THE GREAT ELLIPTIC ARC ON AN ELLIPSOID

R. E. Deakin
School of Mathematical & Geospatial Sciences, RMIT University,
GPO Box 2476V, MELBOURNE VIC 3001, AUSTRALIA
email: rod.deakin@rmit.edu.au

January 2010

ABSTRACT

These notes provide a detailed derivation of the equation for the great elliptic arc on an ellipsoid. Using this equation and knowing the terminal points of the curve, a technique is developed for computing the location of points along the curve. A MATLAB function is provided that demonstrates the algorithm developed.

INTRODUCTION

In geodesy, the great elliptic arc between P_1 and P_2 on the ellipsoid is the curve created by intersecting the ellipsoid with the plane containing P_1, P_2 and O (the centre of the ellipsoid).

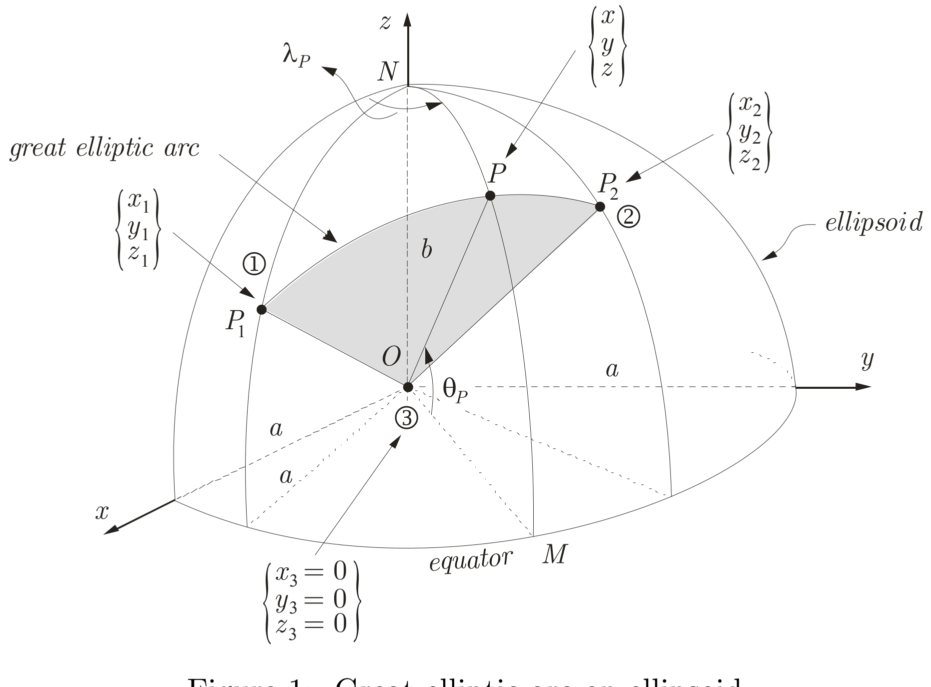

z
λ_P
N
$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$
$\begin{Bmatrix} x_2 \\ y_2 \\ z_2 \end{Bmatrix}$
great elliptic arc
P
P_2
②
$\begin{Bmatrix} x_1 \\ y_1 \\ z_1 \end{Bmatrix}$
①
b
ellipsoid
P_1
O
a
y
θ_P
③
a
a
x
$\begin{Bmatrix} x_3 = 0 \\ y_3 = 0 \\ z_3 = 0 \end{Bmatrix}$
equator
M

Figure 1: Great elliptic arc on ellipsoid

Figure 1 shows *P* on the great elliptic arc between P_1 and P_2. θ_P is the geocentric latitude of *P* and λ_P is the longitude of *P.*

There are an infinite number of planes that cut the surface of the ellipsoid and contain the chord P_1P_2 but only one of these will contain the centre *O.* Two other planes are the normal section plane P_1P_2 (containing the normal at P_1) and the normal section plane P_2P_1 (containing the normal at P_2). All of these curves of intersection (including the great elliptic arc and the two normal section curves) are plane curves that are arcs of ellipses (for a proof of this see Deakin, 2009a). All meridians of longitude on an ellipsoid and the ellipsoid equator are great elliptic arcs. Parallels of latitude – excepting the equator – are not great elliptic arcs. So we could say that the great elliptic arc is a unique plane curve on the ellipsoid – since it is created by the single plane containing P_1, P_2 and *O.* But it is not the shortest distance between P_1 and P_2; this unique property (shortest length) belongs to the geodesic.

Great elliptic arcs are not much used in geodesy as they don't have a practical connection with theodolite observations made on the surface of the earth that are approximated as observations made on an ellipsoid; e.g., normal section curves and curves of alignment. Nor are they the shortest distance between points on the ellipsoid; but, if we ignore earth rotation, they are the curves traced out on the geocentric ellipsoid by the ground point of an earth orbiting satellite or a ballistic missile moving in an orbital plane containing the earth's centre of mass. Here geocentric means *O* (the centre of the ellipsoid) is coincident with the centre of mass.

The equation for the curve developed below is similar to that derived for the curve of alignment in Deakin (2009b) and it is not in a form suitable for computing the distance or azimuth of the curve. But, as it contains functions of both the latitude and longitude of a point on the curve, it is suitable for computing the latitude of a point given a particular longitude; or alternatively the longitude of a point may be computed (iteratively) given a particular latitude.

