The Study of Geodesic Computations on Sphere and Spheroid with Optimized Numerical Methods

Jaruwan Burintharakot
6340320502@psu.ac.th
Faculty of Science and Industrial Technology
Prince of Songkla University, Surat Thani Campus
Thailand

Nathaphon Boonnam
nathaphon.b@psu.ac.th
Faculty of Science and Industrial Technology
Prince of Songkla University, Surat Thani Campus
Thailand

Abstract: *This article explores the problem of determining the shortest distance on spheres and spheroids, which is fundamental for calculating geodesic paths and establishing the computational relationship for all possible routes on a surface. Geodesic paths hold significant importance across a wide range of geometric applications. The study classifies geodesic paths on surfaces into two categories: analytical and numerical and compares the Runge-Kutta and Euler methods as computational techniques. These methods are employed to solve nonlinear ordinary differential equations governing geodesic paths. Based on the comparative analysis, the Runge-Kutta method is identified as the preferred approach for accurate calculations in this study. To ensure efficient computation and swift determination of values, the proposed calculations are implemented using the Google Colab platform, leveraging its capabilities for efficient numerical computations. Convergence rates analysis highlights the Runge-Kutta method's superior accuracy, making it the preferred choice when exact solutions are elusive. In summary, this research underscores the Runge-Kutta method's significance in approximating solutions to differential equations, offering valuable guidance for method selection in various applications involving geodesic paths.*

1. Introduction

Geodesic paths, which represent the shortest distance between two points on a surface [1, 2, 16], hold immense significance across diverse fields like computer graphics, geometric modeling, and surface segmentation [11]. While extensive research has been dedicated to geodesics on spheres, investigating geodesics on more complex surfaces poses a formidable challenge due to their fluctuating curvatures and non-differentiable nature [5, 15]. The curvature of a curve in a plane stands as a fundamental measure of its departure from a straight line [3]. Remarkably, a straight line manifests zero curvature [4], whereas a circle with radius r possesses a curvature of 1/r, clearly illustrating the inverse relationship between curvature and radius. Understanding the interplay of geodesics and curvature yields profound insights into the geometric attributes of surfaces.

The problem of determining the shortest path or distance between two points on a surface S in three-dimensional space, termed the geodesic path and geodesic distance, has undergone extensive exploration in the realms of differential geometry and computational geometry [5]. Traditional methods for computing geodesics can be categorized into three classes: solving the nonlinear differential equations that govern geodesics via numerical integration techniques, such as the Runge-Kutta method [6]; computing discrete geodesics on surfaces through specific rules; or constructing geodesics on smooth surfaces using geometric methods [7, 8, 16]. In the realm of differential geometry, the Euler-Lagrange equations have played a pivotal role, initially introduced by Euler and later refined by Lagrange to establish a mathematical foundation for the Principle of Least Action.

These equations serve as a versatile framework for analyzing a wide array of functions and have found application in the study of geodesic paths, including those subject to nonstationary constraints. While the first approach, employing numerical integration methods, yields precision, the inherent complexity of the differential equations governing geodesics often renders them arduous to solve. Discrete geodesics have garnered attention with the advancement of computational capabilities, yet they cannot be directly computed on the original smooth surface [7, 16]. Moreover, there exists limited research on directly constructing geodesics on smooth surfaces using geometric methods [9, 10].

This paper endeavors to explore geodesics on diverse surfaces and meticulously compare the accuracy of two numerical methods: Euler's method and the fourth-order Runge-Kutta method [6], as applied to the approximation of these geodesic paths. The computation and analysis of geodesics play an indispensable role in unraveling the properties and behaviors of surfaces. This study encompasses the investigation of geodesics on multiple surfaces, accompanied by theoretical elucidations of geodesic theory. The employed numerical methods, Euler's method and the fourth-order Runge-Kutta method, serve to compute geodesic paths [6], and their accuracy is subjected to rigorous testing. Additionally, we delve into the examination of geodesic distances through numerical calculations on several intriguing surfaces.

The successful execution of this experimental endeavor is made possible thanks to the generous financial support extended by Prince of Songkla University, Surat Thani Campus. The authors wish to express their profound gratitude for the invaluable backing and reinforcement that facilitated the realization of this research.

