Straight-line walk direction between two points on earth

An article posted on bathtubphysics.de

Jan Bergner

2022-12-27



The problem

Maybe we are hiking or maybe we are sailing the ocean. Suppose we just got your GPS location and we know the coordinates of our destination, as well.



Figure 1: Suppose we are at startpoint S and want to reach target T on earth's surface. The shortest possible path is along the red great circle.

In the sketch, we are sitting at point S on earth's surface and want to reach target point T.

The shortest possible path is along the red great circle. This great circle is what we need to find.

What is our optimal course (or heading) to reach our target on the shortest possible path?

To answer this question, we simplify it by assuming no obstacles in our path. We can always follow a straight line. (Or rather, a great circle, since earth is approximately a sphere.)

Definitions

Suppose, you are on earth's surface at a starting point $\vec{S} = R\vec{S_0}$ with latitude β_S and longitude λ_S . R is earth's radius.

Your target is $\vec{T}(\lambda_T, \beta_T) =: R\vec{T_0}$.

Notice, that the "auxiliary vectors" $\vec{S_0}$ and $\vec{T_0}$ are pointing in the same direction as \vec{S} and \vec{T} , respectively, but are shortened to a length of 1. We call these **normalized location vectors**.



Figure 2: The situation. Suppose you are located at S and want to reach a point T. (T_A or T_B in this case.)

The above sketch illustrates the situation. We see the great circle around earth, defined by our current location and our destination. The current location is S (the olive point to the left) and we want to reach the point T (red). However, we need to consider *two* cases, seperatly.

That's why this sketch shows *two* points T_A and T_B . The former is closer to you than a quarter of earth's circumference (left to the dotted green line) and the latter is further away (right to the dotted green line). In other words, we consider the cases $\angle \vec{S}, \vec{T} < 90^\circ$ and $\angle \vec{S}, \vec{T} > 90^\circ$, seperately.

If your location S were the north pole, T_A would correspond to your destination also being in the northern hemisphere while T_B would be in the southern hemisphere. The dotted green line would be the equator.

The dashed olive line is the tangent to the great circle through our current position S.

With this, any vector $\vec{D'}$ pointing from S_0 in the direction of our target will be parallel to this line.

Furthermore, this vector $\vec{D'}$ will be perpendicular to our own location vector \vec{S} , since it is a tangent vector.

Knowing any valid "direction vector" $\vec{D'}$ that can uniquely specify the tangent, all we have to do is find out whether to go "up" or "down". (I. e. the proper *orientation* for $\vec{D'}$.)

Let us start by constructing an *auxiliary direction vector* $\vec{D'}$ by making the target vector \vec{T} "longer", so it will hit the tangent straight line specified by $\vec{D'}$ in point T'_A or T'_B (the orange points in the sketch), respectively:

$$\vec{D'} = \vec{T'} - \vec{S}.$$

Now, just look at the two situations. $\vec{T_B}$ points away from the tangent and therefore, we will need to flip it (i. e. multiply by -1) in addition to making it longer.

Unfortunately, in this case the auxiliary direction vector $\vec{D'}$ will point in the direction opposite to the shortest path along the great circle. We will have to flip this one, as well.

For situation A however, it all works out fine and $\vec{D'}$ has the right orientation.

Constructing the proper direction vector

Let us take a step back for now and find out how to rescale the vector \vec{T} , properly.

We can set

$$\vec{T'} = \frac{1}{\eta} \vec{T}$$

and make η the scaling factor. (Putting it into the denominator instead of the numerator will prove beneficial going forward. But of course, you could also just put it as a factor in front of \vec{T} .)

That way, we can require orthogonality with \vec{S} for

$$\vec{D'} = \frac{1}{\eta} \vec{T} - \vec{S}$$

like so:

$$0 \stackrel{!}{=} \vec{S} \cdot \vec{D'}$$
$$= \frac{1}{\eta} \vec{T} \cdot \vec{S} - \vec{S}^2$$
$$= R^2 \left[\frac{1}{\eta} \vec{T_0} \cdot \vec{S_0} - 1 \right]$$
$$\Leftrightarrow \ \eta = \vec{T_0} \cdot \vec{S_0}$$

This is a nice and simple result. η is just the scalar product of the two normalized location vectors. Incidently, this is equal to the cosine of the angle between the two vectors:

$$\eta = \cos\left(\angle \vec{S_0}, \vec{T_0}\right)$$

This means that η actually could be equal to 0! (If the angle is exactly equal to 90°.)

Then, our auxiliary direction vector strictly speaking is undefined due to a 0 in the denominator.

However, this is just due to our construction of the direction vector and not a "real" singularity. If we move our target or start point an infinitely small step along the great circle, this won't be a problem, anymore and thus, we ignore this complication for now.

The next step is to make sure, the direction vector actually has the right orientation.

As stated before, we can do that by flipping the vector $\vec{D'}$, if our target point is in "the other hemi-sphere".

