

Volume 2, Issue 4

Research Article

Date of Submission: 21 Aug, 2026

Date of Acceptance: 21 Sep, 2026

Date of Publication: 06 Oct, 2026

A Comparative Analysis of Forward, Backward and Central Finite Difference Schemes for Numerical Differentiation of Transcendental Function

Yusupha Bah¹, Bai Mbye Cham^{1*} and Dawda Charreh¹

¹Department of Mathematics, University of the Gambia.

Corresponding Author:

Bai Mbye Cham, Department of Mathematics, University of the Gambia.

Citation: Bah, Y., Cham, B, M., Charreh, D. (2026). A Comparative Analysis of Forward, Backward and Central Finite Difference Schemes for Numerical Differentiation of Transcendental Function. *Energy Sci Eng Policy*, 2(4), 01-10.

Abstract

Research has shown that not all mathematical equations can be solved analytically. Numerical methods provide effective approaches for obtaining approximate solutions to complex mathematical problems that cannot be solved analytically using traditional pen-and-paper techniques. In this research, we investigate one of the most commonly used numerical methods, namely, the finite difference method. This is a numerical technique used to approximate the values of derivatives using discrete data points. The finite difference method consists of three main schemes: the forward difference, backward difference, and central difference methods. In this study, we compare these three finite difference schemes to determine which one converges fastest over a range of step sizes. Transcendental functions, including the exponential function (e^x) and the trigonometric function $\sin(x)$, are used to evaluate and compare the performance and convergence of the three schemes. MATLAB is used to analyse the relationship between the step size h and the truncation and round-off errors by examining how changes in the step size affect the convergence rate of each scheme. The results show that the central difference scheme is significantly more accurate than the forward and backward difference schemes, demonstrating second-order convergence. However, when the step size becomes excessively small, round-off error, sometimes associated with the turning-point phenomenon, becomes significant. Consequently, instead of continuing to decrease, the numerical error begins to increase. Based on these findings, researchers seeking higher accuracy and faster convergence in numerical differentiation should consider using the central difference scheme, particularly when an appropriate step size is selected.

Keywords: Finite Difference Method, Forward Difference, Backward Difference, Central Difference, Truncation Error, Round-off Error, Numerical Differentiation

Introduction

Numerical methods are essential for solving mathematical models arising from real-world problems, particularly when analytical solutions are difficult, cumbersome, or unavailable. Conventional differentiation rules are effective for simple continuous functions, whereas experimental and observational data are commonly represented by discrete values rather than explicit continuous expressions. The Finite Difference Method (FDM) provides an important connection between continuous differential calculus and discrete computational data by approximating derivatives using finite step sizes. Its theoretical foundation can be traced to the limit definition of a derivative, which is converted into a computationally implementable form through Taylor series expansions. The development of finite difference calculus dates back to the work of Newton and Taylor, predating modern computing by several centuries; however, advances in computational technology have substantially expanded the range and scale of problems to which these classical numerical techniques can be applied [1–4].

For a sufficiently smooth function, Taylor series expansion provides the mathematical basis for deriving the principal finite difference formulas. Expanding a function about a point x and rearranging the resulting expression gives the forward difference approximation, while expansion about $x-h$ leads naturally to the backward difference approximation. Because a computer cannot evaluate an infinite Taylor series, the expansion must be truncated, introducing a truncation error. For both the forward and backward schemes, the leading truncation error is proportional to the step size h , resulting in first-order convergence, $O(h)$ [5]. The central difference scheme is obtained by combining Taylor expansions at $x+h$ and $x-h$. In this formulation, the leading first-order error terms cancel, leaving a dominant error proportional to h^2 and consequently producing a second-order convergence rate, $O(h^2)$. This theoretical distinction provides the basis for a systematic comparison of the accuracy of the three classical finite difference schemes.

An important numerical issue in finite difference differentiation is the interaction between truncation error and round-off error. Reducing the step size generally decreases truncation error; however, sufficiently small values of h can amplify the effects of finite-precision arithmetic and increase round-off error. Therefore, the most accurate numerical derivative is not necessarily obtained by choosing the smallest possible step size. Previous numerical-analysis studies have examined this balance and derived theoretical estimates of optimal step sizes by considering the competing error contributions, while also demonstrating that the optimal value depends on the numerical scheme being used [6,7]. Finite difference methods have broad applications in fluid dynamics, heat transfer, structural engineering, computational finance, image processing, and weather forecasting, where continuous governing equations must often be represented and solved numerically [8–15]. Through discretization, a continuous domain is replaced by a finite grid of nodes, and differential equations are transformed into algebraic systems that can be solved computationally.

Beyond the basic forward, backward, and central formulations, higher-order and compact finite difference schemes have been developed to improve numerical accuracy. Fourth- and sixth-order schemes use additional neighboring points to reduce higher-order terms in the Taylor expansion, while compact schemes achieve high accuracy using more localized stencils and can be advantageous near boundaries where wider stencils are difficult to apply [16,17]. These improvements, however, generally require additional function evaluations and more complex implementations. Other differentiation approaches, including automatic differentiation and complex-step differentiation, can provide very high accuracy under appropriate conditions. Automatic differentiation can achieve derivatives to approximately machine precision without truncation error, but it generally requires access to the computational graph [18,19]. Complex-step differentiation avoids subtractive cancellation and can permit extremely small step sizes, but it typically requires an analytic function and access to or modification of the underlying computational implementation [20]. Consequently, finite difference methods remain particularly useful for black-box functions, experimental measurements, and situations where only function evaluations are available.

