

Article

Vector-Algebra Algorithms to Draw the Curve of Alignment, the Great Ellipse, the Normal Section, and the Loxodrome

Thomas H. Meyer 

Department of Natural Resources and the Environment, University of Connecticut, 1376 Storrs Rd., U4087, Storrs, CT 06269, USA; thomas.meyer@uconn.edu

Abstract: This paper recasts four geodetic curves—the great ellipse, the normal section, the loxodrome, and the curve of alignment—into a parametric form of vector-algebra formula. These formulas allow these curves to be drawn using simple, efficient, and robust algorithms. The curve of alignment, which seems to be quite obscure, ought not to be. Like the great ellipse and the loxodrome, and unlike the normal section, the curve of alignment from point **A** to point **B** (both on the same ellipsoid) is the same as the curve of alignment from point **B** to point **A**. The algorithm used to draw the curve of alignment is much simpler than any of the others and its shape is quite similar to that of the geodesic, which suggests it would be a practical surrogate when drawing these curves.

Keywords: geometric geodesy; curve of alignment; normal section; geodesic; loxodrome; great ellipse; algorithms



Citation: Meyer, T.H. Vector-Algebra Algorithms to Draw the Curve of Alignment, the Great Ellipse, the Normal Section, and the Loxodrome. *Geomatics* **2024**, *4*, 138–148. <https://doi.org/10.3390/geomatics4020008>

Academic Editor: Filiberto Chiabrando

Received: 29 March 2024

Revised: 29 April 2024

Accepted: 2 May 2024

Published: 8 May 2024



Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In a geographic information system (GIS), linear and polygonal features are usually defined by only their vertices, so the programmer is assigned the task of deciding which curve to use to connect the points; that is, the task of defining the implied edge [1]. If the edge is to be a straight line in “the real world”, then it could be found conceptually by stretching a chalk line between the endpoints and snapping it down, leaving the trace of the curve on the ground—or, for geomatics, on the reference ellipsoid that is used by the geodetic datum for the mapping. Such a curve is called a geodesic, being “the curve that does not veer to the left nor to the right”. Then, the geodesic would be projected onto the map using the forward map-projection formulas.

The formulas for calculating a geodesic curve on an ellipsoid (hereafter just called a geodesic) are very complicated [2–8], and there are many special cases. For example, there are four geodesics connecting any two antipodal points on the Equator: two geodesics following the Equator itself, one to the east and the other to the west, and two others following the northwards and southwards meridians connecting the points. There are other peculiar cases as well, such as the “lift-off” condition [9] for which two points on the Equator are connected by either one or three geodesics depending on how far apart they are. Since there is not always exactly one geodesic connecting two points, and given the programming challenge of implementing them, a programmer might want to consider other curves that are very nearly geodesics but simpler to implement: normal sections, great ellipses, loxodromes, and curves of alignment.

This paper presents a single, unified approach for drawing these curves that is based on a parametric representation using vector algebra. (It would be ideal to provide an algorithm for the geodesic as well; however, there does not appear to be a way to implement general geodesics with this approach.) Parametric solutions have already been found for the great ellipse [10,11], but these approaches use geodetic coordinates for their parameters, which is very helpful for navigation but complicates the formulas if the goal is only to draw the curve. The direct and indirect problems for loxodromes were provided by Meyer

and Rollins [12], but their formulations were also parameterized by geodetic coordinates, resulting in similarly complicated formulas [3] and Deakin [13] report formulas for points along a curve of alignment. Testing revealed that the method below produces the same points as Deakin [13], and the method below is simpler, in the opinion of the author.

Notice that drawing these curves between two points is a separate problem from the typical “direct” and “indirect” problems of geodesy. The former takes a point, an azimuth, and a distance, and the implied forward point is to be determined. The latter takes two points, and the forward and backwards azimuths, and their separation is to be determined. One can use the indirect problem to find the forward azimuth and the separation between the points, and then use the direct problem’s solution parametrically as a function of distance to enumerate the points along the curve to be drawn.

The terminology and notation for the remainder of the paper are now given. The lengths of an ellipsoid’s semi-major and semi-minor axes are a and b , and they are related by the first eccentricity as $b^2 = (1 - e^2) a^2$. The geodetic latitude is $-\pi/2 \leq \phi \leq \pi/2$, longitude is $-\pi \leq \lambda \leq \pi$, and geodetic height is h . See Appendix A for other formulas. The geocentric Cartesian coordinate axes are X , Y , and Z , with Z being the Z -axis of the reference ellipsoid. The X - and Y -axes are perpendicular to Z and to each other, and they also define the Equatorial Plane. The Prime Meridian is the section formed by the X - Z plane. (The reader will recall that a section of an ellipsoid is the curve formed by the intersection of a plane and the ellipsoid.)

The general idea of the algorithms below is to produce a set of XYZ points along the straight-line segment (i.e., the chord) connecting the endpoints of the curve to be drawn. There are simple relationships between these points, which are in the interior of the ellipsoid, and their images on the ellipsoid. The relationship depends upon which curve is desired. An XYZ point, or vector, will be denoted by $\mathbf{P} = (X_p, Y_p, Z_p)$. A point in geodetic longitude, latitude, and height (LBH) will be denoted by $\mathbf{P} = (\lambda_{\bar{P}}, \phi_p, h_p)$.