This is the case when $\eta < 0$ and thus, we can simply use the **sign function** $\sigma(\eta)$, defined as:

$$\sigma(\eta) = \begin{cases} -1 & \eta < 0\\ 0 & \eta = 0\\ +1 & \eta > 0 \end{cases}$$

The definition $\sigma(0) = 0$ is only a formality as we had excluded this case, before. Notice, that this implies

$$\eta = \sigma(\eta) \cdot |\eta| \,.$$

With this definition, we can make sure, that a direction vector \vec{D} will have the proper orientation by setting:

$$\vec{D} = \sigma(\eta) \vec{D'} = \sigma(\eta) \left(\frac{1}{\eta} \vec{T} - \vec{S}\right)$$

Let us normalize the direction vector to a length of 1:

$$\begin{split} \vec{D_0} &= \frac{1}{\left| \vec{D} \right|} \cdot \vec{D} \\ &= \sigma(\eta) \frac{\frac{1}{\eta} \vec{T} - \vec{S}}{\sqrt{(\sigma(\eta))^2 \left(\frac{1}{\eta} \vec{T} - \vec{S}\right)^2}} \\ &= \sigma(\eta) \frac{R \left(\frac{1}{\eta} \vec{T_0} - \vec{S_0}\right)}{\sqrt{R^2 \left(\frac{1}{\eta} \vec{T_0} - \vec{S_0}\right)^2}} \\ &= \sigma(\eta) \frac{\frac{1}{\eta} \vec{T_0} - \vec{S_0}}{\sqrt{\left(\frac{1}{\eta} \vec{T_0} - \vec{S_0}\right)^2}} \\ &= \frac{\sigma(\eta)}{\eta} \frac{\vec{T_0} - \eta \vec{S_0}}{\sqrt{\frac{1}{\eta^2} \vec{T_0}^2 - \frac{2}{\eta} \vec{T_0} \vec{S_0} + \vec{S_0}^2}} \\ &= \frac{\sigma(\eta)}{\sigma(\eta)} \frac{\vec{T_0} - \eta \vec{S_0}}{|\eta| \sqrt{\frac{1}{\eta^2} - \frac{2}{\eta} \cdot \eta + 1}} \\ &= \frac{\vec{T_0} - \eta \vec{S_0}}{\sqrt{\eta^2 \left(\frac{1}{\eta^2} - 1\right)}} \\ &= \frac{\vec{T_0} - \eta \vec{S_0}}{\sqrt{1 - \eta^2}} \end{split}$$

Thus, the **normalized direction vector** $\vec{D_0}$ is:

$$\boxed{\vec{D_0} = \frac{\vec{T_0} - \eta \vec{S_0}}{\sqrt{1 - \eta^2}}}$$

Notice, this vector is well-defined for $\eta=0!$

In fact, if $\eta = 0$, $\vec{D_0} = \vec{T_0}$, which is indeed, the right answer to our question of which direction to go, also considering orientation.

Isn't that neat? With the normalized direction vector $\vec{D_0}$, we get rid of the distinction between the cases, indicated by $\sigma(\eta)$ while still determining the right direction. Furthermore, normalization is inherently useful, itself, and as a bonus, we have handled the previously excluded case of $\eta = 0$ as a "by-product".

I guess, this is what is called killing three birds with one stone!

Position vectors in spherical coordinates

We have been talking quite a bit about the vectors \vec{S} and \vec{T} that specify the location of a corresponding point on earth's surface.

Now, let us express them in terms of longitude λ and latitude β .

There are multiple possible conventions on how to choose two angles λ and β that uniquely describe a point on a sphere's surface, but there is a standing convention for earth.



Figure 3: Sketch of earth's coordinate frame.

In the sketch, the 0° -meridian (see below) is green, while the equator is dark blue. The coordinates λ and β describe point P.

In particular, picture a cartesian coordinate frame whoose origin coincides with earth's center. The *z*-axis passes through the poles, going from south to north.

The equator resides in the x-y-plane.

Next, we can define the latitude β as the angle between a location vector (like \vec{P} , \vec{S} or \vec{T}) and the *x*-*y*-plane. By convention, β is positive if we describe a point in the northern hemisphere and it is

negative if we deal with the southern hemisphere.

With this convention, we can express the *z*-coordinate as

 $z = R\sin\beta.$

Lastly, we will have to decide on where to place our *x*- and *y*-axes.

Once again, there is a convention based on history. In particular, the British Royal Navy had an observatory in Greenwich, London and defined that to be on the 0° - or prime-meridian. (I. e. they decided that the semi-circle-arc from the north pole to the south pole through that observatory should be 0 degrees longitude.)

For some technical reasons, the modern reference meridian slightly varies from that historic choice, but still, this is the origin of the modern convention.

Anyways, the 0° -meridian will hit the equator at some point, which is the point the *x*-axis of our coordinate frame will go through. In particular, it will come from outside earth, hit the point on earth's surface that is exactly opposite to the 0° -meridian-equator-intersection (i. e. the 180° -meridian-equator-intersection), will go through earth's center, i. e. the coordinate frame's origin and will leave earth again at the 0° -meridian-equator-intersection.

The *y*-axis will then be orthogonal to the *x*- and *z*-axes and also go through the origin. As is, again, convention, when looking in positive *x*-direction, the *y*-axis will come from the right and go to the left. (Meaning it hits earth at the 270° -meridian, goes through earth's center and leaves through the 90° -meridian.)

