Aviation Formulary V1.04

Ed Williams

Section 1: Great Circle Navigation Formulae

This introduction is written for pilots (and others) who are interested in great circle navigation and would like to know how to compute courses, headings and other quantities of interest. These formulae can be programmed into your calculator or spreadsheet. I'll attempt to include enough information that those familiar with plane trigonometry can derive additional results if required.

It is a well known that the shortest distance between two points is a straight line. However anyone attempting to fly from Los Angeles to New York on the straight line connecting them would have to dig a very substantial tunnel first. The shortest distance, *following the earth's surface* lies vertically above the aforementioned straight line route. This route can be constructed by slicing the earth in half with an imaginary plane through LAX and JFK. This plane cuts the (assumed spherical) earth in a circular arc connecting the two points, called a *great* circle. Only planes through the center of the earth give rise to great circles. Other planes will cut the sphere in a circle, but the resulting *little* circle is not the shortest distance between the points it connects. A little thought will show that lines of longitude (meridians) are great circles, but lines of latitude, with the exception of the equator, are not.

I will assume the reader is familiar with latitude and longitude as a means of designating locations on the earth's surface. For the convenience of North Americans I will take North latitudes and West longitudes as positive and South and East negative. The longitude is the opposite of the usual mathematical convention. True course is defined as usual, as the angle between the course line and the local meridian measured clockwise.

The first important fact to realize is that in general a great circle route has a true course that varies from point to point. For instance the great circle route between two points of equal (non-zero) latitude does not follow the line of latitude in an E-W direction, but arcs towards the pole. It is possible to fly between two points using an unvarying true course, but the resulting route differs from, and is longer than, the great circle route and is called a *rhumb* line or loxodrome. Unlike a great circle which is a closed curve encircling the earth, a rhumb line spirals indefinitely poleward.

Natural questions are to seek the great circle distance between two specified points and the true course at points along the route. The required spherical trigonometric formulae are greatly simplified if angles and distances are measured in the appropriate natural units, which are both radians! A radian, by definition, is the angle subtended by a circular arc of unit length and unit radius. Since the length of a complete circular arc of unit radius is 2π, ($\pi = 3.1415926535\ldots$) the conversion is 360 degrees equals 2π radians, or:

$$angle_{rad} = \frac{\pi}{180} angle_{deg}$$

and

$$angle_{deg} = \frac{180}{\pi} angle_{rad}$$

Great circle distance can be likewise be expressed in radians by defining the distance to be the angle subtended by the arc at the center of the earth. Since by definition, one nautical mile subtends one minute (=1/60 degree) of arc, we have:

$$distance_{rad} = \frac{\pi}{10800} distance_{nm}$$

and

$$distance_{nm} = \frac{10800}{\pi} distance_{rad}$$

In the subsequent formulae **all distances and angles, such as latitudes, longitudes and true courses will be assumed to be given in radians**, greatly simplifying them. In applications the above formulae and their inverses are necessary to convert back and forth between natural and practical units. Examples of this process are given later.

1.1: Great Circle Distance between two points

The great circle distance d between two points with the coordinates {lat1,lon1} and {lat2,lon2} respectively is given by:

```
d=acos(sin(lat1)*sin(lat2)+cos(lat1)*cos(lat2)*cos(lon1-lon2))
```

$$d = \arccos\left(\sin(lat_1)\sin(lat_2) + \cos(lat_1)\cos(lat_2)\cos(lon_1 - lon_2)\right)$$

A mathematically equivalent formula, which is less subject to rounding error at short distances is:

```
d=2*asin(sqrt((sin((lat1-lat2)/2))^2 +
                 cos(lat1)*cos(lat2)*(sin((lon1-lon2)/2))^2 ))
```

$$d = 2\arcsin((\sin^2((lat_1 - lat_2)/2) + \cos(lat_1)\cos(lat_2)\sin^2((lon_1 - lon_2)/2))^{1/2})$$

Note on mathematical functions:

^ denotes the exponentiation operator, `sqrt` is the square root function, `acos` the arc-cosine (or inverse cosine) function and `asin` is the arc-sine function.

