Computers & Geosciences 36 (2010) 1145–1149

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

# An iterative algorithm to compute geodetic coordinates

Chanfang Shu [a,b,*], Fei Li [b]

[a] College of Civil Engineering, Nanjing University of Technology, 200 Zhongshan North Road, Nanjing 210009, PR China
[b] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, PR China

A R T I C L E  I N F O

*Article history:*
Received 25 July 2009
Received in revised form
4 February 2010
Accepted 4 February 2010

*Keywords:*
Coordinate transformation
Newton–Raphson method
Cartesian coordinates
Geodetic coordinates

A B S T R A C T

By using the Newton–Raphson method to solve a quartic equation of the Lagrange parameter, we propose a new iterative algorithm for the transformation from Cartesian to geodetic coordinates. Numerical experiments show that the new method is sufficiently precise and free from singularity and non-convergency, except within 50 km of the geocentre. After one iteration, the maximum error of latitude is less than $10^{-8}$ arc-sec and the relative height error is less than $10^{-15}$ over the range of geodetic heights from $-10^6$ to $10^{12}$ m. Comparisons of the complexity and performance with some other well-known methods indicate that the algorithm is efficient and accurate.


## 1. Notation

$a$, $b$: semi-major axis, semi-minor axis of geodetic ellipsoid

$e=\dfrac{\sqrt{a^2-b^2}}{a}$ : first eccentricity

$N=\dfrac{a}{\sqrt{1-e^2\sin^2\varphi}}$ : the radius of curvature in the prime vertical

$x,y,z$ : Cartesian coordinates

$\lambda,\varphi,h$ : geodetic longitude, geodetic latitude, geodetic height

## 2. Introduction

The transformation between Cartesian and geodetic coordinates is frequently encountered in many applications, such as GPS navigation, geodesy and astro-geodesy. It is well-known that the Cartesian coordinates can be directly calculated from geodetic coordinates through the following formulas:

$$\begin{aligned} x&=(N+h)\cos\varphi\cos\lambda\\ y&=(N+h)\cos\varphi\sin\lambda\\ z&=[N(1-e^2)+h]\sin\varphi \end{aligned} \tag{1}$$

However, the inverse computation poses certain difficulties, especially when solving for the geodetic latitude and geodetic height. In the past, many scholars have dwelled upon this problem and proposed a number of methods that can be divided into three categories: (1) exact or closed-form approaches (e.g. Borkowski, 1987; Levin, 1988; Vermeille, 2002; Zhang et al., 2005; Zhu, 1993,1994); (2) approximation methods (e.g. Bowring, 1976, 1985; Olson, 1996; Turner, 2009; You, 2000); and (3) iterative methods (e.g. Borkowski, 1989; Feltens, 2008, 2009; Fukushima, 1999, 2006; Heiskanen and Moritz, 1967; Jones, 2002; Lin and Wang, 1995; Pollard, 2002; Wu et al., 2003). Other approaches can be found from a review paper with many references in Featherstone and Claessens (2008). In addition, Laskowski (1991), Gerdan and Deakin (1999) and Fok and Iz (2003) have conducted some numerical comparisons among some of the existing methods. The implementation performance of all algorithms focuses on the following three aspects: accuracy, stability and computational speed.

Zhang et al. (2005) proposed an idea to complete the inverse transformation by establishing and solving a quartic equation of the Lagrange parameter. Unfortunately, their closed algebraic algorithm is complex and its solution requires much calculation time, yet the development of fast, concise, and yet accurate procedures is very much needed for some real-time applications with limited computational resources (such as mobile location based on GPS). Here, we will present an iterative method to solve the Zhang's quartic equation using the Newton–Raphson method. An iterative initial value, which is crucial to the Newton–Raphson method, will be provided logically in Section 4. Numerical tests and comparisons with some other methods will be conducted in Section 5.

* Corresponding author at: College of Civil Engineering, Nanjing University of Technology, 200 Zhongshan North Road, Nanjing 210009, PR China.
Tel.: +86 25 83239935; fax: +86 25 83239616.
*E-mail address:* shuchanfang@gmail.com (C. Shu).