A controlled mathematical benchmark is therefore necessary to assess the accuracy, convergence characteristics, and stability of the forward, backward, and central difference schemes. The present study employs the transcendental functions (e^x) and $\sin(x)$ because their exact analytical derivatives are known, providing reliable reference solutions against which numerical approximations can be evaluated. This enables the systematic calculation of absolute and relative errors over a range of step sizes and facilitates direct investigation of the transition from truncation-error-dominated behavior to round-off-error-dominated behavior. Previous comparative studies have generally confirmed the expected first-order convergence of forward and backward differences and second-order convergence of the central difference scheme [21]. However, comparatively less emphasis has been placed on explicitly identifying the point at which round-off error becomes dominant and determining the step size corresponding to the minimum numerical error. The present investigation addresses this aspect by empirically identifying the accuracy turning point for the selected transcendental functions.

Finite difference methods also remain fundamental in the numerical solution of partial differential equations, including the heat and wave equations, where spatial and temporal derivatives are approximated at discrete grid points [22]. Such problems typically require explicit or implicit time-stepping procedures, and the choice of step size can influence not only accuracy but also numerical stability, as illustrated by stability restrictions such as the Courant–Friedrichs–Lewy (CFL) condition [23]. In addition, finite differences remain relevant in computational science and machine learning, particularly for numerical gradient checking, where finite difference approximations can be compared with analytically computed gradients to identify implementation errors [24]. In the present comparative analysis, MATLAB is used as the principal computational environment because it provides an effective platform for systematic numerical experimentation and the evaluation of floating-point errors. By progressively reducing h and monitoring the resulting numerical errors, the MATLAB implementation enables empirical verification of the theoretical convergence behavior and provides a quantitative comparison of the forward, backward, and central difference schemes [25].

Mathematical Formulation

The finite difference method is a method for estimating the value of derivatives from discrete data points. To mathematically compare the accuracy and convergence of different schemes, we need to mathematically define forward, backward and central difference formulas. Let $f(x)$ be a differentiable function and h a small finite step size.

The forward difference approximation is derived directly from the limit definition of the derivative by evaluating the function at x and a forward point $x + h$:[1]

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

The backward difference approximation, by contrast, uses the function at x and the preceding point $x - h$:[13]

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

The central difference approximation is better balanced by evaluating the function at points on either side of x , effectively averaging the forward and backward approaches:[1]

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

The second derivative approximation reuses the Taylor expansions already given for the first derivative central difference, but combines them differently. Expanding $f(x+h)$ and $f(x-h)$ about x gives

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \dots \quad (1)$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \dots \quad (2)$$

The even-order terms cancel when these are subtracted, giving the first-derivative central difference; if they are *added* instead, the odd-order terms cancel, leaving

$$f(x+h) + f(x-h) = 2f(x) + h^2f''(x) + O(h^4). \quad (3)$$

which can be rearranged to the formula used in this study:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \quad (4)$$

The truncation error is $O(h^2)$, same order as for the first derivative case, so any accuracy difference discussed below is due to round-off behaviour, not truncation order.

Error Analysis

Table 1 summarizes the leading truncation error term, truncation order and roundoff scaling for each of the formulas derived in this section. The forward and backward differences have the same leading error term in magnitude, $\frac{h}{2}f''(\xi)$, but opposite signs. Both have $O(\epsilon/h)$ round-off error, where ϵ is the machine epsilon. The central difference has a higher truncation order, $O(h^2)$, but the same $O(\epsilon/h)$ round-off scaling of the one-sided schemes. The second-derivative central difference also has the same $O(h^2)$ truncation order, but scales as $O(\epsilon/h^2)$ in round-off, giving a direct theoretical explanation of the steeper accuracy breakdown seen empirically in Table 6 and Figure ??.

Scheme	Leading Truncation Term	Truncation Order	Round-off Scaling
Forward difference	$-\frac{h}{2}f''(\xi)$	$O(h)$	$O(\epsilon/h)$
Backward difference	$+\frac{h}{2}f''(\xi)$	$O(h)$	$O(\epsilon/h)$
Central (1st deriv.)	$-\frac{h^2}{6}f'''(\xi)$	$O(h^2)$	$O(\epsilon/h)$
Central (2nd deriv.)	$-\frac{h^2}{12}f^{(4)}(\xi)$	$O(h^2)$	$O(\epsilon/h^2)$

Table 1: Summary of Truncation and Round-off Error Scaling for Each

Results and Discussion

In this section, we present the numerical results obtained by implementing the forward, backward, and central difference schemes in MATLAB. The main objective is to study the behaviour of the absolute error for different step sizes h , and to verify the convergence rates for the transcendental functions $f(x) = e^x$ and $f(x) = \sin(x)$.

Figures 1 and 2 graphically depict the data of Tables ??–4 by plotting absolute error versus step size on a log–log scale over the entire range $h = 10^{-1}$ to 10^{-16} . Both curves exhibit the characteristic V-shape discussed in the Method section: a linear drop due to truncation error dominance, a minimum at the turning point given in Table 4, and a noisy rise as round-off error takes over. The steeper slope of the central difference curve than the forward and backward curves is a direct visual confirmation of the second-order convergence of the central difference curve.

