scientific reports

OPEN

Symbolic iteration method based on computer algebra analysis for Kepler's equation

Ruichen Zhang✉, Shaofeng Bian & Houpu Li

The Kepler's equation of elliptic orbits is one of the most significant fundamental physical equations in Satellite Geodesy. This paper demonstrates symbolic iteration method based on computer algebra analysis (SICAA) to solve the Kepler's equation. The paper presents general symbolic formulas to compute the eccentric anomaly (*E*) without complex numerical iterative computation at run-time. This approach couples the Taylor series expansion with higher-order trigonometric function reductions during the symbolic iterative progress. Meanwhile, the relationship between our method and the traditional infinite series expansion solution is analyzed in this paper, obtaining a new truncation method of the series expansion solution for the Kepler's equation. We performed substantial tests on a modest laptop computer. Solutions for 1,002,001 pairs of (*e*, *M*) has been conducted. Compared with numerical iterative methods, 99.93% of all absolute errors δ_E of eccentric anomaly (*E*) obtained by our method is lower than machine precision ϵ over the entire interval. The results show that the accuracy is almost one order of magnitude higher than that of those methods (double precision). Besides, the simple codes make our method well-suited for a wide range of algebraic programming languages and computer hardware (GPU and so on).

As one of the fundamental physical equations in Geoscience, the Kepler's equation in elliptical orbits describes a relationship of the eccentric anomaly (E), the eccentricity (e), and the mean anomaly (M)[1], in which M is related to the time. For elliptic orbits, it is shown as Eq. (1), and the contours of the Kepler's equation with different eccentricities $(e = 0, 0.1, 0.5, 0.9, 1)$, are shown in Fig. 1. This paper focuses on the situation of nearly-circular motions $(e \ll 1)$. We simplify the problem by limiting E and M within the range of $[0, \pi]$[2], as shown in Eq. (2). General Solutions for $(E, M) \in \mathbb{R} \times \mathbb{R}$ can be easily calculated through $(\pm E + 2n\pi\,,\ \pm M + 2n\pi\,), n \in \mathbb{Z}$.

$$E = M + e \sin E \tag{1}$$

$$E, M \in [0, \pi\,], \quad e \ll 1 \tag{2}$$

Brouwer and Clemence[3], and Danby[4] have described the principles and applications of this equation for artificial Earth satellites and other celestial bodies in great detail. The enormous researches to solve the Kepler's equation underline its significance in many fields of Satellite Geodesy (e.g. precise orbit determination, perturbation theory and Earth model calculation), Geophysics and Trajectory Optimization[5–11]. As all the satellite orbits are nearly-circular, this paper focuses on the situation of nearly-circular motions, providing a symbolic iteration method based on computer algebra analysis for Kepler's equation.

Many scientists have made considerable improvements with high accuracy and low time consumption of various algorithms to solve the equation. Nijenhuis and Albert[1] combined the Mikkola's starter algorithm with a higher-order Newton method, requiring only two trigonometric evaluations. Markley and Landis[12] presented a non-iterative method with four transcendental function evaluations. Colwell[13] systematically analyzed the traditional solutions for the Kepler's equation, such as traditional power serious expansion based on Lagrange inversion theorem, and series solutions in Bessel functions of the first kind, which are still the most common methods for Kepler's equation. Taff and Brennan[14] explored an extensive starting values and solution techniques for iterative methods, giving the best and simplest starting value algorithm. Odell[15] provided an iterative process that always gives 4th order convergence, with error less than 7×10^{-15} rad. Mikkola[16] derived a high order iteration formula by adding an auxiliary variable, which could be used both for elliptical and hyperbolic orbits. Sengupta[17]

Department of Navigation, Naval University of Engineering, 717 Jiefang Avenue, Wuhan, China. ✉email: zrcrc1009@sina.com

