Journal of Education for Pure Science
Vol.15, No. 4 (2025)
DOI: https://doi.org//10.32792/jeps.v15i4.730

Website: jceps.utq.edu.iq Email: jceps@eps.utq.edu.iq

Reducing Rounding Errors and Increasing Numerical Accuracy with High-Order Taylor Series Techniques

Hasanain Beden Jabbar [1*]

[1]Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Kashan, Kashan, Iran

[*] Corresponding email: hasanainbeden@gmail.com

Received 28 / 5 /2025, Accepted 13/ 7 /2025, Published 1 / 12 /2025

Abstract

This research presents an in-depth exploration of minimizing rounding errors and improving numerical precision in the numerical solution of ODEs using high-order Taylor Series Methods (TSM). Rounding errors, inherent in the IEEE 754 floating-point standard, are a universal challenge in computational mathematics as they accumulate over time, particularly in simulations of chaotic or stiff systems, leading to significant deviations from true solutions. TSM surmounts this challenge by approximating ODE solutions by high-order polynomials using Taylor series expansions, which drastically reduce truncation errors. TSM employs a set of advanced techniques, including high-precision arithmetic, automatic differentiation, compensated summation, error-free transformations, and interval arithmetic, to prevent rounding errors. These combined techniques ensure numerical stability and accuracy even in the most computationally demanding situations. The research rigorously tests the theoretical underpinnings of TSM, including full error bounds, convergence behavior, and stability analyses. Practical utility is shown through implementations like the TIDES software, which shows TSM's ability to maintain accuracy over billions of integration steps. Comparisons with traditional numerical methods, such as Runge-Kutta, multistep, and symplectic integrators, showcase TSM's better accuracy and computational efficiency for applications ranging from celestial mechanics to molecular dynamics. Further case studies, such as simulations of N-body problems and stiff biochemical systems, serve to illustrate the flexibility of TSM. Challenges, including computational intensity, memory demands, and scalability, are addressed in detail with new solutions being proposed, e.g., parallel computing frameworks, hybrid integration methods, and machine learning-based adaptivity. This expanded study provides a unifying framework for the advancement of computational mathematics and applications, positioning TSM as a basis for high-accuracy numerical analysis.

Keywords: Taylor Series Methods, Rounding Errors, High-Precision Arithmetic, Numerical Integration, Automatic Differentiation

1- Introduction

Numerical solving of ordinary differential equations (ODEs) is a fundamental computational mathematics tool, which allows sophisticated physical phenomena to be solved in engineering, physics, biology, chemistry, and economics. However, the finite-precision arithmetic conventions established by the IEEE 754 standard introduce rounding errors that can seriously degrade solution quality [1]. The errors are introduced as a result of computers' inability to represent real numbers, manifesting themselves as minute inaccuracies in every mathematical operation. These discrepancies can amplify exponentially in extended time simulations or extremely sensitive systems, e.g., chaotic or stiff ODEs, and make numerical solutions unstable [2]. For example, for chaotic systems like the Lorenz equations, rounding error of 10-16 can lead to order-of-magnitude differences within a few hundred steps.

High-Order Taylor Series Methods (TSM) provide a potent and new approach to these challenges through the combination of high-order polynomial approximations and sophisticated error control techniques. Compared to traditional methods like Runge-Kutta or multistep methods, TSM makes use of Taylor series expansions of approximated solutions to orders of any desired level in order to eliminate truncation errors altogether. Also, TSM uses advanced computational methods—high-precision arithmetic, automatic differentiation, compensated summation, error-free transformations, and interval arithmetic- to precisely control rounding errors, providing unparalleled numerical precision [3]. These capabilities make TSM especially suitable for applications that require extreme precision, such as long-term orbital simulations in celestial mechanics, sensitivity analysis in dynamical systems, and high-fidelity scientific computing calculations like climate modeling. Recent studies have demonstrated the efficacy of advanced numerical techniques, such as the Elzaki Transform-Adomian Polynomial Method for nonlinear coupled ODEs [4] and the application of Taylor series methods to Lane-Emden-Fowler equations [5], highlighting the versatility of high-precision approaches in tackling complex differential equations.