2. Curve of Alignment

The curve of alignment can be visualized in this way (c.f. [13] (Deakin, 2009)). Consider two points of interest, \mathbf{A} and \mathbf{B} , on the ellipsoid, and imagine connecting them with a straight line that penetrates through the ellipsoid. Now, imagine another line segment that is perpendicular to the connecting line and that rises outwards, emerging perpendicularly (normal) to the ellipsoid. This rising line is free to slide along the connecting line and it remains normal to the ellipsoid as it moves. The curve of alignment is the locus of points at the intersection of the rising line and the near side of the ellipsoid.

2.1. Properties

The curve of alignment is undefined if \mathbf{A} and \mathbf{B} are antipodal.

Since \mathbf{A} and \mathbf{B} are on an ellipsoid, their curve of alignment spans less than 180° of longitude.

The curve of alignment from \mathbf{A} to \mathbf{B} is the same as the curve of alignment from \mathbf{B} to \mathbf{A} .

Any portion less than 180° of the Equator is a curve of alignment and is also a geodesic, as is any such portion of a meridian. A portion of any parallel other than the Equator is not a curve of alignment. (The curve of alignment connecting two points at the same latitude will pass polewards of their common parallel; see Figure 1).

The curve of alignment is not in general a planar curve. In fact, it is the intersection of a hyperbolic paraboloid and an ellipsoid [3], p. 67.

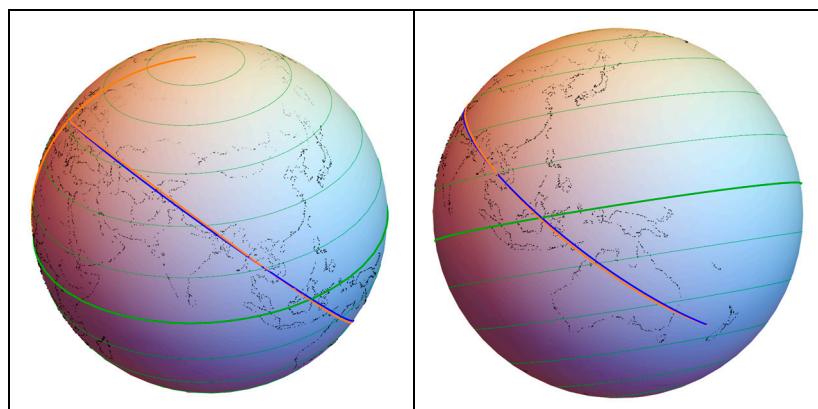


Figure 1. The curve of alignment (orange) and the geodesic (blue) from $(165^\circ \text{ E}, 40^\circ \text{ S})$ and $(0^\circ \text{ E}, 45^\circ \text{ N})$ on the WGS 84 reference ellipsoid. The Equator appears in green and the Prime Meridian in orange.

2.2. Algorithm

The algorithm used to draw curve of alignment is based on geodetic coordinate-system conversions. The algorithm depends on the idea that changing a point's geodetic height only—not geodetic longitude or latitude—moves the point in a direction normal to the ellipsoid. The connecting line segment is given by the difference of \mathbf{A} and \mathbf{B} in geocentric Cartesian coordinates, and we formulate this segment parametrically. The rising segment is found by converting a point on the connecting segment from XYZ into geodetic longitude, latitude, and height, and then setting the geodetic height to zero. This determines a point on the ellipsoid as the other endpoint of the rising segment such that the rising segment is perpendicular to the ellipsoid. The following six steps make this algorithm explicit.

1. Some ellipsoid is specified, such as a geodetic reference ellipsoid. This provides the length of the semi-major axis a and the value of the (first) eccentricity squared e^2 . (See Appendix A for the formulas and parameter values.)
2. Two points of interest, \mathbf{A} and \mathbf{B} , not antipodal and on the ellipsoid are chosen. (Note: \mathbf{A} and \mathbf{B} are on the ellipsoid so their geodetic heights are zero.) Their coordinates are required in geocentric Cartesian coordinates: $\mathbf{A} = (X_A, Y_A, Z_A)$ and $\mathbf{B} = (X_B, Y_B, Z_B)$. If given in geodetic longitude and latitude, they can be readily converted to geocentric (see Appendix A).
3. Compute the vector $\mathbf{v} = \mathbf{B} - \mathbf{A}$.
4. The points of the straight-line segment from \mathbf{A} to \mathbf{B} are given in parametric form by $\mathbf{p}(t) = \mathbf{A} + t \mathbf{v}$ for $0 \leq t \leq 1$.
5. Denote the conversion from geocentric Cartesian coordinates to geodetic longitude, latitude, and height by γ . Then, $\gamma(\mathbf{p}) = (\lambda, \phi, h)$. This conversion depends on a and e^2 . Define a variant of γ called $\bar{\gamma}$ such that $\bar{\gamma}(\mathbf{p}) = (\lambda, \phi, 0)$, where, for some h , $\gamma(\mathbf{p}) = (\lambda, \phi, h)$, i.e., $\bar{\gamma}(\mathbf{p})$ returns the same as $\gamma(\mathbf{p})$ but with h set to zero.
6. The curve of alignment is the curve $\bar{\gamma}(\mathbf{p}(t))$ for $0 \leq t \leq 1$. In other words, the curve of alignment is the set of geodetic points given by the conversion of all the points $\mathbf{p}(t)$ from geocentric Cartesian to geodetic, setting their geodetic heights to zero.