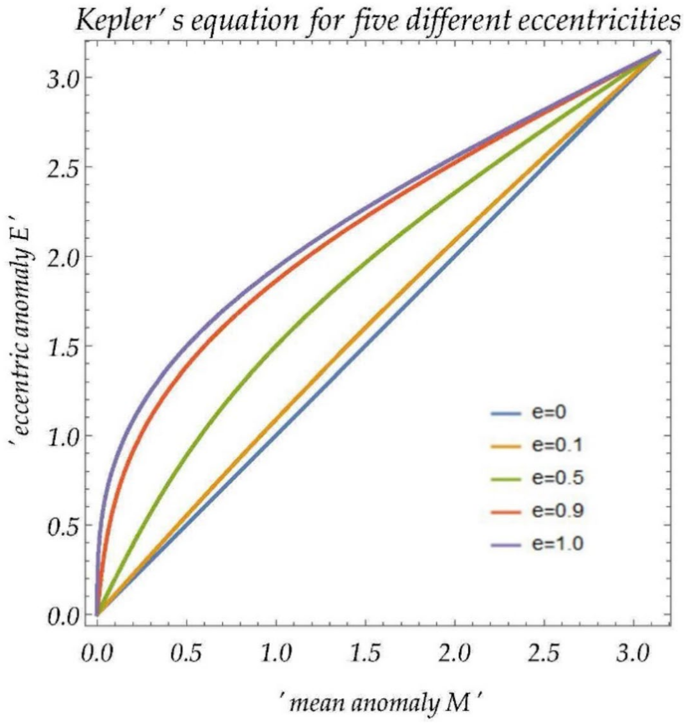

Figure 1. Contours of the Kepler's equation with different eccentricities ($e = 0, 0.1, 0.5, 0.9, 1$).

presented the Lambert W method for the truncation of infinite Fourier–Bessel representations. Calvo and Elipe[18] compared the quality of starting algorithm for the iterative solution of elliptic Kepler's equation, giving new optimal starters. Oltrogge and Daniel[19] derived a solution via families of hybrid and table lookup techniques, achieving the accuracies to machine precision at 3×10^{-11}. Zechmeister[20] presented an algorithm to compute the eccentric anomaly without usage of transcendental functions. Tommasini and Olivieri[21] provided a solution for the inverse of a function based on switching variables followed by spline interpolation.

However, limited by historical scientific conditions, these traditional methods usually contain numerical iterative calculations, such as the standard Newton–Raphson method[2,4] and the Halley's method[2], which have to reset the initial values and iterate from the beginning every time the eccentricity e changes[22]. Meanwhile, some traditional numerical methods [4] contain tremendously complex functions, which costs too much time to compute. In these methods, the CPU time cost by transcendental functions accounts for a large part of the total CPU time consumption[23]. For instance, an iterative method to solve the Kepler's equation usually requires two trig evaluations (sine and cosine) at every iterative step[1]; the CPU time cost increases exponentially with the number of iterations grows. As for the Intel i7 × 920 processor, the calling time of trigonometric functions occupies more than 80% of the total time consumption in measured unmanaged arithmetic[19]. Besides, some traditional formulas are given in the form of concrete numerical values, which lacks generality. Meanwhile, there were also some analytic solutions with a certain tolerance and maximum eccentricity. For example, Markley and Landis[12] provided a non-iterative method with errors better than 10^{-4} requiring four transcendental functions, containing square-root, cube-root and trigonometric operations. Nijenhuis[1] further presented a non-iterative method that represented a gain obtained through a starter algorithm, which required two trigonometric evaluations. Although these methods are classic and efficient at that time, their performances are limited by the historical scientific conditions with a relatively poor accuracy.

The current research is mainly aimed at improving the iterative methods and expanding the scope of the application of algorithms. Meanwhile, since the solution for the Kepler's equation is a traditional problem, the traditional iterative methods are relatively mature. Thus, entering into the new century, there are fewer studies specifically on this problem, letting alone specifically on the circumstances of nearly-circular motions. However, most of the two-body motions in the universe are nearly-circular motions, especially for all satellite motions. The accuracy of the eccentric anomaly (E) is the basis and the determinant for the accuracy of satellite orbit determination. Thus, in order to improve the accuracy of the eccentric anomaly (E) in the field of satellite geodesy, this paper proposes a solution for Kepler's equation specifically for the nearly-circular orbits, which improves the accuracy to the order of 10^{-17}. In this paper, based on the computer algebra system (Mathematica 12.1 and Python 3.8), we present an analytic method with a general symbolic formula to compute the eccentric anomaly (E) without complex numerical iterative computation at run-time for nearly-circular motions. This non-iterative method only requires two trigonometric functions during the whole computation process. Compared with the traditional analytic methods, our analytic method is easy to understand with a higher accuracy and simpler codes; compared with the commonly used traditional iterative methods, our method has higher accuracy and stability;