This full paper gives a comprehensive analysis of TSM's mathematical foundations, error control approaches, and industrial applications; It includes robust mathematical derivations, numerical results, and detailed comparisons to other numerical algorithms. Advanced error control techniques, industrial case studies, and novel computational paradigms are the new subsections. The paper also addresses implementation considerations, i.e., computational cost and memory, and offers future directions, i.e., parallelization, hybrid methods, and integration with machine learning. By being the first to report a comprehensive study of TSM's capability, this research will make it a pillar of high-precision numerical analysis and inspire further work in computational mathematics.

2- Mathematical Foundations of High-Order Taylor Series Methods

2-1- rem, which approximates the smooth function f(t) around a point tn as a polynomial series. For a function with sufficient derivatives, the expansion is: $(t_n + h) = f(t_n) + hf'(t_n) + \frac{h^2}{2} f''(t_n) + \cdots + \frac{h^k}{k!} f^{(k)}(t_n) + R_k(t)$ (1)

Where the remainder term, representing the truncation error, is: $R_k(t) = \frac{h^{k+1}}{(k+1)!} f^{(k+1)}(\xi), \xi \in [t_n, t_n + h]$ (2)

For an ODE of the form:

$$\frac{dy}{dt} = f(t, y), y(t_n) = y_n \quad (3)$$

TSM approximates the solution at t_{n+h} as:

$$y(t_n + h) \approx y_n + hf(t_n, y_n) + \frac{h^2}{2}\frac{d}{dt} f(t_n, y_n) + \cdots + \frac{h^k}{k!}\frac{d^{k-1}}{dt^{k-1}} f(t_n, y_n) \quad (4)$$

Higher-order derivatives are computed recursively. For example, the second derivative is:

$$\frac{d}{dt} f(t, y) = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} f(t, y) \quad (5)$$

Higher derivatives are constructed by the chain rule, so that TSM can construct polynomials of arbitrary order k [6]. The truncation error $R_{k(t)}$ decreases rapidly with increasing k, and one can obtain a high degree of accuracy using small step sizes h. For instance, doubling k reduces the truncation error by orders of magnitude, so TSM can be very useful for problems requiring very high precision.

2-2- Automatic Differentiation

Manual evaluation of derivatives of higher orders is not feasible in complex systems due to its computational complexity and sensitivity to errors. Automatic differentiation (AD) provides an accurate, machine-precision method of evaluating derivatives by constructing a computational graph of the function *f (t, y)* and using the chain rule in a systematic manner [7]. In TSM, AD generates the Taylor coefficients $\frac{1}{k!} \cdot \frac{d^{k-1}}{dt^{k-1}} f(t_n, y_n)$, enabling efficient high-order approximations. Unlike finite differencing, which introduces numerical errors, or symbolic differentiation, which is computationally expensive, AD achieves an optimal balance for nonlinear ODEs [8]. For example, consider the system $f(t, y) = y^2 - t$. The second derivative is computed as:

$$\frac{d^2}{dt^2} f = 2y \cdot \frac{dy}{dt} = 2y \cdot (y^2 - t) \quad (6)$$

AD builds up higher-order terms sequentially with precision and scalability. Additional efficiency comes for high-dimensional systems by employing higher-level AD techniques, operator overloading, and reverse-mode differentiation, bringing TSM within the scope of very large-scale applications such as fluid dynamics or biochemical networks.

3- Rounding Errors in Floating-Point Arithmetic

3-1- IEEE 754 Standard

The IEEE 754 standard governs floating-point arithmetic, and the 64-bit double-precision format can provide 15–17 decimal digits of accuracy [9]. The floating-point number is represented as:

$$(-1)^s \cdot m \cdot 2^{e-\text{bias}} \quad (7)$$

Where s is the sign bit, m is the normalized mantissa ($1 \le m < 2$), and e is the exponent. The unit roundoff, representing the relative error of a single operation, is:

$$u \approx 2.22 * 10^{(-16)} \quad (8)$$

For an operation $x \circ y$, the computed result $\tilde{z}$ satisfies:

$$\tilde{z} = (x \circ y) * (1 + \delta), |\delta| \le u \quad (9)$$

In numerical integration over $N \approx T/h$ steps, where T is the simulation time and h is the step size, rounding errors accumulate, potentially leading to deviations proportional to N * u [10]. For long-term simulations (e.g., $T = 10^6$, $h = 10^{(-3)}$), $N \approx 10^9$ amplifies errors significantly.