EQUATION OF GREAT ELLIPTIC ARC

Figure 1 shows *P* on the great elliptic arc that passes through P_1 and P_2 on the ellipsoid. The semi-axes of the ellipsoid are *a* and *b* $\left(a > b\right)$ and the first-eccentricity squared e^2 and the flattening *f* of the ellipsoid are defined by

$$\begin{aligned} e^2 &= \frac{a^2 - b^2}{a^2} = f\left(2 - f\right) \\ f &= \frac{a - b}{a} \end{aligned} \tag{1}$$

Parallels of latitude ϕ and meridians of longitude λ have their respective reference planes; the equator and the Greenwich meridian, and Longitudes are measured 0° to $\pm 180^\circ$ (east positive, west negative) from the Greenwich meridian and latitudes are measured 0° to $\pm 90^\circ$ (north positive, south negative) from the equator. The *x,y,z* geocentric Cartesian coordinate system has an origin at *O,* the centre of the ellipsoid, and the z-axis is the minor axis (axis of revolution). The xOz plane is the Greenwich meridian plane (the origin of longitudes) and the xOy plane is the equatorial plane. The positive x-axis passes through the intersection of the Greenwich meridian and the equator, the positive y-axis is advanced 90° east along the equator and the positive z-axis passes through the north pole of the ellipsoid.

In Figure 1, θ_P is the geocentric latitude of P and (geodetic) latitude ϕ and geocentric latitude θ are related by

$$\tan\theta = \left(1 - e^2\right)\tan\phi = \frac{b^2}{a^2}\tan\phi = \left(1 - f\right)^2\tan\phi \tag{2}$$

The geometric relationship between geocentric latitude θ and (geodetic) latitude ϕ is shown in Figure 2.

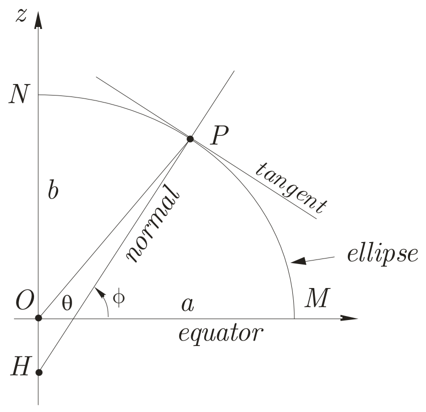


Figure 2: Meridian plane of P

The great elliptic plane in Figure 1 is defined by points ①, ② and ③ that are P_1, P_2 and the centre of the ellipsoid O respectively. Cartesian coordinates of ① and ② are computed from the following equations

$$\begin{aligned}x &= \nu\cos\phi\cos\lambda \\ y &= \nu\cos\phi\sin\lambda \\ z &= \nu\left(1-e^2\right)\sin\phi\end{aligned} \tag{3}$$

where $\nu = PH$ (see Figure 2) is the radius of curvature in the prime vertical plane and

$$\nu = \frac{a}{\sqrt{1-e^2\sin^2\phi}} \tag{4}$$

The Cartesian coordinates of point ③ are all zero.

The General equation of a plane may be written as

$$Ax + By + Cz + D = 0 \tag{5}$$

And the equation of the plane passing through points ①, ② and ③ is given in the form of a 3rd-order determinant

$$\begin{vmatrix} x-x_1 & y-y_1 & z-z_1 \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_3-x_2 & y_3-y_2 & z_3-z_2 \end{vmatrix} = 0 \tag{6}$$

or expanded into 2nd-order determinants

$$\begin{vmatrix} y_2-y_1 & z_2-z_1 \\ y_3-y_2 & z_3-z_2 \end{vmatrix}\left(x-x_1\right) - \begin{vmatrix} x_2-x_1 & z_2-z_1 \\ x_3-x_2 & z_3-z_2 \end{vmatrix}\left(y-y_1\right) + \begin{vmatrix} x_2-x_1 & y_2-y_1 \\ x_3-x_2 & y_3-y_2 \end{vmatrix}\left(z-z_1\right) = 0 \tag{7}$$

Expanding the determinants in equation (7) gives

$$\begin{aligned} &\left(x-x_1\right)\left\{\left(y_2-y_1\right)\left(z_3-z_2\right)-\left(z_2-z_1\right)\left(y_3-y_2\right)\right\} \\ &-\left(y-y_1\right)\left\{\left(x_2-x_1\right)\left(z_3-z_2\right)-\left(z_2-z_1\right)\left(x_3-x_2\right)\right\} \\ &+\left(z-z_1\right)\left\{\left(x_2-x_1\right)\left(y_3-y_2\right)-\left(y_2-y_1\right)\left(x_3-x_2\right)\right\} = 0 \end{aligned} \tag{8}$$

Now since $x_3 = y_3 = z_3 = 0$ and equation (8) becomes

$$\begin{aligned} &\left(x-x_1\right)\left\{\left(y_2-y_1\right)\left(-z_2\right)-\left(z_2-z_1\right)\left(-y_2\right)\right\} \\ &-\left(y-y_1\right)\left\{\left(x_2-x_1\right)\left(-z_2\right)-\left(z_2-z_1\right)\left(-x_2\right)\right\} \\ &+\left(z-z_1\right)\left\{\left(x_2-x_1\right)\left(-y_2\right)-\left(y_2-y_1\right)\left(-x_2\right)\right\} = 0 \end{aligned} \tag{9}$$

Expanding and simplifying equation (9) gives

$$x\left(y_1z_2-y_2z_1\right)-y\left(x_1z_2-x_2z_1\right)+z\left(x_1y_2-x_2y_1\right)=0$$

Replacing *x, y* and *z* with their equivalents, given by equations (3), gives

$$\nu\cos\phi\cos\lambda\left(y_1z_2-y_2z_1\right)-\nu\cos\phi\sin\lambda\left(x_1z_2-x_2z_1\right)+\nu\left(1-e^2\right)\sin\phi\left(x_1y_2-x_2y_1\right)=0$$

and dividing both sides by $\nu \cos\phi$ gives the equation of the great elliptic arc as

$$\boxed{A\cos\lambda - B\sin\lambda + C\left(1-e^2\right)\tan\phi = 0 \qquad (10)}$$

where A, B and C are functions of the coordinates of the terminal points P_1 and P_2

$$A = y_1 z_2 - y_2 z_1 \qquad B = x_1 z_2 - x_2 z_1 \qquad C = x_1 y_2 - x_2 y_1 \tag{11}$$

Equation (10) is not suitable for computing the distance along a great elliptic arc, nor is it suitable for computing the azimuth of the curve, but by certain re-arrangements it is possible to solve (directly) for the latitude of a point on the curve given a longitude somewhere between the longitudes of the terminal points of the curve. Or alternatively, solve (iteratively) for the longitude of a point given a latitude somewhere between the latitudes of the terminal points.

