Journal of Geodesy (1999) 73: 603–610

Journal of Geodesy

Fast transform from geocentric to geodetic coordinates

T. Fukushima

National Astronomical Observatory, 2-21-1, Ohsawa, Mitaka, Tokyo 181-8588, Japan
e-mail: toshio.fukushima@nao.ac.jp; Tel.: +81 422 41 3613; Fax: +81 422 41 3793

Received: 13 November 1998 / Accepted: 27 August 1999

Abstract. A new iterative procedure to transform geocentric rectangular coordinates to geodetic coordinates is derived. The procedure solves a modification of Borkowski's quartic equation by the Newton method from a set of stable starters. The new method runs a little faster than the single application of Bowring's formula, which has been known as the most efficient procedure. The new method is sufficiently precise because the resulting relative error is less than 10^{-15}, and this method is stable in the sense that the iteration converges for all coordinates including the near-geocenter region where Bowring's iterative method diverges and the near-polar axis region where Borkowski's non-iterative method suffers a loss of precision.

Key words. Coordinate transformation · Geocentric coordinates · Geodetic coordinates · Newton method

1 Introduction

The conversion from geocentric rectangular to geodetic coordinates is a basic but nontrivial problem encountered frequently in geodesy and positional astronomy. The existing methods to do it are classified into three categories; (1) the approximate series expansion (e.g. Taff 1985); (2) the numerical iterative procedure (e.g. Heiskanen and Moritz 1967); and (3) the closed analytical formula (e.g. Vanicek and Krakiwski 1982). In his review (Borkowski 1989), Borkowski presented two new methods, one iterative and one non-iterative, and claimed that they are superior to the existing methods in accuracy and/or simplicity. Unfortunately, this work lacks a comparison with the well-known Bowring's formula (Bowring 1976), which has been known as the most efficient method. In fact, Laskowski (1991) shows that Bowring's method, which he defined as the iteration of Bowring's formula twice at most, is 30% faster than the iterative method of Borkowski.

We have recently found a systematic way to construct stable and fast starters for the Newton method and applied it successfully in developing fast procedures for solving Kepler's equations (Fukushima 1997a, b, 1998). The last work, in particular, includes the accelerated solution of Barker's equation, which is a sort of cubic equation. As a natural extension, we applied the same approach to a modification of the quartic equation Borkowski adopted, and developed a new method to solve it. The new method is: (1) fast in the sense it runs a little faster than the single application of Bowring's formula (see Table 1); (2) precise because it achieves relative errors of order of 10^{-15} in the double-precision calculation, which means 10 nm on the Earth's surface; and (3) stable since it converges with any possible combination of input coordinates and the parameters of the reference ellipsoid.

It is noteworthy that the CPU time of the new method, which requires 3–4 iterations in average, is almost the same as that of Bowring's non-iterative method. This mainly comes from the fact that each iteration in the new method needs a small number of arithmetic operations and requires no call for transcendental functions.

2 Method

The core of the transformation from the geocentric rectangular coordinates (x, y, z) to the geodetic coordinates (φ, λ, h) is the conversion from (p, z) to (φ, h), where

$$p \equiv \sqrt{x^2 + y^2} \tag{1}$$

and p and z are restricted to be positive. The conversion consists of: (1) the solution of the latitude equation

$$p \sin\varphi - z \cos\varphi = e^2 N \sin\varphi \cos\varphi \tag{2}$$

with respect to φ or its equivalent; and (2) the computation of h from thus solved φ or others. Here,

Table 1. CPU time to transform (p, z) to (φ, h)

Method	Test set		
	A	B	C
Borkowski non-iterative	7.92	7.09	7.09
Bowring2	4.52	4.08	4.70
Bowring1	3.64	3.26	3.25
New	3.55	3.16	3.19

Note: shown are the averaged CPU times to transform a given pair of geocentric coordinates (p,z) to the corresponding geodetic coordinates (φ,h). They were evaluated by averaging the measurements for three sets of a few millions of equally spaced grid points of (φ,h). In all sets, the range of latitude is $0 < \varphi < 90°$. The ranges of height are (−6300 km, 30 000 km) for set A, (−10 km, 30 000 km) for set B, and (−10 km, 10 km) for set C. The times used for preparing the constants such as $e' = \sqrt{1-e^2}$ or $c \equiv ae^2$ are excluded from the measurements. However, the time used for selecting the starter and computing it is included. The unit CPU time is μs for the Intel Pentium II 450 MHz processor. The programming language used in the measurements was Microsoft Fortran Powerstation 4.00. Compared methods are: (1) the single application of Bowring's formula (Bowring 1976), which we denote Bowring1; (2) the iteration of Bowring's formula twice at most (Laskowski 1991), which we denote Bowring2; (3) Borkowski's non-iterative method (Borkowski 1989); and (4) our new method

$$N = \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}} \tag{3}$$

and a and e are the semi-major axis and the eccentricity of the reference ellipsoid. In his non-iterative method, Borkowski (1989) adopted as the equivalent the following quartic equation:

$$t^4 + 2Et^3 + 2Ft - 1 = 0 \tag{4}$$

with respect to the variable

$$t \equiv \tan\left(\frac{\pi}{4} - \frac{\psi}{2}\right) = \tan\left[\frac{1}{2}\tan^{-1}\left(\frac{1}{e' \tan\varphi}\right)\right] \tag{5}$$