If asin or acos are unavailable they can be implemented using the atan2 function:

```
acos(x)=atan2(sqrt(1-x^2),x)}
asin(x)=atan2(x,sqrt(1-x^2))}
```

Note that atan2 has the conventional (C) ordering of arguments, namely atan2(y,x). This is not universal, Excel for instance uses atan2(x,y), but it has asin and acos anyway. Be warned!

Should your calculator or programming language be so impoverished that only `atan` is available, then you can implement `atan2` using:

```
atan2(y,x)=atan(y/x)      x,y>0
atan2(y,x)=atan(y/x)+pi   x<0
atan2(y,x)=atan(y/x)+2*pi x>0, y<0
atan2(y,x)=pi/2           x=0, y>0
atan2(y,x)=3*pi/2         x=0, y<0
```

(atan2(0,0) is undefined and should return an error.)

In subsequent formulae the `mod` function is used extensively. It is used to ensure that angles end up in the allowed ranges, ie courses between 0 and 2π (360^o), latitudes between $-\pi/2$ (90^oS) and $\pi/2$ (90^oN) and longitudes between $-\pi$ (180^oE) and π ($180^o W$). For instance, turning right

40^o from 340^o degrees puts you on a heading of not 380 degress, but `mod(340+40,360)=20` degrees.

The mod function is implemented differently in different languages. I define it in the usual mathematical sense: Mod(y,x) is the remainder on dividing y by x and always lies in the range $0 <= \text{Mod}(y, x) < x$. Check in particular that mod(-1.5,2) returns 0.5, not -1.5 or even 0! The following should be bulletproof:

```
function mod(y,x)
 if y>=0
  mod=y- x*int(y/x)
 else
  mod=y+ x*(int(-y/x)+1)
 endif
```

1.2 Great circle course from A to B at A

The initial course, tc1, (at point 1) from point 1 to point 2 is given by:

```
 if sin(lon2-lon1)<0
  tc1=acos((sin(lat2)-sin(lat1)*cos(d))/
         (sin(d)*cos(lat1)))
 else
  tc1=2*pi-acos((sin(lat2)-sin(lat1)*cos(d))/
           (sin(d)*cos(lat1)))
 endif
```

The course at B can be found by interchanging the roles of A and B and taking the reciprocal course.

1.3 Lat/long of a point at a known radial and distance from an initial point

A point lat,lon is a distance d out on the tc radial from point 1 if:

```
lat=asin(sin(lat1)*cos(d)+cos(lat1)*sin(d)*cos(tc))
lon=mod(lon1-asin(sin(tc)*sin(d)/cos(lat))+pi,2*pi)-pi
```

$$lat = \text{arcsin}(\text{sin}(lat_1)\,\text{cos}(d) + \text{cos}(lat_1)\,\text{sin}(d)\,\text{cos}(tc))$$

$$lon = \text{mod}((lon_1 - \text{arcsin}(\text{sin}(tc) * \text{sin}(d)/\,\text{cos}(lat)) + \pi), 2\pi) - \pi$$

The above formulae are limited to $d < pi/4$, i.e distances less than one quarter of the earth's circumference. If greater generality is required, use the following:

```
lat=asin(sin(lat1)*cos(d)+cos(lat1)*sin(d)*cos(tc))
if (lat=0)
  lon=lon1    // endpoint a pole
else
  sindlon=sin(tc)*sin(d)/cos(lat)
  cosdlon=(cos(d)-sin(lat1)*sin(lat))/(cos(lat1)*cos(lat))
  dlon=atan2(sindlon,cosdlon)
  lon=mod( lon1-dlon +pi,2*pi )-pi
endif
```

1.4 Latitude of a point on a great circle given its longitude

Intermediate points {lat,lon} lie on the great circle connecting points 1 and 2 when:

```
lat=atan((sin(lat1)*cos(lat2)*sin(lon-lon2)-
    sin(lat2)*cos(lat1)*sin(lon-lon1))/
   (cos(lat1)*cos(lat2)*sin(lon1-lon2)))
```

$$lat = \arctan((\sin(lat_1)\cos(lat_2)\sin(lon - lon_2) - \sin(lat_2)\cos(lat_1)\sin(lon - lon_1))/(\cos(lat_1)\cos(lat_2)\sin(lon_1 - lon_2)))$$

This is not applicable when the great circle is a meridian, i.e. if $\sin(lon_1 - lon_2) = 0$.

