

Haversine formula

The **haversine formula** is an equation important in navigation, giving great-circle distances between two points on a sphere from their longitudes and latitudes. It is a special case of a more general formula in spherical trigonometry, the **law of haversines**, relating the sides and angles of spherical triangles. The first table of haversines in English was published by James Andrew in 1805.^[1] Florian Cajori credits an earlier use by Jose de Mendoza y Ríos in 1801^[2] The term *haversine* was coined in 1835 by Prof. James Inman.^[3]

These names follow from the fact that they are customarily written in terms of the **haversine** function, given by $\text{haversin}(\theta) = \sin^2(\theta/2)$. The formulas could equally be written in terms of any multiple of the haversine, such as the older **versine** function (twice the haversine). Prior to the advent of computers, the elimination of division and multiplication by factors of two proved convenient enough that tables of haversine values and **logarithms** were included in 19th and early 20th century navigation and trigonometric texts.^{[4][5][6]} These days, the haversine form is also convenient in that it has no coefficient in front of the \sin^2 function.

1 The haversine formula

For any two points on a sphere, the haversine of the **central angle** between them is given by

$$\text{haversin}\left(\frac{d}{r}\right) = \text{haversin}(\phi_2 - \phi_1) + \cos(\phi_1) \cos(\phi_2) \text{haversin}(\lambda_2 - \lambda_1)$$

where

- *haversin* is the **haversine** function:

$$\text{haversin}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2}$$

- d is the distance between the two points (along a great circle of the sphere; see **spherical distance**),
- r is the radius of the sphere,
- ϕ_1, ϕ_2 : latitude of point 1 and latitude of point 2
- λ_1, λ_2 : longitude of point 1 and longitude of point 2

On the left side of the equals sign d/r is the central angle, assuming angles are measured in **radians** (note that ϕ and λ can be converted from degrees to radians by multiplying by $\pi/180$ as usual).

Solve for d by applying the inverse haversine (if available) or by using the **arcsine** (inverse sine) function:

$$d = r \text{haversin}^{-1}(h) = 2r \arcsin(\sqrt{h})$$

where h is $\text{haversin}(d/r)$, or more explicitly:

$$\begin{aligned} d &= 2r \arcsin\left(\sqrt{\text{haversin}(\phi_2 - \phi_1) + \cos(\phi_1) \cos(\phi_2) \text{haversin}(\lambda_2 - \lambda_1)}\right) \\ &= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1) \cos(\phi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \end{aligned}$$

When using these formulae, ensure that h does not exceed 1 due to a **floating point** error (d is only **real** for h from 0 to 1). h only approaches 1 for **antipodal** points (on opposite sides of the sphere) — in this region, relatively large numerical errors tend to arise in the formula when finite precision is used. Because d is then large (approaching πR , half the circumference) a small error is often not a major concern in this unusual case (although there are other **great-circle distance** formulas that avoid this problem). (The formula above is sometimes written in terms of the **arctangent** function, but this suffers from similar numerical problems near $h = 1$.)

As described below, a similar formula can be written using **cosines** (sometimes called the **spherical law of cosines**, not to be confused with the **law of cosines** for plane geometry) instead of haversines, but if the two points are close together (e.g. a kilometer apart, on the Earth) you might end up with $\cos(d/R) = 0.99999999$, leading to an inaccurate answer. Since the haversine formula uses sines it avoids that problem.

Either formula is only an approximation when applied to the **Earth**, which is not a perfect sphere: the "Earth radius" R varies from 6356.752 km at the poles to 6378.137 km at the equator. More importantly, the **radius of curvature** of a north-south line on the earth's surface is 1% greater at the poles (≈ 6399.594 km) than at the equator (≈ 6335.439 km)— so the haversine formula and law of cosines can't be guaranteed correct to better than 0.5%. More accurate methods that consider the Earth's ellipticity are given by **Vincenty's formulae** and the other formulae in the **geographical distance** article.

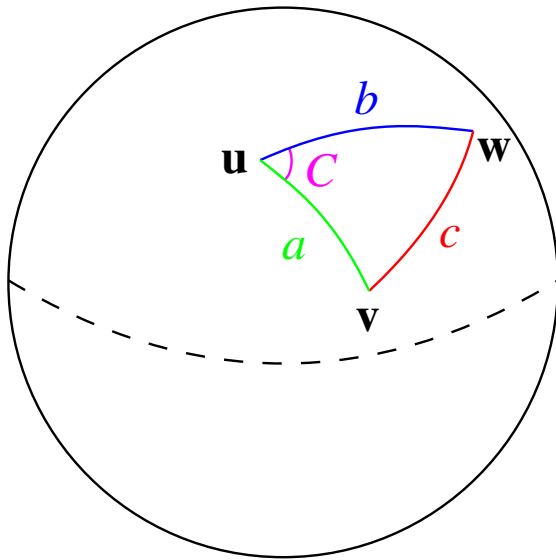
2 The law of haversines

Given a unit sphere, a “triangle” on the surface of the sphere is defined by the **great circles** connecting three points **u**, **v**, and **w** on the sphere. If the lengths of these three sides are a (from **u** to **v**), b (from **u** to **w**), and c (from **v** to **w**), and the angle of the corner opposite c is C , then the law of haversines states:

(the law of haversines)

$$\text{haversin}(c) = \text{haversin}(a-b) + \sin(a) \sin(b) \text{haversin}(C).$$

Since this is a unit sphere, the lengths a , b , and c are simply equal to the angles (in **radians**) subtended by those sides from the center of the sphere (for a non-unit sphere, each of these arc lengths is equal to its **central angle** multiplied by the radius of the sphere).