3-2- Sources of Rounding Errors in TSM

TSM encounters rounding errors at multiple stages:

1. **Computation of Derivatives:** Recursive calculation of high-order derivatives involves numerous floating-point operations, each contributing an error of the order of u**.**
2. **Summation of Polynomials:** Addition of Taylor series terms, ranging from large (low-order) to small (high-order), produces cancellation errors, losing significant digits.
3. **Step Size Effects:** Large h values contribute to truncation errors, adding to rounding errors and contributing to overall error accumulation**.**
4. **Coefficient Storage:** Storage of high-order Taylor coefficients in finite precision introduces additional errors, especially for large k.

5. **Conditioning**: Poorly conditioned systems, whose input perturbations of small magnitude yield significant output variations, amplify the rounding errors in TSM's recursive computations.

These errors contaminate the high-order accuracy of TSM and necessitate advanced error control techniques [11]. For example, in chaotic systems, a single rounding error may alter qualitative behavior, e.g., a transition from a periodic to a chaotic trajectory.

4- Rounding Error Reduction Strategies

TSM employs a comprehensive arsenal of techniques to minimize rounding errors, ensuring robust numerical precision across diverse applications

4-1- High-Precision Arithmetic

High-precision arithmetic goes beyond double precision by at least twice as many significant digits using packages like MPFR or GMP. Increasing precision to, e.g., 100 digits reduces unit roundoff to**:**

$$u \approx 10^{(-100)} \quad (10)$$

This greatly reduces the accumulation of errors, particularly in unstable models where small errors generate enormous divergences [12]. For instance, high-precision TSM maintains accuracy in Lorenz equation models to billions of steps, while double-precision methods break down at thousands [13]. The trade-off is the increased computational cost, which can be compensated for by optimizing algorithms and hardware acceleration. Recent advances in arbitrary-precision libraries, such as MPFR's support for fast polynomial evaluation, also enhance TSM efficiency [14].

4-2- Compensated Summation

Summing polynomial terms in TSM can result in precision loss due to magnitude differences. Kahan summation addresses this by maintaining a compensation term:

$$s_{(i+1)} = s_i + (a_i + c_i),\ c_{(i+1)} = (a_i + c_i) - (s_{(i+1)} - s_i) \quad (11)$$

Where ci offsets lost precision [15]. This reduces rounding error from O (n u) to O(u) when using n terms, crucial in high-order TSM where terms span a large number of orders of magnitude. Such extensions as Priest's double-double arithmetic achieve this a step further, realizing near-quadruple precision on double-precision hardware [16].

4-3- Adaptive Order and Step Size

TSM dynamically adjusts the polynomial order k and step size h to balance accuracy and efficiency. The remainder term is:

$$R_k(t_n + h) = \frac{h^{k+1}}{(k+1)!}\frac{d^k}{dt^k} f(\xi), \xi \in [t_n, t_n + h] \quad (12)$$

Through approximation of the kth derivative, TSM selects k and h to keep Rk within an acceptable tolerance and minimize truncation error [17]. Adaptive methods can increase k or decrease h in stiff systems for stability purposes, while non-stiff systems allow for larger h in economic considerations. Advanced adaptive algorithms, including those through the application of error extrapolation, further improve by predicting optimal k and h values [12].

4-4- Error-Free Transformations

Error-free transformations (EFTs) decompose floating-point operations into exact and error components. For addition a + b:

$$s = a + b,\ e = (a - s) + b \quad (13)$$

Where s is the sum calculated, and e is the exact error [18]. EFTs enhance the precision of polynomial evaluation in TSM without losing a large number of digits to cancellation. This is most relevant to multi-dimensional systems, where summation error is more evident. Later deployments of EFT, such as two-product transforms of multiplication, also enhance high-order term accuracy [19].

4-5- Interval Arithmetic

Interval arithmetic represents numbers as intervals [a, b] containing the true value, ensuring rigorous error bounds. For an operation x ∘ y, the result is:

$$[z_{min}, z_{max}] \quad (14)$$

To enclose the exact solution [20]. Interval arithmetic in TSM quantifies and bounds rounding error, especially in long integrations. Although computationally intensive, it enhances reliability in safety issues, e.g., spacecraft trajectory optimization [21]. Optimized interval libraries with low overhead, e.g., INTLAB, enable the scheme to carry over to high-precision TSM [22].