Thus, if we start at a point on the x-axis that has a positive coordinate value, we can rotate it 90° counterclockwise around the z-axis to reach a point on the y-axis with an also positive coordinate. Rotating again gets us to the negative x-axis.

This convention means that our longitude angle will increase when we walk to the east. What's more, A full circle has 360° and therefore, the 360° -meridian will be identical to the 0° -meridian.

If someone would be interested in more details, they might as well start on wikipedia.

We can finally describe our locations S and T in terms of longitude and latitude. The following is the example for S. T is analogous. (Just replace the index S by an index T.)

$$\vec{S} = R \begin{pmatrix} \cos \lambda_S \cdot \cos \beta_S \\ \sin \lambda_S \cdot \cos \beta_S \\ \sin \beta_S \end{pmatrix} = R \cdot \vec{S_0}$$

Local two-dimensional coordinate frame in ${\boldsymbol S}$

Let us get back to our actual problem.

From a navigational perspective, we cannot move freely in three dimensions but only two, since we are constrained to earth's surface.

Thus, it makes sense to define a local two-dimensional coordinate frame at our current location S.

To be precise, we can define a **local north vector** that will point along earth's surface to the north pole and similarly, a **local east vector** that points right to the east. The former will mean the longitude λ_S being kept constant while the latter means an unchanged latitude β_S .

That makes the north vector $\vec{N_S}$ simply the vector increasing β_S , i. e. it is the derivative of \vec{S} with respect to β_S .

We can also use $\vec{S_0}$ instead of \vec{S} to get a normalized north vector, right away.

$$\vec{V}_{S} = \frac{\partial}{\partial\beta_{S}} \vec{S}_{0}$$
$$= \frac{\partial}{\partial\beta_{S}} \begin{pmatrix} \cos\lambda_{S} \cdot \cos\beta_{S} \\ \sin\lambda_{S} \cdot \cos\beta_{S} \\ \sin\beta_{S} \end{pmatrix}$$
$$= \begin{pmatrix} -\cos\lambda_{S} \cdot \sin\beta_{S} \\ -\sin\lambda_{S} \cdot \sin\beta_{S} \\ \cos\beta_{S} \end{pmatrix}$$

To prove that this vector is normalized, we can square it to find its length to be 1:

$$\vec{N_S}^2 = \left[\begin{pmatrix} -\cos\lambda_S \cdot \sin\beta_S \\ -\sin\lambda_S \cdot \sin\beta_S \\ \cos\beta_S \end{pmatrix} \right]^2$$
$$= (-\cos\lambda_S \cdot \sin\beta_S)^2$$
$$+ (-\sin\lambda_S \cdot \sin\beta_S)^2$$
$$+ (\cos\beta_S)^2$$
$$= \cos^2\lambda_S \cdot \sin^2\beta_S$$
$$+ \sin^2\lambda_S \cdot \sin^2\beta_S$$
$$+ \cos^2\beta_S$$

We can collect like terms and apply the Pythagorean trigonometric identity

 $\sin^2 x + \cos^2 x = 1:$

$$\vec{N_S}^2 = \cos^2 \lambda_S \cdot \sin^2 \beta_S$$
$$+ \sin^2 \lambda_S \cdot \sin^2 \beta_S$$
$$+ \cos^2 \beta_S$$
$$= \left[\cos^2 \lambda_S + \sin^2 \lambda_S\right] \cdot \sin^2 \beta_S + \cos^2 \beta_S$$
$$= \sin^2 \beta_S + \cos^2 \beta_S$$
$$= 1$$

Indeed, $\vec{N_S}$ is normalized.

Now, let us calculate the local east vector $\vec{E'_S}$, again using the derivative of $\vec{S_0}$, but this time with respect to λ_S :

$$\vec{E'_S} = \frac{\partial}{\partial \lambda_S} \vec{S_0}$$
$$= \frac{\partial}{\partial \lambda_S} \begin{pmatrix} \cos \lambda_S \cdot \cos \beta_S \\ \sin \lambda_S \cdot \cos \beta_S \\ \sin \beta_S \end{pmatrix}$$
$$= \begin{pmatrix} -\sin \lambda_S \cdot \cos \beta_S \\ \cos \lambda_S \cdot \cos \beta_S \\ 0 \end{pmatrix}$$

To normalize this one, we have to divide by $\cos\beta_S$:

$$\vec{E}_{S} = \frac{1}{\cos\beta_{S}}\vec{E}_{S}'$$
$$= \begin{pmatrix} -\sin\lambda_{S} \\ \cos\lambda_{S} \\ 0 \end{pmatrix}$$

Let's see, whether these two vectors are orthogonal:

$$\vec{N_S} \cdot \vec{E_S} = \begin{pmatrix} -\cos\lambda_S \cdot \sin\beta_S \\ -\sin\lambda_S \cdot \sin\beta_S \\ \cos\beta_S \end{pmatrix} \cdot \begin{pmatrix} -\sin\lambda_S \\ \cos\lambda_S \\ 0 \end{pmatrix}$$
$$= \begin{bmatrix} -\cos\lambda_S \cdot \sin\beta_S \end{bmatrix} \cdot \begin{bmatrix} -\sin\lambda_S \end{bmatrix}$$
$$+ \begin{bmatrix} -\sin\lambda_S \cdot \sin\beta_S \end{bmatrix} \cdot \begin{bmatrix} \cos\lambda_S \end{bmatrix}$$
$$+ \cos\beta_S \cdot 0$$
$$= \cos\lambda_S \cdot \sin\beta_S \cdot \sin\lambda_S - \sin\lambda_S \cdot \sin\beta_S \cdot \cos\lambda_S$$
$$= 0$$

Indeed, the normalized local noth- and east-vector are perpendicular and we can use them to define a

local two-dimensional coordinate frame at point S to define directions on earth's surface.