1.5 Lat/lon of an intersection.

To compute the latitude, lat3, and longitude, lon3 of an intersection formed by the crs13 true bearing from point 1 (lat1,lon1) and the crs23 true bearing from point 2 (lat2, lon2):

```
dst12=2*asin(sqrt((sin((lat1-lat2)/2))^2+
     cos(lat1)*cos(lat2)*(sin((lon1-lon2)/2))^2 ))
IF sin(lon2-lon1)<0
  crs12=acos((sin(lat2)-sin(lat1)*cos(dst12))/
        (sin(dst12)*cos(lat1)))
ELSE
  crs12=2.*pi-acos((sin(lat2)-sin(lat1)*cos(dst12))/
        (sin(dst12)*cos(lat1)))
ENDIF
IF sin(lon1-lon2)<0
  crs21=acos((sin(lat1)-sin(lat2)*cos(dst12))/
        (sin(dst12)*cos(lat2)))
ELSE
  crs21=2.*pi-acos((sin(lat1)-sin(lat2)*cos(dst12))/
        (sin(dst12)*cos(lat2))))
ENDIF
ang1=abs(mod(crs13-crs12+pi,2.*pi)-pi)
ang2=abs(mod(crs21-crs23+pi,2.*pi)-pi)
if (sin(ang1)*sin(ang2)<=sqrt(TOL))
  ‘‘no intersection exists’’
else
ang3=acos(-cos(ang1)*cos(ang2)+
      sin(ang1)*sin(ang2)*cos(dst12))
dst13=asin(sin(ang2)*sin(dst12)/sin(ang3))
dst23=asin(sin(ang1)*sin(dst12)/sin(ang3))
lat3=asin(sin(lat1)*cos(dst13)+
      cos(lat1)*sin(dst13)*cos(crs13))
lon3=mod(lon1-asin(sin(crs13)*sin(dst13)/
      cos(lat3))+pi,2*pi)-pi
```

```
endif
```

(TOL is a number of order machine precision. 10^{-15} would be fine for standard double precision arithmetic)

1.6 Relation between true course and latitude on a great circle, and highest latitude reached.

Clairaut's formula: This relates the latitude (lat) and true course (tc) along any great circle, namely:

```
sin(tc)*cos(lat)=constant
```

that is, for any two points on a Great Circle:

```
sin(tc1)*cos(lat1)=sin(tc2)*cos(lat2)
```

since at the highest latitude (latmx) reached the tc must be 90/270, we also have:

```
latmx=acos(abs(sin(tc)*cos(lat)))
```

where lat and tc are the latitude and true course at *any* point on the great circle.

1.7 Crossing parallels

Any given great circle (excepting one over the poles) crosses each meridian once and only once. However, any given great circle has a maximum latitude reached at its apex. It crosses lower latitudes twice and never crosses higher latitudes. Thus the algorithm for finding the longitudes at which a given great circle crosses a given parallel is a little more complex.

Suppose a great circle passes through (lat1,lon1) and (lat2,lon2). It crosses the parallel lat3 at longitudes lon3_1 and lon3_2 given by:

```
l12 = lon1-lon2
A = sin(lat1)*cos(lat2)*cos(lat3)*sin(l12)
B = sin(lat1)*cos(lat2)*cos(lat3)*cos(l12) -
        cos(lat1)*sin(lat2)*cos(lat3)
C = cos(lat1)*cos(lat2)*sin(lat3)*sin(l12)
lon = atan2(B,A)    \\ atan2(y,x) convention
if (C >sqrt(A^2 + B^2))
"no crossing"
else
dlon = acos(C/sqrt(A^2+B^2))
lon3_1=mod(lon1+dlon+lon+pi, 2*pi)-pi
lon3_2=mod(lon1-dlon+lon+pi, 2*pi)-pi
endif
```

1.8 Cross Track Error

Suppose you are proceeding on a great circle route from A to B (course =crs_AB) and end up at D, perhaps off course. You can calculate the course from A to D (crs_AD) and the distance from A to D (dist_AD) using the formulae in Sections 1.2 and 1.1 . In terms of these, the cross track error, XTD, (distance off course) is given by:

```
XTD =asin(sin(dist_AD)*sin(crs_AD-crs_AB))
```

Positive XTD means right of course, negative means left.