4-6- Advanced Error Control Techniques

Apart from the basic methods, TSM also employs advanced techniques to minimize rounding errors further:

- **Stochastic Rounding:** Randomizing rounding directions to avoid bias in error accumulation, with particular usefulness for iterative algorithms [23].
- **Symbolic-Numeric Hybrid:** Combining symbolic algebra used for calculating derivatives with numeric computation for reducing rounding errors in coefficient generation [24].
- **Dynamic Precision Scaling:** Dynamically scaling precision based on estimated error, trading off accuracy and computational cost [25].

These methods, employed regularly in codes such as TIDES, make TSM more robust in ultra-high-precision applications, such as integrating exoplanetary orbits over millennia.

5- Improving Numerical Precision

5-1- Theoretical Framework

Numerical precision in TSM is enhanced by minimizing both truncation and rounding errors. The local truncation error for a k_{th}-order TSM is:

$$E_{\text{local}} = O(h^{k+1}) \quad (15)$$

Over N ≈ T/h steps, the global truncation error is:

$$E_{\text{trunc}} \approx N \cdot O(h^{k+1}) = O(Th^k) \quad (16)$$

Rounding errors accumulate as:

$$E_{round} \approx N\,u \approx (T/h)\ \text{u} \quad (17)$$

High-precision arithmetic reduces u, and adaptive order selection optimizes k, ensuring both errors are minimized [26]. TSM's high-order nature allows arbitrarily small truncation errors by increasing k, while compensated summation and EFTs keep rounding errors negligible. Theoretical analyses, such as backward error analysis, further quantify TSM's precision by modeling it as an exact solution to a perturbed ODE [27].

5-2- Practical Implementation

TIDES software demonstrates the accuracy abilities of TSM, using high-accuracy arithmetic and AD to provide errors orders of magnitude smaller than for conventional methods [3]. In celestial mechanics, TIDES sums up the three-body problem over billions of steps with orbital stability preserved, where Runge-Kutta breaks down [28]. TIDES uses variable-order integration with dynamic k adaptation depending on local estimates of error. The recent developments, namely, GPU acceleration of AD and sparse matrix optimizations, extend TIDES to high-dimensional systems, e.g., galaxy formation models [29].

5-3- Error Propagation Analysis

Consider error propagation in a scalar ODE, where the total error at step n, $e_n = y_n - y(t_n)$, evolves as:

$$e_{(n+1)} = e_n + h * [\, f(t_n, y_n) - f(t_n, y(t_n)) \,] + E_{local} + E_{round} \quad (18)$$

with $E_{round} \approx u * |y_{(n+1)}|$. Using a Lipschitz condition:

$$|f(t, y) - f(t, z)| \leq L * |y - z| \quad (19)$$

the error bound is:

$$|e_{(n+1)}| \leq (1 + h L) * |e_n| + |E_{local}| + |E_{round}| \quad (20)$$

Over N steps, the global error is bounded by:

$$|\, e_N \,| \leq e^{LT} \cdot |\, e_0 \,| + \frac{e^{LT}-1}{L} \cdot \left(\frac{Ch^{k+1}}{(k+1)!} + u\right) \quad (21)$$

Where C bounds the (k+1) th derivative [30]. Large k and small u give close bounds. More sophisticated analyses, such as probabilistic error models, make these bounds closer by taking rounding error distributions into account [31].

5-4- Case Study: N-Body Problem

To demonstrate TSM's precision, let's take an N-body simulation of a planetary system. ODEs underlying such a simulation are very sensitive to initial conditions, so rounding errors are crucial. Using k = 50 and 100-digit precision, TIDES maintains energy conservation to $10^{(-80)}$ for 106 years, as compared to $10^{(-5)}$ for RK4 in double precision [32]. Adaptive order selection changes k dynamically to handle close encounters, demonstrating TSM's robustness in chaotic systems.

6- Comparison with Other Numerical Methods

6-1- Runge-Kutta Methods

Runge-Kutta (RK) methods like RK4 are simple and stable but of fixed order (typically 4 or 5). They require repeated function calls per step and introduce more rounding errors. TSM, with adaptive k of 100, achieves orders of magnitude lower error and is up to 15.8 times faster for high-accuracy use [33]. In the simulation of a double pendulum, TSM maintains energy to 105 periods, while RK4 diverges at 103 [34].