To be precise, these vectors will span a tangent plane to earth's surface through S. Since our normalized direction vector $\vec{D_0}$ also specifies a tangent straight line, it must reside inside this plane.

Finally, we can check the scalar products between the local north or east vector and the normalized position vector $\vec{S_0}$.

$$\vec{N}_{S} \cdot \vec{S}_{0} = \begin{pmatrix} -\cos \lambda_{S} \cdot \sin \beta_{S} \\ -\sin \lambda_{S} \cdot \sin \beta_{S} \\ \cos \beta_{S} \end{pmatrix} \cdot \begin{pmatrix} \cos \lambda_{S} \cdot \cos \beta_{S} \\ \sin \lambda_{S} \cdot \cos \beta_{S} \\ \sin \beta_{S} \end{pmatrix}$$
$$= \begin{bmatrix} -\cos \lambda_{S} \cdot \sin \beta_{S} \end{bmatrix} \cdot \begin{bmatrix} \cos \lambda_{S} \cdot \cos \beta_{S} \end{bmatrix}$$
$$+ \begin{bmatrix} -\sin \lambda_{S} \cdot \sin \beta_{S} \end{bmatrix} \cdot \begin{bmatrix} \sin \lambda_{S} \cdot \cos \beta_{S} \end{bmatrix}$$
$$+ \begin{bmatrix} \cos \beta_{S} \end{bmatrix} \cdot \begin{bmatrix} \sin \beta_{S} \end{bmatrix}$$
$$= -\cos^{2} \lambda_{S} \cdot \sin \beta_{S} \cdot \cos \beta_{S}$$
$$- \sin^{2} \lambda_{S} \cdot \sin \beta_{S} \cdot \cos \beta_{S}$$
$$+ \cos \beta_{S} \cdot \sin \beta_{S} \cdot \cos \beta_{S}$$
$$= \cos \beta_{S} \cdot \sin \beta_{S} \cdot \left[-\cos^{2} \lambda_{S} - \sin^{2} \lambda_{S} + 1 \right]$$
$$= 0$$

=0

Now, we check the east vector:

$$\vec{E}_{S} \cdot \vec{S}_{0} = \begin{pmatrix} -\sin \lambda_{S} \\ \cos \lambda_{S} \\ 0 \end{pmatrix} \cdot \begin{pmatrix} \cos \lambda_{S} \cdot \cos \beta_{S} \\ \sin \lambda_{S} \cdot \cos \beta_{S} \\ \sin \beta_{S} \end{pmatrix}$$
$$= \cos \beta_{S} \cdot \underbrace{[-\sin \lambda_{S} \cdot \cos \lambda_{S} + \cos \lambda_{S} \cdot \sin \lambda_{S}]}_{=0}$$

So, the local north and east vector are orthogonal to $\vec{S_0}$ and to each other.

Summary for the local coordinate frame

We found the normalized local north and east vectors to be:

$$\vec{N_S} = \begin{pmatrix} -\cos\lambda_S \cdot \sin\beta_S \\ -\sin\lambda_S \cdot \sin\beta_S \\ \cos\beta_S \end{pmatrix}$$
$$\vec{E_S} = \begin{pmatrix} -\sin\lambda_S \\ \cos\lambda_S \\ 0 \end{pmatrix}$$

They are normalized:

$$\vec{N_S}^2 = \vec{E_S}^2 = \vec{S_0}^2 = 1$$

And they are orthogonal to each other and to $\vec{S_0}$:

$$\vec{N_S} \cdot \vec{E_S} = \vec{N_S} \cdot \vec{S_0} = \vec{E_S} \cdot \vec{S_0} = 0$$

Actually finding the right heading

We can now rewrite the normalized direction vector $\vec{D_0}$ as a linear combination of the local north and east vector:

$$\vec{D_0} =: \nu \vec{N_S} + \varepsilon \vec{E_S}$$

Since $\vec{N_S} \cdot \vec{E_S} = 0$, we can find the parameters ν and ε using the following products:

$$\vec{D_0} \cdot \vec{N_S} = \nu \underbrace{\vec{N_S}^2}_{0} + \varepsilon \underbrace{\vec{E_S} \cdot \vec{N_S}}_{0} \quad \Leftrightarrow \quad \nu = \vec{D_0} \cdot \vec{N_S}$$
$$\vec{D_0} \cdot \vec{E_S} = \nu \underbrace{\vec{N_S} \cdot \vec{E_S}}_{0} + \varepsilon \underbrace{\vec{E_S}^2}_{1} \quad \Leftrightarrow \quad \varepsilon = \vec{D_0} \cdot \vec{E_S}$$