2. Preliminaries

Definition 2.1 [Parameterized Surface]: Consider a continuously differentiable $\mathbb{R}^3$ valued function of two variables. The two straight lines passing through the origin, given by the following equations:

$$G(u, v) = (x(u, v), y(u, v), z(u, v)) \tag{2.1}$$

Definition 2.2 [Geodesic Curve]: A geodesic curve $\gamma \in P(x, y)$ is such that $L(\gamma) = d_M(x, y)$.

Definition 2.3 [Geodesic Distance]: Given some Riemannian space (M, H) with $M \subset \mathbb{R}^s$, the geodesic distance is defined as

$$d_M(x, y) :\overset{\text{def}}{=} \min_{\gamma \epsilon P(x,y)} L(\gamma) \tag{2.2}$$

where $P(x, y)$ denotes the set of piecewise smooth curves joining x and y for any $x, y \in M^2$

$$P(x, y) \overset{\text{def}}{=} \{\gamma \backslash \gamma(0) \text{ and } \gamma(1) = y\}. \tag{2.3}$$

The shortest path between two points with the Riemannian metric is called a geodesic. If the metric H is well chosen, then geodesic curves can be followed salient features on images and surfaces.

A geodesic curve between two points might not be unique for instance about two antipodal points on a sphere. To perform the numerical computation of geodesic distances, we fix a set of starting points $S = (x_k)_k \subset M$ and consider only distance and geodesic curves from this set of points.

Definition 2.4 [Geodesic Euler Method]: Let $\tilde{S}$ be a polyhedral surface with a polyhedral tangential vector field v on S, let $y_0 \in S$ be an initial point, and let $h > 0$ a (possibly varying) stepsize. For each point $p \in S$ let $f(t_i, y_i, y(t_0))$ denote the unique straightest geodesic through p with initial direction $v(p)$ and evaluated at the parameter value t. A single iteration step of the geodesic Euler method is given.

Euler method can be classified as an easy-to-understand method for solving this ordinary differential equation. Principles for solving this ordinary differential equation. We find the approximate solution of y_{i+1} at t_{i+1} from the result. y_i is known at t_i, where h is the width of the calculation. Substitute the values in the equation

$$y_{i+1} = y_i + f(t_i + y_i)h \quad \rightarrow y_{i+1} = y_i \emptyset(t_i, y_i, h)h \, , y(t_0) = y_0$$

$$y_{i+1} = y_i + \emptyset(t_i, y_i, h)h \, , y(t_0) = y_0$$

$$\emptyset = a_1 k_1 + a_2 k_2 + \cdots + a_n k_n$$

where $a_1, a_2, \ldots, a_n$ are constants and

$$\begin{aligned} k_1 &= f(t_i, y_i) \\ k_2 &= f(t_i + p_1 h, y_i + q_{11} k_1 h) \\ k_3 &= f(t_i + p_2 h, y_i + q_{21} k_1 h + q_{22} k_1 h) \\ &\vdots \end{aligned}$$

where $p_1, p_2, \ldots, q_{21}, q_{22}, \ldots$ are constants.

Theorem 2.5 [Geodesic Runge-Kutta Method]: Let $\tilde{S}$ be a polyhedral surface with a polyhedral tangential vector field v on $\tilde{S}$, let $y_0 \in \tilde{S}$ be an initial point, and let $h > 0$ a (possibly varying) stepsize. For each point $p \in \tilde{S}$ let $f(x, y, y(x_0))$ denote the unique straightest geodesic through p with initial direction y_0 and evaluated at the parameter value x. A single iteration step of the geodesic Runge-Kutta method is given. This estimates that is closer to the real value than the Euler method. With the following methods for $y' = f(x, y), y(x_0) = y_0$ where f There is only one possible answer on a range with x_0 in $y_{n+1} = y_{(x_{n+1})}$ at $x_{n+1} = x_0 + (n+1)h$ where $n = 0, \ldots, N-1$.