SOLVING FOR THE LATITUDE

A simple re-arrangement of equation (10) allows the latitude ϕ to be evaluated from

$$\tan\phi = \frac{B\sin\lambda - A\cos\lambda}{C\left(1-e^2\right)} \tag{12}$$

where A and B and C are functions of terminal points P_1 and P_2 given by equations (11).

SOLVING FOR THE LONGITUDE

The longitude λ can be evaluated using Newton-Raphson iteration for the real roots of the equation $f\left(\lambda\right) = 0$ given in the form of an iterative equation

$$\lambda_{(n+1)} = \lambda_{(n)} - \frac{f\left(\lambda_{(n)}\right)}{f'\left(\lambda_{(n)}\right)} \tag{13}$$

where n denotes the n^{th} iteration and $f\left(\lambda\right)$ is given by equation (10) as

$$f\left(\lambda\right) = A\cos\lambda - B\sin\lambda + C\left(1-e^2\right)\tan\phi \tag{14}$$

and the derivative $f'\left(\lambda\right) = \frac{d}{d\lambda}\left\{f\left(\lambda\right)\right\}$ is given by

$$f'\left(\lambda\right) = -A\sin\lambda - B\cos\lambda \tag{15}$$

An initial value of $\lambda_{(1)}$ (λ for $n=1$) can be taken as the longitude of P_1 and the functions $f\left(\lambda_{(1)}\right)$ and $f'\left(\lambda_{(1)}\right)$ evaluated from equations (14) and (15) using λ_1. $\lambda_{(2)}$ (λ for $n=2$) can now be computed from equation (13) and this process repeated to obtain values $\lambda_{(3)}, \lambda_{(4)}, \ldots$. This iterative process can be concluded when the difference between $\lambda_{(n+1)}$ and $\lambda_{(n)}$ reaches an acceptably small value.

Alternatively, the longitude can be evaluated by a trigonometric equation derived as follows. Equation (10) can be expressed as

$$B\sin\lambda - A\cos\lambda = C\left(1-e^2\right)\tan\phi \tag{16}$$

and A, B and C are given by equations (11). Equation (16) can be expressed as a trigonometric addition of the form

$$\begin{aligned} C\left(1-e^2\right)\tan\phi &= R\cos\left(\lambda-\theta\right) \\ &= R\cos\lambda\cos\theta + R\sin\lambda\sin\theta \end{aligned} \tag{17}$$

Now, equating the coefficients of $\cos\lambda$ and $\sin\lambda$ in equations (17) and (16) gives

$$A = -R\cos\theta; \quad B = R\sin\theta \tag{18}$$

and using these relationships

$$R = \sqrt{A^2+B^2}; \quad \tan\theta = \frac{B}{-A} \tag{19}$$

Substituting these results into equation (17) gives

$$\lambda = \arccos\left\{\frac{C\left(1-e^2\right)\tan\phi}{\sqrt{A^2+B^2}}\right\} + \arctan\left\{\frac{B}{-A}\right\} \tag{20}$$

DIFFERENCE IN LENGTH BETWEEN A GEODESIC AND A GREAT ELLIPTIC ARC

There are five curves of interest in geodesy; the geodesic, the normal section, the great elliptic arc the loxodrome and the curve of alignment.

The geodesic between P_1 and P_2 on an ellipsoid is the unique curve on the surface defining the shortest distance; all other curves will be longer in length. The normal section curve P_1P_2 is a plane curve created by the intersection of the normal section plane containing the normal at P_1 and also P_2 with the ellipsoid surface. And as we have shown (Deakin

2009a) there is the other normal section curve P_2P_1. The curve of alignment (Deakin 2009b, Thomas 1952) is the locus of all points P such that the normal section plane at P also contains the points P_1 and P_2. The curve of alignment is very close to a geodesic. The great elliptic arc is the plane curve created by intersecting the plane containing P_1, P_2 and the centre O with the surface of the ellipsoid and the loxodrome is the curve on the surface that cuts each meridian between P_1 and P_2 at a constant angle.

Approximate equations for the difference in length between the geodesic, the normal section curve and the curve of alignment were developed by Clarke (1880, p. 133) and Bowring (1972, p. 283) developed an approximate equation for the difference between the geodesic and the great elliptic arc. Following Bowring (1972), let

$$\begin{aligned} s &= \text{ geodesic length} \\ L &= \text{ normal section length} \\ D &= \text{ great elliptic length} \\ S &= \text{ curve of alignment length} \end{aligned}$$

then

$$\begin{aligned} L - s &= \frac{e^4}{90} s \left(\frac{s}{R}\right)^4 \cos^4\phi_1 \sin^2\alpha_{12} \cos^2\alpha_{12} + \cdots \\ D - s &= \frac{e^4}{24} s \left(\frac{s}{R}\right)^2 \sin^2\phi_1 \cos^2\phi_1 \sin^2\alpha_{12} + \cdots \\ S - s &= \frac{e^4}{360} s \left(\frac{s}{R}\right)^4 \cos^4\phi_1 \sin^2\alpha_{12} \cos^2\alpha_{12} + \cdots \end{aligned} \tag{21}$$

where R can be taken as the radius of curvature in the prime vertical at P_1. Now for a given value of s, $D - s$ will be a maximum if $\phi_1 = 45^\circ$ and $\alpha_{12} = 90^\circ$ in which case $\sin^2\phi_1 \cos^2\phi_1 \sin^2\alpha_{12} = \frac{1}{4}$, thus

$$\left(D - s\right) < \frac{e^4}{96} s \left(\frac{s}{R}\right)^4 \tag{22}$$

For the GRS80 ellipsoid where $f = 1/298.257222101$, $e^2 = f\left(2 - f\right)$, and for $s = 1200000$ m (1200 km) and $R = 6371000$ m, equation (22) gives $D - s < 0.001$ m.