0098-3004/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2010.02.004

## 3. Theory

The geodetic longitude can easily be computed via $\lambda = \tan^{-1}(y/x)$ (an alternative formula with better stability can be found in Vermeille, 2004); the inverse computation for geodetic coordinates can be reduced to a two-dimensional problem from $(r=\sqrt{x^2+y^2}, z)$ to $(\varphi,\ h)$ on the meridian plane (see Fig. 1). Zhang et al. (2005) describe the geodetic height as the minimum distance between the point $P$ and the surface of the reference ellipsoid. Based on their theory, a Lagrange conditional extreme value equation can be obtained as follows:

$$h^2 = \min\left[(r-r_0)^2 + (z-z_0)^2 + abk\left(\frac{r_0^2}{a^2} + \frac{z_0^2}{b^2} - 1\right)\right] \tag{2}$$

where $k$ is the Lagrange parameter. Through partial differentiation of Eq. (2) with respect to $r_0$ and $z_0$, respectively, we can affiliate a space point $P$ with its Helmert projection $P_0$ (project the point $P$ onto the surface of the ellipsoid through the straight ellipsoidal normal) by using the Lagrange parameter $k$

$$r_0 = \frac{ar}{a+bk}$$

$$z_0 = \frac{bz}{b+ak} \tag{3}$$

By substituting Eq. (3) into the ellipse equation $r_0^2/a^2 + z_0^2/b^2 = 1$, we obtain a quartic equation in terms of the Lagrange parameter $k$

$$f(k) \equiv p^2q^2 - r^2q^2 - z^2p^2 = 0 \tag{4}$$

where $p=a+bk$ and $q=b+ak$. The expressions $a+bk$ and $b+ak$ are both positive, because the point $P$ and its Helmert projection $P_0$ should be in the same quadrant. Then, the solution interval of $f(k)$ becomes $k \in (-(b/a), +\infty)$. Once the Lagrange parameter $k$ is solved, the geodetic latitude of point $P$ can be computed directly via the following formulas:

$$\varphi = \tan^{-1}\left(\frac{a^2 z_0}{b^2 r_0}\right) = \tan^{-1}\left(\frac{apz}{bqr}\right) \tag{5}$$

The geodetic height can also be calculated by

$$h = k\sqrt{\frac{b^2r^2}{p^2} + \frac{a^2z^2}{q^2}} \tag{6}$$

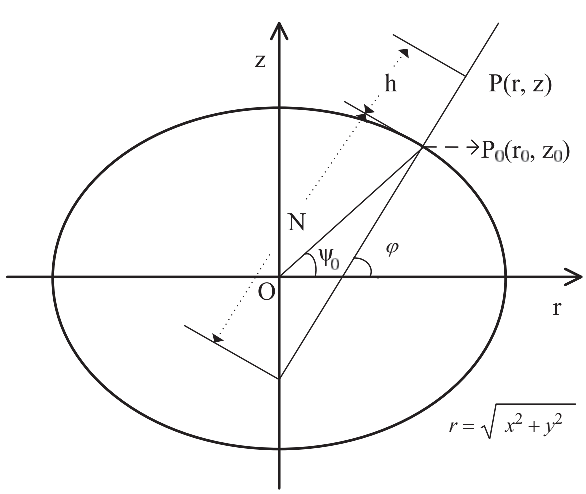


**Fig. 1.** Relationship between Cartesian coordinates and geodetic coordinates.

## 4. Algorithm

In contrast to Zhang et al. (2005) closed algebraic algorithm, we apply Newton–Raphson methods to solve the above quartic Eq. (4), and the resulting iterative formula becomes

$$k_{n+1} = k_n - \frac{f(k)}{f\prime(k)} \tag{7}$$

where the first derivative of $f(k)$ is given as

$$f\prime(k) = 2(bpq^2 + ap^2q - ar^2q - bz^2p) \tag{8}$$

The Newton–Raphson method can often converge remarkably quickly, if the iteration begins sufficiently near the desired root (Yan, 2006). Namely, the key point of the Newton–Raphson method is to provide an iterative starter that is stable, precise and fast.