We will get the formula $y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

where

$$\begin{aligned} k_1 &= hf(x_n, y_n), \\ k_2 &= hf\left(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_1\right), \\ k_3 &= hf\left(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_2\right), \\ k_4 &= hf(x_n + h, y_n + k_3), \\ x_{n+1} &= x_n + h. \end{aligned}$$

Property 2.6 [Sphere]: The properties below are with respect to the parameterization r given above. Parameter-curve tangent vectors the parameter-curve tangent vectors are

$$r_\theta(\theta, \varphi) = \begin{pmatrix} r\cos\theta\cos\varphi \\ r\cos\theta\sin\varphi \\ -r\sin\theta \end{pmatrix}, \quad r_\varphi(\theta, \varphi) = \begin{pmatrix} -r\sin\theta\sin\varphi \\ r\sin\theta\cos\varphi \\ 0 \end{pmatrix}. \tag{2.4}$$

The standard unit normal vector field is

$$\widehat{N}(\theta, \varphi) = \begin{pmatrix} \sin\theta\cos\varphi \\ \sin\theta\sin\varphi \\ \cos\theta \end{pmatrix}. \tag{2.5}$$

The prolate and oblate spheroidal infinite elements are based on a mathematically rigorous formulation, viz. a multiple expansion (a power series in 1 /r, where r is a spheroidal radius, multiplied by an oscillatory factor)

Property 2.7 [Prolate spheroid]: We now turn to the case when S a prolate spheroid is surrounding the structure. Introduce prolate spheroid coordinates $(r, \theta, \emptyset)$ defined by

$$\begin{cases} x = \sqrt{r^2 - f^2}\sin\theta\cos\emptyset \\ y = \sqrt{r^2 - f^2}\sin\theta\sin\emptyset \\ z = r\cos\theta \end{cases} \tag{2.6}$$

and let S be given by $r = r_1$(as before), where $r_1 > f$. That is, S is the surface $\frac{z^2}{r_1^2} + \frac{x^2}{r_1^2} + \frac{y^2}{-f^2} = 1$ in Cartesian coordinates, which is the surface obtained by rotating an ellipse with semi-major axis r_1, and foci at $(0,0,\pm f)$ about its major (z) axis.

Property 2.8 [Oblate spheroid]: Finally, we turn to the case when S is an oblate spheroid, where the appropriate coordinates $(r, \theta, \emptyset)$ are defined by

$$\begin{cases} x = r \sin\theta \cos\emptyset \\ y = r \sin\theta \sin\emptyset \\ z = \sqrt{r^2 - f^2} \cos\theta \,. \end{cases} \tag{2.7}$$

We again let S be given by $r = r_1 (r_1 > f)$, yielding the surface $\frac{x^2+y^2}{r_1^2} + \frac{z^2}{r_1^2 - f^2} = 1$ in Cartesian coordinates, which corresponds to rotating the same ellipse as before about its minor axis.

Numerical analysis is a branch of mathematics that deals with the study of methods and procedures used to obtain approximate solutions to mathematical problems. EndreSull and David Mayer defined the numerical analysis as a branch of mathematics that provides the theoretical foundation for the numerical algorithm, we rely on to solve a multitude of computational problem in mathematical models or the study of algorithms that use numerical approximation [18]. Numerical analysis naturally finds applications in all fields of engineering and the physical science, but in this 21st century, the life science and even the arts have adopted elements of scientific computations [19]. The overall goals of the field of numerical analysis in the design and analysis of techniques to give approximate but accurate solution are hard to get. It is, therefore, important to be able to estimate the error involved in such approximation. Thus, the aims of this work were to compare between Euler and Runge-Kutta methods to a rigorous analysis in order to demonstrate the efficiency of the methods to other similar techniques. It was also examining the effect of the steps on the accuracy of the techniques. Secularity band differences in the results of some numerical methods with the standard Euler's method of order three and four was examined.

To study experimental numerical methods such as implementing and running computational codes. It is necessary to understand in detail the properties/behavior of the numerical algorithms considered. Which can be represented in the form of an algorithm for computation and ease of programming. By setting the constant values as x_0 = 0, y_0 = 1, h = 0.025, and x = 1. Importing the calculation process through the program side in the equation (2.5).

From our previous discussion, it is clear that errors play an important role in numerical analysis. The error is of various types and arise from different sources. We have already seen two types of errors namely, truncation errors and roundoff errors. Among the other types of errors are input errors, which enter the computation through input data. This could happen, for example, if some inputs are known only approximately or determined, say, experimentally. Such errors are not the subject of study in numerical analysis. Neither are mistakes or blunders, because they cause the whole computation to become worthless and numerical analysis can do nothing to save the situation.

