geodetic transformations within the Peridetic software module. It presents the specific mathematical technique used, and provides an overview of associated data value interrelationships.

The "Cartesian from Geodetic" transformation used by the "peri::xyzForLpa()" function is described compactly by expression 3. The more involved "Geodetic from Cartesian" transformation used by the "peri::lpaForXyz()" function is described by the computational recipe in section III.

Appendix

A. Geodetic Condition Metric Function

Equation 4 may be derived as follows.

Consider the geodetic vector decomposition of the point of interest location, x, into the sum of two vectors,

$$x = p + a$$

Here vector, p, is understood to be a point that lies exactly on the specified geodetic ellipsoid, and vector, a, is directed (positive outward) exactly perpendicular to the surface at location, p.

The magnitude of a, contains the geodetic altitude value, η. I.e. $\eta = \pm |a|$ with the sign resolved by incorporating vector direction into the computation (e.g. $\eta = a \cdot \hat{a}$). In principal, the direction, $\hat{a} = \frac{a}{\sqrt{a^2}}$, directly encodes the geodetic angles, λ and ϕ. However, neither computation is directly practical since the direction, $\hat{a}$, is only available for $a \neq 0$. Therefore, a more general approach is to represent the vertical direction in terms of the gradient vector associated with a scalar-valued field that is well defined everywhere.

Consider the ellipsoidal scalar field function, ς, that is defined (everywhere) as a function of vector location, x

$$\varsigma(x) = \sum \frac{x_k^2}{\mu_k^2} - 1$$

This field has zero value everywhere on the surface of the ellipsoid defined by semi-axis magnitudes, μ_k. I.e. for p on the ellipsoid surface,

$$\varsigma(p) = \sum \frac{p_k^2}{\mu_k^2} - 1 = 0 \tag{6}$$

On this surface, the gradient of the field, is perpendicular to the (level) surface of the ellipsoid and therefore parallel to the local vertical direction. I.e. the gradient vector, g, defined as

$$g = \nabla \varsigma$$

contains complete information for the geodetic angles, λ and ϕ (e.g. via expressions 1 and 2 applied using the components of g_k instead of the u_k), and also allows computing signed value of altitude via $\eta = a \cdot \hat{u} = (x - p) \cdot \hat{u}$).

The gradient, when evaluated on the surface of the ellipsoid, may be expressed in terms of the components of p, as

$$g = 2 \sum \mu_k^{-2} p_k e_k$$

Combining the above considerations, the vector decomposition may be expressed as

$$x = p + \eta \frac{g}{\sqrt{g^2}}$$

Partition into individual components, and substitute for the g_k, to represent as

$$x_k = p_k + \eta \frac{\mu_k^{-2}}{\sqrt{\sum \left(\mu_k^{-2} p_k e_k\right)^2}} p_k \tag{7}$$

Introduce a scalar pseudo altitude parameter, σ, defined by

$$\sigma \equiv \frac{\eta}{\frac{1}{2}\sqrt{g^2}} = \frac{\eta}{\sqrt{\sum \left(\mu_k^{-2} p_k e_k\right)^2}}$$

The parameter, σ, is an alternative representation for the single degree of freedom associated with altitude, η. The altitude, η, has linear units, and the gradient magnitude is associated with inverse linear units. Therefore, σ, is a parameter with area-like units.

In terms of free parameter, σ, express the preceding geodetic decomposition relationship 7 as

$$\mu_k^2 x_k = \left(\mu_k^2 + \sigma\right) p_k$$

The quantity, $\mu_k^2 + \sigma$, is strictly positive for legitimate geodetic applications. Therefore multiply by the inverse of this value to express the individual p_k coordinates for the projection point on the ellipsoid surface,

$$p_k = \frac{\mu_k^2 x_k}{\mu_k^2 + \sigma}$$

Substitute this expression into the ellipsoid surface constraint of equation 6 to arrive at the geodetic solution metric 4, viz.