Figure 3 superimposes the first- and second-derivative central-difference error curves for $f(x) = e^x$ at $x = 1$, showing in a direct visual form the comparison summarized in Table 6. The second derivative curve diverges sharply from the first derivative curve when round-off error starts to dominate, reaching errors as large as $10^{15} - 10^{20}$ for the smallest step sizes tested, although both curves follow the same $O(h^2)$ truncation error slope for larger h . This is consistent with the underlying error scaling: since the second difference formula divides by h^2 rather than by h , machine round-off noise of order 10^{-16} is amplified by a factor of approximately $1/h^2$, and thus there is catastrophic growth of error as $h \rightarrow 10^{-16}$.

In Table 2, we report the absolute error and observed order of convergence of the three finite difference schemes applied to $f(x) = e^x$ at $x = 1$, where the exact derivative is known analytically as $f'(1) = e^1 \approx 2.718281828$. Step sizes ranging from $h = 10^{-1}$ to $h = 10^{-8}$ were used, a range over which truncation error remains the dominant source of error for all three schemes. The observed order of convergence agrees with theory down to about $h = 10^{-4}$, with the forward and backward schemes at about 1.00 and the central scheme at about 2.00. Below $h = 10^{-5}$, the observed order of the central scheme starts to deviate from 2.00 as discussed further in the Discussion.

Table 3 shows the same analysis for $f(x) = \sin(x)$ at $x = \pi/4$, where $f'(\pi/4) = \cos(\pi/4) \approx 0.707106781$. The same range of step sizes, $h = 10^{-1}$ to $h = 10^{-8}$, was used for direct comparison with Table ?. The results follow an almost identical pattern: first-order convergence for the forward and backward schemes and second-order convergence for the central scheme until about $h = 10^{-4}$, then the same departure from theory at smaller step sizes. The agreement for two different functions indicates that the round-off breakdown is a generic property of the schemes rather than a peculiarity of e^x .

We summarize in Table 4 the step size h at which each scheme achieves the minimum attainable error, obtained by scanning the full range $h = 10^{-1}$ to $h = 10^{-16}$ for both benchmark functions. The forward and backward schemes give the same turning points and minimum errors for e^x at $x = 1$. They are both first-order, one-sided approximations of the same magnitude but opposite sign. The central scheme achieves a minimum error that is about 3 to 4 orders of magnitude smaller than either of the one-sided schemes for a step size that is about 2 orders of magnitude larger. This pattern also holds for $\sin(x)$ at $x = \pi/4$, although in this case the turning point of the backward scheme is found to occur at a much smaller h than that of the forward scheme, a difference which is discussed in more detail in the Discussion.

For $f(x) = e^x$ at $x = 1$ and $f(x) = \sin(x)$ at $x = \pi/4$, the absolute error of the central second-difference approximation $f''(x) \approx [f(x+h) - 2f(x) + f(x-h)]/h^2$ is given in Table 5, where the exact second derivatives are given by $f''(1) = e^1$ and $f''(\pi/4) = -\sin(\pi/4)$, respectively. Compared to the first-derivative results in Tables 2 and 3, here the error decreases only up to approximately $h = 10^{-4}$, after which it increases rapidly, reaching values worse than at $h = 10^{-1}$ when $h = 10^{-7}$ is reached. This earlier and steeper breakdown is discussed further in the Discussion, along with a direct comparison to the first-derivative results in Table 6.

Table 6 directly compares the turning point and the minimum error attainable by the central scheme for the first and second derivatives of both benchmark functions. For e^x the second-derivative turning point is at $h \approx 2.46 \times 10^{-4}$, which is about 500 times larger than the first-derivative turning point at $h \approx 4.81 \times 10^{-7}$, and the corresponding minimum error is about four orders of magnitude larger. The same disagreement holds for $\sin(x)$. This agrees with the underlying error theory: because the second difference formula divides by h^2 not h , round-off error grows faster as h decreases, so that the optimal step size must be significantly larger, and the best achievable accuracy is limited.

Table 7 presents the turning point of the central difference scheme for three different points for each benchmark function. This is to test whether the behavior observed in Tables ??–4 is specific to the single point tested originally, or if it is more general. The turning point and minimum error for e^x are of the same order of magnitude at $x = 0, 1, 2$ although the function value and derivative increase considerably in this range. The same consistency is observed for $\sin(x)$, except the result at $x = 0$ is a measured error of exactly zero; this is coincidental (between the central-difference approximation and the double-precision value of $\cos(0) = 1$ at this particular step size) and not an exact result and should not be interpreted as the scheme being error-free.

Tables 2 and 3 show that the observed order has a matching to the theory within almost 10^{-4} , after which it deviates from the expected value for all three schemes. The departure is much more dramatic for the central scheme, for which

the order drops from 2.00 to negative values within one or two decades of h , while the forward and backward schemes drop only slightly below 1.00 for the same range. This asymmetry is because the truncation error of the central scheme is already very small, of the order of 10^{-11} , when round-off starts to interfere. It needs comparatively little round-off noise to swamp the remaining signal. This does not mean that the theory is broken; the deviation is an effect of the finiteprecision computation, not of the approximation.