6-2- Multistep Methods

Multistep methods like Adams-Bashforth reduce function calls but are unstable for stiff systems. TSM's high-order adaptability and high-precision arithmetic yield greater accuracy and stability. For stiff chemical kinetics problems, TSM converges to errors of 10-20 while multistep methods diverge because of instability [35].

6-3- Symplectic Integrators

Symplectic integrators preserve the geometric structure of Hamiltonian systems but are lower than TSM orders. TSM's higher-order solutions yield higher quantitative precision for planetary orbit determination, with 10^{-50} errors as opposed to 10^{-10} for symplectic algorithms [36].

6-4- Emerging Methods

More recent techniques, like exponential integrators and spectral deferred correction, are very accurate but do not provide TSM's flexibility in the choice of order. TSM's capacity to scale k arbitrarily and the use of high-precision arithmetic make it especially well-suited to extreme-precision applications [37].

7- Applications in Mathematical Research

TSM's precision has transformative implications across mathematical research:

- **Dynamical Systems**: TSM computes Lyapunov exponents in chaotic systems like the Hénon-Heiles model, identifying chaotic regions with errors below 10^{-30} [38].
- **Celestial Mechanics**: Long-term simulations of exoplanet orbits over 10^9 years maintain stability with TSM, critical for habitability studies [39].
- **Numerical Analysis**: TSM benchmarks new integrators, providing near-exact solutions for computational fluid dynamics and electromagnetics [40].
- **Scientific Computing**: In molecular dynamics, TSM ensures accurate energy conservation, improving simulation reliability for protein folding [41].
- **Control Theory**: TSM optimizes spacecraft trajectories with sub-micron precision, essential for deep-space missions [42].

7-1- Case Study: Biochemical Networks

In stiff biochemical ODEs modeling enzyme kinetics, TSM's adaptive k and high-precision arithmetic achieve errors of 10^{-25}, enabling reliable prediction of reaction rates. Standard methods like implicit RK diverge due to stiffness, highlighting TSM's advantage [42].

8- Challenges and Future Directions

TSM faces several challenges:

- **Computational Intensity**: High-order derivative computations require significant resources, especially for $k > 100$.
- **Memory Requirements**: High-precision arithmetic increases memory usage, challenging for large-scale systems.
- **Error Estimation**: Rigorous rounding error bounds for high-dimensional systems remain elusive.
- **Scalability**: TSM's sequential nature limits its application to distributed systems like PDE solvers.

Future directions include:

- **Parallelization**: GPU and distributed computing frameworks to accelerate TSM, enabling real-time simulations.
- **Hybrid Methods**: Combining TSM with low-order methods for non-critical regions, reducing costs while maintaining precision.
- **Robust Error Bounds**: Probabilistic and machine learning-based error models to quantify rounding errors.
- **Machine Learning Integration**: Deep learning to optimize k and h selection, improving efficiency in adaptive TSM.

9- Conclusion

High-order Taylor Series Methods constitute a breakthrough in numerical integration, offering unprecedented accuracy, clever error control, and high-order approximations. Through the union of high-precision arithmetic, compensated summation, error-free transformations, interval arithmetic, and adaptivity, TSM delivers hard-won accuracy in applications from celestial mechanics to biochemical models. Deepening studies, case studies, and contrasts with traditional techniques emphasise TSM's preeminence, while new methods such as stochastic rounding and machine learning integration expand its possibilities. With advanced computational technology, the strength of TSM in mathematical work will grow, fueled by discoveries that address its shortcomings and enhance its application, establishing it as a cornerstone of high-precision numerical computation.

References

[1] **D. Goldberg**, "What every computer scientist should know about floating-point arithmetic," *ACM Computing Surveys (CSUR),* vol. 23, no. 1, pp. 5-48, 1991, doi: https://doi.org/10.1145/103162.103163.

[2] **N. J. Higham**, *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[3] **R. Barrio,** "Performance of the Taylor series method for ODEs/DAEs," *Applied Mathematics and Computation,* vol. 163, no. 2, pp. 525-545, 2005, doi: https://doi.org/10.1016/j.amc.2004.02.015.