1.9 Some worked examples.

Suppose point 1 is LAX: (33deg 57min N, 118deg 24min W)
Suppose point 2 is JFK: (40deg 38min N, 73deg 47min W)
In *radians* the latitude and longitude of LAX are:
latitude=(33+57/60)*pi/180= 0.592539,
longitude=(118+24/60)*pi/180= 2.066470,
and those of JFK are (0.709186,1.287762)

The distance from LAX to JFK is given by (Sec 1.1):

```
d=acos(sin(lat1)*sin(lat2)+
       cos(lat1)*cos(lat2)*cos(lon1-lon2))
 =acos(sin(0.592539)*sin(0.709186)+
       cos(0.592539)*cos(0.709186)*cos(0.778708))
 =acos(0.811790)
 =0.623585 radians
 =0.623585*180*60/pi=2144nm
```

The initial true course out of LAX is given by (sec 1.2):

```
sin(-0.778708)=-0.702<0 so

tc1=acos((sin(lat2)-sin(lat1)*cos(d))/
         (sin(d)*cos(lat1)))
   =acos((sin(0.709186)-
          sin(0.592539)*cos(0.623585))/
          (sin(0.623585)*cos(0.592535))
   =acos(0.408455)
   =1.150035 radians
        =66 degrees
```

An enroute waypoint 100nm from LAX on the 66 degree radial (100nm along the GC to JFK) has lat and long given by (Sec 1.3):

```
100nm= 100*pi/(180*60)=0.0290888radians

 lat=asin(sin(lat1)*cos(d)+
          cos(lat1)*sin(d)*cos(tc))
   =asin(sin(0.592539)*cos(0.020888)+
          cos(0.592539)*sin(0.020888)*cos(1.150035))
   =asin(0.568087)
   =0.604180 radians
   =34 degrees 37 min N

 lon=mod(lon1-asin(sin(tc)*sin(d)/cos(lat))+pi, 2*pi)-pi
   =mod(2.066470-
   asin(sin(1.150035)*sin(0.020888)/cos(0.604180))+pi,2*pi)-pi
   =mod(2.034206+pi, 2*pi)-pi
   =2.034206 radians
```

```
=116 degrees 33 min W
```

The great circle route from LAX to JFK crosses the 111 degree W meridian at a latitude of (Sec 1.4) :

```
(111 degrees=1.937315 radians)

 lat=atan((sin(lat1)*cos(lat2)*sin(lon-lon2)
    -sin(lat2)*cos(lat1)*sin(lon-lon1))/
    (cos(lat1)*cos(lat2)*sin(lon1-lon2)))
   =atan((sin(0.592539)*cos(0.709186)*sin(0.649553)
    -sin(0.709186)*cos(0.592539)*sin(-0.129154))/
    (cos(0.592539)*cos(0.709186)*sin(0.778708)))

   =atan(0.737110)
   =0.635200radians
   =36 degrees 24 min N
```

Cross track error (Sec 1.8):

Enroute from JFK to LAX, you find yourself at the point, D, N$34^o 30'$ W$116^o 30'$, which in radians is (0.6021386,2.033309) (See earlier example for LAX, JFK coordinates and course)

From LAX to D the distance is:

```
dist_AD=acos(sin(0.592539)*sin(0.6021386)+
      cos(0.592539)*cos(0.6021386)*
      cos(2.066470-2.033309))
     =0.02905 radians (99.8665 nm)
```

From LAX to D the course is:

```
crs_AD=acos((sin(0.6021386)-sin(0.592539)*cos(0.02905))/
     (sin(0.02905)*cos(0.592539)))
    =1.22473 radians  (70.17 degrees)
```

At point D the cross track error is:

```
xtk= asin(sin(0.02905)*sin(1.22473-1.15003))
  = 0.00216747 radians
  = 0.00216747*180*60/pi =7.4512 nm right of course
```

Intersection example (in brief) (Sec 1.5):