Data conversion also introduces errors, which are input errors. Data conversion error belongs to the category of round-off errors, to be discussed presently. We are interested here in numerical errors which, as we have seen, are of two kinds, namely, round-off errors and truncation errors.

Definition 2.9 [Error] : We formally define

$$\text{Error} = \text{True value} - \text{Approximate value} \tag{2.8}$$

and

$$\text{Relative Error} = \frac{\text{Error}}{\text{True value}} \tag{2.9}$$

In this study we use an algorithm to infer the relationship between Euler's method and Rung-Kutta method by the value used to calculate each method of the formula, respectively.

3. Methods and Results

To compute the geodesic path using the Euler's and Runge-Kutta methods, we first need to define the geodesic equations for the specific surface on which we want to find the geodesic path. The geodesic equations are differential equations that describe the curves representing the shortest distance between points on the surface.

Let us consider a surface S with parameterization given by two parameters u and v, and the geodesic path is represented by a curve with coordinates $(u(t), v(t))$, where t is the parameter along the geodesic path. The geodesic equations can be expressed as a set of first-order differential equations:

$$du/dt = f(u, v) \tag{3.1}$$
$$dv/dt = g(u, v) \tag{3.2}$$

where $f(u, v)$ and $g(u, v)$ are functions that depend on the specific curvature properties of the surface S. To compute the geodesic path using the Euler method, we discretize the geodesic equations and update the coordinates (u, v) iteratively for small time steps. Here are the general processes:

Algorithm 3.1 Euler's Method
1. Start
2. Define function
3. Get the values of x0, y0, h and xn : Here x0 and y0 are the initial conditions
h is the interval
xn is the required value
4. n = (xn – x0)/h + 1
5. Start loop from i=1 to n
6. y = y0 + h*f(x0,y0)
x = x + h
7. Print values of y0 and x0
8. Check if x < xn
If yes, assign x0 = x and y0 = y, Otherwise, go to 9.
9. End loop i
10. Stop

It is essential to choose a small enough time step to ensure accuracy in the approximation. Larger time steps may lead to inaccuracies, especially if the geodesic path has sharp curves or encounters regions of high curvature. It is important to note that the Euler method is a first-order numerical method and may not be as accurate as higher-order methods like the Runge-Kutta method. However, for simple surfaces and short geodesic paths, the Euler method can still provide reasonable approximations. Keep in mind that the specific form of f(u,v) and g(u,v) will vary depending on the surface S and its curvature properties, so the geodesic equations need to be adapted accordingly for each surface under consideration.

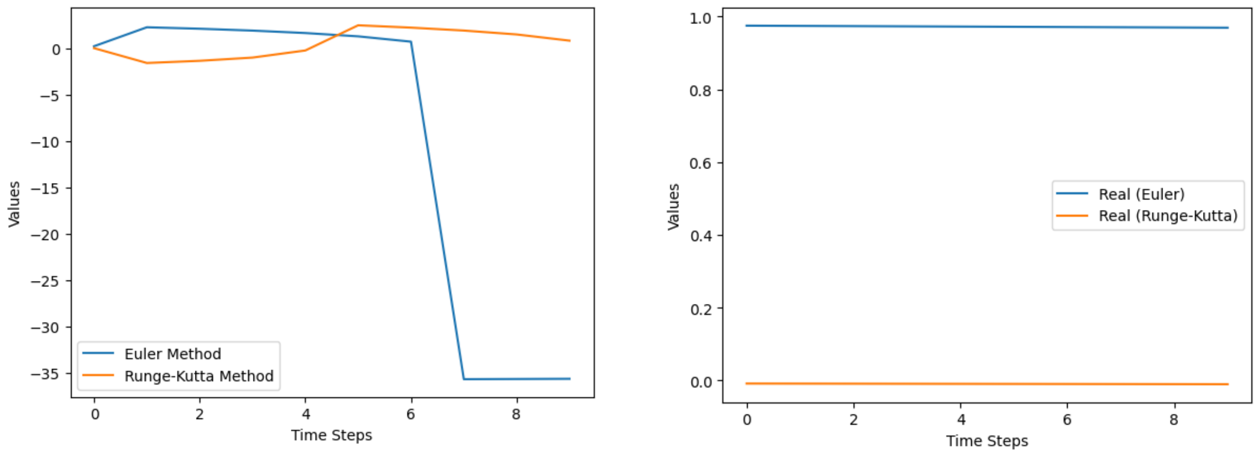
Algorithm 3.2 Runge-Kutta Method
1. Start
2. Define function f(x,y)
3. Read values of initial condition (x0 and y0)
number of steps (n)
calculation point (xn)
4. Calculate step size (h) = (xn - x0)/n
5. Set i=0
6. Loop k1 = h * f(x0, y0)
k2 = h * f(x0+h/2, y0+k1/2)
k3 = h * f(x0+h/2, y0+k2/2)
k4 = h * f(x0+h, y0+k3)
yn = y0 + k
i = i + 1
x0 = x0 + h
y0 = yn
While i < n
7. Display yn as result
8. Stop