Here, ψ is the reduced (or parametric) latitude and

$$e' \equiv \sqrt{1-e^2}, \quad E \equiv \frac{z'-c}{p}, \quad F \equiv \frac{z'+c}{p}, \quad z' \equiv e'z, \quad c \equiv ae^2 \tag{6}$$

Borkowski's non-iterative method is to solve this quartic equation by Ferrari's method. We note that the above equation is ill-represented when p is small. In fact, Fig. 1 shows that the error of Borkowski's non-iterative method grows significantly when the co-latitude reduces to zero.

To avoid this singularity, we rewrite the equation as

$$f(t) \equiv pt^4 + ut^3 + vt - p = 0 \tag{7}$$

where

$$u \equiv 2pE = 2(z' - c), \quad v \equiv 2pF = 2(z' + c) \tag{8}$$

It is easy to show that f has only one root in the solution interval (0,1); see Appendix D. We solve the equation by the Newton method, details of which are given in Appendix A. Its key part is the selection of starting values, which will be described in the next section. After the solution t is obtained, the geodetic coordinates are computed as

$$\varphi = \mathrm{atan2}\left(1 - t^2, 2e't\right), \quad h = \frac{2pe't + z\left(1-t^2\right) - ae'\left(1+t^2\right)}{\sqrt{\left(1+t^2\right)^2 - 4e^2t^2}} \tag{9}$$

Note that only one call of `sqrt` is required in the evaluation of h. This contributes to the acceleration of the current procedure.

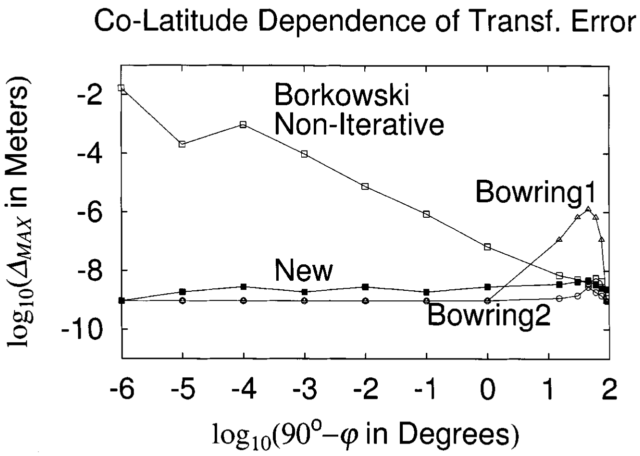

Fig. 1. Co-latitude dependence of transformation error. This figure shows the dependency of maximum of transformation error on the co-latitude. The methods compared are: the single application of Bowring's formula (Bowring1); the iteration of Bowring's formula twice at most (Bowring2); Borkowski's non-iterative method; and our method. The errors shown are the maximum of difference between the input pair of (p,z) and that computed from (φ,h) solved, where the input pairs are prepared such that the corresponding φ is fixed and h is varied uniformly in the range $|h| < 10$ km with a separation of 1 m. The parameters of reference ellipsoid were set as those of the GRS 1980 system; namely $a = 6\,378\,137$ m and $f = 1/298.257222101$. The computer used was a Windows 98 PC with an Intel Pentium II processor. The computer language used was Microsoft Fortran Powerstation ver. 4

3 Starter

In order to find a stable starter for the Newton method to solve Eq. (7), we examine the first and second derivatives of the function f

$$f'(t) = 4pt^3 + 3ut^2 + v \tag{10}$$

and

$$f''(t) = 12pt^2 + 6ut \tag{11}$$

Note that the zeros of $f''(t)$ are explicitly given as

$$t = 0, \quad t = t_M \equiv \frac{-u}{2p} = \frac{c - z'}{p} \tag{12}$$

Depending on the location of t_M with respect to the solution interval (0,1), we separate the procedure into the following three cases.