We can insert for $\vec{D_0}$. (We do both equations at once.)

$$\begin{split} \stackrel{\nu}{\varepsilon} &| = \vec{D_0} \cdot \left| \stackrel{\vec{N_S}}{\vec{E_S}} \right| \\ &= \frac{1}{\sqrt{1 - \eta^2}} \left(\vec{T_0} - \eta \vec{S_0} \right) \cdot \left| \stackrel{\vec{N_S}}{\vec{E_S}} \right| \\ &= \frac{1}{\sqrt{1 - \eta^2}} \left(\vec{T_0} \cdot \left| \stackrel{\vec{N_S}}{\vec{E_S}} \right| - \eta \cdot \underbrace{\vec{S_0} \cdot \left| \stackrel{\vec{N_S}}{\vec{E_S}} \right|}_{0} \right) \end{split}$$

The second summand is 0 by perpendicularity.

Side note:

Let us replace $\vec{N_S}$ and $\vec{E_S}$ by their derivative-definition. (Notice, $\vec{T_0}$ is independent of the derivative variables.)

$$\begin{aligned} \begin{aligned} \xi' &= \frac{1}{\sqrt{1 - \eta^2}} \vec{T_0} \cdot \left| \frac{\vec{N_S}}{\vec{E_S}} \right| \\ &= \frac{1}{\sqrt{1 - \eta^2}} \vec{T_0} \cdot \left| \frac{\frac{\partial}{\partial \beta_S} \vec{S_0}}{\frac{1}{\cos \beta_S} \frac{\partial}{\partial \lambda_S} \vec{S_0}} \right| \\ &= \frac{1}{\sqrt{1 - \eta^2}} \cdot \left| \frac{\frac{\partial}{\partial \beta_S} (\vec{T_0} \cdot \vec{S_0})}{\frac{1}{\cos \beta_S} \frac{\partial}{\partial \lambda_S} (\vec{T_0} \cdot \vec{S_0})} \right| \\ &= \frac{1}{\sqrt{1 - \eta^2}} \cdot \left| \frac{\frac{\partial}{\partial \beta_S} \eta}{\frac{1}{\cos \beta_S} \frac{\partial}{\partial \lambda_S} \eta} \right| \end{aligned}$$

Thus, we finally obtain:

ν =	$=\frac{1}{\sqrt{1-\eta^2}}$	$\frac{\partial \eta}{\partial \beta_S}$	
	1	1	$\partial \eta$
8 =	$\overline{\sqrt{1-\eta^2}}$	$\overline{\cos\left(\beta_S\right)}$	$\partial \lambda_S$

We managed to reduce our equations to the one fundamental object η , which is an interesting insight on its own. We need latitudes and longitudes of our start and target point and then, only construct the η -function to solve our problem.

However, the vector product definitions will turn out to be more useful going forward:

To summarize:

$$\begin{aligned} \nu = & \frac{1}{\sqrt{1 - \eta^2}} \vec{T_0} \cdot \vec{N_S} \\ \varepsilon = & \frac{1}{\sqrt{1 - \eta^2}} \vec{T_0} \cdot \vec{E_S} \end{aligned}$$

Now, how do ν and ε actually help us?

Ultimately, we aim for a **heading** (or a course, but in any case, the compass direction to our target), which is an angle ϕ telling us our direction.

In fact, ϕ is the angle between the north vector and the direction vector, when going clockwise around the compass. In particular,

ϕ direction		direction	ϕ
south	180°	north	0°
south-west	225°	north-east	45°
west	270°	east	90°
north-west	315°	south-east	135°

With this fact and the north, east and direction vector being normalized, we can deduce

$$\nu = \cos \phi$$

and

 $\varepsilon = \sin \phi.$

Unfortunately, we cannot simply compute either of the two parameters and just take the inverse cosine or inverse sine function, since two angles between 0° and 360° can have the same sine- or cosine-value. (In fact this is always the case, if the sine- or cosine value do not happen to be equal to 1 or -1.)

We cannot obtain an unambiguous result like that!

Since we can calculate both, the sine and cosine of ϕ , we can indeed find the one unambiguous angle that satisfies both equations. There is more than one way to do this, but I will go with the **tangent** approach. (This time, we are not referencing the straight line but the trigonometric function by the same name.)

The heading using the tangent function

We can divide ε by ν to find

$$\frac{\varepsilon}{\nu} = \frac{\sin\phi}{\cos\phi} = \tan\phi$$

The tangent of an angle is a bit special, as it is not periodic in 360° but in 180° .

In fact, let us plot the tangent function and the inverse tangent function, the so-called arctangent function:



Figure 4: The tangent function. The angle ϕ is in degrees.

This is the tangent function.

As we can see, it is periodic in $180^{\circ}.$

If we only consider the range $\phi \in (-90^\circ, 90^\circ)$, we have a bijection, meaning we can invert it here. This is, indeed, the "branch" that is used for the arctangent function definition:



Figure 5: The arctangent function.