This is the result of catastrophic cancellation. As h becomes small, $f(x + h)$ and $f(x)$ (or $f(x - h)$) agree in most of their leading digits, and subtracting nearly identical numbers destroys that shared information, leaving only the last few digits, which are dominated by storage error rather than real signal. This limitation is directly inherited from IEEE 754 double-precision arithmetic, which represents real numbers with only about 15–17 significant decimal digits [26]. Since this effect influences both terms in the numerator of each scheme, the resulting round-off error increases as h decreases, opposite to the decreasing truncation error, causing the turning point observed in Tables 2–4 and Figures 1–2.

Table 4 is a straightforward translation to practical guidance. The central scheme achieves errors of the order 10^{-12} – 10^{-13} at step sizes close to 10^{-6} – 10^{-7} , whereas the forward and backward schemes stagnate at 10^{-9} for similarly small h , so central is clearly superior in both accuracy and step-size sensitivity. Since the turning point of the two-sided schemes is at relatively larger h , the user does not have to carefully tune h to stay near-optimal as in the one-sided schemes with a narrower safe operating window. These empirical turning points provide a concrete example for the general truncation–round-off trade-off discussed in the optimal step-size literature, here identified explicitly for two particular functions rather than only in the abstract [2].

As shown in Table 6 and Figure ??, the second-derivative central difference reaches its turning point at a step size of around 500 times larger than the first-derivative case, with a minimum error around four orders of magnitude worse. Not because the second-difference formula is less accurate in theory; its truncation error is $O(h^2)$, the same order as the first-derivative central scheme. This difference is entirely due to a round-off effect: because the formula divides by h^2 instead of h , the same order of machine noise, $\sim 10^{-16}$, is multiplied by a factor of $1/h^2$ instead of $1/h$, so round-off error increases much more steeply as h decreases. The practical consequence is obvious; the numerical approximation of higher-order derivatives requires proportionally larger steps to avoid this penalty.

As we can see in Table 7, the turning point and minimum error for the central scheme are in the same order of magnitude for both functions and for all tested x -values, although the range of magnitudes of e^x between $x = 0$ and $x = 2$ is quite large. Thus, the results presented in Tables 2–4, which were obtained at one evaluation point per function, are general and not restricted to that single point. The exact-zero error for $\sin(x)$ at $x = 0$ is, as noted above, a floating-point coincidence, not something that should be taken as indicating error-free computation, and shouldn't be read to contradict this general pattern.

The trade-off discussed in this study is specific to methods that approximate derivatives from real-valued function evaluations. This is avoided entirely by techniques such as automatic differentiation and complex-step differentiation, the first by applying the chain rule to the computational graph of the function, and the latter by using a complex step to entirely remove subtractive cancellation [16,18]. But neither one applies if you only have a black-box function evaluation that returns a real number. This is exactly the setting where finite differences come in. Contrary to prior work that analytically derives an optimal step size, this work empirically finds an optimal step size for two transcendental functions, offering a concrete complement to this theoretical foundation.