Let point 1 be REO (42.60N,117.866W)=(0.74351,2.05715)rad

Let point 2 be BKE (44.84N,117.806W)=(0.782606,2.056103)rad

The 51 degree (=0.890118 radians) bearing from REO intersects with 137 degree (=2.391101 radians) from BKE at (lat3,lon3):

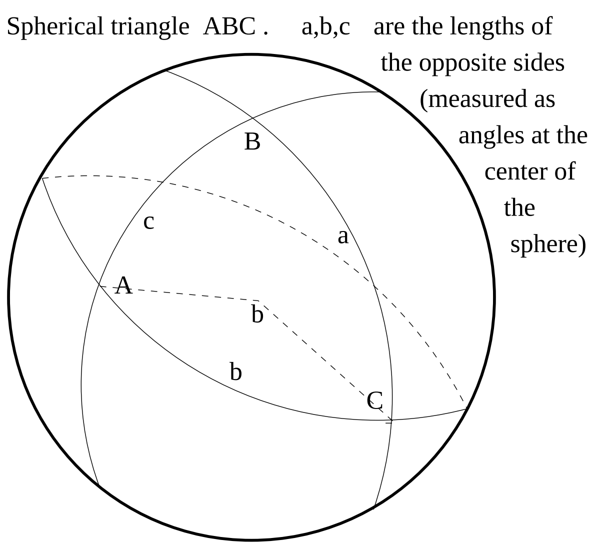
Then:

```
dst12=0.039103
crs12=0.018996
crs21=3.161312
```

```
ang1=0.871122
ang2=0.770211
ang3=1.500667
dst13=0.02729
dst23=0.029986
lat3=0.760473   =43.5N
lon3=2.027876   =116.2W at BOI!
```

1.9 Some general spherical triangle formulae

A spherical triangle is one whose sides are all great circular arcs. Let the sides have lengths a,b and c *radians*, and the opposite angles be A, B and C *radians*. Recall that the length of a side in *radians* is its length divided by the radius of the sphere.



(None of the angles need be right angles.)
The following relations apply:

$$\frac{\sin(a)}{\sin(A)} = \frac{\sin(b)}{\sin(B)} = \frac{\sin(c)}{\sin(C)}$$

$$\cos(a) = \cos(b) * \cos(c) + \sin(b)\sin(c)\cos(A)$$

$$\cos(b) = \cos(c)\cos(a) + \sin(c)\sin(a)\cos(B)$$

$$\cos(c) = \cos(a)\cos(b) + \sin(a)\sin(b)\cos(C)$$

$$\cos(A) = -\cos(B)\cos(C) + \sin(B)\sin(C)\cos(a)$$

$$\cos(B) = -\cos(C)\cos(A) + \sin(C)\sin(A)\cos(b)$$

$$\cos(C) = -\cos(A)\cos(B) + \sin(A)\sin(B)\cos(c)$$

Using the above, given *any* three of the sides or angles $\{a, b, c, A, B, C\}$, it is possible to find the remaining sides and angles from the above formulae.

Note that for a spherical triangle the three angles add up to more than π radians or 180 degrees.

1.10 Rhumb line formulae

Rhumb lines or loxodromes are tracks of constant true course. Except for N/S courses, they differ from great circles, which are shorter, but have varying true course.

When two points (lat1,lon1), (lat2,lon2) are connected by a rhumb line with true course, tc:

```
lon2-lon1=-tan(tc)*(log((1+sin(lat2))/cos(lat2))+
             log((1+sin(lat1))/cos(lat1)))
```

(logs are "natural" logarithms to the base e)

The dist, d, between the points is given by:

```
d= abs((lat2-lat1)/cos(tc))    (cos(tc) > 0)
 = mod(lon2-lon1,2*pi)*cos(lat1)  (tc=3*pi/2, lat1=lat2)
 = mod(lon1-lon2.2*pi)*cos(lat1)  (tc=pi/2, lat1=lat2)
```

