

Approximation solutions to the Cartesian to geodetic coordinate transformation problem

Hatem Hmam

Defence Science and Technology Group, Australia
hatem.hmam@dsto.defence.gov.au

ABSTRACT

This paper presents several approximation algorithms, which convert Cartesian coordinates to geodetic coordinates. All the proposed methods are based on the regular perturbation expansion of the reduced latitude tangent. After substitution in the latitude equation, it is shown that the developed solutions can readily be expressed as a bilinear form on the vector space, \mathbb{R}^n , where n is the perturbation expansion order. The 5th order perturbation expansion is later used to generate a more computationally efficient algorithm and yet achieves sub-millimeter-level coordinate conversion accuracy. Finally the conversion accuracy and computational efficiency of all proposed methods are compared with those of popular iterative algorithms.

KEYWORDS: Geodetic coordinates, Perturbation, latitude equation, WGS84, Geodetic ellipsoid

1. INTRODUCTION

The ubiquitous use of satellite based positioning receivers embedded in smart phones, vehicles, drones and many other devices, has emphasized the need to implement numerically stable and efficient algorithms for the transformation between the Cartesian rectangular coordinates (x, y, z) and the geodetic coordinates (φ, λ, h) . This requirement becomes critical in applications where such a transformation is executed at high frequency, particularly when only limited processing resources are available on-board the device or machine.

The computation of Cartesian coordinates given their geodetic counterparts is straightforward, but the reverse transformation requires some computational effort. Similar to many research approaches in this area, focus is placed on solving for the latitude, φ , and altitude, h , in the meridian plane (ie. determination of (φ, h) given $p \equiv \sqrt{x^2 + y^2}$ and z). Computing the longitude, λ , is straightforward and non-iterative (see Ligas et al (2011)).

The literature on the Cartesian to geodetic transformation is extensive and ranges from exact methods to iterative or approximation based algorithms (see for example Gerdan (1999) for a

brief survey of earlier works). The article of Ligas et al (2011) provides a performance comparison of some of the most efficient and popular iterative algorithms. Other approaches not covered in Ligas (2011) include the exact algebraic method of Vermeille (2002, 2004 and 2011), Zhang et al (2005) and Zeng (2013). The solutions of the two latter methods are based on minimizing the distance between a point in space and its projection on the geodetic ellipsoid and solving a quartic equation. Although these approaches lead to exact transformation inversion, they are not computationally efficient.

Given the near spheroid shape of planet earth (ie. very small eccentricity), a number of approximation methods use this property to an advantage (Turner 2009 and 2015). By writing all coordinates as regular perturbation expansions, one is able to iteratively express the expansion coefficients as a function of the Cartesian coordinates. In practice a low order perturbation expansion is often sufficient to yield a highly accurate coordinate conversion. However, if the expansion coefficients are themselves expensive to compute, then this leads to a slower coordinate inversion process. The perturbation based works of Turner (2009 and 2015) are not computationally as efficient as Halley one-iteration method of Fukushima (2006).

Fukushima (1999 and 2006) has contributed a number of iterative algorithms, the latest of which is the Halley iteration method to solve the latitude equation where the unknown is the tangent of the reduced latitude. The convergence rate of the algorithm is boosted to cubic at the expense of a slight increase in computational effort. He also revised the Bowring Newton-based iterative method (Bowring 1976) and suggested a faster implementation of it. It should be pointed out that both these algorithms run into numerical problems if the Cartesian point is on the polar axis (or z-axis).

In this paper a conversion method is presented, which is based on the regular perturbation expansion of the reduced latitude tangent. This expansion is substituted into the latitude equation to deduce the perturbation coefficients. By increasing the expansion order, more accurate conversion approximations are obtained. In this work accurate coordinate conversion approximations are derived based on the 3rd, 4th and 5th perturbation order expansion. A fourth algorithm which strikes a compromise between extreme conversion accuracy and computational effort is given. It is shown that this algorithm achieves sub-millimeter conversion accuracy for altitudes between -10 and 100000 km.