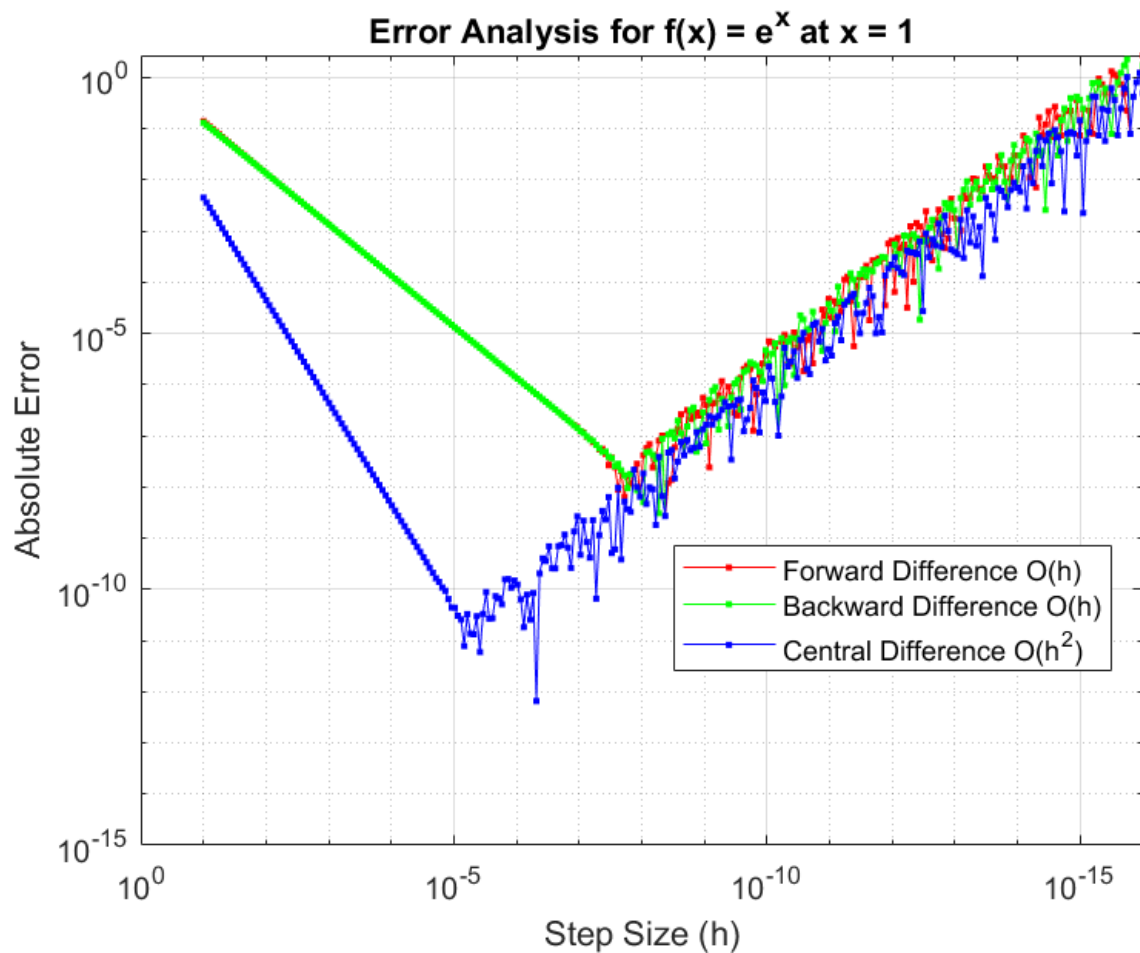


Figure 1: Log–log Error Curves for $f(x) = e^x$ at $x = 1$, Spanning $h = 10^{-1}$ to 10^{-16}

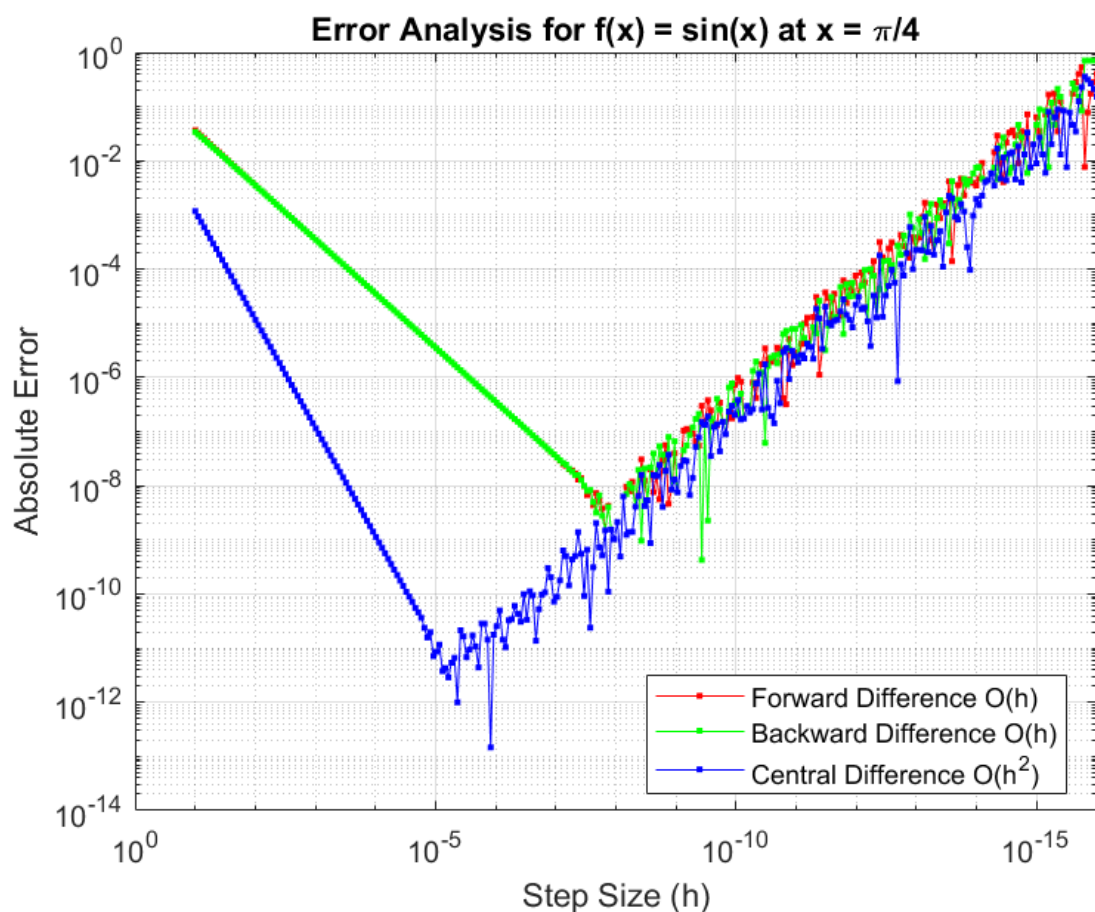


Figure 2: Log–log Error Curves for $f(x) = \sin(x)$ at $x = \pi/4$, Spanning $h = 10^{-1}$ to 10^{-16}