[4] OLUFEMI ELIJAH IGE1, RAZAQ ADEKOLA ODERINU, AND TARIG M. ELZAKI, NUMERICAL SIMULATION OF THE NONLINEAR COUPLED JAULENT-MIODEK EQUATION BY ELZAKI TRANSFORM-ADOMIAN POLYNOMIAL METHOD, Advances in Mathematics: Scientific Journal **9** (2020), no.12, 10335–10355, ISSN: 1857-8365 (printed); 1857-8438 (electronic). https://doi.org/10.37418/amsj.9.12.25.

[5]Chamekh, M., Elzaki, T.M**.** & Ahmed, S.A. Solution of Lane Emden–Fowler equations by Taylor series method. *J.Umm Al-Qura Univ. Appll. Sci.* (2023). https://doi.org/10.1007/s43994-023-00100-2. (https://link.springer.com/article/10.1007/s43994-023-00100-2).

[6] J.-M. Guihal, F. Auger, N. Bernard, and E. Schaeffer, "Efficient implementation of continuous-discrete extended Kalman filters for state and parameter estimation of nonlinear dynamic systems," *IEEE Transactions on Industrial Informatics,* vol. 18, no. 5, pp. 3077-3085, 2021, doi: https://doi.org/10.1109/TII.2021.3109095.

[7] A. Griewank and A. Walther, *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

[8 F. B. R. Barrio and M. Lara, no. 8, pp. 1896–1908, 2009., "VSVO formulation of the Taylor series method for nonlinear ODEs," *Appl. Numer. Math,* vol. 59, no. 8, pp. 1896-1908, 2009.

[9] E. Hairer, "SP N rsett, and G. Wanner. Solving ordinary differential equations I, Nonsti problems," ed: Springer-Verlag, 1993.

[10] D. H. Bailey, "High-precision floating-point arithmetic in scientific computation," *Computing in science & engineering,* vol. 7, no. 3, pp. 54-61, 2005, doi: https://doi.org/10.1109/MCSE.2005.52. .

[11] R. Barrio, M. Rodríguez, A. Abad, and F. Blesa, "Breaking the limits: the Taylor series method," *Applied mathematics and computation,* vol. 217, no. 20, pp. 7940-7954, 2011, doi: https://doi.org/10.1016/j.amc.2011.02.080.

[12] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann, "MPFR: A multiple-precision binary floating-point library with correct rounding," *ACM Transactions on Mathematical Software (TOMS),* vol. 33, no. 2, pp. 13-es, 2007, doi: https://doi.org/10.1145/1236463.1236468.

[13] M. Rodríguez, A. A. Medina, R. B. Gil, and F. Blesa, "TIDES: A free software based on the Taylor series method," *Monografías de la Real Academia de Ciencias Exactas, Físicas, Químicas y Naturales de Zaragoza,* no. 35, pp. 83-95, 2011. [Online]. Available: https://dialnet.unirioja.es/servlet/articulo?codigo=5996234.

[14] G. I. Montecinos, "Error extrapolation for adaptive Taylor methods," *J. Comput. Appl. Math,* vol. 340, no. 3, pp. 567-582, 2018.

[15] W. Kahan, "Pracniques: further remarks on reducing truncation errors," *Communications of the ACM,* vol. 8, no. 1, p. 40, 1965, doi: https://doi.org/10.1145/363707.363723.

[16] D. M. Priest, "Algorithms for arbitrary precision floating point arithmetic," *University of California, Berkeley,* 1991. [Online]. Available: http://www.cs.cmu.edu/afs/cs.cmu.edu/project/cmcl/OldFiles/link.iwarp/OldFiles/archive/quake-papers/related/Priest.ps.

[17] A. Baeza, S. Boscarino, P. Mulet, G. Russo, and D. Zorío, "Approximate Taylor methods for ODEs," *Computers & Fluids,* vol. 159, pp. 156-166, 2017, doi: https://doi.org/10.1016/j.compfluid.2017.10.001.

[18] B. Christianson, "Reverse accumulation and accurate rounding error estimates for taylor series coefficient," *Optimization Methods and Software,* vol. 1, no. 1, pp. 81-94, 1992, doi: https://doi.org/10.1080/10556789208805508.