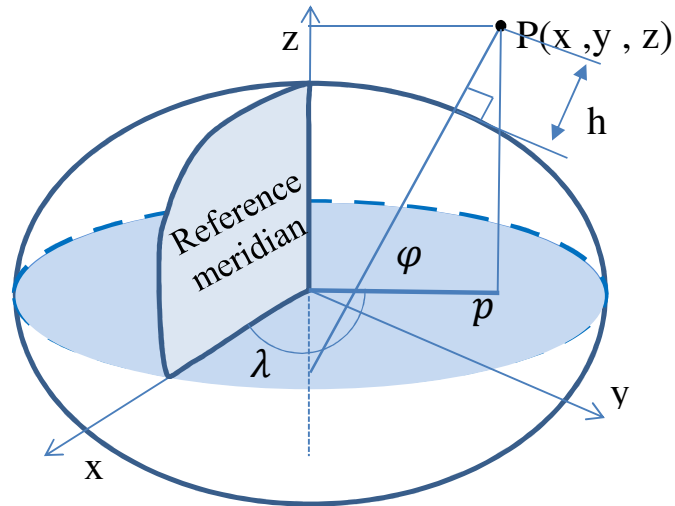


Figure 1: Cartesian and Geodetic coordinates

2. REGULAR PERTURBATION BASED APPROXIMATIONS

2.1 The Latitude Equation

First it is convenient to define $p \equiv \sqrt{x^2 + y^2}$ and carry out the conversion process on the meridian plane. The specific task in the coordinate conversion process is to determine φ and h given p and z . Figure 1 provides a visual illustration of these coordinates. The latitude equation, which is given in Fukushima (1999 and 2006) is the starting point of our conversion approach. This equation reads as

$$PT - Z - E \frac{T}{\sqrt{1+T^2}} = 0 \quad (1)$$

where $T \equiv \tan \psi$, $E \equiv e^2$, $P \equiv p/a$, $Z \equiv e_c z/a$ and $e_c \equiv \sqrt{1-E}$. Here a and e are respectively the semi-major axis and the first eccentricity. The angle, ψ , is called the parametric or reduced latitude and is shown in Figure 2. To help understand how this angle is defined, an auxiliary circle whose radius is the ellipse semimajor axis, is first drawn and superimposed on the ellipse of Figure 2. The normal projection of the point (p, z) on the ellipse is next obtained as the point, Q , as illustrated in the figure. Finally the intersection point of the auxiliary circle and the line crossing Q and parallel to the z -axis, is determined as R . The angle, ψ , is then defined as the angle of the radial line crossing the point R . The relationship between ψ and φ follows from the geometric properties of the ellipse and its external auxiliary circle as $\tan \psi = e_c \tan \varphi$. For more details see Deakin et al (2013).

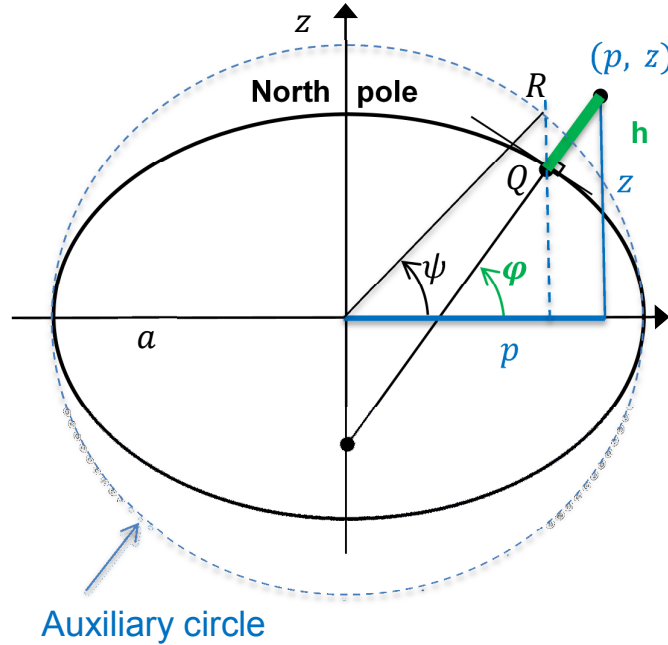


Figure 2: Parametric latitude angle definition. The angle, ψ , is determined as the intersection point of the radial line with the external auxiliary circle.

