Global Journal of Engineering and Technology Advances

eISSN: 2582-5003

Cross Ref DOI: 10.30574/gjeta

Journal homepage: https://gjeta.com/

(Research Article)

Check for updates

Performance evaluation of the python Mpmath and math libraries in computing ellipsoidal distances using the inverse problem of geodesy

Isaac Ramos Junior [1,*], Leonard Niero da Silveira [1], Vinícius Bergmann Martins [1], Andréa de Seixas [2], Sílvio Jacks dos Anjos Garnés [2] and Lucas Gonzales Lima Pereira Calado [2]

[1] *Federal University of Pampa, Itaqui Campus, Rio Grande do Sul, Brazil.*

[2] *Department of Cartographic Engineering, Federal University of Pernambuco, Recife, Brazil.*

Global Journal of Engineering and Technology Advances, 2026, 27(01), 106-115

Publication history: Received on 09 March 2026; revised on 19 April 2026; accepted on 22 April 2026

Article DOI: https://doi.org/10.30574/gjeta.2026.27.1.0095

Abstract

Since the works of Clairaut in the 18th century and Bessel in the 19th century, the Direct and Inverse Problems of Geodesy have been central topics in this science, due to their relevance in both scientific and engineering applications that require high positional precision. Despite the vast literature on geodesic lines, studies comparing the performance and limitations of the different available numerical methods are still relatively scarce. Furthermore, the advancement of dynamic geodesy and the establishment of high-precision reference frames have imposed requirements on the solutions to these problems, which justifies the evaluation of modern computational tools. In this context, Python has stood out as a widely used language in scientific computing, including in Geodetic Sciences, due to its versatility and ability to integrate with numerical libraries. Thus, the objective of this work was to analyze Python's performance in computing geodetic distances by solving the Inverse Problem. For this, Vincenty's equations were implemented in two programs: one based on the *math* library, with fixed precision, and another on the *mpmath* library, with arbitrary precision. The results were validated against a set of one hundred thousand high-precision geodesic lines, with the largest discrepancies on the order of 10^{-5} m. When compared in a practical application, calculating distances between the Municipality of Paulista, in Pernambuco, Brazil, and ten globally distributed locations, the differences between the libraries reached at most 10^{-6} m, demonstrating numerical performance consistent with high-precision geodetic applications.

Keywords: High Precision Geodesic Lines; Vincenty Method; Computational Performance; Arbitrary Precision Arithmetic; Geodetic Distance Computation

1. Introduction

Since the time of Clairaut in the 18th century and the studies of Bessel in the 19th century, the Direct and Inverse Problems of Geodesy have been subjects of interest among geodesists (Sjöberg and Shirazian, 2012). More specifically, the solution of these problems holds scientific importance in research and engineering applications involving high-precision geodetic measurements, reference systems, precise navigation, tectonic plate motion, among others (Wang et al., 2022).

Despite the existence of many works on geodesic lines, such as those by Rainsford (1955), Robbins (1962), Sodano (1965), Kivioja (1971), Bowring (1971, 1977a, 1978, 1981, 1996), Vincenty (1975), Saito (1979), Jank and Kivioja (1980), and Pittman (1986), few compare the relative advantages and limitations of each method. However, at the beginning of this century, some new computational methods were proposed, such as in Sjöberg (2008) and Bing and Shaofeng (2007); new algorithms, as in Sjöberg (2012) and Xiaohan et al. (2012); and, with the more recent

* Corresponding author: Isaac Ramos Junior.

development of computational technology, the geodetic problem has become easier to solve, as shown, for example, in Yuhao et al. (2020). Furthermore, with Karney (2013) and Karney (2024), applications have become more extensive, and, with the development of dynamic geodesy, as in Zhengquan (2021), combined with the emergence of high-precision reference frames, such as in Pengfei and Yingya (2017), more stringent requirements have been established for solving both Problems.

Parallel to these aspects, with regard to computational implementation, Python is a versatile language widely used in scientific and engineering computing for data analysis, graphical visualization, and numerical solutions of complex problems. Moreover, it is a general-purpose language applied in several areas, such as web development, finance, and system automation. In recent years, its use has grown among scientists and engineers, including as an interface for software written in other languages, such as Fortran and C (Pine, 2025).

In this context, within the field of Geodetic Sciences, Python has also been employed. For example, Tulu et al. (2025) developed a user-friendly Python-based desktop application to automate the processing and extraction of information from thermal and multispectral images acquired by unmanned aerial vehicles; Deng et al. (2025) used Python to study new expressions for the gravitational potential, gravitational vector, and gravitational gradient tensor of a vertical cylindrical prism using a second-order 3D Taylor series expansion for practical gravity modeling applications; Haque et al. (2025) developed a Python program to segment images, align them with the field of view of an active sensor, and extract reflectance data to compute the Normalized Difference Vegetation Index (NDVI), which assesses vegetation health and density; and Barazzetti (2025) developed Python scripts to perform complex calculations in Field Astronomy and Spherical Trigonometry, automating the process and simulating measurement accuracy.