MATLAB FUNCTIONS

Two MATLAB functions are shown below; they are: *great_ elliptic_ arc_ lat.m* and *great_ elliptic_ arc_ lon.m* Assuming that the terminal points of the curve are known, the first function computes the latitude of a point on the curve given a longitude and the second function computes the longitude of a point given the latitude.

Output from the two functions is shown below for points on a great elliptic arc between the terminal points of the straight-line section of the Victorian–New South Wales border. This straight-line section of the border, between Murray Spring and Wauka 1978, is known as the Black-Allan Line in honour of the surveyors Black and Allan who set out the border line in 1870-71. Wauka 1978 (Gabo PM 4) is a geodetic concrete border pillar on the coast at Cape Howe and Murray Spring (Enamo PM 15) is a steel pipe driven into a spring of the Murray River that is closest to Cape Howe. The straight line is a normal section curve on the reference ellipsoid of the Geocentric Datum of Australia (GDA94) that contains the normal to the ellipsoid at Murray Spring. The GDA94 coordinates of Murray Spring and Wauka 1978 are:

Murray Spring:	$\phi\ -37^\circ\, 47'\, 49.2232''$	$\lambda\ 148^\circ\, 11'\, 48.3333''$
Wauka 1978:	$\phi\ -37^\circ\, 30'18.0674''$	$\lambda\ 149^\circ\, 58'\, 32.9932''$

The normal section azimuth and distance are:

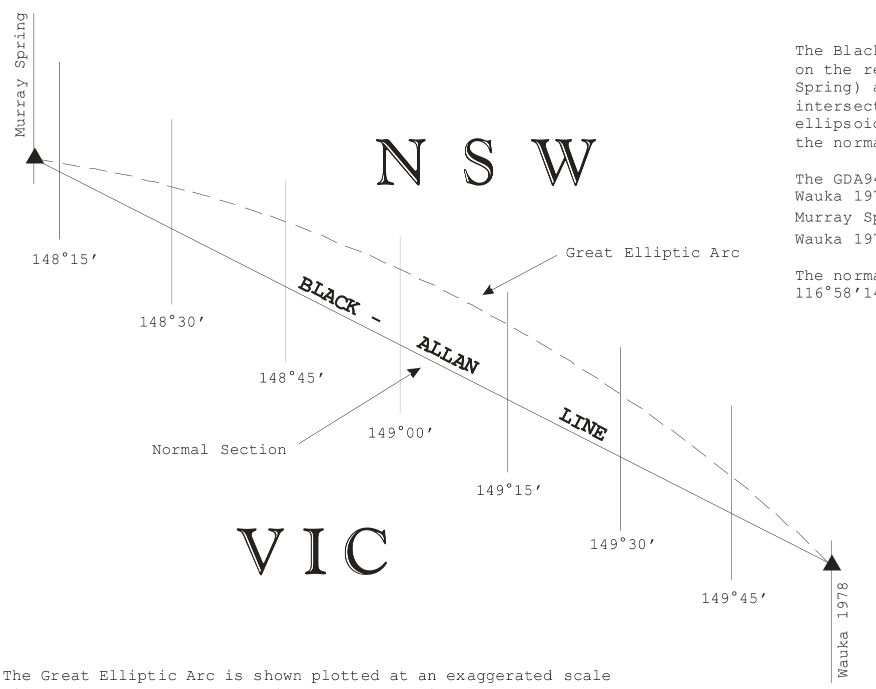
$$116^\circ\, 58'14.173757'' \qquad 176495.243760 \text{ m}$$

The geodesic azimuth and distance are:

$$116^\circ\, 58'14.219146'' \qquad 176495.243758 \text{ m}$$

Figure 3 shows a schematic view of the Black-Allan line (normal section) and the great elliptic arc. The relationships between the great elliptic arc and the normal section have been computed at seven locations along the line (A, B, C, etc.) where meridians of longitude at $0^\circ\, 15'$ intervals cut the line. These relationships are shown in Table 1.

BLACK-ALLAN LINE: VICTORIA/NSW BORDER



The Black-Allan Line is a normal section curve on the reference ellipsoid between P1 (Murray Spring) and P2 (Wauka 1978). This curve is the intersection of the normal section plane and the ellipsoid, and the normal section contains P1, the normal to the ellipsoid at P1, and P2.

The GDA94 coordinates of Murray Spring and Wauka 1978 are:
Murray Spring: ϕ -37°47′49.2232″ λ 148°11′48.3333″
Wauka 1978: ϕ -37°30′18.0674″ λ 149°58′32.9932″

The normal section azimuth and distance are:
116°58′14.173757″ 176495.243760 m.

The Great Elliptic Arc is shown plotted at an exaggerated scale with respect to the Border Line (normal section).
At longitude 149°00′E. the Great Elliptic Arc is 1.939 m north of the Border Line.
At longitude 149°30′E. the Great Elliptic Arc is 1.522 m north of the Border Line.