compared with modern solutions for Kepler's equation, on the basis of retaining the accuracy of the algorithm, our method takes into account the advantages of analytical methods and the simplicity of codes.

©

Basic scheme.

The SICAA method is to symbolically iterate the Kepler's equation in the form of Taylor series expansion without concrete numerical values. Due to the limitation of the computer platform precision, we could control the error $\varepsilon_E = E_{n+1} - E_n$ when the residual $\varepsilon_M = M_{n+1} - M_n$ is within the accuracy. Nonetheless, the cancellation problem is hard to avoid for the Kepler's equation when the limits of platform precision are pushed. As for this problem, we use the Taylor series expansion for every symbolic operational step, as shown in Eq. (5). The starter of E is shown in Eq. (3). If the variable e in Eq. (4) is regarded as an argument of the function $E(e)$, the variable M is regarded as the parameter. We suppose $e \ll 1$, so $E_{n+1}(e)$ is expanded at $e = 0$ by Taylor Series expansion, and then Eq. (4) can be expanded as Eq. (5).

$$E_0 \approx M \quad (e \ll 1) \tag{3}$$

$$E_{n+1}(e, M) \approx M + e \sin E_n(e, M) \tag{4}$$

$$E_{n+1}(e,M) \approx E_{n+1}(0,M) + E^{'}_{n+1}(0,M)e + \frac{E^{''}_{n+1}(0,M)}{2!}e^2 + \ldots + \frac{E^{k}_{n+1}(0,M)}{k!}e^k \tag{5}$$

where $n = 0, 1, 2, \ldots$ $k = 0, 1, 2, \ldots 18$.

The final analytic formula to calculate the eccentric anomaly obtained by our SICAA method is shown in Eq. (6), where series is extended to infinitely small quantity $O[e]^{18}$. The specific coefficients of sin nM are shown in supplementary material S.4.

$$E = f(e, M) = M + a_1 \sin M + a_2 \sin 2M + a_3 \sin 3M + a_4 \sin 4M + a_5 \sin 5M + \cdots + a_i \sin nM \tag{6}$$

It can be found that our analytic solution is really intuitive, and the process of the SICAA method is easy to conduct and simple to understand.

Convergence criteria and convergence domain.

According to the convergence rule, we need to discuss the convergence of Eq. (4). Thus, we rearrange Eq. (4). The function $f_1(E)$ is defined as $f_1(E) = E$. The function $f_2(E)$ is defined as $f_2(E) = M + e \sin E$. When $e < 1$, for $\exists E_a \in [0, \pi]$, $\exists E_b \in [0, \pi]$, so $|\cos E_b| \leq 1$ and eventually we can get the Eq. (7).

$$\frac{\left|f_2^{'}(E_b)\right|}{\left|f_1^{'}(E_a)\right|} = \frac{|e \cos E_b|}{|1|} = |e \cos E_b| < 1 \tag{7}$$

The above iterative equation converges at $E \in [0, \pi]$ and $e < 1$. The convergence rate (also called error disappearance rate) is equivalent to α^n. The specific formula of α is shown in Eq. (8).

$$\alpha \approx \frac{E_2 - E_1}{E_1 - E_0} \tag{8}$$

The relationship between the SICAA method and the Bessel function.