An implementation in *Mathematica*TM produced the image in Figure 1, which shows the curve of alignment (red) and the geodesic (green) from $(0^\circ \text{ E}, 45^\circ \text{ N})$ to $(165^\circ \text{ E}, 40^\circ \text{ S})$ on the WGS 84 reference ellipsoid. The arc length of the geodesic is 18,669,335.84 m, and the arc length of the curve of alignment is 18,671,843.56 m, which is 2507.72 m (134 ppm) longer. The arc length of the geodesic was computed using [5], and the arc length of the curve of alignment was computed by recursively bisecting the curve into linear segments and summing the lengths of the segments until the length converged to submillimeter levels. In this example, the curve of alignment starts out north of the geodesic, crosses exactly once near the middle, and is south thereafter.

Notes:

- The vector \mathbf{v} is equally well defined as $\mathbf{v} = \mathbf{A} - \mathbf{B}$, but then $\mathbf{p}(t) = \mathbf{B} + t \mathbf{v}$. The same curve of alignment results either way.
- Pseudocode for the curve of alignment is given below. The algorithm assumes there is a function named `XYZ_to_LBH` that converts XYZ coordinates to geodetic longitude, latitude, and height LBH . This code is assumed to be written in a language with built-in vector algebra operators such as *Mathematica* or *Matlab*, or with a vector-algebra library such as *NumPy*.

```

 $\mathbf{A} := (\mathbf{XA}, \mathbf{YA}, \mathbf{ZA})$  # FROM point in XYZ
 $\mathbf{B} := (\mathbf{XB}, \mathbf{YB}, \mathbf{ZB})$  # TO point in XYZ
t: floating point scalar in [0, 1]
 $\mathbf{V} := \mathbf{B} - \mathbf{A}$  # vector subtraction
 $\mathbf{pt} := \mathbf{A} + t * \mathbf{V}$  #  $\mathbf{pt}$  will be inside the ellipsoid
 $\mathbf{p2} := \text{XYZ\_TO\_LBH}(\mathbf{pt})$  # convert  $\mathbf{pt}$  to LBH  $(\lambda_{p2}, \phi_{p2}, h_{p2})$ .
 $\mathbf{CoA} := (\lambda_{p2}, \phi_{p2}, 0)$  # set  $h_{p2}$  to zero.

```

3. Normal Sections

Normal sections can be defined with a single point on an ellipsoid and an azimuth [14]. There is a vector that is normal to the ellipsoid at this point. The sectioning plane contains the point and the normal vector, and the sectioning plane is oriented by the azimuth. The normal section is the intersection of that plane and the ellipsoid. This definition is not well suited for the purposes of this paper, so an alternative based on two points is provided. A normal section from \mathbf{A} to \mathbf{B} is a section formed from the intersection of a plane containing \mathbf{A} , \mathbf{B} , and another point \mathbf{V} , where \mathbf{VA} is normal to the ellipsoid at \mathbf{A} (see Figure 2). There are many choices for \mathbf{V} , but it is convenient and helpful to choose \mathbf{V} to be the point on the Z-axis intercepted by the inwards-directed normal vector at \mathbf{A} (details are given below).

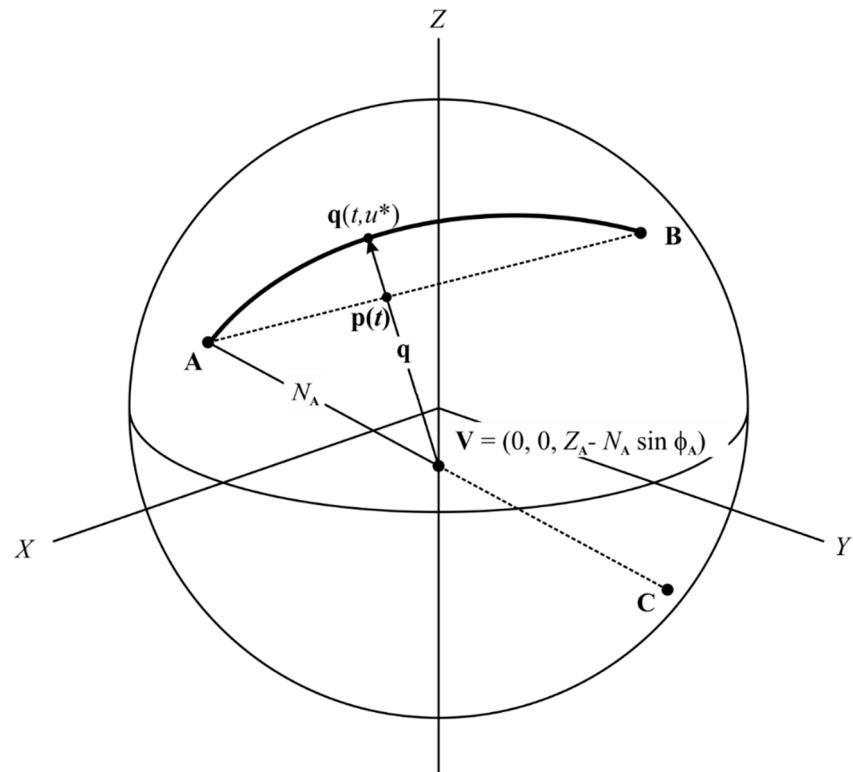


Figure 2. The construction of the (shorter limb of the) normal section from \mathbf{A} to \mathbf{B} .