Figure 3

BLACK-ALLAN LINE: VICTORIA/NSW BORDER

NAME	GDA94		Ellipsoid values		
	LATITUDE	LONGITUDE	dφ	ρ	dm = ρ×dφ
Murray Spring	-36°47´49.223200″	148°11´48.333300″			
A	-36°49´07.598047″ N -36°49´07.590584″ GEA	148°15´00.000000″	+00´00.007463″	6358356.102	+0.2301
B	-36°55´13.876510″ N -36°55´13.840305″ GEA	148°30´00.000000″	+00´00.036205″	6358465.209	+1.1161
C	-37°01´17.289080″ N -37°01´17.234433″ GEA	148°45´00.000000″	+00´00.054647″	6358573.577	+1.6846
D	-37°07´17.845554″ N -37°07´17.782643″ GEA	149°00´00.000000″	+00´00.062911″	6358681.204	+1.9394
E	-37°13´15.555723″ N -37°13´15.494607″ GEA	149°15´00.000000″	+00´00.061116″	6358788.089	+1.8841
F	-37°19´10.429372″ N -37°19´10.379991″ GEA	149°30´00.000000″	+00´00.049381″	6358894.232	+1.5224
G	-37°25´02.476276″ N -37°25´02.448453″ GEA	149°45´00.000000″	+00´00.027823″	6358999.632	+0.8578
Wauka 1978	-37°30´18.067400″	149°58´32.993200″			

TABLE 1: Points where the Great Elliptic Arc cuts meridians of A, B, C, etc at 0°15´ intervals of longitude along Border Line. N = Normal Section, GEA = Great Elliptic Arc

```
>> great_elliptic_arc_lat

==================
Great Elliptic Arc
==================
Ellipsoid parameters
a  = 6378137.0000
f  = 1/298.257222101

Terminal points of curve
Latitude  P1 =  -36 47 49.223200 (D M S)
Longitude P1 =  148 11 48.333300 (D M S)

Latitude  P2 =  -37 30 18.067400 (D M S)
Longitude P2 =  149 58 32.993200 (D M S)

Cartesian coordinates
            X               Y                Z
P1   -4345789.609716  2694844.030716 -3799378.032024
P2   -4386272.668061  2534883.268540 -3862005.992252

Given longitude of P3
Longitude P3 =  149 30  0.000000 (D M S)

Latitude of P3 computed from trigonometric equation
Latitude  P3 =  -37 19 10.379991 (D M S)

>>
```

```
>> great_elliptic_arc_lon

==================
Great Elliptic Arc
==================
Ellipsoid parameters
a  = 6378137.0000
f  = 1/298.257222101

Terminal points of curve
Latitude  P1 =  -36 47 49.223200 (D M S)
Longitude P1 =  148 11 48.333300 (D M S)

Latitude  P2 =  -37 30 18.067400 (D M S)
Longitude P2 =  149 58 32.993200 (D M S)

Cartesian coordinates
            X               Y                Z
P1   -4345789.609716  2694844.030716 -3799378.032024
P2   -4386272.668061  2534883.268540 -3862005.992252

Given latitude of P3
Latitude  P3 =  -37 19 10.379991 (D M S)

Longitude of P3 computed from Newton-Raphson iteration
Longitude P3 =  149 30  0.000001 (D M S)
iterations   =    5

Longitude of P3 computed from trigonometric equation
Longitude P3 =  149 30  0.000001 (D M S)
theta P3     =    8 39 58.683516 (D M S)

>>
```

MATLAB function *great_ elliptic_ arc_ lat.m*

```
function great_elliptic_arc_lat
%
% great_elliptic_arc_lat: Given the terminal points P1 and P2 of a great
% elliptic arc on an ellipsoid, and the longitude of a point P3 on the
% curve, this function computes the latitude of P3.

%--------------------------------------------------------------------------
% Function:  great_elliptic_arc_lat
%
% Usage:     great_elliptic_arc_lat
%
% Author:    R.E.Deakin,
%            School of Mathematical & Geospatial Sciences, RMIT University
%            GPO Box 2476V, MELBOURNE, VIC 3001, AUSTRALIA.
%            email: rod.deakin@rmit.edu.au
%            Version  1.0  3 October 2009
%            Version  1.1  5 January 2010
%
% Purpose:   Given the terminal points P1 and P2 of a great elliptic arc on
%  an ellipsoid, and the longitude of a point P3 on the curve, this
%  function computes the latitude of P3.
%
% Functions required:
%      [D,M,S] = DMS(DecDeg)
%      [X,Y,Z] = Geo2Cart(a,flat,lat,lon,h)
%      [rm,rp] = radii(a,flat,lat);
%
% Variables:
%  A,B,C          - constants of great elliptic arc
%  a              - semi-major axis of ellipsoid
%  b              - semi-minor axis of ellipsoid
%  d2r            - degree to radian conversion factor 57.29577951...
%  e2             - eccentricity of ellipsoid squared
%  f              - f = 1/flat is the flattening of ellipsoid
%  flat           - denominator of flattening of ellipsoid
%  h1,h2          - ellipsoid heights of P1 and P2
%  lat1,lat2,lat3 - latitude of P1, P1, P3 (radians)
%  lon1,lon2,lon3 - longitude of P1, P2, P3 (radians)
%  nu             - radius of curvature in prime vertical plane
%  rho            - radius of curvature in meridain plane
%  X1,Y1,Z1       - Cartesian coordinates of P1
%  X2,Y2,Z2       - Cartesian coordinates of P2
%
% Remarks:
%
% References:
%  [1] Deakin, R.E., 2010, 'The Great Elliptic Arc on an Ellipsoid',
%         Lecture Notes, School of Mathematical and Geospatial Sciences,
%         RMIT University, January 2010
%
%--------------------------------------------------------------------------

% Degree to radian conversion factor
d2r   = 180/pi;

% Set ellipsoid parameters
a    = 6378137;    % GRS80
flat = 298.257222101;
% a    = 6378160;     % ANS
% flat = 298.25;
% a = 20926062;  % CLARKE 1866
% b = 20855121;
% f = 1-(b/a);
% flat = 1/f;

% Compute ellipsoid constants
```

```
f   = 1/flat;
e2  = f*(2-f);

% Set lat, lon and height of P1 and P2 on ellipsoid
lat1 = -(36  + 47/60 + 49.2232/3600)/d2r;   % Spring
lon1 =  (148 + 11/60 + 48.3333/3600)/d2r;
lat2 = -(37  + 30/60 + 18.0674/3600)/d2r;   % Wauka 1978
lon2 =  (149 + 58/60 + 32.9932/3600)/d2r;
h1 = 0;
h2 = 0;
% Compute Cartesian coords of P1 and P2
[X1,Y1,Z1] = Geo2Cart(a,flat,lat1,lon1,h1);
[X2,Y2,Z2] = Geo2Cart(a,flat,lat2,lon2,h2);

% Compute constants of Curve of Alignment
A = Y1*Z2-Y2*Z1;
B = X1*Z2-X2*Z1;
C = X1*Y2-X2*Y1;

% Set longitude of P3
lon3 = (149 + 30/60)/d2r;

% Compute latitude of P3
lat3 = atan((B*sin(lon3)-A*cos(lon3))/(C*(1-e2)));

%-----------------------
% Print result to screen
%-----------------------

fprintf('\n==================');
fprintf('\nGreat Elliptic Arc');
fprintf('\n==================');
fprintf('\nEllipsoid parameters');
fprintf('\na  = %12.4f',a);
fprintf('\nf  = 1/%13.9f',flat);

fprintf('\n\nTerminal points of curve');
% Print lat and lon of P1
[D,M,S] = DMS(lat1*d2r);
if D == 0 && lat1 < 0
    fprintf('\nLatitude  P1 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\nLatitude  P1 = %4d %2d %9.6f (D M S)',D,M,S);
end
[D,M,S] = DMS(lon1*d2r);
if D == 0 && lon1 < 0
    fprintf('\nLongitude P1 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\nLongitude P1 = %4d %2d %9.6f (D M S)',D,M,S);
end
% Print lat and lon of P2
[D,M,S] = DMS(lat2*d2r);
if D == 0 && lat2 < 0
    fprintf('\n\nLatitude  P2 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\n\nLatitude  P2 = %4d %2d %9.6f (D M S)',D,M,S);
end
[D,M,S] = DMS(lon2*d2r);
if D == 0 && lon2 < 0
    fprintf('\nLongitude P2 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\nLongitude P2 = %4d %2d %9.6f (D M S)',D,M,S);
end

% Print Coordinate table
fprintf('\n\nCartesian coordinates');
fprintf('\n            X               Y               Z');
fprintf('\nP1   %15.6f %15.6f %15.6f',X1,Y1,Z1);
fprintf('\nP2   %15.6f %15.6f %15.6f',X2,Y2,Z2);
```

```
% Print lat and lon of P3
fprintf('\n\nGiven longitude of P3');
[D,M,S] = DMS(lon3*d2r);
if D == 0 && lon3 < 0
    fprintf('\nLongitude P3 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\nLongitude P3 = %4d %2d %9.6f (D M S)',D,M,S);
end
fprintf('\n\nLatitude of P3 computed from trigonometric equation');
[D,M,S] = DMS(lat3*d2r);
if D == 0 && lat3 < 0
    fprintf('\nLatitude  P3 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\nLatitude  P3 = %4d %2d %9.6f (D M S)',D,M,S);
end

fprintf('\n\n');
```