[19] T. Ogita, S. M. Rump, and S. i. Oishi, "Accurate sum and dot product," *SIAM Journal on Scientific Computing,* vol. 26, no. 6, pp. 1955-1988, 2005, doi: https://doi.org/10.1137/030601818.

[20] R. E. Moore, *Interval analysis*. Prentice-Hall, 1966.

[21] S. M. Rump, "Verification methods: Rigorous results using floating-point arithmetic," in *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, 2010, pp. 3-4, doi: https://doi.org/10.1145/1837934.1837937.
[22] S. M. Rump, "INTLAB—interval laboratory," in *Developments in reliable computing*: Springer, 1999, pp. 77-104.
[23] M. Croci, M. B. Giles, and v. H. H. Ronnlund, no. 4, pp. A2583–A2608, 2021., "Stochastic rounding: Algorithms and applications," *SIAM J. Sci. Comput,* vol. 43, no. 4, pp. 2583-2608, 2021.
[24] M. Berz and K. Makino, "Symbolic-numeric integration of ODEs using Taylor series," *J. Comput. Phys,* vol. 227, no. 23, pp. 9876-9893, 2008.
[25] F. Johansson, "Dynamic precision scaling for high-precision computations," *ACM Trans. Math Softw,* vol. 46, no. 3, pp. 1-25, 2020.
[26] G. I. Montecinos and D. S. Balsara, " , vol. 156, pp. 402–420, 2017., "High-order Taylor series methods for ODEs," *Comput. Fluids,* vol. 156, pp. 402-420, 2017.
[27 P. Amodio and F. Mazzia, "A new approach to backward error analysis for ODEs,," *J. Comput. Appl. Math,* vol. 156, no. 1, pp. 183–199, 2003.
[28] J. R. Scott and M. C. Martini, "Rapid Calculation of Spacecraft Trajectories Using Efficient Taylor Series Integration," 2011. [Online]. Available: https://ntrs.nasa.gov/citations/20110002995
[29] **R. Barrio**, "**GPU-accelerated Taylor series integration**," ***Parallel Comput,*** vol. **88**, pp. **102–115**, **2019**.
[30] J.-M. Muller and B. Salvy, "Effective Quadratic Error Bounds for Floating-Point Algorithms Computing the Hypotenuse Function," *arXiv preprint arXiv:2405.03588,* 2024, doi: https://doi.org/10.48550/arXiv.2405.03588.
[31] N. J. Higham, "Probabilistic error analysis for floating-point computations," *IMA J. Numer. Anal,* vol. 41, no. 2, pp. 1213–1239, 2021.
[32] R. B. A. Abad and n. F. Blesa vol. 834, p. 123, 2017., "High-precision simulation of the N-body problem," *Astrophys. J.,* vol. 834, no. 2, p. 123, 2017.
[33] S. Boscarino and G. Russo, "On a class of hybrid Taylor methods for ODEs," *J. Comput. Appl. Math,* vol. 316, pp. 74-86, 2017.
[34] A. Abad, R. Barrio, and F. Blesa, "High-precision simulation of the double pendulum," *Phys. Lett A,* vol. 376, no. 45, pp. 3015–3021, 2012.
[35] J. R. Cash, "“Efficient numerical methods for stiff differential equations," *J. Comput. Appl. Math,* vol. 125, pp. 153–165, 2000.
[36] M. Betancourt, S. Byrne, S. Livingstone, and M. Girolami, "The geometric foundations of hamiltonian monte carlo," *Bernoulli,* pp. 2257-2298, 2017.
[37] A. Dutt, L. Greengard, and V. Rokhlin, "Spectral deferred correction methods for ordinary differential equations," *BIT Numerical Mathematics,* vol. 40, pp. 241-266, 2000.
[38] R. Barrio, "Sensitivity analysis of ODEs/DAEs using the Taylor series method," *SIAM Journal on Scientific Computing,* vol. 27, no. 6, pp. 1929-1947, 2006, doi: https://doi.org/10.1137/030601892.
[39] G. Rein, "Long-term stability of planetary orbits using high-order numerical integrators," *Celest. Mech. Dyn. Astron,* vol. 130, no. 3, p. 23, 2018.
[40] Y. Zhang and A. A. Ding, "High-order numerical methods for ODEs with applications to wave propagation," *J. Sci. Comput,* vol. 76, no. 2, pp. 1123–1145, 2018.
[41] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2023.
[42] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.