The traditional series expansion of the eccentric anomaly[9,13] is shown as the Eqs. (24) and (25) in "Mathematics essence of our method and relationship with traditional Fourier–bessel series" section. By comparing the coefficients of our method (SICAA) and those of Fourier–Bessel series expansion of the eccentric anomaly, the relationship between SICAA and the Bessel function are shown in Eqs. (9) and (10).

$$E_{\text{SICAA}} \approx M + 2\sum_{k=1}^{k_{max}} \frac{1}{k} J_k^{\lceil k_{max}-k+1 \rceil}(ke) \sin kM \tag{9}$$

$$J_k^{\lceil k_{max}-k+1 \rceil}(ke) = \sum_{j=0}^{\lceil k_{max}-k+1 \rceil} (-1)^j \frac{\left(\frac{ke}{2}\right)^{k+2j}}{j!(k+j)!} \tag{10}$$

where $\lceil * \rceil$ is the Ceiling function. From the Eqs. (9) and (10), we can see that the coefficients of the SICAA method are the truncation ($\lceil k_{max} - k + 1 \rceil$) of the first kind infinite Bessel function.

Variants and refinements

As mentioned above, the code of SICAA method is simple and easy to implement with strong expansibility. We also present some generalized types of our SICAA method.

The Newton-like SICAA method.

The convergence of the Eq. (4) is linear. Based on Newton numerical iterative theory, the convergence of the Eq. (11) is quadratic. Our algorithm can serve derivatives and starters at

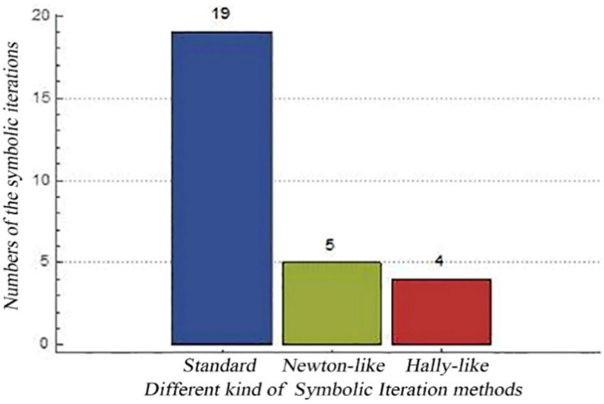

Figure 2. Number of iterations through standard SICAA method, Newton-like SICAA method and Halley-like SICAA method (double precision), until the ith iterative coefficients of e^n $(n = 1, 2, \ldots, 18)$ and the $(i-1)$th iterative coefficients of e^n $(n = 1, 2, \ldots, 18)$ are the same.

any stage in a symbolic way. The quadratic convergence could get the accuracy from, for example, 10^{-6} down to 10^{-8}, costing mainly one division. It would be preferred over the optimization mentioned in "Symbolic iteration method based on computer algebra analysis (SICAA)" section. The starter of E is shown in Eq. (3). We modify Eq. (4) into the symbolic form of the standard Newton–Raphson method, as shown in Eq. (12). The function $E_{n+1}(e, M)$ in Eq. (12) is expanded at $e = 0$ by Taylor Series expansion in every operational step, shown in Eq. (13). The main schemes of the Newton-like SICAA method are shown in Eqs. (12) and (13).

$$E_{n+1} = E_n - \frac{f(E_n)}{f'(E_n)}, \quad n = 0, 1, 2, \ldots \tag{11}$$

$$E_{n+1}(e, M) \approx E_n(e, M) - \frac{E_n(e, M) - M - e\sin E_n(e, M)}{1 - e\cos E_n(e, M)} \tag{12}$$

$$E_{n+1}(e, M) \approx E_{n+1}(0, M) + E_{n+1}^{'}(0, M)e + \frac{E_{n+1}^{''}(0, M)}{2!}e^2 + \cdots + \frac{E_{n+1}^{k}(0, M)}{k!}e^k \tag{13}$$

where $n = 0, 1, 2, \ldots$ $k = 0, 1, 2, \ldots, 18$.

The Halley-like SICAA method.