3.1 Case 1: $t_M \le 0$

The corresponding region in (p, z) is quite large; $z' \ge c$. Thus most (more than 99%) practical problems fall into this case. In this case, $f''(t)$ is always positive in the interval (0,1). Thus $f'(t)$ is monotonically increasing there. Since $f'(0) = v > 0$, $f'(t)$ is always positive in that interval (see Fig. 2). Therefore, an upper bound of the solution is a stable starter in this case. Note that the CPU time of each Newton iteration is relatively small since the function is a quartic polynomial. It is better to select a simple starter rather than a more precise but complicated one. Therefore, we adopt the upper bound of the solution interval, $t = 1$, as the starter. In the actual implementation, we start from the point Newton-mapped once

$$t_1 \equiv f^*(1) = \frac{4p + 2u}{4p + 3u + v} = \frac{p - c + z'}{p - c + 2z'} \tag{13}$$

3.2 Case 2: $t_M \ge 1$

The corresponding region in (p, z') is a neighborhood of the geocenter; $p + z' \le c$. Practically, this case will be met very rarely. In this case, $f''(t)$ is always negative in the interval (0,1). Thus $f'(t)$ is monotonically decreasing there. Since $f'(0) = v > 0$, $f'(t)$ can be negative in that interval (see Fig. 3). Therefore, we adopt the lower bound of the interval, $t = 0$, as the starter. In the actual implementation, we start from the point Newton-mapped once

$$t_0 \equiv f^*(0) = \frac{p}{v} = \frac{p}{z' + c} \tag{14}$$

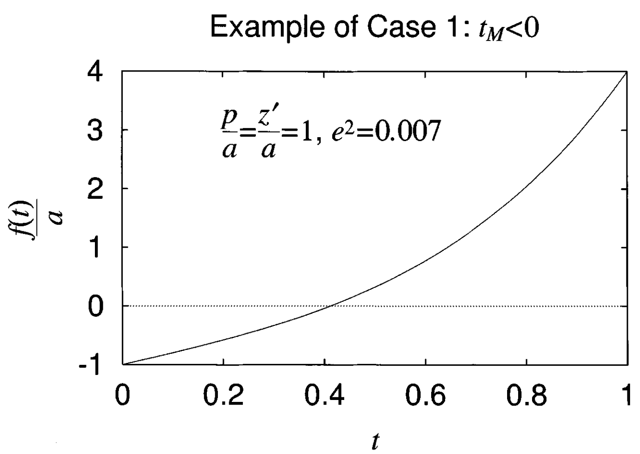

Fig. 2. Example of case 1: $t_M \le 0$. Shown is the function $f(t) = pt^4 + ut^3 + vt - p$ for an example of case 1, a case set in selecting the starter to find the root of $f(t)$ by the Newton method. The input coordinates are $p/a = z'/a = 1$ while the parameter is $e^2 = 0.007$. In this case, $t_M = (c - z')/p = -0.993 < 0$

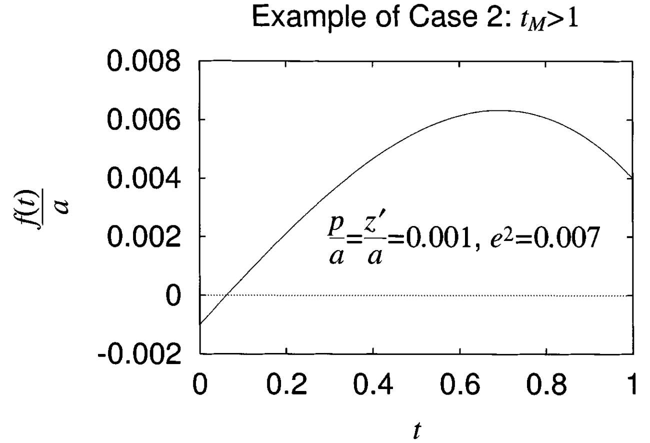

Fig. 3. Example of case 2: $t_M \ge 1$. Same as Fig. 2 but as an example of case 2. The input coordinates are $p/a = z'/a = 0.001$. In this case, $t_M = 6 > 0$

3.3 Case 3: $0 < t_M < 1$

The corresponding region in (p, z') is a thin layer on the equatorial plane with a small hollow at the center: $c - p < z' < c$. The probability of this case is small (less than 1%) but finite. In this case, we evaluate $f_M \equiv f(t_M)$. And we separate the procedure into the following two subcases depending on its signature.

3.3.1 Case 3a: $f_M \ge 0$

In this subcase, the solution locates in the subinterval $(0, t_M)$ (see Fig. 4). Since $f''(t)$ is nonpositive there, we adopt the lower bound t_0 as the starter as in case 2.

3.3.2 Case 3b: $f_M < 0$

In this subcase, the solution locates in the subinterval $(t_M, 1)$ (see Fig. 5). Since $f''(t)$ is positive there, we adopt the upper bound t_1 as the starter as in case 1.