Here, we see the arctangent function. (The principal branch, that is.)

As we can see, the arctangent values range from -90° to 90° . (-90° corresponds to west.)

Therefore, using the plain arctangent function, we will always get a result going to some degree to the north.

But what if we actually need to go south?

Well, we need to go into the exact opposite direction from what we have calculated, i. e. we have to add 180° to our heading angle ϕ .

Let us get back to our sign function.

If

$$\sigma(\nu) = 1,$$

we can simply use the result from our arctangent function, but if

$$\sigma(\nu) = -1$$

we need to add 180° .

In any case, we must shift any resulting ϕ by 360°, until it is truly in between 0° and 360°.

Going forward, we ignore this minor complication.

We can encapture this in the following formula:

$$\phi = \arctan\left(\frac{\varepsilon}{\nu}\right) - \frac{1}{2}\left[\sigma\left(\nu\right) - 1\right] \cdot 180^{\circ}$$

This is, indeed, our "result". But recall, that we do not have explicit representations of ν and ε , yet.

We define the argument of the arctangent-function as *x*:

$$\begin{aligned} \varepsilon := & \frac{\varepsilon}{\nu} \\ &= \frac{\frac{1}{\sqrt{1 - \eta^2}} \cdot \frac{1}{\cos \beta_S} \cdot \frac{\partial \eta}{\partial \lambda_S}}{\frac{1}{\sqrt{1 - \eta^2}} \cdot \frac{\partial \eta}{\partial \beta_S}} \\ &= \frac{\frac{\partial \eta}{\partial \lambda_S}}{\cos \beta_S \cdot \frac{\partial \eta}{\partial \beta_S}} \end{aligned}$$

This is nice. We actually got rid of the parameter η . Recall the vector representations

X

$$\begin{split} \eta = &\vec{T_0} \cdot \vec{S_0}, \\ &\frac{\partial}{\partial \lambda_S} \eta = &\vec{T_0} \cdot \frac{\partial}{\partial \lambda_S} \vec{S_0} = \vec{T_0} \cdot \vec{E'_S} \\ &\frac{\partial}{\partial \beta_S} \eta = &\vec{T_0} \cdot \frac{\partial}{\partial \beta_S} \vec{S_0} = \vec{T_0} \cdot \vec{N_S} \end{split}$$

Sweet. We know all these vectors already!

We can do some additional simplifications:

$$x = \frac{\frac{\partial \eta}{\partial \lambda_S}}{\cos \beta_S \cdot \frac{\partial \eta}{\partial \beta_S}}$$
$$= \frac{\vec{T_0} \cdot \vec{E'_S}}{\cos \beta_S \cdot \vec{T_0} \cdot \vec{N_S}}$$
$$= \frac{\vec{T_0} \cdot \vec{E_S}}{\vec{T_0} \cdot \vec{N_S}}$$

Isn't this amazing? After all the pain we had to endure in calculating this result, it finally boils down to two vector products, their ratio and applying the arctangent function.

Now, let's roll up our sleeves and do this!

$$\vec{T_0} \cdot \vec{E_S} = \begin{pmatrix} \cos \lambda_T \cdot \cos \beta_T \\ \sin \lambda_T \cdot \cos \beta_T \\ \sin \beta_T \end{pmatrix} \cdot \begin{pmatrix} -\sin \lambda_S \\ \cos \lambda_S \\ 0 \end{pmatrix}$$

 $= \cos \beta_T \cdot \left[-\sin \lambda_S \cdot \cos \lambda_T + \cos \lambda_S \cdot \sin \lambda_T \right]$

$$\vec{I_0} \cdot \vec{N_S} = \begin{pmatrix} \cos \lambda_T \cdot \cos \beta_T \\ \sin \lambda_T \cdot \cos \beta_T \\ \sin \beta_T \end{pmatrix} \cdot \begin{pmatrix} -\cos \lambda_S \cdot \sin \beta_S \\ -\sin \lambda_S \cdot \sin \beta_S \\ \cos \beta_S \end{pmatrix}$$
$$= \sin \beta_S \cdot \cos \beta_T \cdot [-\cos \lambda_S \cdot \cos \lambda_T - \sin \lambda_S \cdot \sin \lambda_T]$$
$$+ \sin \beta_T \cdot \cos \beta_S$$

Next, we can consider two really useful theorems:

$$\sin (x \pm y) = \sin x \cdot \cos y \pm \cos x \cdot \sin y$$
$$\cos (x \pm y) = \cos x \cdot \cos y \mp \sin x \cdot \sin y$$

This implies:

$$-\sin\lambda_S \cdot \cos\lambda_T + \cos\lambda_S \cdot \sin\lambda_T = \sin(\lambda_T - \lambda_S)$$
$$-\cos\lambda_S \cdot \cos\lambda_T - \sin\lambda_S \cdot \sin\lambda_T = -\cos(\lambda_T - \lambda_S)$$

Now, our equations simplify quite a bit:

$$\vec{T_0} \cdot \vec{E_S} = \cos \beta_T \cdot \sin (\lambda_T - \lambda_S)$$

$$\vec{T_0} \cdot \vec{N_S} = -\sin \beta_S \cdot \cos \beta_T \cdot \cos (\lambda_T - \lambda_S) + \sin \beta_T \cdot \cos \beta_S$$

Interesting! Our result will not depend on the individual longitudes of target and start point but only on their difference.