The Halley's numerical iterative method has a cubic convergence, as shown in Eq. (14), while it is more sensitive to the starters. The starter of E is shown in Eq. (3). We modify Eq. (4) into the symbolic form of the Halley's method, as shown in Eq. (15). The function $E_{n+1}(e, M)$ in Eq. (15) is expanded at $e = 0$ by Taylor Series expansion in every step, shown in Eq. (16). The main scheme of the Halley-like SICAA method is shown in Eqs. (15) and (16).

$$E_{n+1} = E_n - \frac{f}{f'}\left[1 + \frac{ff''}{2f'^2}\right] \quad k = 1, 2, \ldots \tag{14}$$

$$E_{n+1}(e, M) \approx E_n - \frac{E_n - M - e\sin E_n}{1 - e\cos E_n}\left[1 + \frac{(E_n - M - e\sin E_n)(e\sin E_n)}{2(1 - e\cos E_n)^2}\right] \tag{15}$$

$$E_{n+1}(e, M) \approx E_{n+1}(0, M) + E_{n+1}'(0, M)e + \frac{E_{n+1}^{''}(0, M)}{2!}e^2 + \cdots + \frac{E_{n+1}^{k}(0, M)}{k!}e^k \tag{16}$$

where $n = 0, 1, 2, \ldots$ $k = 0, 1, 2, \ldots, 18$.

When the coefficients of $\sin iM$ are the same between two symbolic iterations, which means $E_{n+1} = E_n$, then E_{n+1} is regarded as the final analytic formula calculated by our SICAA methods. Figure 2 shows the number of symbolic iterations through three SICAA methods (double precision). As we can see, the standard SICAA method needs 19 steps to get the precision of e^{18} series expansion, while the Newton-like SICAA method and the Halley-like SICAA method only need 5 steps and 4 steps, respectively. Although the final coefficients of $\sin iM$ calculated by any of these SICAA methods are the same, the Newton-like SICAA method has the advantages of both fewer symbolic operational steps than the standard SICAA method and simpler in symbolic form than the Halley-like one. Thus, we use the Newton-like SICAA method to get the final analytic formula.

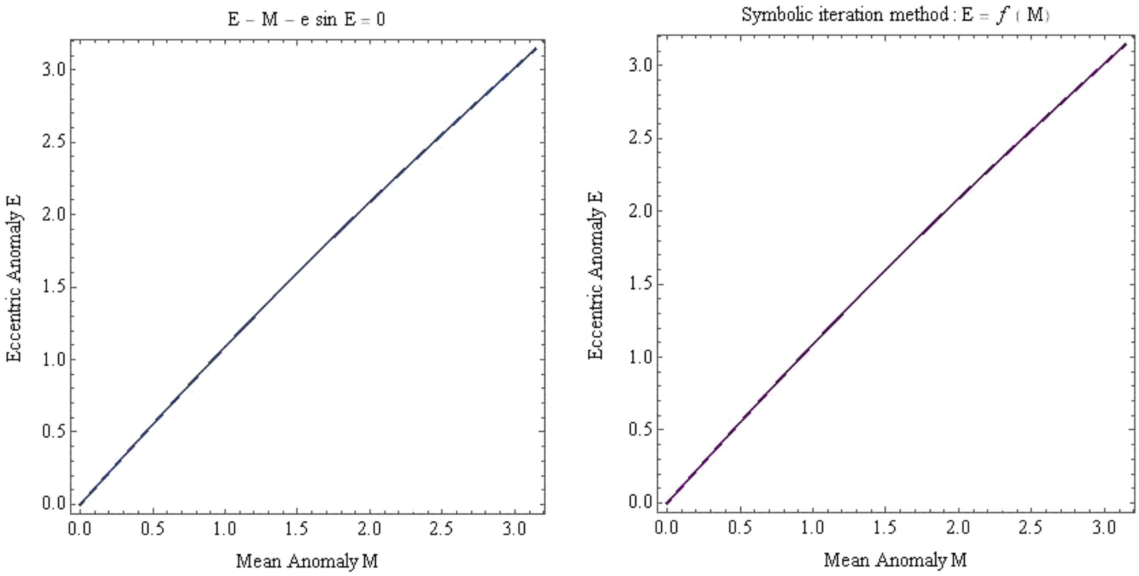

Figure 3. Explicit expression $E = f_2(M)$ of the SICAA method applied to the function $f_1(M, E) = E - M - 0.1 \sin E$ over the domain $M \in [0, \pi]$ (top right), matching nearly-circular orbit of eccentricity 0.1. The implicit function contour (top left) of the Kepler's equation is shown for comparison.