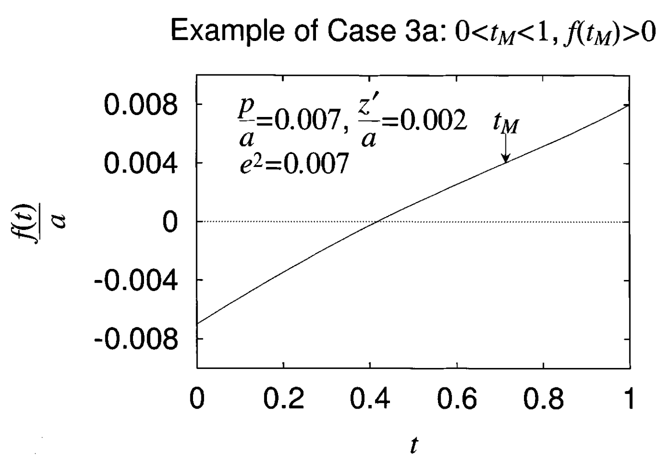

Fig. 4. Example of case 3a: $0 < t_M < 1$, $f(t_M) > 0$. Same as Fig. 2 but as an example of case 3a. The input coordinates are $p/a = 0.007$, $z'/a = 0.002$. In this case, $t_M \sim 0.714$ and $f(t_M)/a \sim 0.004 > 0$

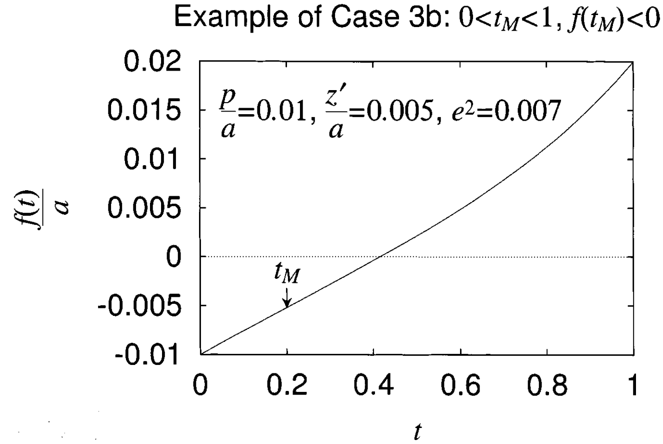

Fig. 5. Example of case 3b: $0 < t_M < 1$, $f(t_M) < 0$. Same as Fig. 2 but as an example of case 3b. The input coordinates are $p/a = 0.01$, $z'/a = 0.005$. In this case, $t_M = 0.2$ and $f(t_M)/a \sim -0.0052 < 0$

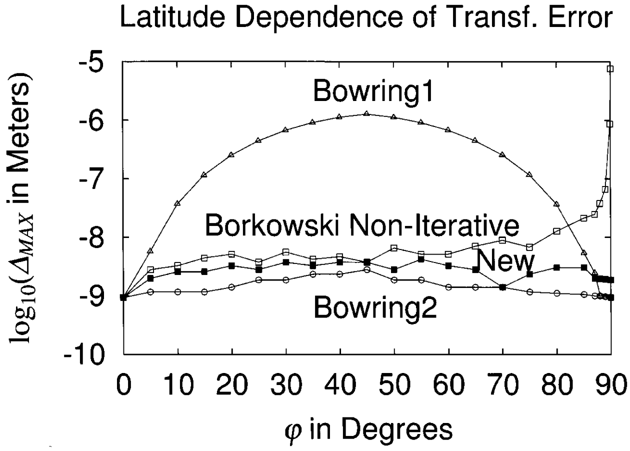

Fig. 6. Latitude dependence of transformation error. Same as Fig. 1 but plotted in the linear scale of the latitude

4 Numerical comparisons

In order to evaluate our method, we did a series of test conversions from (p, z) to (φ, h) for millions of grid points of (p, z) covering the area of the Moon's orbit and including the geocenter, the equatorial plane, and the polar axis, where some methods are unstable and/or less precise. For the parameters of reference ellipsoid, a and e, we adopted those of the GRS 1980 system. The tests were also done for some existing methods: (1) the single application (or non-iterative use) of Bowring's formula (Bowring 1976), which we denote Bowring1; (2) the iteration of the same Bowring formula twice at most (Laskowski 1991), which we denote Bowring2; and (3) Borkowski's non-iterative method (Borkowski 1989).

First, let us discuss the precision and the stability. Figure 6 shows the latitude dependence of the transformation errors of these four methods for the Earth-based coordinates. Here, we define the error as

$$\Delta \equiv |p - p^*| + |z - z^*| \tag{15}$$

where (p, z) are the input pair of geocentric coordinates and (p^*, z^*) are those computed from the transformed geodetic coordinates (φ^*, h^*). In preparing each input pair in Fig. 6, we fixed the corresponding latitude φ and varied the corresponding height h uniformly in the range $|h| \leq 10$ km, and took the maximum of the errors obtained. It is clear that, for the Earth-bound transformation, even a single application of Bowring's formula is sufficiently precise, producing an error of a few μm at most. Note that Borkowski's method suffers a significant loss of precision near the polar axis (refer to Fig. 1). For the other two methods, the magnitude of transformation errors is negligibly small, of nm order, and is almost independent of the latitude.