Thus, with the objective of analyzing the performance of Python in computing ellipsoidal (or geodesic) distances based on the solution of the Inverse Geodetic Problem, this work applied the well-established Vincenty equations in two programs: one based on the *math* library and the other on the *mpmath* library. In the first, a fixed precision between 15 and 17 decimal places was used. In the second, arbitrary precision was employed, set to 30 decimal places, which corresponds to approximately twice that of the first. As a reference parameter for validating the programs, results from one hundred thousand geodesic lines randomly distributed over the ellipsoid were used. These belong to a larger dataset comprising results for five hundred thousand lines, computed with high precision. This dataset is known as the geodesic test set, and can be found in Karney (2010).

After validating the programs, geodesic distances were computed, for each implementation, between the Municipality of Paulista, in Pernambuco State, Brazil, and ten other locations around the world: Shanghai, China; Anchorage, Alaska; Perth, Australia; Ferraz Commander Station, Antarctica; Moscow, Russia; Lisbon, Portugal; Manokwari, Indonesia; Lima, Peru; Monterrey, Mexico; and Nairobi, Kenya, with distances ranging from four to eighteen thousand kilometers. In the comparison between the libraries, the largest differences were in the order of 10^{-6} m, demonstrating consistency with high-precision geodetic applications.

2. Methodology

The Inverse Problem of Geodesy can be formulated as follows: given two points (φ_1, λ_1) and (φ_2, λ_2) on the surface of the ellipsoid, determine the geodesic distance s between them, as well as the geodesic azimuth α_1 and the geodesic back-azimuth α_2 of the geodesic line connecting the points (φ_1, λ_1) and (φ_2, λ_2), respectively (Thomas and Featherstone, 2005). Several authors have studied the determination of geodesic distances through the Inverse Problem of Geodesy. However, the solution proposed by Vincenty appears to be the preferred approach for solving this problem (Sjöberg and Shirazian, 2012).

Thus, Vincenty's equations, which can be found in Vincenty (1975), were implemented in two different Python programs. One of them used only one of Python's most traditional libraries, the *math* library, while the other relied exclusively on the *mpmath* library.

Python's *math* library performs operations using the *float* type, which is implemented as double-precision floating-point (64 bits), corresponding to approximately 15 to 17 significant decimal digits. This implementation directly uses the computer's hardware resources, namely the Floating-Point Unit (FPU) of the Central Processing Unit (CPU) and does not allow increasing precision beyond this limit (Python, 2026). The *mpmath* library is a Python library for arbitrary-precision real and complex floating-point arithmetic, and it works with both Python 2 and Python 3 without depending on other libraries. It can be used either as a library or interactively through the Python interpreter (Mpmath, 2026).

Accordingly, in this study, for the computations performed with the *mpmath* library, a value of 30 significant digits after the decimal point was chosen, as this corresponds approximately to twice the total number of significant digits available in the *math* library, ensuring that any numerical contrast between the libraries could be properly analyzed.

For the validation of the two programs, part of a geodesic test dataset established by Karney (2010) was used. This dataset contains information on five hundred thousand geodesic lines computed with high precision, divided into nine subsets. Since one of these subsets consists of one hundred thousand geodesic lines randomly distributed over the ellipsoid, it was selected for the validation tests.

After validation, the city of Paulista was chosen as the origin point for the geodesic distances, which were computed between this location and ten other points around the world. The coordinates used for the calculations were the geodetic latitudes and longitudes of a point in Paulista obtained from IBGE (2026), and one point in each of the other locations, obtained from Geodatos (2026), as shown in Table 1. These locations were selected due to their homogeneous distribution across the Earth, as illustrated in Figure 1.

It should be noted that, in Table 1, the coordinate values are given with four decimal places. However, since the objective of this study was to compare the performance of two computational libraries rather than the accuracy of a survey based on traditionally collected field data, this number of decimal places did not influence the calculations or the results.

Table 1 Geodetic Latitudes and Longitudes, in WGS84

Point	Latitude (°)	Longitude (°)
Paulista	-7.9005	-34.8265
Alaska	61.2181	-149.9003
Antártida	-62.0833	-58.4000
Lima	-12.0464	-77.0428
Lisboa	38.7223	-9.1393
Manokwari	-0.8615	134.0620
Monterrey	25.6866	-100.3161
Moscou	55.7558	37.6173
Nairobi	-1.2921	36.8219
Perth	-31.9505	115.8605
Xangai	31.2304	121.4737