$$f(x, \sigma) = \sum \left(\frac{\mu_k x_k}{\mu_k^2 + \sigma}\right)^2 - 1$$

Let $\breve{\sigma}$, represent an exact solution to this equation such that, $f(x, \breve{\sigma}) = 0$. An interesting interpretation of $\breve{\sigma}$ is as follows. If each axis of the actual ellipsoid were to be *additively* increased by a fractional amount proportional to the value, $\breve{\sigma}$, then the hypothetical new ellipsoid would contain point x exactly on its surface. E.g. point x lies on the surface of the hypothetical ellipsoid described by semi-axes magnitudes, μ_k', where,

$$\mu_k' = \mu_k + \frac{\breve{\sigma}}{\mu_k}$$

B. Initial Value Model

The free parameter, σ, is a function of both the (to be determined) magnitude of the ellipsoid gradient vector, $|g|$, and the (to be determined) altitude value, η. Thus, an initial estimate for σ can be obtained by first obtaining crude estimated values for each of these two individual entities..

Both values, $|g|$, and η, can be computed exactly for a spherical ellipsoid. Since the Figure of Earth is nearly spherical, an obvious approach is to estimate approximate values $|g_0|$, and η_0, by simply assuming a spherical Earth with some overall characteristic radius, ρ. For rotation-symmetric Earth models, a useful value for ρ is to use the geometric mean of the two unique values for ellipsoid semi-axis magnitudes, i.e. $\rho = \sqrt{\mu_{1,2}\mu_3}$.

An initial estimate value, $|g_0|$, can be computed easily from a spherical Earth model[13]. Let p_0, be the point in line with direction to x, (i.e. $p_0 \propto x$), and with magnitude that places it on the sphere surface (i.e. $|p_0| = \rho$). I.e. use,

$$p_0 = \sqrt{\frac{\mu_{1,2}\mu_3}{x^2}} x$$

This location is on the sphere surface, but in general is not on the desired ellipsoid surface. However, the ellipsoid gradient can be approximated reasonably by assuming the spherically estimated p_0, is not far (relative to characteristic size of the ellipsoid) from the location of the true point on the ellipsoid. I.e. for the gradient use,

$$g_0 = 2 \sum \mu_k^{-2} p_{0k} e_k$$

The gradient vector may next be expressed in terms of the Cartesian coordinates, x_k, and several constant geometric shape factors associated with the given ellipsoid. After substitution for the components, p_{0k}, estimate the gradient vector via,

$$g_0 = 2 \sum \left(\sqrt{\frac{\mu_{1,2}\mu_3}{\mu_k^4}}\right) \frac{x_k}{|x|} e_k$$

This provides an approximation to the desired ellipsoid gradient. The approximation is reasonably good for low eccentricity ellipsoids such as those used to represent the Figure of Earth.

An initial estimate, η_0, for the altitude value may also be obtained from the spherical approximation by using the radial excess distance,

$$\eta_0 = |x| - \rho$$

where ρ is the radius of the sphere.

Overall, an initial value for σ_0, may be expressed as,

$$\sigma_0 = \frac{\eta_0}{\frac{1}{2}|g_0|}$$

which after substitution of the above values leads to an expression for σ_0, directly in terms of the x_k^2, and μ_k,

$$\sigma_0 = \frac{\left(\sum x_j^2\right) - \sqrt{\mu_{1,2}\mu_3}\sqrt{\sum x_j^2}}{\sqrt{\sum \frac{\mu_{1,2}\mu_3}{\mu_k^4} x_k^2}}$$

References

[1] Fok, Hok Sum & Iz, H.. (2003). A Comparative Analysis of the Performance of Iterative and Non-iterative Solutions to the Cartesian to Geodetic Coordinate Transformation. 5.

[2] Lakshmanan, Swetha, "How, When, and Why Should You Normalize / Standardize / Rescale Your Data?", https://medium.com/towards-artificial-intelligence/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff, visited Feb 2021.

[3] Stellacore Corporation, "Conversion of Geodetic and Cartesian Coordinates", Internal document - unpublished as of Feb 2021.

[13] A better estimate can be obtained by using an ellipsoidal model in combination with a geocentric direction. However, in practice, evaluation of the associated quantities are more expensive than an extra iteration of the Newton Raphson process.

[4] Stellacore Corporation, "Peridetic - Easy and Effective Geodetic Transformations", GitHub repository, https://github.com/Stellacore/peridetic.
[5] Toms, R. "An improved algorithm for geocentric to geodetic coordinate conversion". United States: N. p., 1996. Web.