The Runge-Kutta method uses four intermediate slopes (k1, k2, k3, k4) to compute the update, resulting in a more accurate approximation compared to the Euler method. The method effectively considers multiple points within each time step, capturing more information about the behavior of the geodesic path on the surface. The choice of a small-time step Δt is still important for accuracy in the approximation. By using the fourth order Runge-Kutta method, we can achieve better accuracy even for complex surfaces and longer geodesic paths. Again, it's important to adapt the specific functions f(u, v) and g(u, v) according to the surface S under consideration, as the curvature properties and geodesic equations will vary for different surfaces.

Table 3.3 Exact numerical values from our study

Round	Euler	Runge	Real (E)	Real(R)	error(E)	error(R)
0	0.22958	0.02464	0.97500	-0.00833	0.74542	0.03297
1	2.27594	-1.57748	0.97438	-0.00853	1.30157	1.56894
2	2.11864	-1.34199	0.97375	-0.00874	1.14489	1.33325
3	1.91677	-0.99420	0.97313	-0.00894	0.94364	0.98526
4	1.65444	-0.22553	0.97250	-0.00915	0.68194	0.21638
5	1.29667	2.48888	0.97188	-0.00936	0.32479	2.49823
6	0.72231	2.23764	0.97125	-0.00956	0.24894	2.24720
7	-35.68496	1.92211	0.97063	-0.00977	36.65559	1.93188
8	-35.66473	1.50234	0.97000	-0.00997	36.63473	1.51231
9	-35.64273	0.83337	0.96938	-0.01018	36.61210	0.84355
AVE. %error					11.52936	1.31700

In the result of our study, from Table 3.3, it shows that the results were significantly improved with the Runge-Kutta method. The table below summarizes all the results for the method. In comparing the two methods. This is trivial that the Runge-Kutta method is better than Euler especially when multiple xn values are required. If only a few xn values are required, the Runge-Kutta technique is superior to the Euler method as it produces slightly better results: See the following figure:



(a) (b)

Figure 3.1 Graph of approximate solution of standard Euler and Runge-Kutta compare to (a) exact solution and (b) real value

The concept of convergence rate serves as a fundamental criterion for assessing the performance of numerical methods. It quantifies how quickly a numerical method converges to the true solution as the step size decreases, with higher values indicating faster convergence. In our analysis, we will examine the convergence rates of two numerical methods: Euler and Runge-Kutta. These rates will provide valuable insights into the methods' respective abilities to approach accurate solutions with decreasing step sizes. Subsequently, we will discuss the implications of these convergence rates, considering the trade-offs between accuracy and computational efficiency, to make informed decisions regarding the choice between the Euler and Runge-Kutta methods in specific applications. See the following figure:

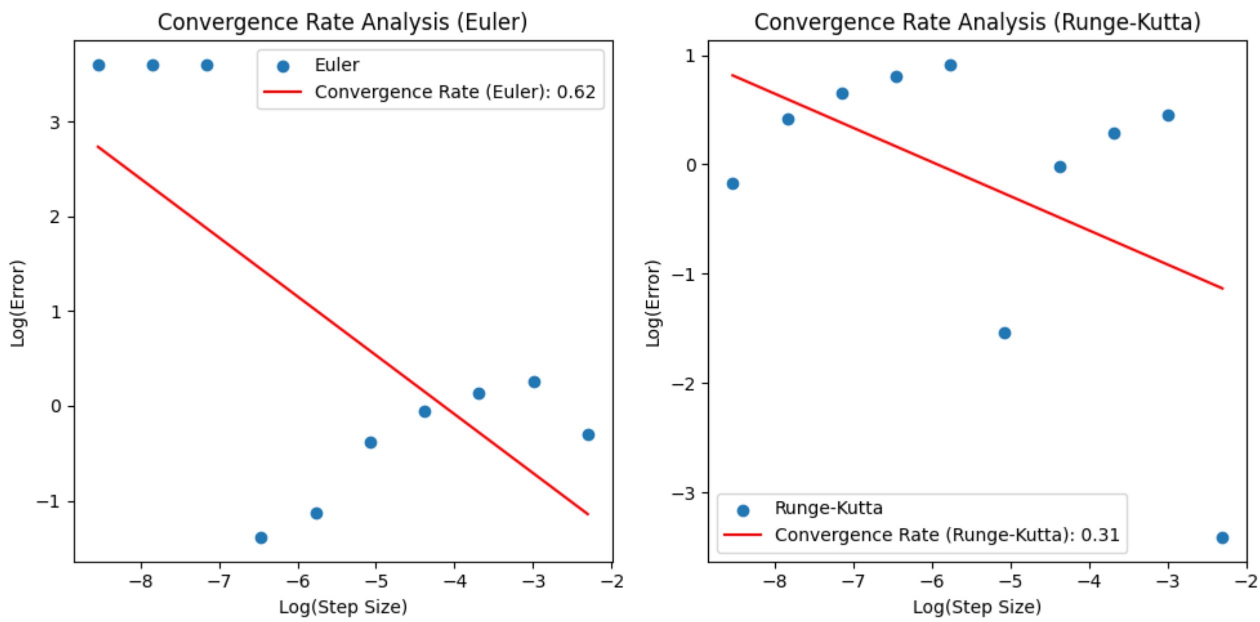


Figure 3.2 Graph of standard Euler and Runge-Kutta compare to Percentage-error

The convergence rates, Convergence Rate (Euler): 0.62 and Convergence Rate (Runge-Kutta): 0.31, provide valuable insights into the performance of these numerical methods. Convergence rate measures how quickly a method approaches the true solution as the step size decreases, with higher values indicating faster convergence. In this context, the Euler method outperforms the Runge-Kutta method with a higher convergence rate (0.62), signifying faster convergence as step size decreases. Conversely, the Runge-Kutta method has a lower convergence rate (0.31), indicating slower convergence and requiring finer discretization for the same accuracy level. These convergence rates prompt considerations of practicality; the Euler method converges faster but may demand smaller step sizes and increased computational effort, while the Runge-Kutta method, with slower convergence, may offer efficiency advantages with larger step sizes. Therefore, selecting the most suitable method depends on balancing accuracy and computational efficiency in your specific application, with these convergence rates guiding your decision-making process.

4. Conclusion

In this study, we have conducted a thorough comparison between two numerical methods, Euler's method and the Runge-Kutta method, for approximating solutions to differential equations. These methods share the common iterative approach of updating variables over time; however, they diverge in crucial aspects that profoundly affect their accuracy. Euler's method initiates with an initial variable value and subsequently updates it based on the derivative at each step, signifying the rate of change at a specific point. The new value is calculated by multiplying the derivative by a small time interval and adding it to the previous value, iterating until the desired solution is reached. In contrast, the Runge-Kutta method introduces a new default value and systematically refines it. By incorporating multiple derivative approximations at various points within each time interval and calculating weighted averages of these approximations, it achieves heightened accuracy. Our experiments, conducted in a web service environment such as Google Colab, consistently demonstrate the Runge-Kutta method's superior performance over Euler's method. This superiority arises from its capacity to leverage a wealth of information regarding variable behavior within each time interval, resulting in significantly more precise approximations. Consequently, the Runge-Kutta method consistently exhibits smaller error values compared to Euler's method, indicating its superior accuracy.