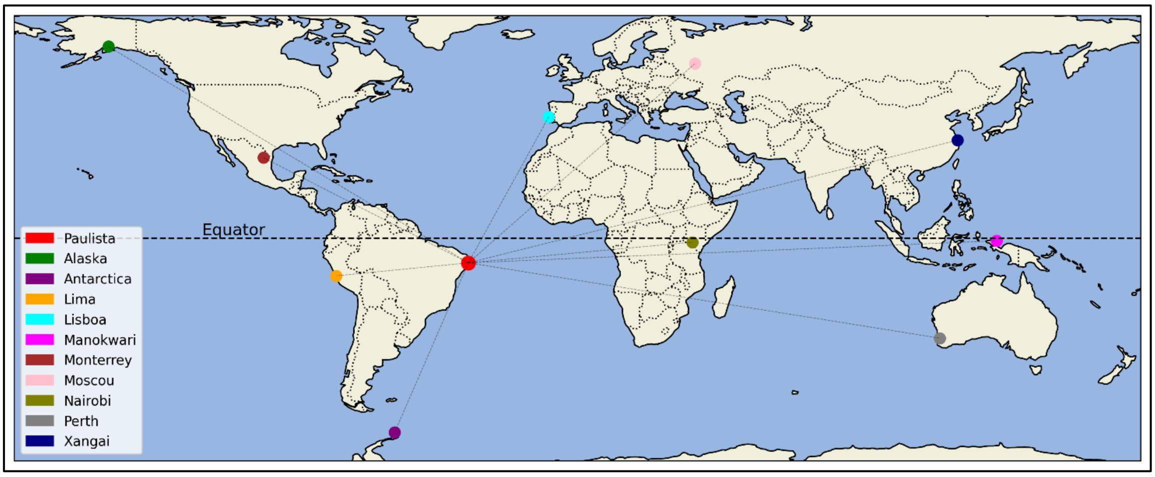

Figure 1 Distributions of the locations involved in the distance computations

For the calculations, the ellipsoid of the World Geodetic System 1984 (WGS84) was used. The choice of this ellipsoid was made because the coordinates in Table 1 and the geodesic test dataset employed (Karney, 2010) are referenced to this system. Accordingly, the constants used were $a = 6{,}378{,}137.000$ m for the semi-major axis of the ellipsoid and $f = 1/298.257223563$ for the flattening, as given in Karney (2013).

3. Results and discussions

The first results obtained were those related to the validation of the programs. To perform this task, one hundred thousand geodesic lines randomly distributed over the Earth were selected as the reference for comparison, belonging to the geodesic test dataset, whose values were computed using high-precision geodetic calculations (Karney, 2010).

After the computations, the largest differences found for both libraries were on the order of 10^{-5}m, as can be observed in Figures 2 and 3, where the one hundred thousand absolute differences between the random geodesic distances from the test dataset and the geodesic distances computed using the *math* library and the *mpmath* library, respectively, were plotted.

In Figure 2, it can be observed that for shorter lines, the errors are mainly concentrated between $10^{-7}$and 10^{-6}meters; for intermediate distances, they shift to the range of $10^{-6}$to 10^{-5}meters. For the longest distances, the most significant deviations appear; however, they still do not exceed approximately 8×10^{-5}meters, as indicated by the dashed line in the graph. The vertical dispersion of the data for a given distance suggests that geometric factors, such as latitude, geodesic direction, and critical configurations (for example, nearly antipodal paths) influence the results. Despite this, there are situations in which the agreement between the methods reaches very high levels of precision, between $10^{-9}$and 10^{-10}meters. Overall, the results show that the error increases with distance but remains small across the entire analyzed range, being practically negligible from a practical standpoint and consistent with numerical differences between high-precision geodetic algorithms.

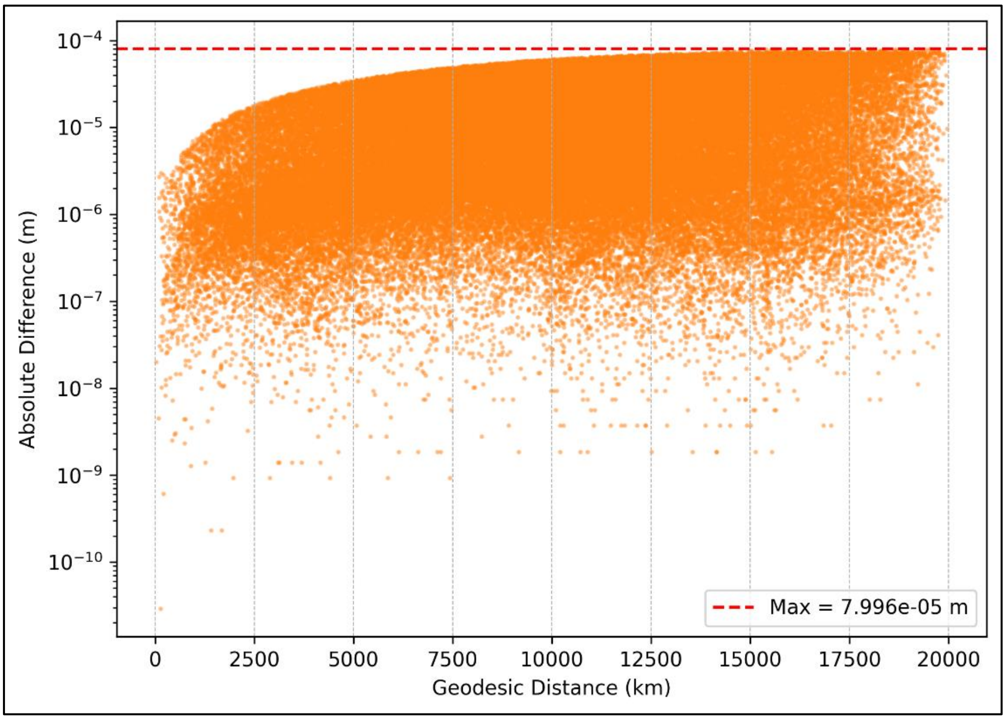

Figure 2 Absolute differences between the geodesic distances of the test set and the *math* library

When Figure 3 is also considered, it becomes evident that the overall behavior is essentially the same, indicating that both methods exhibit similar numerical stability over the entire range of geodesic distances. In both cases, the absolute difference increases with distance, with the upper envelope of the error progressively increasing from shorter to longer distances.