MATLAB function *great_ elliptic_ arc_ lon.m*

```
function great_elliptic_arc_lon
%
% great_elliptic_arc_lon: Given the terminal points P1 and P2 of a great
% elliptic arc on an ellipsoid, and the latitude of a point P3 on the
% curve, this function computes the longitude of P3.

%--------------------------------------------------------------------------
% Function:  great_elliptic_arc_lon
%
% Usage:     great_elliptic_arc_lon
%
% Author:    R.E.Deakin,
%            School of Mathematical & Geospatial Sciences, RMIT University
%            GPO Box 2476V, MELBOURNE, VIC 3001, AUSTRALIA.
%            email: rod.deakin@rmit.edu.au
%            Version  1.0  3 October 2009
%            Version  1.1  5 January 2010
%
% Purpose:   Given the terminal points P1 and P2 of a great elliptic arc on
%  an ellipsoid, and the latitude of a point P3 on the curve, this
%  function computes the longitude of P3.
%
% Functions required:
%      [D,M,S] = DMS(DecDeg)
%      [X,Y,Z] = Geo2Cart(a,flat,lat,lon,h)
%      [rm,rp] = radii(a,flat,lat);
%
% Variables:
%  A,B,C          - constants of great elliptic arc
%  a              - semi-major axis of ellipsoid
%  b              - semi-minor axis of ellipsoid
%  d2r            - degree to radian conversion factor 57.29577951...
%  e2             - eccentricity of ellipsoid squared
%  f              - f = 1/flat is the flattening of ellipsoid
%  flat           - denominator of flattening of ellipsoid
%  f_lat3         - function of latitude of P3
%  fdash_lat3     - derivative of function of latitude of Pp3
%  h1,h2          - ellipsoid heights of P1 and P2
%  iter           - number of iterations
%  lambda         - longitude of P3 computed from trigonometric equation
%  lat1,lat2,lat3 - latitude of P1, P1, P3 (radians)
%  lon1,lon2,lon3 - longitude of P1, P2, P3 (radians)
%  new_lat3       - next latiude in Newton-Raphson iteration
%  nu             - radius of curvature in prime vertical plane
%  rho            - radius of curvature in meridain plane
%  theta          - auxiliary angle in the computation of lambda
%  X1,Y1,Z1       - Cartesian coordinates of P1
%  X2,Y2,Z2       - Cartesian coordinates of P2
%
% Remarks:
%
% References:
%  [1] Deakin, R.E., 2010, 'The Great Elliptic Arc on an Ellipsoid',
%         Lecture Notes, School of Mathematical and Geospatial Sciences,
%         RMIT University, January 2010
%
%--------------------------------------------------------------------------

% Degree to radian conversion factor
d2r   = 180/pi;

% Set ellipsoid parameters
a    = 6378137;    % GRS80
flat = 298.257222101;
% a    = 6378160;     % ANS
% flat = 298.25;
% a = 20926062;  % CLARKE 1866
% b = 20855121;
% f = 1-(b/a);
% flat = 1/f;

% Compute ellipsoid constants
f   = 1/flat;
e2  = f*(2-f);

% Set lat, lon and height of P1 and P2 on ellipsoid
lat1 = -(36  + 47/60 + 49.2232/3600)/d2r;   % Spring
lon1 =  (148 + 11/60 + 48.3333/3600)/d2r;
lat2 = -(37  + 30/60 + 18.0674/3600)/d2r;   % Wauka 1978
lon2 =  (149 + 58/60 + 32.9932/3600)/d2r;
h1 = 0;
h2 = 0;
% Compute Cartesian coords of P1 and P2
[X1,Y1,Z1] = Geo2Cart(a,flat,lat1,lon1,h1);
[X2,Y2,Z2] = Geo2Cart(a,flat,lat2,lon2,h2);

% Compute constants of Curve of Alignment
A = Y1*Z2-Y2*Z1;
B = X1*Z2-X2*Z1;
C = X1*Y2-X2*Y1;

% Set latitude of P3
lat3 = -(37 + 19/60 + 10.379991/3600)/d2r;

%------------------------------------------------------------
% Compute the longitude of P3 using Newton-Raphson iteration
%------------------------------------------------------------
% Set starting value of lon3 = longitude of P1
lon3 = lon1;
iter = 1;
while 1
    % Compute radii of curvature
    f_lon3     = A*cos(lon3)-B*sin(lon3)+C*(1-e2)*tan(lat3);
    fdash_lon3 = -A*sin(lon3)-B*cos(lon3);
    new_lon3   = lon3-(f_lon3/fdash_lon3);
    if abs(new_lon3 - lon3) < 1e-15
        break;
    end
    lon3 = new_lon3;
    if iter > 100
        fprintf('Iteration for longitude failed to converge after 100 iterations');
        break;
    end
    iter = iter + 1;
```

```
end;

%----------------------------------------------------------
% Compute the longitude of P3 using trigonometric equation
%----------------------------------------------------------
theta  = atan2(B,-A);
lambda = acos(C*(1-e2)*tan(lat3)/sqrt(A^2+B^2))+theta;

%------------------------
% Print result to screen
%------------------------

fprintf('\n==================');
fprintf('\nGreat Elliptic Arc');
fprintf('\n==================');
fprintf('\nEllipsoid parameters');
fprintf('\na  = %12.4f',a);
fprintf('\nf  = 1/%13.9f',flat);

fprintf('\n\nTerminal points of curve');
% Print lat and lon of P1
[D,M,S] = DMS(lat1*d2r);
if D == 0 && lat1 < 0
    fprintf('\nLatitude  P1 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\nLatitude  P1 = %4d %2d %9.6f (D M S)',D,M,S);
end
[D,M,S] = DMS(lon1*d2r);
if D == 0 && lon1 < 0
    fprintf('\nLongitude P1 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\nLongitude P1 = %4d %2d %9.6f (D M S)',D,M,S);
end
% Print lat and lon of P2
[D,M,S] = DMS(lat2*d2r);
if D == 0 && lat2 < 0
    fprintf('\n\nLatitude  P2 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\n\nLatitude  P2 = %4d %2d %9.6f (D M S)',D,M,S);
end
[D,M,S] = DMS(lon2*d2r);
if D == 0 && lon2 < 0
    fprintf('\nLongitude P2 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\nLongitude P2 = %4d %2d %9.6f (D M S)',D,M,S);
end

% Print Coordinate table
fprintf('\n\nCartesian coordinates');
fprintf('\n            X               Y               Z');
fprintf('\nP1   %15.6f %15.6f %15.6f',X1,Y1,Z1);
fprintf('\nP2   %15.6f %15.6f %15.6f',X2,Y2,Z2);

% Print lat and lon of P3
fprintf('\n\nGiven latitude of P3');
[D,M,S] = DMS(lat3*d2r);
if D == 0 && lat3 < 0
    fprintf('\nLatitude  P3 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\nLatitude  P3 = %4d %2d %9.6f (D M S)',D,M,S);
end

fprintf('\n\nLongitude of P3 computed from Newton-Raphson iteration');
[D,M,S] = DMS(lon3*d2r);
if D == 0 && lon3 < 0
    fprintf('\nLongitude P3 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\nLongitude P3 = %4d %2d %9.6f (D M S)',D,M,S);
end
```

```
fprintf('\niterations   = %4d',iter);

fprintf('\n\nLongitude of P3 computed from trigonometric equation');
[D,M,S] = DMS(lambda*d2r);
if D == 0 && lambda < 0
    fprintf('\nLongitude P3 =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\nLongitude P3 = %4d %2d %9.6f (D M S)',D,M,S);
end
[D,M,S] = DMS(theta*d2r);
if D == 0 && theta < 0
    fprintf('\ntheta P3     =   -0 %2d %9.6f (D M S)',M,S);
else
    fprintf('\ntheta P3     = %4d %2d %9.6f (D M S)',D,M,S);
end

fprintf('\n\n');
```