3.1. Properties

The normal section is a planar curve. On an ellipsoid, the normal section is the entire ellipse (a circle for the Equator), but it is common practice to consider the normal section to be the shorter limb from **A** to **B**. The algorithm below produces the shorter limb, which is consistent with the goal of this paper.

A, **B**, and **V** will not define a normal section if they are collinear, but selecting **B** to be antipodal to **A** is not a problem (generally), as will now be shown. Point **C** in Figure 2 is the point where **VA** is extended to intersect the ellipsoid on the hemisphere opposite **A**. **C** is collinear with **A** and **V**, but **C** is not antipodal to **A** unless **A** is on the Equator or is a Pole (antipodal points are on a line through the origin). **VA** is normal to the ellipsoid at **A**, but **VA** does not generally contain the origin due to the ellipsoid having non-zero eccentricity. Therefore, the normal section is uniquely defined if **A** and **B** are antipodal but not on the Equator and not Poles. In that case, there is a singularity at the exact midpoint, which is at the origin and thus does not possess a unique normal vector.

The normal section from **A** to **B** is not the same as the normal section from **B** to **A** unless **A** and **B** are at the same latitude or at the same longitude.

The Equator and the meridians are normal sections. Any parallel other than the Equator is not a normal section (the normal section connecting two distinct points at the same latitude other than zero will pass polewards of their common parallel; see Figure 3).

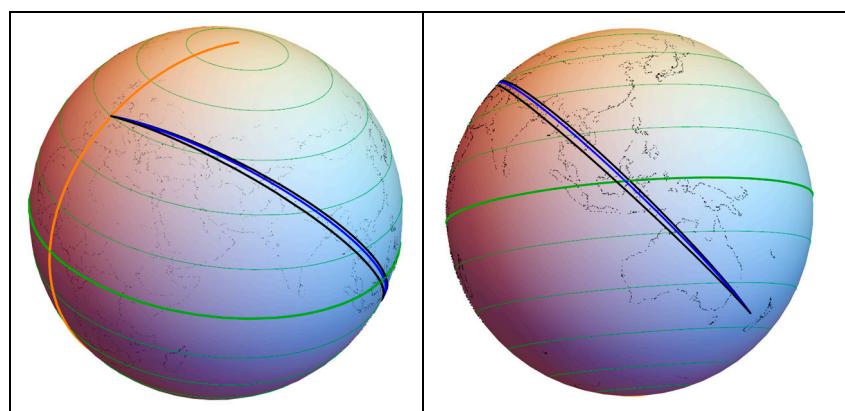


Figure 3. The two normal sections (black) and the geodesic (blue) between $(0^\circ \text{ E}, 45^\circ \text{ N})$ and $(165^\circ \text{ E}, 40^\circ \text{ S})$ on the WGS 84 reference ellipsoid. The Equator appears in green and the Prime Meridian in orange.

The normal section from **A** to **B** does not have the same arc length as the normal section from **B** to **A** unless $\phi_A = \phi_B$ or $\phi_A = -\phi_B$.

3.2. Algorithm

The algorithm used to draw a normal section is similar to that for curve of alignment, but it is a little more complex, so it warrants a figure to help explain it (see Figure 2). Some of the steps given here are expounded upon below in the notes.

1. Some ellipsoid is specified, such as a geodetic reference ellipsoid. This provides the length of the semi-major axis a and the value of the (first) eccentricity squared ϵ^2 . (See Appendix A for the formulas and parameter values.)
2. There is a point **V** on the Z-axis at a distance N_A from **A**, where $N_A = a / \sqrt{1 - \epsilon^2 \sin^2 \phi_A}$ is the radius of curvature in the prime vertical at **A**. **V**'s geocentric coordinates are $(0, 0, Z_A - N_A \sin \phi_A) = (0, 0, -N_A \epsilon^2 \sin \phi_A)$ and **VA** is normal to the ellipsoid at **A**. The plane **AVB** contains **VA** and so **AVB** is the normal sectioning plane at **A**.
3. A point of the straight-line segment (chord) from **A** to **B** is given in parametric form by $\mathbf{p}(t) = \mathbf{A} + t(\mathbf{B} - \mathbf{A})$ for $0 \leq t \leq 1$.

4. Any scalar multiple of a vector from \mathbf{V} to $\mathbf{p}(t)$ is in the normal sectioning plane defined by **AVB**. Let \mathbf{q} denote a vector in the normal sectioning plane defined by $\mathbf{q}(t, u) = \mathbf{V} + u(\mathbf{p}(t) - \mathbf{V})$, where u is a positive real number.
5. There is one value for u , say u^* , such that the geodetic height of $\gamma(\mathbf{q}(t, u^*))$ is zero for $u \geq 0$. The point $\mathbf{q}(t, u^*)$ is on the normal-section curve from **A** to **B** because $\mathbf{q}(t, u^*)$ is in the sectioning plane and the geodetic height of $\gamma(\mathbf{q}(t, u^*))$ is zero. Notice that this is different from the curve of alignment: there, the geodetic height is set to zero, but here, a value for u must be found that produces a geodetic height of zero. Setting the height to zero does not work for normal sections because \mathbf{q} is not generally normal to the ellipsoid. The points on an ellipsoid satisfy $\frac{x^2}{a^2} + \frac{y^2}{a^2} + \frac{z^2}{(1-\epsilon^2)a^2} = 1$, so $\mathbf{q}(t, u^*)$ must also. Define $\mathbf{n} = \mathbf{p}(t) - \mathbf{V}$. Substituting in $\mathbf{q}(t, u^*)$'s components and recalling that $V_x = V_y = 0$ gives:

$$\frac{(u^* n_x)^2}{a^2} + \frac{(u^* n_y)^2}{a^2} + \frac{(V_z + u^* n_z)^2}{(1-\epsilon^2)a^2} = 1$$

which is a quadratic in u^* . After some algebra, u^* can be computed from

$$u^* = \frac{-\kappa_1 + \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0}}{2\kappa_2} \quad (1)$$

where

$$\kappa_0 = a^{-2} (1 - \epsilon^2)^{-1} V_z^2 - 1$$

$$\kappa_1 = 2a^{-2} (1 - \epsilon^2)^{-1} n_z V_z$$

$$\kappa_2 = a^{-2} \left(n_x^2 + n_y^2 + (1 - \epsilon^2)^{-1} n_z^2 \right)$$

6. The normal section from **A** to **B** is the curve $\gamma(\mathbf{q}(t, u^*))$ for $0 \leq t \leq 1$. In other words, the normal section from **A** to **B** is the set of points given by the set of all the points $\mathbf{q}(t, u^*)$ converted from geocentric Cartesian to geodetic coordinates.

Notes:

- The normal-section curve is parameterized by t , the same as the curve of alignment.
- In general, each t will have its own u^* .
- The point on the normal section is given by $\mathbf{V} + u^*(\mathbf{p}(t) - \mathbf{V})$.
- The vector $\mathbf{q}(0, u)$ is normal to the ellipsoid but no other vector $\mathbf{q}(t, u)$ for $0 < t \leq 1$ is also normal unless **A** and **B** have the same latitude, and then $\mathbf{q}(0, u)$ and $\mathbf{q}(1, u)$ are both normal (at **A** and **B**).
- Pseudocode for the normal section is given below, with the same assumptions as for the curve of alignment.

A := (XA, YA, ZA) # starting point in XYZ

B := (XB, YB, ZB) # ending point in XYZ

t: floating point scalar in [0, 1]

$\mathbf{p}_t := \mathbf{A} + t(\mathbf{B} - \mathbf{A})$

$\mathbf{V} := (0, 0, -N_A \epsilon^2 \sin \phi_A)$ # get ϕ_A from XYZ_TO_LBH(A)

$\mathbf{n} := \mathbf{p}_t - \mathbf{V}$

$\kappa_0 := a^{-2} (1 - \epsilon^2)^{-1} V_z^2 - 1$

$\kappa_1 := 2a^{-2} (1 - \epsilon^2)^{-1} n_z V_z$

$\kappa_2 := a^{-2} \left(n_x^2 + n_y^2 + (1 - \epsilon^2)^{-1} n_z^2 \right)$ # N.B. $\kappa_2 > 0$

$u^* := -\kappa_1 + \sqrt{\kappa_1^2 - 4\kappa_2\kappa_0} / (2\kappa_2)$

$\mathbf{p}_{NS} := \mathbf{V} + u^*(\mathbf{p}_t - \mathbf{V})$

The two normal sections (black) and the curve of alignment (red) between $(0^\circ \text{ E}, 45^\circ \text{ N})$ and $(165^\circ \text{ E}, 40^\circ \text{ S})$ on the WGS 84 reference ellipsoid are shown in Figure 3. The normal sections have arc lengths of (N–S) 18,669,545.69 m and (S–N) 18,670,163.62 m, making the N–S section 617.94 m longer than the other. The lengths of the normal sections were computed with recursive subdivision, as above.

4. Great Ellipses

For non-antipodal points **A** and **B**, a great ellipse from **A** to **B** is a section formed from the intersection of a plane containing **A**, **B**, and the origin; therefore, the great ellipse is available from the normal-section algorithm simply by setting $\mathbf{V} = (0,0,0)$. The equation for $\mathbf{q}(t, u^*)$ simplifies to $\mathbf{q}(t, u^*) = u^* \mathbf{p}(t)$. The value for:

$$u^* = a^2 \sqrt{(1 - \epsilon^2) / ((1 - \epsilon^2)(n_x^2 + n_y^2) + n_z^2)}$$

The algorithm is otherwise unchanged.

Properties

1. The great ellipse is a planar curve.
2. The great ellipse is the entire ellipse (a circle for the Equator). Every great ellipse contains **A**'s antipodal point by construction; however, the antipodal point cannot be used to define the great ellipse because, then, **A**, **B**, and **V** would be collinear. The algorithm below produces the shorter limb between **A** and **B**, which is consistent with the goal of this paper.
3. The great ellipse from **A** to **B** is the same as the great ellipse from **B** to **A**.
4. The Equator and the meridians are great ellipses. Any parallel other than the Equator is not a great ellipse (the great ellipse connecting two distinct points at the same latitude other than zero will pass poleward of their common parallel; see Figure 4).