The main quantitative difference lies in the maximum observed error. In Figure 2, the maximum value reaches approximately 7.996×10^{-5} m, whereas in Figure 3 the maximum is slightly smaller, around 7.533×10^{-5} m. This reduction is modest but indicates a slight improvement in numerical agreement in the second case under the most extreme scenarios, typically associated with very long distances and unfavorable geometric configurations.

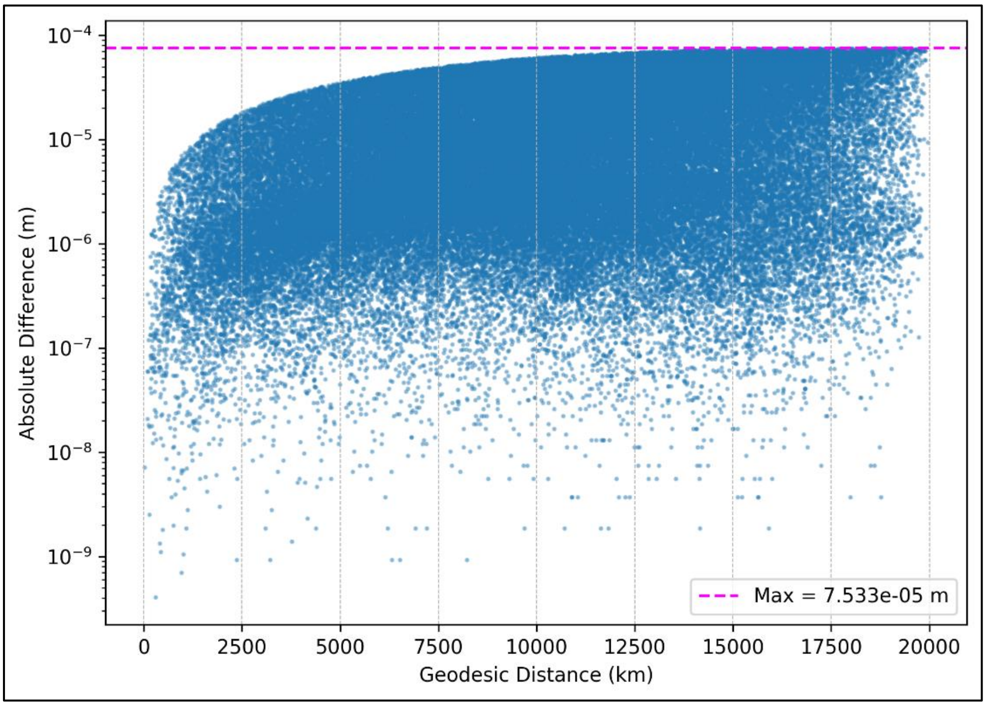

Figure 3 Absolute differences between the geodesic distances of the test set and the *mpmath* library

The dispersion of the points is also very similar between Figures 2 and 3: for a given distance, the errors vary over several orders of magnitude, again reflecting the influence of geometric factors such as latitude, geodesic orientation, and proximity to critical cases, such as nearly antipodal situations. In both figures, the lower bound of the point cloud reaches values on the order of 10^{-9} m, showing that, under favorable configurations, the agreement between the methods approaches the limit of numerical precision.

In general, the comparison shows that Figures 2 and 3 are qualitatively equivalent, with only marginal differences in the extreme values. Both confirm that the errors remain very small, even for long distances, and that the observed discrepancies are consistent with expected numerical effects in high-precision geodetic algorithms, without indicating any significant degradation in performance in either case.

After the validation procedure, the differences between the libraries for the distances from Paulista to each of the ten locations were computed, and the results can be seen in Table 2, where it is observed that the largest differences are in the order of micrometers. In this regard, the three largest absolute differences, in descending order, were found for Nairobi, Monterrey, and Lima. Conversely, the smallest absolute differences, in ascending order, were found for Antarctica, Perth, and Alaska. The results for the *mpmath* library are presented with thirty decimal places to explicitly demonstrate the numerical performance of this library.

Table 2 Results for the geodesic distances

Point	*math* distance (m)	*mpmath* distance (m)	Dif. (m)
Alaska	12098069.962285884	12098069.962285796180367469787597656250	-8.754×10^{-8}
Antártida	6313489.858702957	6313489.858702966012060642242431640625	9.313×10^{-9}
Lima	4647191.640184474	4647191.640186077915132045745849609375	1.603×10^{-6}
Lisboa	5805783.47664085	5805783.476641090586781501770019531250	2.402×10^{-7}
Manokwari	18455781.124983016	18455781.124983310699462890625000000000	2.942×10^{-7}
Monterrey	7993715.797128523	7993715.797130296938121318817138671875	1.774×10^{-6}

Moscou	9650847.54410633	9650847.54410641267895698547363281250 0	8.195×10^{-8}
Nairobi	7975400.604782502	7975400.60478561930358409881591796875 0	3.117×10^{-6}
Perth	14614545.158791533	14614545.158791504800319671630859375000	-2.793×10^{-8}
Xangai	16450186.922524853	16450186.922524714842438697814941406250	-1.378×10^{-7}

Figure 4 shows a graphical representation of the geodesic distances from the Municipality of Paulista, arranged from the shortest to the longest. It can be observed that, as the destinations become more distant, the measured distance increases in a nearly continuous manner. The shortest distances correspond to cities such as Lima and Lisbon, located in South America and Western Europe, respectively.