In practical applications where exact solutions are often elusive, numerical methods like Euler's and the Runge-Kutta method prove invaluable. While exact solutions can serve as benchmarks for accuracy assessment, they are frequently unavailable, especially for intricate or nonlinear differential equations. In such scenarios, the accuracy of these methods becomes paramount, and it can be gauged effectively through rigorous testing and error analysis.

Our comprehensive analysis demonstrates that the Runge-Kutta method consistently excels, providing heightened accuracy across a spectrum of scenarios, including both linear and nonlinear solutions, even extending to fourth-order differential equations. This exceptional accuracy positions the Runge-Kutta method as the preferred choice in situations where precision plays a pivotal role.

In summary, our findings strongly support the Runge-Kutta method's superiority in terms of accuracy when approximating solutions to differential equations. Its advanced approximation techniques and consistent excellence in precision underscore its significance in practical applications demanding superior accuracy.

Acknowledgements The author would like to thank her colleague and reviewers for their valuable comments and input.

References

[1] Andrew, T. (2026). *Obtaining the Shortest Distance Between Two Points on a Surface*. Calculus III Honors Component on Geodesics, 1–13.

[2] Andrew, T. (2020). *An introduction to geodesics: the shortest distance between two points*. Functional Analysis (math.FA), 1–13.

[3] Richard, J., Lisle, J., and Robinson, M. (1995). *The Mohr circle for curvature and its application to fold description*. Journal of Structural Geology, 17(5), 739–750.

[4] Bradley, E. (1975). *Defining the Curvature of a Statistical Problem (with Applications to Second Order Efficiency)*. The Annals of Statistics, 3(6), 1189–1242.

[5] Bose, P., Maheshwari, A., Shu, C., and Wuhrer, S. (2011). *A survey of geodesic paths on 3D surfaces*. Computational Geometry, 44(9), 486–498.

[6] Sharp, N. A. (1981). *Comparison of numerical methods for the integration of the black hole geodesic equations*. Journal of Computational Physics, 41(2), 295–308.

[7] Peng, Z., Ronglei, S., and Tao, H. (2015). *A geometric method for computation of geodesic on parametric surfaces*. Computer Aided Geometric Design, 38, 24–37.

[8] Hotz, I. and Hagen, H. (2000). *Visualizing geodesics*. Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145).

[9] Peng, Z., Ronglie, S., and Tao, H. (2015). *A geometric method for computation of geodesic on parametric surface*. Computational Geometry, 35, 1–14.

[10] Michael, R., Tim, H., and Olga, S. H. (2018). *Discrete Geodesic Nets for Modeling Developable Surfaces*. ACM Transactions on Graphics, 37(2), 1–17.

[11] Joon-Kyung, S., Won-Ki, J., and Elaine, C. (2008). *Anisotropic Geodesic Distance Computation for Parametric Surfaces*. IEEE International Conference on Shape Modeling and Applications - Anisotropic geodesic distance computation for parametric surfaces, 179–186.

[12] Nabutovsky, A. and Rotman, R. (2002). *The Length of the Shortest Closed Geodesic on a 2-Dimensional Sphere*. IMRN International Mathematics Research Notices, 23, 1211–1222.

[13] Kimmel, R. and Sethian, J. A. (1998). *Computing geodesic paths on manifolds*. Proceedings of the National Academy of Sciences, 95(15), 8431–8435.

[14] Kumar, R. S. (2014). *Geodesic equation in spherical surface*. viXra:1404.0016: 1–7.

[15] Herman, R. (2018). *Euler Equation and Geodesics*. UNCW,1–16.

[16] Panou, G. and Korakitis, R. (2017). *Geodesic equations and their numerical solutions in geodetic and Cartesian coordinates on an oblate spheroid*, Journal of Geodetic Science, 7(1), 31–42.

[17] David, S. B. and Richard, L. H. (1996). *Prolate and oblate spheroidal acoustic infinite elements*. Computer Methods in Applied Mechanics and Engineering, 158(1–2), 117–141.

[18] Endre, S. and Mayer, D. F. (2003). *An introduction to Numerical Analysis*, Cambridge University Press, 444 pages.

[19] Hussain, K., Ismail, F., and Senua, N. (2016). *Solving Direct Special First-Ordinary Differential Equations using Runge-Kutta*, Journal of Computational and Applied Mathematics, 306, 179–199.