However, the dependence with respect to h is strong (see Fig. 7); this was prepared similarly to Fig. 6 but the errors shown were the maximum of those for around 10 000 uniformly separated grid points of φ in the range $0 < \varphi < 90°$ while h was fixed. Note that Bowring's formula diverges in some region near the geocenter. See Appendix B for further discussion. Except near this unstable region, at most two iterations of Bowring's formula are sufficiently precise for all other elevations, say $h > -6300$ km; Borkowski's non-iterative method fails both near the geocenter and in the region with large h, say $h > 10^5$ km. There a naive application of Ferrari's formula, such as the realization of cubic root by the combination of logarithmic and exponential functions, confronts with the overflow of these functions. On the other hand, the new method keeps the error as low as 10 nm from the geocenter up to the global positioning system (GPS) orbit. Then the error grows linearly with respect to the geocentric distance due to round-off errors, which the iterative use of Bowring's formula also suffers.

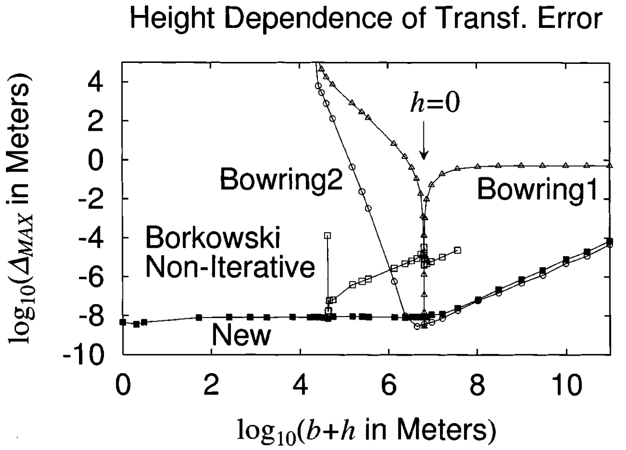

Fig. 7. Height dependence of transformation error. Same as Fig. 1 but shown as functions of h while the error was taken as the maximum for $0 < \varphi < 90°$

Next, compare the speed. Table 1 illustrates the comparison of averaged CPU times of these four methods. In measuring the CPU times, we included the time used for selecting the starter and computing it. In

Table 1, the results are shown for three different sets of test coordinates; set A for almost all possible cases[1], set B for all practical cases, and set C for all Earth-bound cases. Here, we used a fast implementation for Bowring's formula, details of which are given in Appendix C. For all sets, the new method runs a little faster than the single application of Bowring's formula. This is a remarkable result in view of the fact that the average number of iterations is 3 to 4 in the new method.

This is mainly because the new method requires fewer calls of transcendental functions than does the fast implementation of Bowring's formula. Actually, the new method needs only two such calls: (1) no calls of transcendental functions in preparing the starter; (2) no calls in evaluating f and its derivative in the iteration; and (3) one call of `sqrt` and one call of `atan2` in computing φ and h (see also Sect. 3 and Appendix A). On the other hand, the single application of Bowring's method deploys four calls: (1) no calls of transcendental functions in preparing the starter; (2) one call of `sqrt` in the formula; and (3) two calls of `sqrt` and one call of `atan2` in computing φ and h (see details in Appendix C; also refer to Table C1). Judging from Table 1, we conclude that the effect of this difference in the number of transcendental functions called between the new method and Bowring's formula outweighs the effects of the difference in the number of iterations and of the difference in the complexity of the algorithms.

5 Conclusion

A new method to convert geocentric rectangular coordinates to geodetic ones has been created. It runs a little faster than the single application of Bowring's formula for a PC with a Pentium II processor[2]. It is sufficiently precise because its error is of order of 10 nm in the neighborhood of the Earth, and it is stable since it converges for all given coordinates, including those in the near-geocenter area where Bowring's formula diverges and those in the near-polar axis region where Borkowski's non-iterative method suffers a significant loss of precision.

Since the algorithm of the new method is designed independently of the magnitude of eccentricity, it is applicable to any kind of reference ellipsoid. The $f77$ routine of the new method is available from the author.

Acknowledgements. The author thanks the referees for providing information on Bowring's formula and its implementations and for their valuable suggestions to improve the quality of this paper.

[1] We must limit the lower and upper bounds since some of the methods diverge and/or cause execution error beyond these limitations.

[2] It should be noted that this type of comparison is dependent on the nature of the computer's architecture.