Here, considering that $k=0$ when the point $P$ is on the surface of the reference ellipsoid, $k=r-a/b$ when $z=0$ and $k=|z|-b/a$ when $r=0$, an approximate solution of Eq. (4), which satisfies the above conditions exactly, can be given as follows:

$$k_{appro} = \frac{(\sqrt{a^2z^2+b^2r^2}-ab)(a^nz^m+b^nr^m)}{a^{n+2}z^m+b^{n+2}r^m}, \quad \begin{matrix} n=0,1,2,3\cdots \\ m=1,2,3\cdots \end{matrix} \tag{9}$$

By comparison with the exact solution by substituting different combinations of $(n,\ m)$ into Eq. (9), we find that the approximate solutions with the combination of (0, 2), (1, 2) and (2, 2), respectively, are more precise. Thus, the simpler approximation of the combination (0, 2) is used as the initial guess as follows:

$$k_{ini} = \frac{(\sqrt{a^2z^2+b^2r^2}-ab)R^2}{a^2z^2+b^2r^2} \tag{10}$$

where $R^2 = x^2+y^2+z^2$

## 5. Numerical experiments and comparisons

In this section, the implementation performances of the new algorithm, namely its accuracy, stability and computational speed, are tested using two numerical experiments and compared with some existing methods. In the first test, the space points are obtained by varying latitude from 0° to 90° by 0.5° steps and height $h$ from $-10^6$ to $10^6$ m by 100 m steps. In the second test, height $h$ ranged from 10° to $10^{12}$ m. Because of their symmetry, only the cases for positive latitude are considered in both tests. The test results were obtained on a Dell INSPIRON 500 m computer with a Pentium M 1300 MHz processor and 256 MB RAM. The code was programmed using Matlab 7.0.

### 5.1. Accuracy

To estimate the accuracy of the algorithm, we first converted the real geodetic coordinates of the test points to Cartesian coordinates using the exact formula (1). Then, the calculation of geodetic coordinates could be obtained from these Cartesian coordinates using the inverse algorithm. The latitude error (units: arc-sec) and height error (units: mm) are defined as the absolute value of the difference between the real geodetic coordinate and their calculated values. The results of both tests show that the new algorithm is sufficiently precise, because its error is less than $10^{-8}$ arc-sec in latitude (see Fig. 2.) and 0.1 mm in height ranging from $-10^6$ to $10^6$ m (see Fig. 3a) after only one iteration. Though the height error increases as the height increases in middle latitudes (see Fig. 3b), its relative error (the ratio of height error to height) is less than $10^{-15}$. Of course, higher accuracy in geodetic