Let us think about why that makes sense.

As pointed out, chosing a particular $\lambda = 0^{\circ}$, i. e. a 0° - or prime-meridian was arbitrary. In fact, it is a mere convention.

This means, that we could have chosen the meridian of our start point to be 0° just as easily, i. e. introduce a new coordinate frame $\lambda' = \lambda - \lambda_S$.

If we did that, the target meridian would become exactly $\lambda'_T = \lambda_T - \lambda_S$, i. e. our result depends on the target meridian λ'_T , but not on the start meridian λ'_S , as this one would be 0° by definition.

But let's get back to work, now.

We wanted to calculate

$$\begin{aligned} \varepsilon &= \frac{\varepsilon}{\nu} \\ &= \frac{\vec{T}_0 \cdot \vec{E}_S}{\vec{T}_0 \cdot \vec{N}_S} \\ &= \frac{\cos \beta_T \cdot \sin (\lambda_T - \lambda_S)}{\sin \beta_T \cdot \cos \beta_S - \sin \beta_S \cdot \cos \beta_T \cdot \cos (\lambda_T - \lambda_S)} \\ &= \frac{\sin (\lambda_T - \lambda_S)}{\tan \beta_T \cdot \cos \beta_S - \sin \beta_S \cdot \cos (\lambda_T - \lambda_S)}, \\ &= \frac{1}{\cos \beta_S} \cdot \frac{\sin (\lambda_T - \lambda_S)}{\tan \beta_T - \tan \beta_S \cdot \cos (\lambda_T - \lambda_S)} \end{aligned}$$

which, sadly, is the end of the rope.

Unfortunately, we are still not done. Recall our formula for ϕ :

$$\phi = \arctan\left(x\right) - \frac{1}{2}\left[\sigma\left(\nu\right) - 1\right] \cdot 180^{\circ}$$

We need to find $\sigma(\nu)$.

Since

$$\nu = \frac{1}{\sqrt{1-\eta^2}} \cdot \vec{T_0} \cdot \vec{N_S},$$

we need to check the sign of this expression.

However, we can just drop the first factor, because the square-root-term is always positive. This implies:

$$\begin{aligned} \sigma(\nu) = &\sigma\left(\vec{T}_0 \cdot \vec{N}_S\right) \\ = &\sigma\left(\sin\beta_T \cdot \cos\beta_S - \sin\beta_S \cdot \cos\beta_T \cdot \cos\left(\lambda_T - \lambda_S\right)\right) \end{aligned}$$

Let us pull the cosine terms of the latitudes in front. Since β_S and β_T only range from -90° to 90° , the cosines will always be positive and we can drop them, as well:

$$\sigma(\nu) = \sigma \left(\sin \beta_T \cdot \cos \beta_S - \sin \beta_S \cdot \cos \beta_T \cdot \cos (\lambda_T - \lambda_S) \right)$$
$$= \sigma \left(\cos \beta_S \cos \beta_T \cdot \left[\tan \beta_T - \tan \beta_S \cdot \cos (\lambda_T - \lambda_S) \right] \right)$$
$$= \sigma \left(\tan \beta_T - \tan \beta_S \cdot \cos (\lambda_T - \lambda_S) \right)$$

Sadly, there is not much else to be simplified.

Let us therefore insert:

$$\phi = \arctan\left(\frac{1}{\cos\beta_S} \cdot \frac{\sin(\lambda_T - \lambda_S)}{\tan\beta_T - \tan\beta_S \cdot \cos(\lambda_T - \lambda_S)}\right) - \frac{1}{2} \left[\sigma \left(\tan\beta_T - \tan\beta_S \cdot \cos(\lambda_T - \lambda_S)\right) - 1\right] \cdot 180^\circ$$

Discussion of the result

Personally, I always consider it satisfying if I can derive an explicit formula for something I want to investigate.

However, in this case, this formula might be impractical.

With all those sine-, cosine-, and tangent-terms, this is not really something you can easily work out without a calculator.

Calculation in the real world

If you are computer literate, you would probably write a short program that you provide the coordinates β_S , β_T , λ_S and λ_T which would then construct the vectors $\vec{T_0}$, $\vec{N_S}$ and $\vec{E_S}$ numerically to work this out.

In fact, you can find a short and clumsy demo code at the end of this article.

Limitations

North and south pole

Notice, that the formula (and the algorithm) break down at the poles, because you cannot define an umambiguous north vector there.

It is, however, clear that from any of the poles, you would "just" have to follow your target meridian to get to your destination.

If you really wanted to use our formula, you should do one step into any arbitrary direction and go from there.

${\rm Denominator}\ {\rm being}\ 0$

The denominator $\vec{T_0} \cdot \vec{N_S}$ being 0 is not really a problem, as long as you perform a limit, because then, the whole argument of the arctangent-function goes to either $+\infty$ or $-\infty$, for which the arctangent approaches either $+90^\circ$ or -90° , respectively. (Meaning you can just go west or east.)