Appendix A

Newton method

We use the Newton method to solve our quartic equation, Eq. (7). The Newton method is to iterate the mapping

$$t \to f^*(t) \equiv t + \Delta t,$$

$$\Delta t \equiv \frac{-f(t)}{f'(t)} = \frac{p - \left(pt^4 + ut^3 + vt\right)}{4pt^3 + 3ut^2 + v} \tag{A1}$$

until convergence. The reason why we adopt the quartic equation as the form of latitude equation to be solved is that the expression of $f(t)$ and $f'(t)$ contains no transcendental functions, and therefore, can be evaluated rapidly. Now, the key point of the Newton method is to provide a starter which is stable, precise, *and* fast. Namely, a good starter is to be such that: (1) it makes the following Newton iterations to convergence for any possible combination of parameters, which is the meaning of *stability*; (2) the obtained solution must be sufficiently precise; and (3) the convergence must be fast. Among these the stability is the most important property.

Let us present an example of a stable starter; see also Fukushima (1997a). Assume that we look for a root of a real continuous function $F(x)$ in the given interval (x_L, x_R) and the function satisfies the following conditions:

$$F(x_L) < 0, \quad F(x_R) > 0 \tag{A2}$$

and

$$F'(x) > 0, \quad F''(x) > 0, \quad (x_L \le x \le x_R) \tag{A3}$$

Namely, the function is concave and monotonically increasing in the interval, and therefore, the function has only one root there and the root is simple. Consider the associated Newton mapping

$$x \to F^*(x) \equiv x - \frac{F(x)}{F'(x)} \tag{A4}$$

Once the mapping converges, the converged point coincides with the root of $F(x)$. In fact

$$F^*(x) = x \Leftrightarrow -\frac{F(x)}{F'(x)} = 0 \Leftrightarrow F(x) = 0 \tag{A5}$$

since F' remains positive and finite in the interval given.

Now, it is easy to show that the initial guess $x = x_R$ or any other upper bound of the solution makes the Newton iteration converge. Let us prove that the associated Newton mapping is diminishing if approaching from an upper bound of the solution.

If we denote the solution by x^*, then the starter x must satisfy $x > x^*$. Then $F(x) > 0$ since F is monotonically increasing. Therefore

$$F^*(x) = x - \frac{F(x)}{F'(x)} < x \tag{A6}$$

since both $F(x)$ and $F'(x)$ are positive. On the other hand, we can rewrite $F(x^*)$ by using the first and second derivatives as

$$0 = F(x^*) = F(x) + F'(x)(x^* - x) + \tfrac{1}{2}F''(x + \theta(x^* - x))(x^* - x)^2 \tag{A7}$$

where $0 \le \theta \le 1$. Since $F'' > 0$, we learn that

$$F(x) + F'(x)(x^* - x) < 0 \tag{A8}$$

Therefore

$$F^*(x) - x^* = x - x^* - \frac{F(x)}{F'(x)} = \frac{F''(x + \theta(x^* - x))}{2F'(x)}(x^* - x)^2 > 0 \tag{A9}$$

Thus

$$x^* < F^*(x) < x \tag{A10}$$

which means the mapping is diminishing. Also, the above expression shows that the convergence is quadratic since

$$|F^*(x) - x^*| \le M(x^* - x)^2 \tag{A11}$$

where

$$M = \max_{x_L \le x \le x_R} \frac{F''(x)}{2F'(x)} \tag{A12}$$

is a certain numerical constant. As long as neither $F'(x)$ nor $F''(x)$ change their signs in the interval considered, a similar starter can be constructed by changing the signs of x and F' appropriately. Thus, the key point in preparing a stable starter is, before the Newton iteration, to bracket the solution interval so as to satisfy such properties.

Appendix B

Stability of Bowring's formula

In the conclusion of his article, Laskowski (1991) considered that Bowring implicitly incorporated a single step of Newton's iteration into his final formula. In fact, Bowring (1976, p. 325) actually reached his formula by applying the Newton method. Therefore, Bowring's formula is nothing other than a Newton iteration and, therefore, is of quadratic convergence as long as the starter adopted is stable. Let us illustrate this more clearly.

If we introduce a new variable

$$T \equiv \tan\psi, \quad 0 \le T < \infty \tag{B1}$$

the latitude equation, Eq. (2), is rewritten as

$$g(T) \equiv pT - \frac{cT}{\sqrt{1+T^2}} - z' = 0 \tag{B2}$$

where we used the relations

$$N\cos\varphi = a\cos\psi, \quad N\left(1 - e^2\right)\sin\varphi = b\sin\psi \tag{B3}$$

It is easy to show that this equation has only one solution; see Appendix D. The derivative of g becomes

$$g'(T) = p - \frac{c}{\left(\sqrt{1+T^2}\right)^3} \tag{B4}$$

If we apply the Newton method in solving the rewritten equation, the corresponding Newton mapping becomes

$$T \to g^*(T) \equiv T - \frac{g(T)}{g'(T)} = \frac{z' + cS^3}{p - cC^3} \tag{B5}$$

where

$$C \equiv \cos\psi = \frac{1}{\sqrt{1+T^2}}, \quad S \equiv \sin\psi = CT \tag{B6}$$

This is nothing other than Bowring's formula.