Next are locations in the Southern Hemisphere and at mid-latitudes, such as Antarctica and Nairobi. There is a nearly constant range between Nairobi and Monterrey, indicating that these two cities are almost at the same distance from Paulista. From Moscow onward, a more pronounced increase is observed, reflecting the large spatial separation of regions in the Northern Hemisphere and the Pacific. The greatest distances are to Perth, Shanghai, and especially Manokwari, which is the most distant point analyzed.

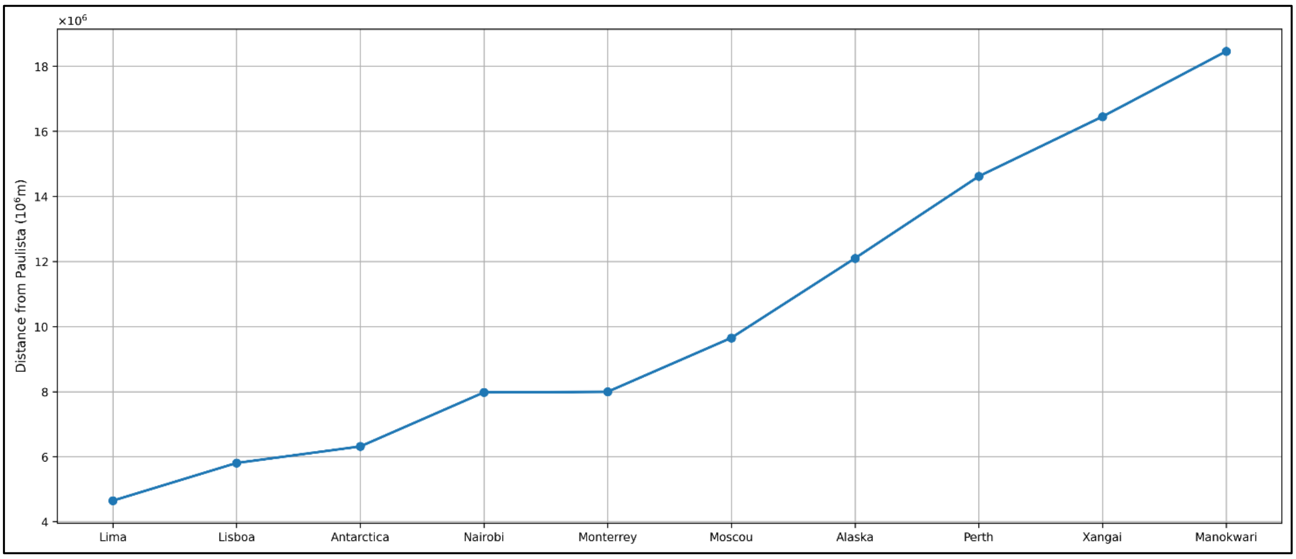

Figure 4 Geodesic distances from Paulista to the locations involved

To better visualize the differences between the two libraries, Figure 5 was prepared, which graphically presents these results for all locations.

The analysis of Figure 5 reveals an important aspect. It can be observed that there is a considerable difference between the three largest values of the differences in geodesic distances (Nairobi, Monterrey, and Lima) and the others. At the same time, it can be noted that these three points are among the four closest to the equator, as shown in Figure 6. However, the point closest to the equator is Manokwari, which, in turn, ranks fourth in the list of the largest differences in geodesic distances.

This occurs because the difference in longitude between Manokwari and Paulista, in absolute terms, is the largest among all locations, that is, it is the point farthest from Paulista in the east–west direction. This longitudinal difference is important because it is one of the initial computations performed by Vincenty's method, which operates with an iterative scheme at this stage of the calculations.