If T is a solution to (1), then the latitude and altitude follow as

$$\varphi = \tan^{-1}(T/e_c) \quad (2)$$

$$h = \frac{e_c p + zT - b\sqrt{1+T^2}}{\sqrt{e_c^2 + T^2}} \quad (3)$$

2.2 Perturbation Expansion of T

Given that $E \approx 0.00669437999 \ll 1$ (WGS84), regular perturbation theory is used to approximately solve (1). By approximating T by the sequence, $T_n = \sum_{i=0}^n \alpha_i E^i$, substituting it in (1), collecting all powers of E and setting their coefficients to zero, one obtains expressions for α_i , $0 \leq i \leq n$ in terms of the Cartesian coordinates. With the help of a symbolic processing package such as Matlab Symbolics Toolbox, it turns out that

$$T_n = (Z/P)(1 + \mathbf{v}_n^T G_n \mathbf{u}_n) = (e_c z/p)(1 + \mathbf{v}_n^T G_n \mathbf{u}_n) \quad (4)$$

where $\mathbf{u}_n = [u^0, u^2, \dots, u^{2(n-1)}]^T$, $\mathbf{v}_n = [v^1, v^2, \dots, v^n]^T$ and G_n is an $n \times n$ lower triangular square matrix, whose entries are given in Table 1. The elements u and v are respectively $c Z/P$ and $c E/P$ where $c \equiv 1/\sqrt{1 + (Z/P)^2}$. The parameters, u and v , may conveniently be expressed as $u \equiv \kappa \frac{e_c}{a} z$ and $v \equiv \kappa E$ where $\kappa \equiv a/\sqrt{p^2 + (e_c z)^2}$.

G_n	u^0	u^2	u^4	u^6	u^8	u^{10}
v	1					
v^2	1	-1				
v^3	1	-7/2	5/2			
v^4	1	-8	15	-8		
v^5	1	-15	427/8	-273/4	231/8	
v^6	1	-25	146	-330	320	-112

Table 1: Matrix entries of G_n

To give an example, the third order perturbation expansion, T_3 , uses the 3×3 matrix, G_3 , whose entries are shown in the shaded section of Table 1. The corresponding perturbation solution, T_3 , is $(Z/P)(1 + \mathbf{v}_3^T G_3 \mathbf{u}_3)$ and its associated expansion error is $\mathcal{O}(E^4)$.

If we put $V_n \equiv pT_n / e_c$, then it follows that

$$V_n = z(1 + \mathbf{v}_n^T G_n \mathbf{u}_n) \quad (5)$$

where the expression on the right hand side is no longer rational.

Remark:

Note that with the exception of the first row, the sum of each row in Table 1 is zero.

The geodetic altitude follows as

$$h_n = \sqrt{V_n^2 + p^2} \left(1 - \frac{a}{\sqrt{e_c^2 V_n^2 + p^2}} \right) \quad (5a)$$

where V_n is given by (5). To compute the latitude, it is numerically convenient to consider two regions, the equatorial region, which comprises all space points where $p \geq |z|$ and the polar region, which encompasses all remaining space points. The latitude follows as

$$\varphi_n = \tan^{-1}(V_n/p), \text{ if } p \geq |z| \text{ and } \varphi_n = \text{sign}(z)(\pi/2) - \tan^{-1}(p/V_n) \text{ otherwise} \quad (5b)$$

Using (5b) it is ensured that the argument magnitude of the arc tangent is less or nearly 1.

2.3 Third, Fourth and Fifth Order Approximations

By restricting the perturbation expansion to the third order, we obtain $V_3 = z(1 + \mathbf{v}_3^T G_3 \mathbf{u}_3)$, which after algebraic manipulations leads to

$$V_3 = z(1 + v(1 + (v - w)(1 + v - 2.5w))) \quad (6)$$

where $v \equiv aE/\sqrt{p^2 + (e_c z)^2}$ and $w \equiv vu^2 = (1 - E)z^2 v^3 / (aE)^2$.