Now, we have proved that Bowring's formula is the Newton method to solve a variant of latitude equation $g(T) = 0$. Its stability can be discussed similarly as done for the new method; see Appendix A. To do this, we evaluate the second derivative of g as

$$g''(T) = \frac{c}{\left(\sqrt{1+T^2}\right)^5} \tag{B7}$$

Since g'' is non-negative in the domain $0 \le T < \infty$, starting from an upper bound assures stability. Unfortunately, the starter of Bowring's formula

$$T_B = \frac{z}{e'p} \tag{B8}$$

is neither a lower nor an upper bound. In fact

$$g(T_B) = \frac{e^2 z}{e'} - \frac{cT_B}{\sqrt{1+T_B^2}} = \frac{cz}{\sqrt{(e'p)^2 + z^2}}\left(\sqrt{\frac{p^2}{a^2} + \frac{z^2}{b^2}} - 1\right) \tag{B9}$$

Namely, $g(T_B) > 0$ when $h > 0$ and $g(T_B) < 0$ when $h < 0$. This is because T_B is correct when $h = 0$. Therefore, Bowring's formula can be unstable when $h < 0$. For example, $g'(T_B)$ can be zero or negative. In that case, the Newton-mapped point, $g^*(T_B)$, goes out of the solution interval and, therefore, causes a divergence. The region of (p, z) leading to such instability is evaluated by solving the inequality $g'(T_B) \le 0$, which turns out as

$$z \le e'\sqrt[3]{p^2}\sqrt{\sqrt[3]{c^2} - \sqrt[3]{p^2}} \tag{B10}$$

See Fig. B1 for the critical curve separating the stable and unstable regions.

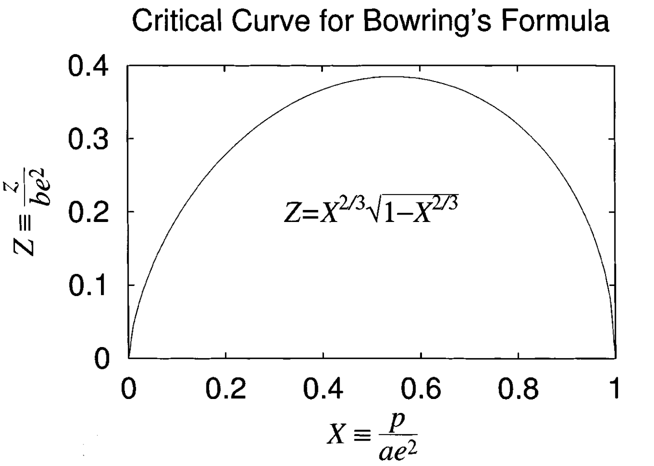

Fig. B1. Critical curve of Bowring's formula. Shown is the curve separating the stable and unstable regions for Bowring's formula

Table C1. CPU time of basic operations for Intel Pentium II

Operation	Single	Double
$x = y + z$	0.36	1.00
$x = y * z$	0.75	1.00
$x = y/z$	11.3	11.7
$x = \mathrm{sqrt}(y)$	31.9	31.9
$x = \log(y)$	39.4	39.0
$x = \sin(y)$	40.2	40.0
$x = \tan(y)$	48.0	47.9
$x = \mathrm{atan}(y)$	48.8	48.7
$x = \exp(y)$	57.3	56.9

Note: the CPU time of major operations are shown for the Intel Pentium II processor. The unit is dflops, the simple mean of the addition (the second row) and the multiplication (the third row) of double precision operation. The actual time of dflops is 0.0127 μs for the Intel Pentium II 450 MHz. The results for single and double precisions are listed in parallel. The programming language used in measurements was Microsoft Fortran Powerstation 4.00

Appendix C

Fast implementation of Bowring's formula

Naive implementations of Bowring's formula cause a severe slow-down of the conversion. If we introduce

$$\varepsilon \equiv \frac{a^2 - b^2}{b^2} = \frac{e^2}{1 - e^2} \tag{C1}$$

then, one such example is to iterate

$$\psi \leftarrow \tan^{-1}\left(\frac{e'(z + b\varepsilon \sin^3 \psi)}{p - c\cos^3 \psi}\right) \tag{C2}$$

from the starter

$$\psi_B = \tan^{-1}\left(\frac{z}{e'p}\right) \tag{C3}$$

and to finish the conversion by

$$\varphi = \tan^{-1}\left(\frac{\tan \psi}{e'}\right) \tag{C4}$$

and if $p > z$ then

$$h = \frac{p}{\cos \varphi} - \frac{a}{\sqrt{1 - e^2 \sin^2 \varphi}} \tag{C5}$$

else

$$h = \frac{z}{\sin \varphi} - \frac{a(1 - e^2)}{\sqrt{1 - e^2 \sin^2 \varphi}} \tag{C6}$$

Even for a single application (or non-iterative usage) of Bowring's formula, this implementation contains eight or nine calls of transcendental functions; three calls of `atan`, one call of `tan`, three or four calls of `sin` and/or `cos`, and one call of `sqrt`. Further, each additional iteration costs three more calls; one call of `atan` and two calls of `sin` and/or `cos`.