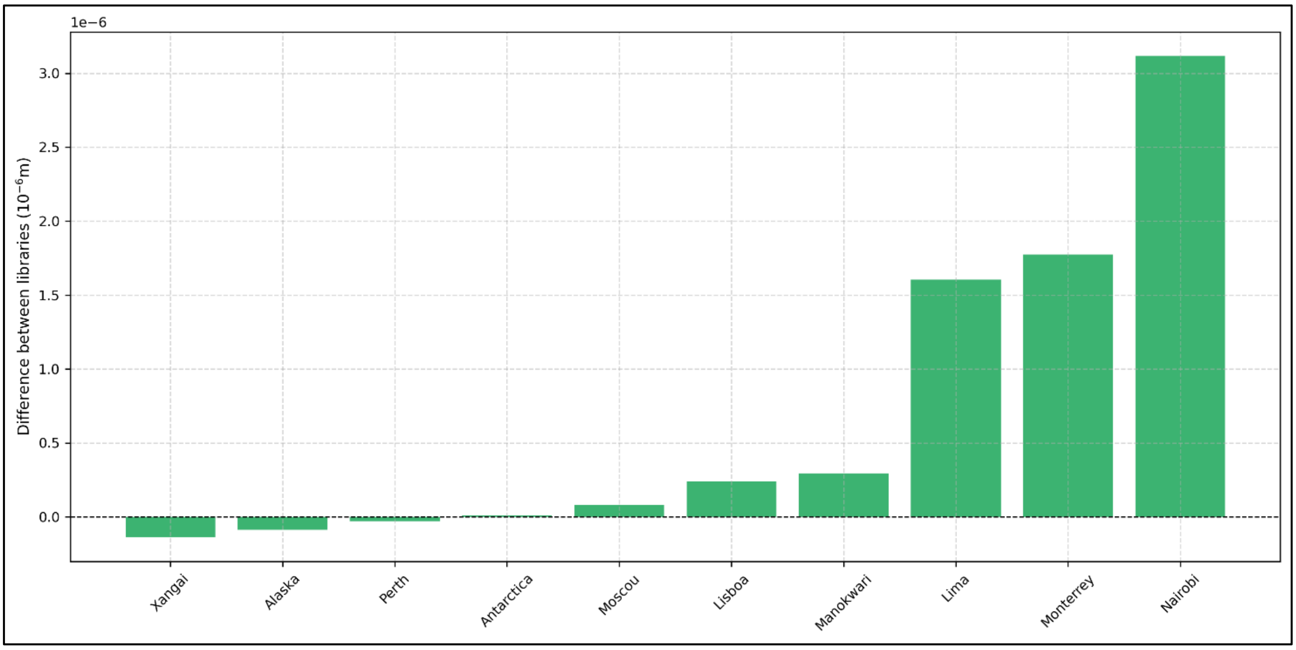

Figure 5 Geodesic distance differences between the two libraries

From the perspective of latitudes, by observing the horizontal axis of Figure 6, and with the exception of the already discussed case of Manokwari, it can be seen that the largest discrepancies occur at low latitudes, near the Equator, as in Nairobi and Lima, whereas at mid and high latitudes the differences tend to decrease significantly, approaching zero.

This behavior is directly related to the geometry of the ellipsoid and the way the algorithms handle it. In equatorial regions, where the radius of curvature in the prime vertical reaches its maximum values, small differences in the computation of sines and cosines of low latitudes or in the convergence of iterative methods may result in slightly larger deviations in the result. At mid-latitudes, such as Lisbon and Moscow, the combination of the radii of curvature in the prime vertical and in the meridional section tends to smooth out these effects, leading to smaller discrepancies.

At very high latitudes, near the poles, such as Antarctica and Alaska, the differences become practically negligible or even slightly negative. In these cases, despite the greater meridional curvature, the strong convergence of the meridians and the reduced influence of the longitudinal component decrease the numerical sensitivity between the implementations.

Therefore, the graph suggests that the maximum discrepancies between the libraries are mainly associated with low latitudes, while mid and high latitudes exhibit greater numerical stability, reinforcing that the observed effect is both geometric and computational in nature.

As can be seen in Figure 7, the smallest differences in geodesic distances occur precisely at the points that are farthest from Paulista in the east–west direction, namely: Antarctica, Alaska, Perth, and Shanghai. Manokwari presents a slightly higher value for the difference between geodesic distances due to its greater proximity to the Equator compared to these four points.

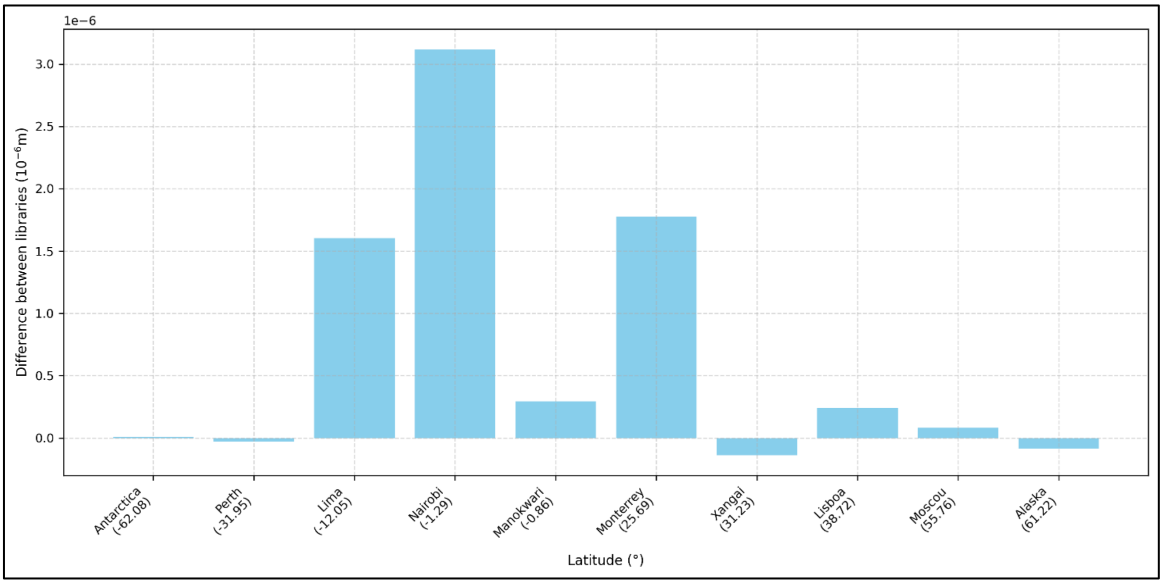

Figure 6 Latitudes of the locations involved and respective differences

Thus, for example, the point in Nairobi, which is closer to Paulista in the east–west direction and also very close to the Equator, resulted in the highest value for the difference between the geodesic distances computed by the two libraries.