Similarly the fourth and fifth order expansions follow as

$$V_4 \equiv z\left(1 + v\left(1 + (v - w)(1 + v + v^2 + w(8w - 7v - 5/2))\right)\right) \quad (7)$$

$$V_5 \equiv z\left(1 + v\left(1 + (v - w)\left((1 + v)(1 + v^2) + w\left(-\frac{5}{2} + w\left(8 - \frac{231}{8}w\right) + v\left(\frac{315}{8}w - 14v - 7\right)\right)\right)\right)\right) \quad (8)$$

The geodetic altitude and latitude are computed using (5a) and (5b).

3. FAST COORDINATE CONVERSION APPROXIMATION

3.1 An Alternative Latitude Equation

It is convenient at this point to introduce an alternative formulation to the latitude equation (1), which we will use later in this section to derive a fast coordinate conversion algorithm.

We transform T into its inverse by putting $T \equiv 1/F$ and substituting $1/F$ into the latitude equation (1). After algebraic rearrangements one obtains,

$$ZF - P + E \frac{F}{\sqrt{1+F^2}} = 0, \quad (9)$$

where the unknown variable, F , is the cotangent of the reduced latitude, ψ . This equation looks very similar to (1) except for swapping P with Z and changing the sign of E . Hence its perturbation-based solution is similar to (4) except that P and Z are swapped and E is replaced by $-E$. In other words we let $c \equiv 1/\sqrt{1 + (P/Z)^2}$, $t \equiv cP/Z$ and $s \equiv -cE/Z$ and compute

$$F_n \equiv \frac{1}{T_n} = (P/Z)(1 + \mathbf{s}_n^T G_n \mathbf{t}_n)$$

where $\mathbf{t}_n = [t^0, t^2, \dots, t^{2(n-1)}]^T$ and $\mathbf{s}_n = [s^1, s^2, \dots, s^n]^T$. If we put $H_n \equiv e_c z F_n$, then it follows that

$$H_n = p(1 + \mathbf{s}_n^T G_n \mathbf{t}_n) \quad (10)$$

The parameters, t and s , may explicitly be expressed as $t \equiv \kappa p/a$ and $s \equiv -\kappa E$ where $\kappa \equiv a/\sqrt{p^2 + (e_c z)^2}$.

The geodetic altitude and latitude are given by

$$p \geq |z|: h_n = \sqrt{(z/H_n)^2 + 1} \left(p - \frac{a}{\sqrt{1+e_c^2(z/H_n)^2}} \right), \quad \varphi_n = \tan^{-1}(z/H_n) \quad (10a)$$

$$p < |z|: h_n = \frac{p(H_n/z) + |z| - b\sqrt{1+e_c^{-2}(H_n/z)^2}}{\sqrt{1+(H_n/z)^2}}, \quad \varphi_n = \text{sign}(z)(\pi/2) - \tan^{-1}(H_n/z) \quad (10b)$$

where H_n is given by (10).

Similar to the 5th order expansion parameter, V_5 , one may derive its corresponding analogue on the horizontal or equatorial plane, as

$$H_5 \equiv p \left(1 + s \left(1 + (s - w) \left((1 + s) (1 + s^2) + w \left(-\frac{5}{2} + w \left(8 - \frac{231}{8} w \right) + s \left(\frac{315}{8} w - 14s - 7 \right) \right) \right) \right) \right) \quad (11)$$

where $s \equiv -aE/\sqrt{p^2 + (e_c z)^2}$ and $w \equiv st^2 = p^2 s^3 / (aE)^2$.

3.2 Fast Coordinate Conversion Algorithm

The main idea in this subsection is to demonstrate how to eliminate two computationally expensive operations from the altitude expressions (5a) and (10a). These consist of the square root and the division operators. This, however, is achieved at the expense of a slight degradation of the conversion accuracy, which is a small price to pay as the 5th order conversion approximation is extremely accurate (refer to Table 2).