The call of any transcendental function costs many, say 30 or 60, additions and/or multiplications, in current CPUs such as the Intel Pentium II; see Table C1. Thus, one should minimize the number of such calls as much as possible in order to realize a rapid implementation.

One example[3] would be to iterate

$$T \leftarrow \frac{z' + cS^3}{p - cC^3} \tag{C7}$$

from the starter

$$T_B = \frac{z}{e'p} \tag{C8}$$

where

$$C = \frac{1}{\sqrt{1 + T^2}}, \quad S = CT \tag{C9}$$

and to finish the conversion by

$$\varphi = \tan^{-1}\left(\frac{T}{e'}\right) \tag{C10}$$

and if $p > z$ then

$$h = \frac{\sqrt{(1 - e^2) + T^2}}{e'}\left(p - \frac{a}{\sqrt{1 + T^2}}\right) \tag{C11}$$

else

$$h = \sqrt{(1 - e^2) + T^2}\left(\frac{z}{T} - \frac{b}{\sqrt{1 + T^2}}\right) \tag{C12}$$

[3] Bowring's original implementation is close to this except that he computes S from C by calling sqrt as $S = \sqrt{1 - C^2}$. Thus his implementation is a little slower than the current one.

Then, for the single application, the number of calls of transcendental functions reduces to 4; three calls of `sqrt` and one call of `atan`. And each additional iteration costs only one more call of `sqrt`. We used this fast implementation of Bowring's formula in the comparison we conducted.

Appendix D

Uniqueness of solution of latitude equation

Here we will prove that the latitude equation has only one solution. To do this simply, we begin with its expression as it appeared in deriving Bowring's formula, Eq. (B2), which we quote here as

$$g(T) \equiv pT - \frac{cT}{\sqrt{1+T^2}} - z' = 0 \qquad \text{(D1)}$$

We exclude trivial cases; $T = 0$ when $z' = 0$, and $T = \infty$ when $p = 0$. Then the solution interval becomes

$$0 < T < \infty \qquad \text{(D2)}$$

It is easy to show that g has at least one root there. In fact

$$g(0) = -z' < 0,$$
$$\lim_{T\to\infty} g(T) \sim pT - (z' + c) \to +\infty > 0 \qquad \text{(D3)}$$

Consider its uniqueness. As we see in Appendix B, g'' is positive. Thus, g' has at most one root in the solution interval. Namely, g has at most one local optimum there. Thus, in the following, we separate the problem into two cases depending on the number of optima.

Case A: no optimum
If g' has no root in the solution interval, it is always positive there since it becomes positive for sufficiently large T as we saw in the above. Then g has only one root in the solution interval.

Case B: single optimum
If g has one local optimum in the solution interval, this optimum is actually a minimum since g'' is positive. Name the root of g' as T_0. Then, g' is negative in the subinterval $(0, T_0)$. Since $g(0) = -z'$ is negative, g is always negative in this subinterval, and therefore it has no root there. Thus the root(s) of g must exist in the other subinterval (T_0, ∞). Since g' is positive there, g has only one root.

Thus it has been proved that g has only one root in the solution interval. This result is easily applied to the original latitude equation, Eq. (2), or to the quartic equation, Eq. (7), because the variable transformations among them are one-to-one.

References

Borkowski KM (1989) Accurate algorithms to transform geocentric to geodetic coordinates. Bull Géod 63: 50–56

Bowring BR (1976) Transformation from spatial to geographical coordinates. Sur Rev XXIII: 181, 323–327

Fukushima T (1997a) A method solving Kepler's equation without transcendental function evaluation. Celest Mech Dynam Astron 66: 309–319

Fukushima T (1997b) A method solving Kepler's equation for hyperbolic case. Celest Mech Dynam Astron 68: 121–137

Fukushima T (1998) A fast procedure solving Gauss' form of Kepler's equation. Celest Mech Dynam Astron 70: 115–130

Heiskanen WA, Moritz H (1967) Physical geodesy. WH Freeman, New York

Laskowski P (1991) Is Newton's iteration faster than simple iteration for transformation between geocentric and geodetic coordinates? Bull Géod 65: 14–17

Taff LG (1985) Celestial mechanics. A computational guide for the practitioner. Wiley, New York

Vanicek P, Krakiwski EJ (1982) Geodesy: the concepts. North Holland, Amsterdam