In this case, you just have to look at start and target longitude to find the right way. (However, this might be something to consider in a computer algorithm.)

Basically, if $\vec{T_0} \cdot \vec{N_S} = 0$, check $\lambda_T - \lambda_S$. If it is smaller than 180°, go eastwards. Otherwise, go west.

Small differences in latitude and longitude

Suppose your start and your target are only a few meters to kilometers apart.

In that case, the difference in longitudes and latitudes is small.

Let us check, what that does for our arctangent's argument x. We will use the following approximations for small values of h:

 $\sin h \approx h$ $\cos h \approx 1$

Notice, that these approximations only work for decently small values of h in radians.

Therefore, we temporarily express latitudes and longitudes in radians.

First, we write

$$\Delta \lambda = \lambda_T - \lambda_S,$$

which is, by assumption, a small value. Thus, we start by approximating

$$x = \frac{1}{\cos \beta_S} \cdot \frac{\sin \Delta \lambda}{\tan \beta_T - \tan \beta_S \cdot \cos \Delta \lambda}$$
$$= \frac{\sin \Delta \lambda}{\tan \beta_T \cdot \cos \beta_S - \sin \beta_S \cdot \cos \Delta \lambda}$$
$$\approx \frac{\Delta \lambda}{\tan \beta_T \cdot \cos \beta_S - \sin \beta_S}$$
$$\approx \frac{\Delta \lambda \cos \beta_T}{\sin \beta_T \cdot \cos \beta_S - \sin \beta_S \cdot \cos \beta_T}$$
$$\approx \frac{\Delta \lambda \cos \beta_T}{\sin (\beta_T - \beta_S)}.$$

In the last step, we once again used a trigonometric identity.

Approximation of the sine in the denominator yields:

$$x \approx \frac{\Delta\lambda \cos\beta_T}{\sin\left(\beta_T - \beta_S\right)}$$
$$\approx \frac{\Delta\lambda \cos\beta_T}{\Delta\beta}$$
$$\approx \cos\beta_T \cdot \frac{\Delta\lambda}{\Delta\beta}$$

Now, it does not matter anymore, whether we use *radians* or *degrees*, since converting between the two just involves multiplying by a factor. And since we had to convert numerator and denominator alike, the conversion factors would just cancel.

Let's make this explicit:

$$\tan\phi\approx\cos\beta_T\cdot\frac{\Delta\lambda}{\Delta\beta}$$

Without the cosine term, this would look like a case of "rise over run". In fact, around the equator, where the cosine is exactly 1, this fits best.

This is reflective of the fact, that a given meridian and the equator really are perpendicular, while for any other latitude, this only holds for an infinitely small area.

That is, what is corrected for by the cosine term.

In any case, we need to do numerically precise calculations when using the exact formula, so we don't "approximate away" the real discrepancy between start and target.

If we cannot do that, we might be better off with this approximation.

Python code

I got around to write some rudimentary python code that gets the job done.

For the sake of simplicity, I hard-coded start point and destination point to be the city of Muenster, Germany (where I happen to live) and the city of Cologne, Germany (which is my favourite, so naturally, I want to go there).

There is no error-handling for "special" inputs like we discussed. Consider that an exercise for the reader.

```
!/usr/bin/env python3
1
2
3
   import numpy as np
4
5
   def location_vector(latitude, longitude):
6
       x = np.cos(longitude*np.pi/180.)*np.cos(latitude*np.pi/180.)
7
       y = np.sin(longitude*np.pi/180.)*np.cos(latitude*np.pi/180.)
8
       z = np.sin(latitude*np.pi/180.)
9
       return np.array([x,y,z])
10
   def north_vector(latitude,longitude):
11
12
       x = - np.cos(longitude*np.pi/180.)*np.sin(latitude*np.pi/180.)
       y = - np.sin(longitude*np.pi/180.)*np.sin(latitude*np.pi/180.)
13
       z = np.cos(latitude*np.pi/180.)
14
15
       return np.array([x,y,z])
16
   def east_vector(longitude):
17
18
       x = - np.sin(longitude*np.pi/180.)
19
       y = np.cos(longitude*np.pi/180.)
       return np.array([x,y,0])
21
22
   def main():
23
       # Cologne
24
       beta_T=50.935173
25
       lambda_T=6.953101
26
       # Muenster
       beta_S=51.961563
27
28
       lambda_S=7.628202
29
       # Location vector Cologne
       vec_T = location_vector(beta_T,lambda_T)
31
       # North and east vector Muenster
32
33
       vec_N = north_vector(beta_S,lambda_S)
34
       vec_E = east_vector(lambda_S)
       epsilon = vec_T.dot(vec_E)
37
       nu = vec_T.dot(vec_N)
       # Heading angle
40
       phi = np.arctan(epsilon/nu)*180/np.pi
41
       if nu < 0:
42
            phi = phi + 180
43
44
       # Normalize heading angle
       while phi < 0:</pre>
45
            phi = phi + 360
46
47
       while phi >= 360:
48
            phi = phi - 360
49
50
       print(phi)
51
52
   if __name__ == "__main__":
53
       main()
```