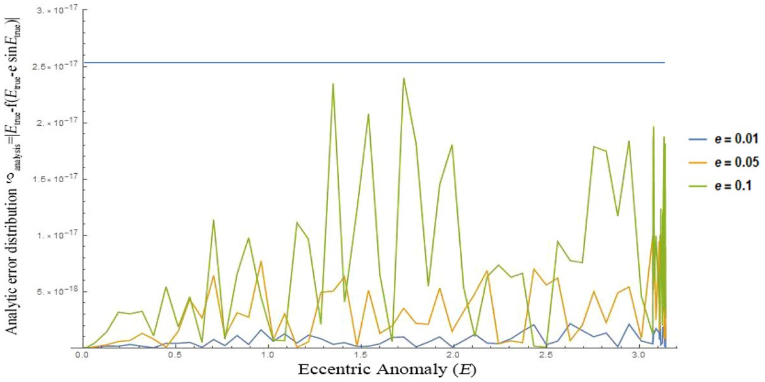

Figure 4. Analytic formula of the error distributions given by $\delta_{analysis} = \left|E_{true} - f(E_{true} - e \sin E_{true})\right|$ over the domain $E \in [0, \pi]$, matching nearly-circular orbits of eccentricity 0.01, 0.05, 0.1.

Analysis and comparison

Accuracy and performance analysis.

Analytic accuracy. After symbolic operation process, we change the implicit expression $f_1(M, E) = 0$ into the explicit expression $E = f_2(M)$, as shown in Eq. (6), thus obtaining the analytic solution of the Kepler's equation when $e \ll 1$ to directly study the relationship between E and M, as shown in Fig. 3.

It is convenient to study the characteristics of the solution by explicit expression Eq. (6). To analyze the error properties, the error expression $\delta_{analysis}$ is defined, as shown in Eq. (17). The parameter M in Eq. (17) is calculated by the Eq. (18). After injecting Eqs. (6) and (18) into Eq. (17), the specific form of error $\delta_{analysis}$ is shown in Eq. (19), and the coefficient of s_i is the same with that of the Eq. (6). Figure 4 shows the error distribution lines of the Eq. (19) over $E_{true} \in [0, \pi]$ for the cases of eccentricity $e = 0.01, 0.05, 0.1$. As we can see in Fig. 4, the theoretical prediction of error $\delta_{analysis}$ for Eq. (6) is no more than 2.5×10^{-17} (shown by the blue line). Obviously, owing to the computer machine precision limits and truncation errors caused in every step, the numerical errors distribution will be slightly different with what are shown in Fig. 4, which will be discussed in the following part.

$$\delta_{analysis} = \left|E_{true} - f_2(e, M)\right| \tag{17}$$

$$M = E_{true} - e \sin E_{true} \tag{18}$$

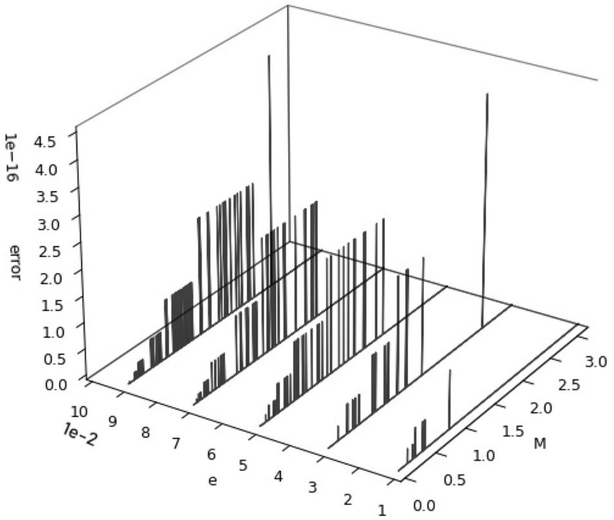

Figure 5. Numerical result of the SICAA method over $M \in [0, \pi]$, matching nearly-circular orbits of eccentricity 0.01, 0.03, 0.05, 0.07, 0.09. The SICAA method is shown for 1001 × 5 grid points within the accuracy for $O(e)^{18}$, together with the evaluation of the numerical error: $\delta_E = \left|E_{symbol}(M) - E\right|$, where the error is computed by the method mentioned above.