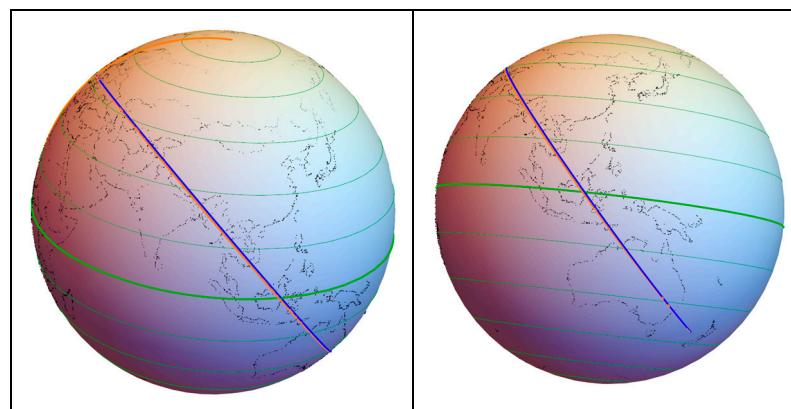


Figure 4. Great ellipse (orange) and the geodesic (blue) between $(0^\circ \text{ E}, 45^\circ \text{ N})$ and $(165^\circ \text{ E}, 40^\circ \text{ S})$ on the WGS 84 reference ellipsoid. The Equator appears in green.

Figure 4 shows the great ellipse, the curve of alignment, and the normal sections between $(0^\circ \text{ E}, 45^\circ \text{ N})$ and $(165^\circ \text{ E}, 40^\circ \text{ S})$ on the WGS 84 reference ellipsoid. The arc length of the great ellipse is 18,669,406.160 m, as computed by recursive subdivision. For this example, the curve of alignment starts out north of the great ellipse, then crosses once near the middle, and remains south thereafter, which is also true for the geodesic.

5. Loxodromes

The loxodrome is a curve of constant azimuth [12], so all the parallels of constant latitude (i.e., “parallels”) are loxodromes. A loxodrome is drawn as a straight line on a

Mercator-projection map, which permits a method of drawing loxodromes in geodetic coordinates using this paper's approach.

Properties

1. The parallels and meridians are loxodromes.
2. A loxodrome is a spiral if its starting azimuth is not a cardinal direction. In that case, loxodromes spiral infinitely many times around both Poles but do not reach them—yet, they have finite length. For example, for the WGS 84 ellipsoid and azimuth 45° , the total length works out to be 28,289,831.17 m [12].
3. The loxodrome from **A** to **B** is the same as the loxodrome from **B** to **A**.
4. The loxodrome is not a planar curve in general.
5. The loxodrome is defined if **A** and **B** are antipodal.

The algorithm used to draw loxodromes given below is based on the loxodromes being straight lines when drawn using a Mercator projection. See [15] or [3] for traditional implementations of the Mercator projection. However, Rollins and Meyer [16] provide a more succinct implementation, which is used here. Set the central meridian of the projection λ_0 to be the average of λ_A and λ_B , $\bar{\lambda}$, which avoids issues that arise if the loxodrome crosses 180° . However, the arithmetic average of two angles can fail, so let $s = \sin(\lambda_A) + \sin(\lambda_B)$ and $c = \cos(\lambda_A) + \cos(\lambda_B)$. Then, $\lambda_0 = \bar{\lambda} = \text{atan2}(c, s)$, and

$$x = a(\lambda - \lambda_0)$$

$$y = aq$$

$$q = \text{atanh}(\sin \phi) - \epsilon \text{atanh}(\epsilon \sin \phi)$$

where atan2 is a 2-argument arctangent function, atanh is the inverse hyperbolic tangent, x is the Mercator easting, y is the Mercator northing, and q is isometric latitude. The Mercator forward mapping formulas are applied to **A** and **B**, producing Mercator grid coordinates (x, y) , and the vector **v** is given by their difference, as before. Points **p**(t) in the Mercator grid are computed as a multiple of **v** and then converted back to geodetic coordinates. The inverse formulas are from [16], as provided in Appendix A. The following six steps make this algorithm explicit.

1. Some ellipsoid is specified, such as a geodetic reference ellipsoid. This provides the length of the semi-major axis a and the value of the (first) eccentricity squared ϵ^2 . (See Appendix A for the formulas and parameter values.)
2. Two points of interest on the ellipsoid are chosen. Their coordinates are required in geodetic coordinates; e.g., $\mathbf{A} = (\lambda_A, \phi_A)$ and $\mathbf{B} = (\lambda_B, \phi_B)$. If given in geocentric Cartesian coordinates, they can be readily converted to geodetic.
3. Using the forward mapping equations (see above), convert **A** and **B** into Mercator coordinates $\mathbf{a} = (x_a, y_a)$ and $\mathbf{b} = (x_b, y_b)$.
4. Compute the vector $\mathbf{v} = \mathbf{b} - \mathbf{a}$.
5. The points of the straight-line segment from **a** to **b** are given in parametric form by $\mathbf{p}(t) = \mathbf{a} + t \mathbf{v}$ for $0 \leq t \leq 1$.
6. Denote the inverse Mercator mapping equations (see Appendix A) by γ^M . Then, $\gamma^M(\mathbf{p}(t)) = (\lambda, \phi)$.
7. The loxodrome is curve $\gamma^M(\mathbf{p}(t))$ for $0 \leq t \leq 1$. In other words, the loxodrome is the set of points given by the conversion of all the points **p**(t) from Mercator grid coordinates to geodetic coordinates.

An implementation in *Mathematica*TM produced Figure 5, which shows the geodesic (green) and loxodrome (blue) from $(0^\circ \text{ E}, 45^\circ \text{ N})$ to $(165^\circ \text{ E}, 40^\circ \text{ S})$ on the WGS 84 reference ellipsoid. The arc length of the geodesic is 18,669,335.840 m, and the arc length of the loxodrome is 19,066,164.69 m, as computed by recursive subdivision, which is 396,828.850 m longer.