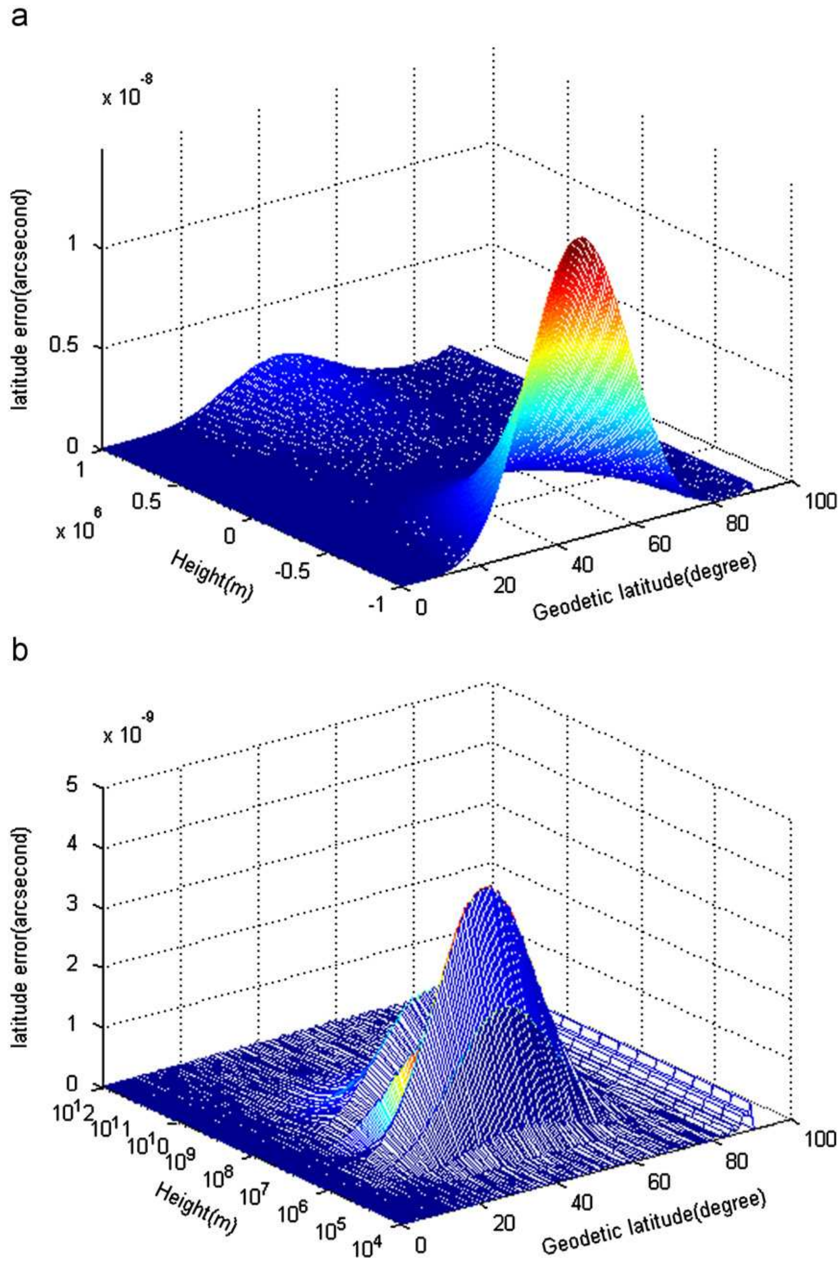


**Fig. 2.** Geodetic latitude error after one iteration in (a) first test and (b) second test.

height can be obtained after more iterations, but at the expense of increased computation time.

### 5.2. Stability

Numerical results show that the algorithm works rigorously for most points in space, excluding those near the centre of the Earth. Non-convergent points are all wrapped in a sphere of radius about 50 km from the geocentre (Fig. 4). This also occurs in other algorithms, e.g. Vermeille (2002), Bowring (1976). As with Wu et al. (2003), we define non-convergence as and when the algorithm converges to a wrong solution (denoted by '+') or the maximum iteration number is reached (denoted by 'o'). Here, the allowed maximum iteration number $k_{\max}$ is set to 20. Obviously, this non-convergent problem can be ignored for all practical terrestrial applications.

### 5.3. Computational speed of algorithm

Fukushima (1999) has listed the CPU times of some major operations, e.g. addition or subtraction, multiplication, division, square root and trigonometric functions (see Table C1 of Fukushima, 1999). According to that table, one multiplication operation requires the same CPU time as one addition or subtraction operation in double-precision arithmetic. A call for division, square root and trigonometric function operations costs 12, 30 and 40–50 times the addition and/or multiplication operation. Thus, divisions and especially square root and trigonometric function operations should be avoided as much as possible for the establishment of fast algorithms.

For evaluating the computational speed of the new algorithm, we compared the main arithmetic operations (including the computation of both latitude and height) of the presented algorithm with some well-known methods, excluding the common