To figure the rhumb line course, tc, and distance, d, from (lat1,lon1) to (lat2,lon2) allowing for the possibility of date-line crossings and E-W courses (neither point a pole!):

```
dlon_W=mod(lon2-lon1,2*pi)
dlon_E=mod(lon1-lon2,2*pi)
dphi=log((1+sin(lat2))/cos(lat2))
   -log((1+sin(lat1))/cos(lat1))
if (dlon_W<dlon_E){// West is the shortest
tc=mod(atan2(-dlon_W,dphi),2*pi
 }
else{
 tc=mod(atan2(dlon_E,dphi),2*pi)
 }
if (abs(lat1-lat2)<sqrt(tol)){
 d=min(dlon_W,dlon_E)*cos(lat1) // distance along parallel
 }
else{
 d=abs((lat2-lat1)/cos(tc))
 }
}
```

tol should be approximately machine precision, 10^{-15} should be OK for normal double precision arithmetic.

To find the lat/lon of a point on true course tc, distance d from (lat1,lon1) along a rhumbline (initial point cannot be a pole!):

```
lat = lat1+d*cos(tc);
dphi= log((1+sin(lat))/cos(lat))
   -log((1+sin(lat1))/cos(lat1))
if (abs(cos(tc)) > sqrt(tol)){
 dlon=dphi*tan(tc)
```

```
  }
 else{ // along parallel
  dlon=sin(tc)*d/cos(lat1)
  }
  lon =mod(lon1-dlon+pi,2*pi)-pi
 }
```

1.10 Rhumb line: numerical example

Example: rhumb line course from LAX to JFK: LAX lat/lon=(0.592539,2.066470) and JFK lat/lon is (0.709186,1.287762)

```
dlon_W = mod(1.287762-2.066470,2*pi)=5.50448
dlon_E = mod(2.066470-1.287762,2*pi)=0.778708
 dphi = log((1+sin(0.709186))/cos(0.709186))
    - log((1+sin(0.592539))/cos(0.592539))
    = 0.146802
   tc = mod(atan2(0.778708,0.146802),2*pi)
    = 1.38446
    = 79.3 degrees
   d = abs((0.709186-0.592539)/cos(1.38446))
    = 0.62965
    = 2164.6nm
```

Compare this with the great circle course of 66 degrees and distance of 2144 nm.

Section 2: Wind Triangle Formulae

In all formulae, all angles are in radians. Convert back and forth as in the Great Circle section. (This is unnecessary on calculators which have a "degree mode" for trig functions. Most programming languages provide only "radian mode".) As before:

```
angle_radians=(pi/180)*angle_degrees
angle_degrees=(180/pi)*angle_radians
```

and if using dd:mm:ss

```
angle_degrees=degrees+(minutes/60.)+(seconds/3600.)
   degrees=int(angle_degrees)
   minutes=int(60*(angle_degrees-degrees))
  seconds=int(3600*(angle_degrees-degrees-60*minutes))
```

You may have a HH <-> HH:MM:SS conversion to do this efficiently.

Let CRS=course, HD=heading, WD=wind direction from, TAS=True airspeed, GS=ground speed, WS=wind speed. The units of the speeds do not matter as long as they are all the same.

2.1: Unknown Wind

```
WS=sqrt( (TAS-GS)^2+ 4*TAS*GS*(sin((HD-CRS)/2))^2 )
WD=CRS + atan2(TAS*sin(HD-CRS), TAS*cos(HD-CRS)-GS) (**)
if (WD<0) then WD=WD+2*pi
if (WD>2*pi) then WD=WD-2*pi
```

((**) assumes atan2(y,x), reverse arguments if your implementation has atan2(x,y). See Great Circle section if you don't have atan2)

2.2: Find HD and GS

```
SWC=(WS/TAS)*sin(WD-CRS)
 if (abs(SWC)>1) then "course cannot be flown-- wind too strong"
 else HD=CRS+asin(SWC)
    if (HD<0) HD=HD+2*pi
    if (HD>2*pi) HD=HD-2*pi
    GS=TAS*sqrt(1-SWC^2)-WS*cos(WD-CRS)
 endif
```

Note:

The purpose of the "if (HD<0) HD=HD+2*pi; if (HD>2*pi) HD=HD-2*pi" is to ensure the final heading ends up in the range (0, 2*pi). Another way to do this, with the MOD function (see earlier) available is: HD=MOD(HD,2*pi)