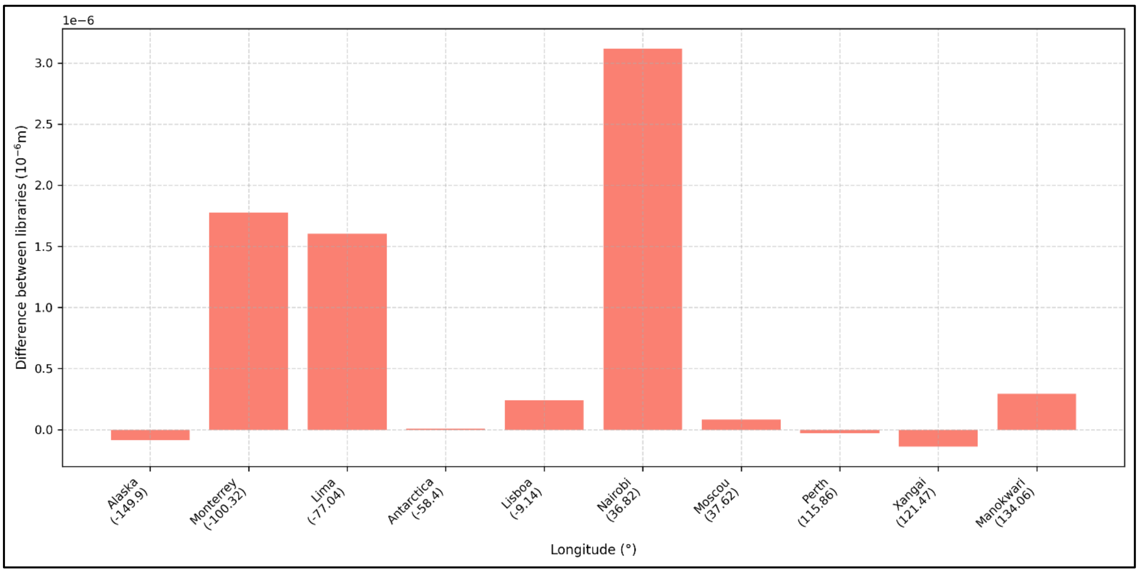

Figure 7 Longitudes of the locations involved and respective differences

An observation that supports the interpretation presented is that Manokwari is the point farthest from Paulista in the east–west direction, while at the same time being very close to the Equator. This means that it would be among the worst results if only latitudes were considered, and among the best results if only longitudes were considered.

However, if the average differences in geodesic distances from Table 2 are computed, the value obtained is 6.867×10^{-7} m, and when identifying which point has a value closest to this average, it turns out to be Manokwari.

In any case, since all the differences between the two libraries analyzed in this study are in the order of 10^{-6} m, discussions at this level of precision go beyond the accuracy required for most geodetic applications, which generally do not exceed the order of 10^{-4} m.

4. Conclusion

The comparative analysis between Python's *math* and *mpmath* libraries, using Vincenty's equations for the Inverse Problem of Geodesy, revealed that the differences in the computed geodesic distances are very small, in the order of micrometers, which, in terms of decimal arcseconds of latitude and longitude, correspond to nanometers. The application results showed that points closer to the Equator tend to accentuate the differences between the library outputs, whereas points located farther from the Municipality of Paulista in the east–west direction contribute to reducing these discrepancies.

The largest differences between the geodesic distances computed by the libraries were observed for the locations of Nairobi, Monterrey, and Lima, all situated near the Equator and with small longitudinal differences relative to Paulista. Conversely, the smallest differences were recorded for Antarctica, Perth, and Alaska, that is, points that are geographically farther from the origin in the east–west direction, reinforcing the influence of longitudinal differences on computational accuracy, since this aspect underlies Vincenty's method.

A particularly interesting case is that of Manokwari, which exhibited hybrid characteristics: although it is very close to the Equator (which would tend to produce larger differences), it is also the point farthest from Paulista in terms of longitude, which balances this effect. As a result, among the analyzed points, Manokwari is the one whose difference is closest to the average of the differences.

Therefore, it can be concluded that the numerical performance of the analyzed libraries is highly consistent with most high-precision geodetic applications. However, the observed variations (generally at the micrometer level) can be predominantly explained by the geometric relationships between the latitudes and longitudes of the points considered. For this reason, the *mpmath* library can be regarded as more efficient than the *math* library, due to its greater number of significant digits, which provides higher numerical precision in the convergence computations of Vincenty's equations.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Barazzetti L. Revitalizing astronomical azimuth determination: integrating modern computing with traditional techniques. *Sensors*. 2025;25(6):1871. doi:10.3390/s25061871.

[2] Bing J, Shaofeng B. The new non-iterative solution to the geodetic problem. *Acta Geodaetica et Cartographica Sinica*. 2007;36(3):269–273.

[3] Bowring B. The normal section—forward and inverse formulae at any distance. *Surv Rev*. 1971;21(161):131–136. doi:10.1179/sre.1971.21.161.131.

[4] Bowring B. Solution for azimuth of the geodesic in near antipodal situations with special reference to the behaviour of lines for which the azimuth is in the region of 90°. *Bull Géod*. 1977;51(1):17–32. doi:10.1007/bf02521538.

[5] Bowring B. The antipodal normal section. *Bull Géod*. 1978;52(3):213–222. doi:10.1007/bf02521774.

[6] Bowring B. The direct and inverse problems for short geodesic lines on the ellipsoid. *Surv Map*. 1981;41(2):135–141.

[7] Bowring B. Total inverse solutions for the geodesic and great elliptic. *Surv Rev*. 1996;33(261):461–476. doi:10.1179/sre.1996.33.261.461.