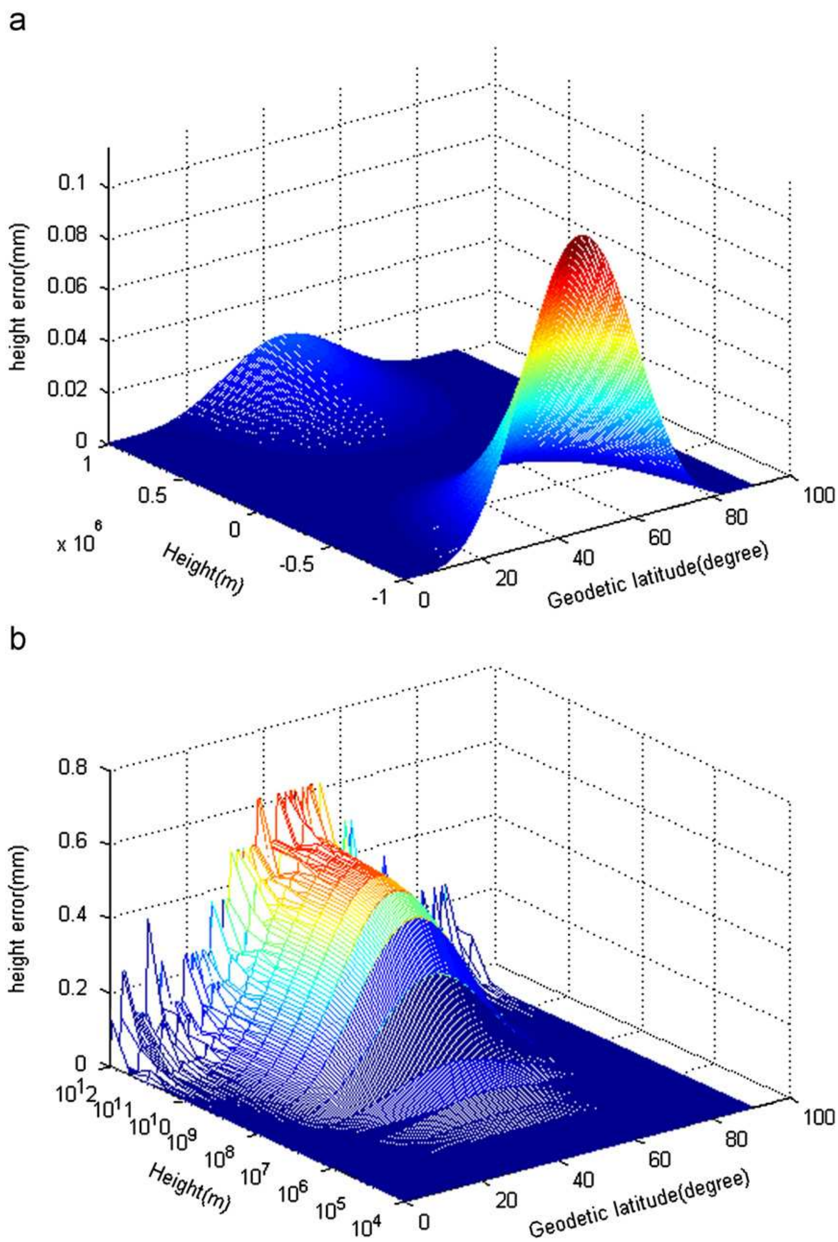


**Fig. 3.** Geodetic height error after one iteration in (a) first test and (b) second test.

operations between constants (see Table 1). The arithmetic operations of the Bowring's method are counted based on a fast implementation proposed by Fukushima (1999), Appendix C. Of the six variants presented by Fukushima (2006), the optimal algorithm, ''Halley's iteration in terms of S and C'' (variant ''$f$'' in his paper), was chosen. The operation counts in Table 1 show that the time-consuming operations, such as trigonometric functions, square roots and division, are reduced adequately in the new algorithm as much as two Fukushima's methods. For 3–4 iterations in average are required in Fukushima's (1999), one more square root operation in our method would counteract the CPU time saved by having only one iteration required. Furthermore, since the precision of our algorithm in latitude is about $10^2$ higher than Fukushima's (2006) after only one iteration, while two more division operations are required, we recommend the new algorithm as an alternative method of good performance.

## 6. Conclusion

We have presented an iterative algorithm to convert Cartesian coordinates to geodetic coordinates. The approach solves a quartic equation of the Lagrange parameter using the Newton–Raphson method. An iterative starter, which is significant to the Newton–Raphson method, is provided through a simplification. Numerical experiments show that the proposed algorithm is: (1) precise, because latitude error is less than $10^{-8}$ arc-sec and relative error of height is less than $10^{-15}$ over the range from $-10^6$ to $10^{12}$ m after only one iteration; (2) stable, since it works rigorously for most points in space, excluding those within 50 km of the geocentre; and (3) fast, as some time-consuming arithmetic operations are reduced sufficiently and only one iteration is required for practical applications. Comparisons with some well-known methods indicate that the new algorithm is able to compete with them.