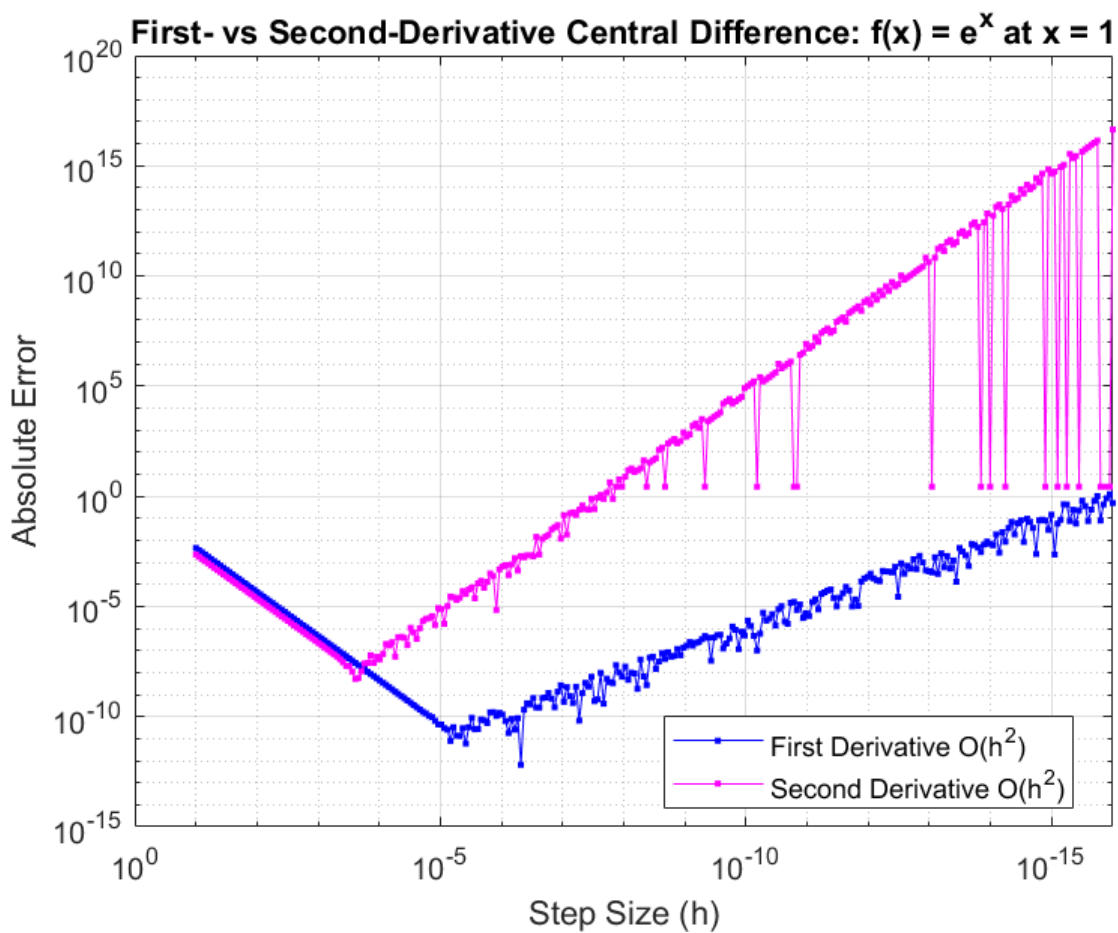


Figure 3: Comparison of First- and Second-Derivative Central-Difference Error for $f(x) = e^x$ at $x = 1$, Illustrating the much Steeper Round-off Blowup for the Second Derivative at Small h

h	Forward Error	Order	Backward Error	Order	Central Error	Order
10^{-1}	1.4056×10^{-1}	—	1.3149×10^{-1}	—	4.5327×10^{-3}	—
10^{-2}	1.3637×10^{-2}	1.01	1.3546×10^{-2}	0.99	4.5305×10^{-5}	2.00
10^{-3}	1.3596×10^{-3}	1.00	1.3587×10^{-3}	1.00	4.5305×10^{-7}	2.00
10^{-4}	1.3592×10^{-4}	1.00	1.3591×10^{-4}	1.00	4.5306×10^{-9}	2.00
10^{-5}	1.3591×10^{-5}	1.00	1.3591×10^{-5}	1.00	5.8586×10^{-11}	1.89
10^{-6}	1.3585×10^{-6}	1.00	1.3589×10^{-6}	1.00	1.6346×10^{-10}	-0.45
10^{-7}	1.3551×10^{-7}	1.00	1.3539×10^{-7}	1.00	5.8587×10^{-11}	0.45
10^{-8}	5.1012×10^{-8}	0.42	3.7806×10^{-8}	0.55	6.6028×10^{-9}	-2.05

Table 2: Absolute Error and Observed Convergence Order for $f(x) = ex$ at $x = 1$

h	Forward Error	Order	Backward Error	Order	Central Error	Order
10^{-1}	3.6504×10^{-2}	—	3.4148×10^{-2}	—	1.1779×10^{-3}	—
10^{-2}	3.5473×10^{-3}	1.01	3.5237×10^{-3}	0.99	1.1785×10^{-5}	2.00
10^{-3}	3.5367×10^{-4}	1.00	3.5344×10^{-4}	1.00	1.1785×10^{-7}	2.00
10^{-4}	3.5357×10^{-5}	1.00	3.5354×10^{-5}	1.00	1.1786×10^{-9}	2.00
10^{-5}	3.5355×10^{-6}	1.00	3.5355×10^{-6}	1.00	1.3964×10^{-11}	1.93
10^{-6}	3.5344×10^{-7}	1.00	3.5355×10^{-7}	1.00	5.2650×10^{-11}	-0.58
10^{-7}	3.4697×10^{-8}	1.01	3.4136×10^{-8}	1.02	2.8042×10^{-10}	-0.73
10^{-8}	3.0503×10^{-9}	1.06	3.0503×10^{-9}	1.05	3.0503×10^{-9}	-1.04