Spherical triangle solved by the law of haversines.

In order to obtain the haversine formula of the previous section from this law, one simply considers the special case where **u** is the **north pole**, while **v** and **w** are the two points whose separation d is to be determined. In that case, a and b are $\pi/2 - \varphi_{1,2}$ (i.e., $90^\circ - \text{latitude}$), C is the longitude separation $\Delta\lambda$, and c is the desired d/R . Noting that $\sin(\pi/2 - \varphi) = \cos(\varphi)$, the haversine formula immediately follows.

To derive the law of haversines, one starts with the **spherical law of cosines**:

(spherical law of cosines)

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

As mentioned above, this formula is an ill-conditioned way of solving for c when c is small. Instead, we substitute the identity that $\cos(\theta) = 1 - 2 \text{haversin}(\theta)$, and also employ the **addition identity** $\cos(a - b) = \cos(a) \cos(b) + \sin(a) \sin(b)$, to obtain the law of haversines, above.

3 References

- U. S. Census Bureau Geographic Information Systems FAQ, (content has been moved to **What is the best way to calculate the distance between 2 points?**)
- R. W. Sinnott, “Virtues of the Haversine”, *Sky and Telescope* **68** (2), 159 (1984).
- **Deriving the haversine formula**, *Ask Dr. Math* (Apr. 20–21, 1999).
- Romuald Ireneus 'Scibor-Marchocki, **Spherical trigonometry**, *Elementary-Geometry Trigonometry* web page (1997).
- W. Gellert, S. Gottwald, M. Hellwich, H. Kästner, and H. Küstner, *The VNR Concise Encyclopedia of Mathematics*, 2nd ed., ch. 12 (Van Nostrand Reinhold: New York, 1989).

- [1] Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry By Glen Van Brummelen
- [2] A History of Mathematical Notations: Vol. II By Florian Cajori
- [3] Oxford English Dictionary. Oxford University Press. 2nd ed. 1989. Cites coinage of term “Haversine” by Prof. Jas. Inman, D. D., in his *Navigation and Nautical Astronomy*, 3rd ed. (1835).
- [4] H. B. Goodwin, *The haversine in nautical astronomy*, *Naval Institute Proceedings*, vol. 36, no. 3 (1910), pp. 735–746: *Evidently if a Table of Haversines is employed we shall be saved in the first instance the trouble of dividing the sum of the logarithms by two, and in the second place of multiplying the angle taken from the tables by the same number. This is the special advantage of the form of table first introduced by Professor Inman, of the Portsmouth Royal Navy College, nearly a century ago.*
- [5] W. W. Sheppard and C. C. Soule, **Practical navigation** (World Technical Institute: Jersey City, 1922).
- [6] E. R. Hedrick, *Logarithmic and Trigonometric Tables* (Macmillan, New York, 1913).

4 External links

- Implementations of the Haversine formula in 67 languages at rosettacode.org and in 17 languages on codecadex.com
- Other implementations in C++, C (MacOS), Pascal, Python, Ruby, JavaScript, Matlab, MySQL

5 Text and image sources, contributors, and licenses

5.1 Text

- **Haversine formula** *Source:* <http://en.wikipedia.org/wiki/Haversine%20formula?oldid=653577603> *Contributors:* Roadrunner, Michael Hardy, Ronz, Stevenj, Kaihsu, Charles Matthews, Giftlite, Fropuff, Wwoods, Pne, Doops, TiMike, Alby, Oleg Alexandrov, Acone, Mathbot, Fresheneesz, CiaPan, Mhartl, Dbfirs, Mike Dillon, Allens, Kaimbridge, Bluebot, Quinsareth, Rainer198, Cuneas, Hispalois, Utopiah, Magioladitis, Swpb, Glmory, LokiClock, TheOtherJesse, TXiKiBoT, Halfiehsu, Hannycow, Naleh, Dgwsoft, XLinkBot, Addbot, Lightbot, Yobot, Kipoc, AnomieBOT, Piano non troppo, HistPhd, Xqbot, Aroraparag, Sigi fikanz, Spatialthoughts, LucienBOT, Esawdust, Kaarenmax19, Jaimerios, Tim Zukas, Mentibot, ClueBot NG, Muehlburger, Jweisblat, Spolu, Slaing, Vswitchs, Kiusao, Pdecalculus, E E Ballew, Markcaudill and Anonymous: 50

5.2 Images

- **File:Law-of-haversines.svg** *Source:* <http://upload.wikimedia.org/wikipedia/commons/3/38/Law-of-haversines.svg> *License:* Public domain *Contributors:* self-made, based on Image:Law-of-haversines.png *Original artist:* TheOtherJesse (SVG version), Steven G. Johnson (PNG original)

5.3 Content license

- Creative Commons Attribution-Share Alike 3.0