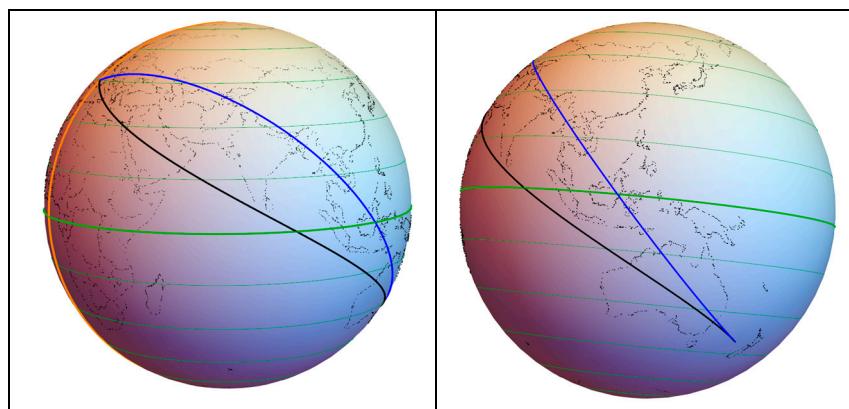


Figure 5. The geodesic (blue) and loxodrome (black) from $(0^\circ \text{ E}, 45^\circ \text{ N})$ to $(165^\circ \text{ E}, 40^\circ \text{ S})$ on the WGS 84 reference ellipsoid. The Equator appears in green and the Prime Meridian in orange.

Notes:

- The vector \mathbf{v} is equally well defined as $\mathbf{v} = \mathbf{a} - \mathbf{b}$, but then $\mathbf{p}(t) = \mathbf{b} + t \mathbf{v}$. The same loxodrome results either way.
- The inverse formula for latitude requires finding a fixed point. This can be cast as a root-finding problem by finding the latitude such that the difference of the input latitude minus the output latitude equals zero.
- Pseudocode for the loxodrome is given below. The reference-ellipsoid object RE is assumed to have a method ecc1 that returns the first eccentricity of the reference ellipsoid; otherwise, the same assumptions as above pertain.

Define a function $Mf(\lambda, \phi) \rightarrow (x, y) := (a\lambda_A, a(\operatorname{atanh}(\sin \phi_A) - e \operatorname{atanh}(e \sin \phi_A)))$. Define a function $Mr(x, y) \rightarrow (\lambda, \phi)$ to be the reverse Mercator mapping formulas.

See Appendix A.

```
t: floating point scalar in [0, 1]
A := ( $\lambda_A, \phi_A$ ) # starting point
a := Mf(A) # forward Mercator mapping formulas
B := ( $\lambda_B, \phi_B$ ) # ending point
b := Mf(B) # forward Mercator mapping formulas
v := b - a # vector subtraction
pt := a + tv # linear interpolation
pL := Mr(pt) # reverse Mercator mapping formula
```

6. Summary

Four new algorithms are presented that draw loxodromes, great ellipses, curves of alignment, and normal sections. The algorithms are parameterized so that a simple implementation based on vector algebra is available, given a supporting geometric geodesy library to calculate the coordinate system. There are many, more or less equivalent, choices for the conversion from geocentric Cartesian coordinates to geodetic longitude, latitude, and height [17,18]. However, for long lines, the chord between the beginning and ending points can come near to the geocenter, which can cause some algorithms to fail. For such situations, a robust algorithm such as [19] is recommended. Also, the algorithms herein are language neutral and do not depend on external packages. However, it should be noted that efficient algorithms and packages exist for geodesics [20] when the actual geodesic is required.

The examples herein were computed using the WGS84 reference ellipsoid. Ref. [21] showed that the minute difference in the flattening of these ellipsoids cannot result in a difference in coordinates at the millimeter level. So, these examples are properly illustrative for GRS80, as well.

Funding: This research received no external funding.

Data Availability Statement: Two Python 3.9 scripts that implement the algorithms (curves.py) and provide some testing and usage demonstrations (curves_demo.py) are available at https://github.com/thomas6meyer/geodetic_curves.git. Last accessed 4 March 2024.

Acknowledgments: The author is grateful to the anonymous reviewers who helped improve this manuscript.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A

For any geodetic reference ellipsoid, the semi-major axis a is a defining parameter. The ellipsoid's shape is given either by the semi-minor axis b , by the flattening f , or by the square of the first eccentricity ϵ^2 . See [22] for a list of the parameter values defining geodetic reference ellipsoids. The value for a is 6,378,137 m for the GRS 80 and WGS 84 reference ellipsoids. The values for f are 1/298.257222101 and 1/298.257223563 for the GRS 80 and WGS 84 reference ellipsoids. (This minute difference in the flattenings creates no practical difference in the positions [21].

The flattening of an oblate ellipse is:

$$f = (a - b)/b$$

The first eccentricity squared of an oblate ellipsoid is:

$$\epsilon^2 = (a^2 - b^2)/a^2 = 2f - f^2$$

The semi-minor axis b can be computed from either the flattening or the first eccentricity.