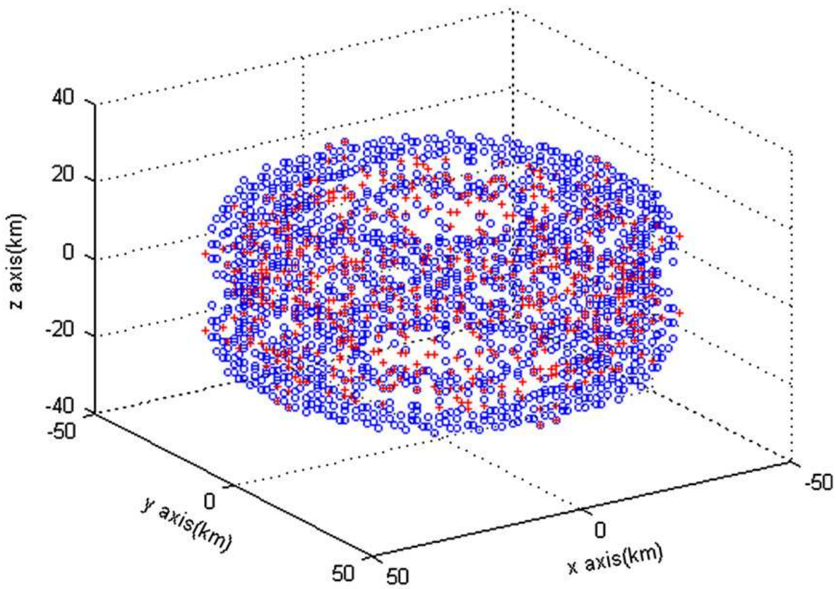


**Fig. 4.** Non-convergent points in near-geocentre region. '+' denotes that algorithm converges to a wrong solution, 'o' denotes that maximum iteration number $k_{\max}$ is reached.

**Table 1**
Comparison of arithmetic operations (latitude and height).

| Algorithms | Cubic roots | Square roots | Division | Trigonometric |
|---|---|---|---|---|
| Borkowski (1987)[a] | 1 | 3 | 6 | 3 |
| Vermeille (2002)[a] | 1 | 5 | 6 | 1 |
| Zhang et al. (2005)[a] | 1 | 8 | 22 | 1 |
| Bowring (1976)[b] | 0 | 3+$k$ | 1+2$k$ | 1 |
| Heiskanen and Moritz (1967) | 0 | 2+$k$ | 2+3$k$ | 2+2$k$ |
| Lin and Wang (1995) | 0 | 3 | 6+5$k$ | 3 |
| Jones (2002) | 0 | 2 | 3+3$k$ | 3+3$k$ |
| Wu et al. (2003) | 0 | 3 | 3+$k$ | 2 |
| Fukushima (1999) | 0 | 2 | 3+$k$ | 1 |
| Fukushima (2006)[c] | 0 | 2+$k$ | 2 | 1 |
| This paper | 0 | 3 | 3+$k$ | 1 |

[a] Denotes exact algorithms.
[b] Arithmetic operations of a fast implementation of Bowring's formula (see Appendix C, Fukushima, 1999).
[c] The optimal algorithm ($f$) described in Fukushima (2006); $k$ denotes the iteration number.

## Acknowledgements

The authors thank the editors and two anonymous reviewers for their excellent suggestions and thorough review, which improved our original manuscript. This work has been supported by grants from the National High Technology Research and Development Program of China (No. 2009AA12Z318, 2009AA121401), the National Science Foundation (No. 40974014) and the Open Fund of Key Laboratory of Polar Survey and Mapping, SBSM (No. 200808).