MATLAB function *Geo2Cart.m*

```
function [X,Y,Z] = Geo2Cart(a,flat,lat,lon,h)
%
% [X,Y,Z] = Geo2Cart(a,flat,lat,lon,h)
%   Function computes the Cartesian coordinates X,Y,Z of a point
%   related to an ellipsoid defined by semi-major axis (a) and the
%   denominator of the flattening (flat) given geographical
%   coordinates latitude (lat), longitude (lon) and ellipsoidal
%   height (h).  Latitude and longitude are assumed to be in radians.

%--------------------------------------------------------------------------
% Function:  Geo2Cart()
%
% Usage:     [X,Y,Z] = Geo2Cart(a,flat,lat,lon,h);
%
% Author:    R.E.Deakin,
%            School of Mathematical & Geospatial Sciences, RMIT University
%            GPO Box 2476V, MELBOURNE, VIC 3001, AUSTRALIA.
%            email: rod.deakin@rmit.edu.au
%            Version  1.0   6 April  2006
%            Version  1.0  20 August 2007
%
% Functions required:
%    radii()
%
% Purpose:
%    Function Geo2Cart() will compute Cartesian coordinates X,Y,Z
%    given geographical coordinates latitude, longitude (both in
%    radians) and height of a  point related to an ellipsoid
%    defined by semi-major axis (a) and denominator of flattening
%    (flat).
%
% Variables:
%    a       - semi-major axis of ellipsoid
%    e2      - 1st eccentricity squared
%    f       - flattening of ellipsoid
%    flat    - denominator of flattening f = 1/flat
%    h       - height above ellipsoid
%    lat     - latitude (radians)
%    lon     - longitude (radians)
%    p       - perpendicular distance from minor axis of ellipsoid
%    rm      - radius of curvature of meridian section of ellipsoid
%    rp      - radius of curvature of prime vertical section of ellipsoid
%
% References:
% [1] Gerdan, G.P. & Deakin, R.E., 1999, 'Transforming Cartesian
%     coordinates X,Y,Z to geogrpahical coordinates phi,lambda,h', The
%     Australian Surveyor, Vol. 44, No. 1, pp. 55-63, June 1999.
%--------------------------------------------------------------------------

% calculate flattening f and ellipsoid constant e2
f   = 1/flat;
e2  = f*(2-f);

% compute radii of curvature for the latitude
[rm,rp] = radii(a,flat,lat);

% compute Cartesian coordinates X,Y,Z
p = (rp+h)*cos(lat);
X = p*cos(lon);
Y = p*sin(lon);
Z = (rp*(1-e2)+h)*sin(lat);
```