The radius of curvature in the prime vertical is $N = a/\sqrt{1 - \epsilon^2 \sin^2 \phi}$.
Conversion from geodetic to geocentric:

$$X = (N + h)\cos \lambda \cos \phi$$

$$Y = (N + h)\sin \lambda \cos \phi$$

$$Z = ((1 - \epsilon^2)N + h)\sin \phi$$

but for $h = 0$:

$$Z = (1 - \epsilon^2) N \sin \phi$$

The Mercator forward (direct) mapping equations take geodetic coordinates (λ, ϕ) to Mercator grid coordinates (x, y) . They are as follows:

$$x = a(\lambda - \lambda_0)$$

$$y = a q$$

$$q = \operatorname{atanh}(\sin \phi) - \epsilon \operatorname{atanh}(\epsilon \sin \phi)$$

There is no exact algebraic expression that provides the geodetic latitude directly from the Mercator northing, so either iterations or approximations are needed. The formulas below follow from the mathematics in [16] and they use iteration. For longitude,

$$\lambda = \lambda_0 + x/a$$

For latitude,

$$q = y/a$$

$$s = \text{Limit of } s^{(0)}, s^{(1)}, s^{(2)}, \dots$$

$$\phi = \arcsin(s)$$

where:

$$s^{(0)} = \tanh(q)$$

$$s^{(i+1)} = \tanh(q + \epsilon \operatorname{atanh}(\epsilon s^{(i)}))$$

and $\arcsin(\dots)$ is the inverse sine function. The iteration is avoidable by using truncated infinite series: “first calculate the conformal latitude chi based on the inverse of the spherical Mercator, then use the method found in <https://arxiv.org/pdf/2212.05818.pdf> to calculate geodetic latitude” (accessed 17 April 2024) (anonymous reviewer 2024).

References

1. Kallay, M. Defining edges on a round Earth. In Proceedings of the 15th International Symposium on Advances in Geographic Information Systems, ACM GIS 2007, Seattle, DC, USA, 7–9 November 2007. [[CrossRef](#)]
2. Vincenty, T. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Surv. Rev.* **1975**, *23*, 88–93. [[CrossRef](#)]
3. Thomas, P.D. *Conformal Projections in Geodesy and Cartography, Special Publication 251*; U.S. Naval Oceanographic Office: Washington, DC, USA, 1979.
4. Sjöberg, L.E. New solutions to the direct and indirect geodetic problems on the ellipsoid. *Z. Für Geodäsie Geoinf. Und Land Manag.* **2006**, *131*, 35–39.
5. Rollins, C.M. An integral for geodesic length. *Surv. Rev.* **2010**, *42*, 20–26. [[CrossRef](#)]
6. Karney, C.F. Algorithms for geodesics. *J. Geod.* **2013**, *87*, 43–55. [[CrossRef](#)]
7. Panou, G.; Korakitis, R. Geodesic equations and their numerical solution in Cartesian coordinates on a triaxial ellipsoid. *J. Geod. Sci.* **2019**, *9*, 1–12. [[CrossRef](#)]
8. Karney, C.F. Geodesics on an arbitrary ellipsoid of revolution. *J. Geod.* **2024**, *98*, 4. [[CrossRef](#)]
9. Thien, G. A Solution to the Inverse Problem for Nearly Antipodal Points on the Equator of the Ellipsoid of Revolution. Master’s Thesis, The Ohio State University, Columbus, OH, USA, 1967.
10. Earle, M.A. A vector solution for navigation on a great ellipse. *J. Navig.* **2000**, *53*, 473–481. [[CrossRef](#)]
11. Tseng, W.-K.; Guo, J.-L.; Liu, C.-P.; Wu, C.-T. The vector solutions for the great ellipse on the spheroid. *J. Geod.* **2012**, *6*, 103–109. [[CrossRef](#)]
12. Meyer, T.H.; Rollins, C.M. The Direct and Indirect Problem for Loxodromes. *Navigation* **2011**, *58*, 1–6. [[CrossRef](#)]
13. Deakin, R.E. *The Curve of Alignment on an Ellipsoid*; RMIT University, School of Mathematical & Geospatial Sciences: Melbourne, Australia, 2009.
14. Deakin, R.E. *The Normal Section Curve on an Ellipsoid*; RMIT University, School of Mathematical & Geospatial Sciences: Melbourne, Australia, 2009.
15. Snyder, J.P. *Map Projections—A Working Manual*; Professional Paper 1395p; U.S. Geological Survey: Washington, DC, USA, 1987; 1395p.
16. Rollins, C.M.; Meyer, T.H. Conformal mapping at elevation with implications for low-distortion projections. *Surv. Rev.* **2024**, *in press*.
17. Claessens, S.J. Efficient transformation from Cartesian to geodetic coordinates. *Comput. Geosci.* **2019**, *133*, 104307. [[CrossRef](#)]
18. Blewitt, G. An improved equation of latitude and a global system of graticule distance coordinates. *J. Geod.* **2024**, *98*, 6. [[CrossRef](#)] [[PubMed](#)]
19. Panou, G. Cartesian to geodetic coordinates conversion on an oblate spheroid using the bisection method. *Comput. Geosci.* **2019**, *133*, 104308. [[CrossRef](#)]
20. Karney, C.F. GeographicLib. 2015. Available online: <http://geographiclib.sourceforge.net> (accessed on 17 April 2024).
21. Rollins, C.M.; Meyer, T.H. The effect of delta epsilon on XYZ coordinates. *Surv. Rev.* **2012**, *44*, 282–284. [[CrossRef](#)]
22. Meyer, T.H. *Introduction to Geometrical and Physical Geodesy: Foundations of Geomatics*; ESRI Press: Redlands, CA, USA, 2010; 246p.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.