Table 3: at $x = n$ Absolute Error / 4 and Observed Convergence Order for $f(x) = \sin(x)$

Function	Scheme	h^*	Minimum Error
e^x at $x = 1$	Forward	4.221×10^{-9}	2.697×10^{-9}
e^x at $x = 1$	Backward	4.221×10^{-9}	2.697×10^{-9}
e^x at $x = 1$	Central	4.811×10^{-7}	6.533×10^{-13}
$\sin(x)$ at $x = \pi/4$	Forward	8.442×10^{-9}	4.922×10^{-10}
$\sin(x)$ at $x = \pi/4$	Backward	3.732×10^{-10}	4.248×10^{-10}
$\sin(x)$ at $x = \pi/4$	Central	1.212×10^{-6}	1.471×10^{-13}

Table 4: Step Size and Minimum Attainable Error at the Turning Point

h	e^x error ($x = 1$)	$\sin(x)$ error ($x = \pi/4$)
10^{-1}	2.2660×10^{-3}	5.8906×10^{-4}
10^{-2}	2.2652×10^{-5}	5.8925×10^{-6}
10^{-3}	2.2566×10^{-7}	5.9011×10^{-8}
10^{-4}	5.1012×10^{-8}	8.0520×10^{-9}
10^{-5}	2.8932×10^{-6}	1.2959×10^{-6}
10^{-6}	8.9995×10^{-4}	1.1676×10^{-4}
10^{-7}	9.3376×10^{-3}	1.8769×10^{-2}

Table 5: Absolute Error for the Central Second-Difference Approximation

	h^*	Minimum Error
e^x , first derivative	4.811×10^{-7}	6.533×10^{-13}
e^x , second derivative	2.462×10^{-4}	5.103×10^{-9}
$\sin(x)$, first derivative	1.212×10^{-6}	1.471×10^{-13}
$\sin(x)$, second derivative	1.097×10^{-4}	3.027×10^{-9}

Table 6: Comparison of First- and Second-Derivative Turning Points (Central Scheme)

Function	x	h^*	Minimum Error
e^x	0	1.361×10^{-6}	9.865×10^{-13}
e^x	1	4.811×10^{-7}	6.533×10^{-13}
e^x	2	5.442×10^{-6}	8.429×10^{-12}
$\sin(x)$	0	2.127×10^{-8}	0.000
$\sin(x)$	$\pi/4$	1.212×10^{-6}	1.471×10^{-13}
$\sin(x)$	1	6.109×10^{-6}	1.380×10^{-13}

Table 7: Turning Point of the Central Scheme at Multiple Evaluation Points

Conclusion

In this research, a complete study of the forward, backward, and central finite difference schemes has been presented using the transcendental functions $f(x) = e^x$ and $f(x) = \sin(x)$. Systematic testing in MATLAB led to several important conclusions about numerical accuracy and stability.

In this work, the empirical validation of the theoretical convergence rates of forward, backward, and central finite difference schemes was conducted for two transcendental benchmark functions, e^x and $\sin(x)$. The forward and backward schemes were shown to be first-order convergent, $O(h)$, and the central scheme second-order convergent, $O(h^2)$. The observed convergence orders were in good agreement with theory until round-off error began to dominate. Further analysis of the second derivative revealed that the round-off sensitivity increases significantly for higher-order derivatives. The turning point for the central scheme was found to be at a step size about 500 times larger than that for the first derivative. To confirm that these results were not tied to a single point on either function, a robustness check was performed over multiple evaluation points.

Of the three schemes, the central difference method still gives the best tradeoff between computational effort and accuracy. Its error is three to four orders of magnitude smaller than that of either one-sided scheme, at a step size about two orders of magnitude larger, and therefore it is both more accurate and less sensitive to the exact choice of h , provided function values are available on both sides of the evaluation point. This advantage is less pronounced for higher-order derivatives: The results for the second derivative show that the best accuracy attainable even with the

central scheme decreases by several orders of magnitude compared to the first-derivative case. Thus, the practical benefit of finite differences decreases with the order of the derivative, which is a limitation to consider in applications needing higher derivatives.

This work is still limited to two smooth and well-behaved transcendental functions; behavior may differ for functions with sharp gradients, discontinuities, or noisy data, which are common in real experimental settings. Future work could try to extend the higher-order stencils discussed in the introduction, such as fourth- or sixth-order schemes, to see whether they meaningfully shift the round-off turning point seen here. The framework can be extended to partial differential equations, like the heat or wave equation, where the step-size choice influences numerical stability and accuracy. Finally, an empirical comparison with complex-step differentiation or automatic differentiation, both discussed in this paper but not tested directly, would clarify how much practical advantage those alternatives offer over the finite difference schemes examined here.