After substituting the 5th order expansion, $T = T_5 = \sum_{i=0}^5 \alpha_i E^i$, into the latitude equation (1), one may write $\frac{T}{\sqrt{1+T^2}} \approx \frac{PT-Z}{E}$, which allows the elimination of one square root and one division operation from the altitude expression in (5a). From $V_n \equiv T_n p/e_c$ and replacing

$\frac{T_n}{\sqrt{1+T_n^2}}$ by $\frac{PT_n-Z}{E}$ in (5a), it follows that $h_n = \left(|V_n| - \frac{a}{\sqrt{\left(\frac{e_c}{T_n}\right)^2 + e_c^2}} \right) \sqrt{\left(\frac{p}{V_n}\right)^2 + 1}$, which after a sequence of algebraic manipulations reduces to $h_n = \frac{1}{E} (|z| + (E - 1)|V_n|) \sqrt{(p/V_n)^2 + 1}$.

Similarly from (9) and noting that $\frac{F_n}{\sqrt{1+F_n^2}} \approx \frac{P-ZF_n}{E}$, then substituting this approximation in the alternative altitude expression of (10a), it can be shown that $h_n = \frac{1}{E} ((E - 1)p + H_n) \sqrt{(z/H_n)^2 + 1}$, where $H_n = e_c z F_n$.

To summarize, the faster coordinate conversion algorithm uses V_n or H_n ($n \geq 5$) according to

$$p \geq |z|: h_n = \frac{1}{E} ((E-1)p + H_n) \sqrt{(z/H_n)^2 + 1}, \varphi_n = \tan^{-1}(z/H_n) \quad (12a)$$

$$p < |z|: h_n = \frac{1}{E} (|z| + (E-1)|V_n|) \sqrt{(p/V_n)^2 + 1}, \varphi_n = \text{sign}(z) \pi/2 - \tan^{-1}(p/V_n) \quad (12b)$$

4. PERFORMANCE ASSESSMENT

4.1 Coordinate Conversion Accuracy

In this work the metric used to assess algorithm performance accuracy is the gap distance between the input point, (p, z) , and the reconstructed point obtained by subjecting the computed coordinates, (φ, h) , to the forward transformation. The resulting Euclidean distance gap or error is expressed in mm over the whole region, \mathcal{R} , of points for which the altitude ranges between -10 km and 100000 km. The performance measure for each coordinate conversion algorithm is the maximum error over the region \mathcal{R} .

As is clear from Table 2 for all perturbation-based conversion approximations, the maximum error is generally lower at higher altitudes. This behaviour at higher altitudes is desirable and not mimicked by many other Cartesian to geodetic conversion approaches. Furthermore as expected, the conversion approximation based on the 5th order perturbation expansion is extremely accurate. However, such extreme accuracy is of little use in satellite-based positioning because the Cartesian coordinates are typically estimated with meter-level accuracy. If the carrier phase is employed to fine-tune pseudo-range measurements, then cm-level Cartesian coordinate accuracy may be obtained, which require sub-millimeter-level or better coordinate conversion accuracy.

Conversion Method	Altitude (km)				
	-10 to 10	10 to 1000	1000 to 20000	20000 to 35000	35000 to 100000
3 rd order App.	2.6	2.6	1.66	0.037	0.0094
4 th order App.	0.016	0.016	0.009	6.1e-5	8.3e-5
5 th order App.	1.2e-4	1.2e-4	5.9e-5	3.3e-5	8.3e-5
Fast Appr.	0.022	0.022	0.011	0.002	0.0047
Bowring	0.0013	9.5	295	367	453
Fukushima1	3.8e-6	0.004	0.7	0.96	1.32
Fukushima2	0.2	0.2	0.077	0.066	0.135

Table 2: Coordinate conversion accuracy (in mm) of our proposed and popular iterative approaches.

A close examination of Table 2, reveals that the 4th order, 5th order and the fast approximations are suitable to compute the geodetic coordinates in the entire altitude region -10 to 100000 km. The 3rd order approximation may be used but at the expense of a small bias, in the altitude range, -10 to 20000 km. Referring to the last row of Table 2, the accelerated one iteration algorithm of Fukushima (2006) may also be used but its conversion accuracy is lower than that of the fast approximation. Furthermore, as we shall see next, the fast approximation algorithm is computationally more efficient.