[8] Deng X, Sneeuw N, Tsoulis D. Optimized formulas of the gravitational field of a vertical cylindrical prism. *Geo-Spatial Inf Sci*. 2025:1–14. doi:10.1080/10095020.2024.2448227.

[9] Geodatos. Geographic coordinates finder [Internet]. 2026 [cited 2025 Jun 13]. Available from: https://www.geodatos.net/en/coordinates

[10] Haque M, Reza M, Karim M, Ali M, Samsuzzaman, Lee K, Kang Y, Chung S. Geometric alignment improves wheat NDVI calculation from ground-based multispectral images. *Remote Sens*. 2025;17(5):743. doi:10.3390/rs17050743.

[11] Instituto Brasileiro de Geografia e Estatística (IBGE). Relatório da estação 93191. Rio de Janeiro: IBGE; 2026.

[12] Jank W, Kivioja L. Solution of the direct and inverse problems on reference ellipsoids by point-by-point integration using programmable pocket calculators. *Surv Map*. 1980;40(3):325–337.

[13] Karney C. Test set for geodesics [dataset]. Zenodo; 2010. doi:10.5281/zenodo.32156.

[14] Karney C. Algorithms for geodesics. *J Geod*. 2012;87(1):43–55. doi:10.1007/s00190-012-0578-z.

[15] Karney C. Geodesics on an arbitrary ellipsoid of revolution. *J Geod*. 2023;98(1). doi:10.1007/s00190-023-01813-2.

[16] Kivioja L. Computation of geodetic direct and indirect problems by computers accumulating increments from geodetic line elements. *Bull Géod*. 1971;99(1):55–63. doi:10.1007/bf02521679.

[17] Mpmath. Mpmath [Internet]. 2026 [cited 2026 Mar 13]. Available from: https://mpmath.org/

[18] Pengfei C, Yingyan C. The current status and tendency of China millimeter coordinate frame implementation and maintenance. *Acta Geodaetica et Cartographica Sinica*. 2017;46(10):1327–1335.

[19] Pine D. *Introduction to Python for science and engineering*. 2nd ed. London: CRC Press; 2025.

[20] Pittman M. Precision direct and inverse solutions of the geodesic. *Surv Map*. 1986;46(1):47–54.

[21] Python. Floating-point arithmetic: issues and limitations [Internet]. 2026 [cited 2026 Mar 13]. Available from: https://docs.python.org/3/tutorial/floatingpoint.html

[22] Rainsford H. Long geodesics on the ellipsoid. *Bull Géod*. 1955;37(1):12–22. doi:10.1007/bf02527187.

[23] Robbins A. Long lines on the spheroid. *Empire Survey Review*. 1962; 16(125):301–309. doi:10.1179/sre.1962.16.125.301.

[24] Saito T. The computation of long geodesics on the ellipsoid through Gaussian quadrature. *Bull Géod*. 1979;53(2):165–177. doi:10.1007/bf02521087.

[25] Sjöberg L. New solutions to classical geodetic problems on the ellipsoid. In: *International Association of Geodesy Symposia*. Berlin: Springer; 2009. p. 781–784. doi:10.1007/978-3-540-85426-5_89.

[26] Sjöberg L. Solutions to the direct and inverse navigation problems on the great ellipse. *J Geod Sci*. 2012;2(3):200–205. doi:10.2478/v10156-011-0040-9.

[27] Sjöberg L, Shirazian M. Solving the direct and inverse geodetic problems on the ellipsoid by numerical integration. *J Surv Eng*. 2012;138(1):9–16. doi:10.1061/(asce)su.1943-5428.0000061.

[28] Sodano E. General non-iterative solution of the inverse and direct geodetic problems. *Bull Géod*. 1965;75(1):69–89. doi:10.1007/bf02530662.

[29] Thomas C, Featherstone W. Validation of Vincenty's formulas for the geodesic using a new fourth-order extension of Kivioja's formula. *J Surv Eng*. 2005;131(1):20–26. doi:10.1061/(asce)0733-9453(2005)131:1(20).

[30] Tulu B, Teshome F, Ampatzidis Y, Hailegnaw N, Bayabil H. AgriSenAI: automating UAV thermal and multispectral image processing for precision agriculture. *SoftwareX*. 2025;30:102083. doi:10.1016/j.softx.2025.102083.

[31] Vincenty T. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Surv Rev*. 1975;23(176):88–93. doi:10.1179/sre.1975.23.176.88.

[32] Wang J, Sun Y, Dai Y. Inverse geodetic problem for long distances based on improved Vincenty's formula. *J Appl Geod*. 2022;16(3):241–246. doi:10.1515/jag-2021-0057.

[33] Xiaohan X, Yukai X, Yajun L. Practical algorithm of geodetic problem solution. *Sci Technol Eng*. 2012;12(9):2062–2068. doi:10.3969/j.issn.1671-1815.2012.09.017.

[34] Yuhao S, Xiaodong Y, Shidong X. Geodetic distance calculation model based on spherical constraints. *J Ordnance Equip Eng*. 2020;41(1):200–204. doi:10.11809/bqzbgcxb2020.01.040.

[35] Zhengquan Z. Establishment and accuracy of high precision horizontal velocity field model in Yunnan and its surrounding areas. *J Navig Position*. 2021;9(2):114–117.