CRedit Authorship Contribution Statement

Yusupha Bah: Writing – original draft, Validation, Methodology, Investigation, Formal analysis.

Bai Mbye Cham: Supervision, Resources, Conceptualization.

Dawda Charreh: Writing – review & editing, Resources, Investigation, Conceptualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Data associated with the study has not been deposited into a publicly available repository. Data will be made available on request.

Acknowledgments

The authors acknowledge the efforts of the research team from the Mathematics Unit, University of The Gambia (UTG).

References

1. Burden, R. L., & Faires, J. D. (2011). Numerical differentiation & integration numerical differentiation I. Numerical analysis, 174-184.
2. Jacquemin, T., Tomar, S., Agathos, K., Mohseni-Mofidi, S., & Bordas, S. P. (2020). Taylor-Series Expansion Based Numerical Methods: A Primer, Performance Benchmarking and New Approaches for Problems with Non-smooth Solutions: T. Jacquemin et al. Archives of Computational Methods in Engineering, 27(5), 1465-1513.
3. Goldstine, H. H. (2012). A History of Numerical Analysis from the 16th through the 19th Century. Springer Science & Business Media.
4. Othmane, A., Kiltz, L., & Rudolph, J. (2022). Survey on algebraic numerical differentiation: historical developments, parametrization, examples, and applications. International Journal of Systems Science, 53(9), 1848-1887.
5. Thomas, J. W. (2013). Numerical partial differential equations: finite difference methods. Springer Science & Business Media.
6. Iott, J., Haftka, R. T., & Adelman, H. M. (1985). Selecting step sizes in sensitivity analysis by finite differences (No. NASA-TM-86382).
7. Vignes, J., & Alt, R. (2005, May). An efficient stochastic method for round-off error analysis. In Accurate Scientific Computations: Symposium, Bad Neuenahr, FRG, March 12–14, 1985 Proceedings (pp. 183-205). Berlin, Heidelberg: Springer Berlin Heidelberg.
8. Volkov, E. A. (1990). Numerical methods. Taylor & Francis.
9. Griffiths, D. V., & Smith, I. M. (2006). Numerical methods for engineers. CRC press.
10. Cont, R., & Voltchkova, E. (2005). A finite difference scheme for option pricing in jump diffusion and exponential Lévy models. SIAM Journal on Numerical Analysis, 43(4), 1596-1626.
11. Jain, A. K. (1977). Partial differential equations and finite-difference methods in image processing, part 1: Image representation. Journal of optimization Theory and Applications, 23(1), 65-91.
12. Isaacson, E., & Keller, H. B. (2012). Analysis of numerical methods. Courier Corporation.
13. Chapra, S. C., & Canale, R. P. (2011). *Numerical methods for engineers* (Vol. 1221). New York: Mcgraw-hill.
14. Dahlquist, G., & Björck, Å. (2003). Numerical methods. Courier Corporation.
15. Acton, F. S. (1990). Numerical methods that work. Maa.
16. Morinishi, Y., Lund, T. S., Vasilyev, O. V., & Moin, P. (1998). Fully conservative higher order finite difference schemes for incompressible flow. Journal of computational physics, 143(1), 90-124.
17. Gamet, L., Ducros, F., Nicoud, F., & Poinso, T. (1999). Compact finite difference schemes on non-uniform meshes. Application to direct numerical simulations of compressible flows. International Journal for Numerical Methods in Fluids, 29(2), 159-191.
18. Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning:

- a survey. *Journal of machine learning research*, 18(153), 1-43.
19. Verma, A. (2000). An introduction to automatic differentiation. *Current Science*, 804-807.
 20. Martins, J. R., Sturdza, P., & Alonso, J. J. (2003). The complex-step derivative approximation. *ACM Transactions on Mathematical Software (TOMS)*, 29(3), 245-262.
 21. Ketkar, N., & Moolayil, J. (2021). Automatic differentiation in deep learning. In *Deep Learning with python: learn best practices of deep learning models with PyTorch* (pp. 133-145). Berkeley, CA: Apress.
 22. Khan, I. R., & Ohba, R. (1999). Closed-form expressions for the finite difference approximations of first and higher derivatives based on Taylor series. *Journal of Computational and Applied Mathematics*, 107(2), 179-193.
 23. Wong, D. D., Fuglsang, S. A., Hjortkjær, J., Ceolini, E., Slaney, M., & De Cheveigne, A. (2018). A comparison of regularization methods in forward and backward models for auditory attention decoding. *Frontiers in neuroscience*, 12, 531.
 24. Zuazua, E. (2005). Propagation, observation, and control of waves approximated by finite difference methods. *SIAM review*, 47(2), 197-243.
 25. Mathews, J. H., & Fink, K. D. (2004). *Numerical methods using MATLAB* (Vol. 4). Upper Saddle River: Pearson prentice hall.
 26. Goldberg, D. (1991). What every computer scientist should know about floating-point arithmetic. *ACM computing surveys (CSUR)*, 23(1), 5-48.
 27. Muller, J. M., Brunie, N., De Dinechin, F., Jeannerod, C. P., Joldes, M., Lefèvre, V., ... & Torres, S. (2018). *Handbook of floating-point arithmetic* (Vol. 1). Basel, Switzerland:: Birkhäuser.