4.2 Coordinate Conversion Efficiency

One way to assess the computational efficiency of each method, is to count all arithmetic operations weighted by their CPU time usage. This, however, only gives a partial picture of what goes on inside the CPU as modern computers may execute instructions simultaneously

and often use pipelining to speed up processing. Fukushima (1999) gave some indicative performance numbers comparing the CPU time of different arithmetic operations. In reality such numbers vary depending on the CPU hardware architecture and how instructions are queued and executed in the CPU. However, it is generally true that reducing the number of expensive arithmetic operations, namely the square and higher roots, divisions, trigonometric, and inverse trigonometric operations, lead to a noticeable increase in processing speed.

Conversion Method	Expensive Arithmetic Operations		
	Division	Square root	Arc Tangent
3 rd order Approx.	3	4	1
4 th order Approx.	3	4	1
5 th order Approx.	3	4	1
Fast Approx.	2	3	1
Bowring (1 iter.)	4	4	1
Fukushima1 (1 iter.)	4	4	1
Fukushima2 (1 iter.)	2	4	1

Table 3: Count of expensive arithmetic operations for each coordinate conversion method

Examining Table 3, it is clear that the fast approximation algorithm uses the least number of expensive arithmetic operations, followed by Fukushima's accelerated one iteration algorithm. A Fortran program was written to compare the algorithm runtimes on a desktop with the processor/memory specifications: Intel i7-2600 3.4GHz, RAM 8.00 GB. The obtained runtimes are approximately (listed in the order of Table 3) 61.3, 61.4, 65.5, 60.7, 74.3, 79.0, and 72.2 ns, where it is clear that the Fast Approximation algorithm achieves the most computationally efficient coordinate transformation.

5. CONCLUSIONS

A number of approximation algorithms, which convert Cartesian coordinates to geodetic coordinates, are presented in this paper. All the proposed methods are based on the regular perturbation expansion of the reduced latitude tangent. The developed solutions are shown to be readily expressed as a bilinear form on the vector space, \mathbb{R}^n , where n is the perturbation expansion order. The 5th order perturbation expansion is particularly useful as it enables the generation of a reasonably efficient algorithm, which achieves sub-millimeter-level coordinate conversion accuracy. This algorithm is then modified to generate a more computationally efficient coordinate transform that achieves sub-millimeter-level coordinate conversion accuracy. Finally the conversion accuracy and computational efficiency of all proposed methods are compared with those of Bowring and Fukushima's one iteration algorithms.

REFERENCES

- Ligas, M. and Banasik, P., (2011) Conversion between Cartesian and geodetic coordinates on a rotational ellipsoid by solving a system of nonlinear equations, *Geodesy and Cartography*, Vol. 60, pp 145-159
- Bowring B. R., (1976), Transformation from spatial to geographical coordinates, *Survey Review*, Vol. 23, pp 323-327
- Deakin, R. E. and Hunter, M.N., *Geometric Geodesy: Part A*, School of Mathematical Geospatial Sciences, RMIT University, 2013
- Fukushima T., (1999): Fast transform from geocentric to geodetic coordinates, *Journal of Geodesy*,

Vol. 73, pp. 603–610

Fukushima T., (2006) Transformation from Cartesian to geodetic coordinates accelerated by Halley's method, *Journal of Geodesy*, Vol. 79, pp. 689–693

Gerdan GP, Deakin RE (1999) Transforming Cartesian coordinates to geographical coordinates. *Australian Surveyor* 44:55–63

Vermeille H., (2002): Direct transformation from geocentric to geodetic coordinates, *Journal of Geodesy*, Vol. 76, pp. 451–454.

Vermeille H., (2004): Computing geodetic coordinates from geocentric coordinates, *Journal of Geodesy*, Vol. 78, pp. 94–95

Vermeille H., (2011): An analytical method to transform geocentric into geodetic coordinates, *Journal of Geodesy*, Vol. 85, pp. 105–117

Zhang C.D., Hsu H.T., Wu X.P., Li S.S., Wang Q.B., Chai A.Z., Du L., (2005): An alternative algebraic algorithm to transform Cartesian to geodetic coordinates, *Journal of Geodesy*, Vol. 79, pp. 413–420

Zeng, H. (2013), Explicitly computing geodetic coordinates from Cartesian coordinates, *Earth Planets Space*, Vol. 65, pp. 291–298