## References

Borkowski, K.M., 1987. Transformation of geocentric to geodetic coordinates without approximations. Astrophysics and Space Science 139, 1–4.
Borkowski, K.M., 1989. Accurate algorithms to transform geocentric to geodetic coordinates. Bulletin Géodésique 63, 50–56.
Bowring, B.R., 1976. Transformation from spatial to geographical coordinates. Survey Review 23, 323–327.
Bowring, B.R., 1985. The accuracy of geodetic latitude and height equations. Survey Review 28, 202–206.
Featherstone, W.E., Claessens, S.J., 2008. Closed-form transformations between geodetic and ellipsoidal coordinates. Studia Geophysica et Geodaetica 52, 1–18.
Feltens, J., 2008. Vector methods to compute azimuth, elevation, ellipsoidal normal, and the Cartesian ($X$, $Y$, $Z$) to geodetic ($\varphi$, $\lambda$, $h$) transformation. Journal of Geodesy 82, 493–504.
Feltens, J., 2009. Vector methods to compute the Cartesian ($X$, $Y$, $Z$) to geodetic ($\varphi$, $\lambda$, $h$) transformation on a triaxial ellipsoid. Journal of Geodesy 83, 129–137.
Fok, H.S., Iz, H.B., 2003. A comparative analysis of the performance of iterative and non-iterative solutions to the Cartesian to geodetic coordinate transformation. Journal of Geospatial Engineering 5, 61–74.
Fukushima, T., 1999. Fast transform from geocentric to geodetic coordinates. Journal of Geodesy 73, 603–610.
Fukushima, T., 2006. Transformation from Cartesian to geodetic coordinates accelerated by Halley's method. Journal of Geodesy 79, 689–693.
Gerdan, G.P., Deakin, R.E., 1999. Transforming Cartesian coordinates to geographical coordinates. Australian Surveyor 44, 55–63.
Heiskanen, W.A., Moritz, H., 1967. Physical Geodesy. Freeman, San Francisco 364 pp.
Jones, G.C., 2002. New solutions for the geodetic coordinate transformation. Journal of Geodesy 76, 437–446.
Laskowski, P., 1991. Is Newton's iteration faster than simple iteration for transformation between geocentric and geodetic coordinates? Bulletin Géodésique 65 14–17.
Levin, J.Z., 1988. A rational parametric approach to latitude, longitude, and altitude. Navigation 35, 361–370.
Lin, K.C., Wang, J., 1995. Transformation from geocentric to geodetic coordinates using Newton's iteration. Bulletin Géodésique 69, 14–17.
Olson, D.K., 1996. Converting Earth-centered, Earth-fixed coordinates to geodetic coordinates. IEEE Transactions on Aerospace and Electronic Systems 32, 473–476.
Pollard, J., 2002. Iterative vector methods for computing geodetic latitude and height from rectangular coordinates. Journal of Geodesy 76, 36–40.
Turner, J.D., 2009. A non-iterative and non-singular perturbation solution for transforming Cartesian to geodetic coordinates. Journal of Geodesy 83, 139–145.
Vermeille, H., 2002. Direct transformation from geocentric coordinates to geodetic coordinates. Journal of Geodesy 76, 451–454.
Vermeille, H., 2004. Computing geodetic coordinates from geocentric coordinates. Journal of Geodesy 78, 94–95.
Wu, Y.X., Wang, P., Hu, X.P., 2003. Algorithm of Earth-centered, Earth-fixed coordinates to geodetic coordinates. IEEE Transactions on Aerospace and Electronic Systems 39, 1457–1461.
Yan, Q.J. (Ed.), 2006. Numerical Analysis third edn. BUAA Press, Beijing.
You, R.-J., 2000. Transformation of Cartesian to geodetic coordinates without iterations. Journal of Surveying Engineering 126, 1–7.
Zhang, C.-D., Hsu, H.T., Wu, X.P., Li, S.S., Wang, Q.B., Chai, H.Z., Du, L., 2005. An alternative algebraic algorithm to convert Cartesian to geodetic coordinates. Journal of Geodesy 79, 413–420.
Zhu, J., 1993. Exact conversion of Earth-centered, Earth-fixed coordinates to geodetic coordinates. Journal of Guidance, Control, and Dynamics 16, 389–391.
Zhu, J., 1994. Conversion of Earth-centered, Earth-fixed coordinates to geodetic coordinates. IEEE Transactions on Aerospace and Electronic Systems 30, 957–961.