MATLAB function *radii.m*

```
function [rm,rp] = radii(a,flat,lat)
%
% [rm,rp]=radii(a,flat,lat)  Function computes radii of curvature in
%   the meridian and prime vertical planes (rm and rp respectively) at a
%   point whose latitude (lat) is known on an ellipsoid defined by
%   semi-major axis (a) and denominator of flattening (flat).
%   Latitude must be in radians.
%   Example: [rm,rp] = radii(6378137,298.257222101,-0.659895044);
%            should return rm = 6359422.96233327 metres and
%                          rp = 6386175.28947842 metres
%            at latitude -37 48 33.1234 (DMS) on the GRS80 ellipsoid

%--------------------------------------------------------------------------
% Function:  radii(a,flat,lat)
%
% Syntax:    [rm,rp] = radii(a,flat,lat);
%
% Author:    R.E.Deakin,
%            School of Mathematical & Geospatial Sciences, RMIT University
%            GPO Box 2476V, MELBOURNE, VIC 3001, AUSTRALIA.
%            email: rod.deakin@rmit.edu.au
%            Version  1.0  1 August 2003
%            Version  2.0  6 April  2006
%            Version  3.0  9 February 2008
%
% Purpose:   Function radii() will compute the radii of curvature in
%            the meridian and prime vertical planes, rm and rp respectively
%            for the point whose latitude (lat) is given for an ellipsoid
%            defined by its semi-major axis (a) and denominator of
%            flattening (flat).
%
% Return value: Function radii() returns rm and rp
%
% Variables:
%  a      - semi-major axis of spheroid
%  c      - polar radius of curvature
%  c2     - cosine of latitude squared
%  ep2    - 2nd-eccentricity squared
%  f      - flattening of ellipsoid
%  lat    - latitude of point (radians)
%  rm     - radius of curvature in the meridian plane
%  rp     - radius of curvature in the prime vertical plane
%  V      - latitude function defined by V-squared = sqrt(1 + ep2*c2)
%  V2,V3  - powers of V
%
% Remarks:
%  Formulae are given in [1] (section 1.3.9, page 85) and in
%  [2] (Chapter 2, p. 2-10) in a slightly different form.
%
% References:
% [1] Deakin, R.E. and Hunter, M.N., 2008, GEOMETRIC GEODESY, School of
%        Mathematical and Geospatial Sciences, RMIT University, Melbourne,
%        AUSTRALIA, March 2008.
% [2] THE GEOCENTRIC DATUM OF AUSTRALIA TECHNICAL MANUAL, Version 2.2,
%        Intergovernmental Committee on Surveying and Mapping (ICSM),
%        February 2002 (www.anzlic.org.au/icsm/gdatum)
%---------------------------------------------------------------------------

% compute flattening f eccentricity squared e2
f   = 1/flat;
c   = a/(1-f);
ep2 = f*(2-f)/((1-f)^2);

% calculate the square of the sine of the latitude
c2 = cos(lat)^2;

% compute latitude function V
V2 = 1+ep2*c2;
V  = sqrt(V2);
V3 = V2*V;

% compute radii of curvature
rm = c/V3;
rp = c/V;
```

MATLAB function $DMS.m$

```
function [D,M,S] = DMS(DecDeg)
% [D,M,S] = DMS(DecDeg)  This function takes an angle in decimal degrees and returns
%   Degrees, Minutes and Seconds

val = abs(DecDeg);
D = fix(val);
M = fix((val-D)*60);
S = (val-D-M/60)*3600;
if(DecDeg<0)
  D = -D;
end
return
```

REFERENCES

Bowring, B. R., (1972), 'Distance and the spheroid', Correspondence, *Survey Review,* Vol. XXI, No. 164, April 1972, pp. 281-284.

Clarke, A. R., (1880), *Geodesy,* Clarendon Press, Oxford.

Deakin, R. E., (2009a), 'The Normal Section Curve on an Ellipsoid', Lecture Notes, School of Mathematical & Geospatial Sciences, RMIT University, Melbourne, Australia, November 2009, 53 pages.

——————— (2009b), 'The Curve of Alignment on an Ellipsoid', Lecture Notes, School of Mathematical & Geospatial Sciences, RMIT University, Melbourne, Australia, December 2009, 20 pages.

Thomas, P. D., (1952), *Conformal Projections in Geodesy and Cartography,* Special Publication No. 251, Coast and Geodetic Survey, United States Department of Commerce, Washington, D.C.