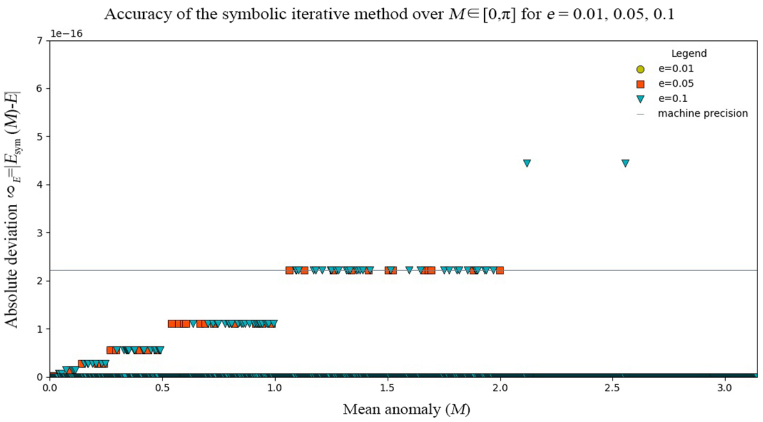

Figure 6. Accuracy of the SICAA method $\delta_E = \left|E_{symbol}(M) - E\right|$ as mentioned before over $M \in [0, \pi]$ for the cases $e = 0.01, 0.05, 0.1$ with machine precision $\epsilon = 2.220446049250313 \times 10^{-16}$ by the grey line.

$$\begin{aligned}\delta_{analysis} = &-e\sin E_{true} + a_1 s_1 + a_2 s_2 + a_3 s_3 + a_4 s_4 + a_5 s_5 + a_6 s_6 + a_7 s_7 + a_8 s_8 + a_9 s_9 + a_{10} s_{10} \\ &+ a_{11} s_{11} + a_{12} s_{12} + a_{13} s_{13} + a_{14} s_{14} + a_{15} s_{15} + a_{16} s_{16} + a_{17} s_{17}\end{aligned} \tag{19}$$

where $s_i = \sin(iE_{true} - ie\sin E_{true})$.

Numerical accuracy. To prove if the SICAA method can achieve double precision practically, we pre-calculated 1001 × 1001pairs of $(E - e\sin E, E)$ uniformly sampled over $(e, E) \in [0, 0.1] \times [0, \pi]$. The eccentric anomaly E is regarded as true values. Then, the $M(E) = E - e\sin E$ and e are the input of our algorithm, outputting the inverse function $E_{symbol}(M)$ as the solution of the Kepler's equation. Differences between the true eccentric anomalies and the re-calculated ones $\delta_E = \left|E_{symbol}(M) - E\right|$ are shown in Fig. 5 for each case $e = 0.01, 0.03, 0.05, 0.07, 0.09$. To better present the changing trends of the error δ_E with the increase of M and e, a waterfall picture of the error δ_E has been drawn in Fig. 5. With the increase of M, the errors increase and then decrease gradually, except for two outliers. With the increase of e, the errors increase slightly. Hence, the change of e has less influences on the precision in the SICAA method, while the accuracy of the SICAA solution is more sensitive to M.

For $M \in [0, \pi]$ and $e \in [0, 0.1]$, most of the deviations δ_E are no more than machine precision $\epsilon = 2.220446049250313 \times 10^{-16}$ as indicated by the grey line in Fig. 6. As shown in Fig. 6, when $e = 0.1$, there are only two points higher than the machine accuracy ϵ; when $e = 0.01$ and 0.05, all the errors are no more than the machine accuracy ϵ; there are many errors even close to 0.

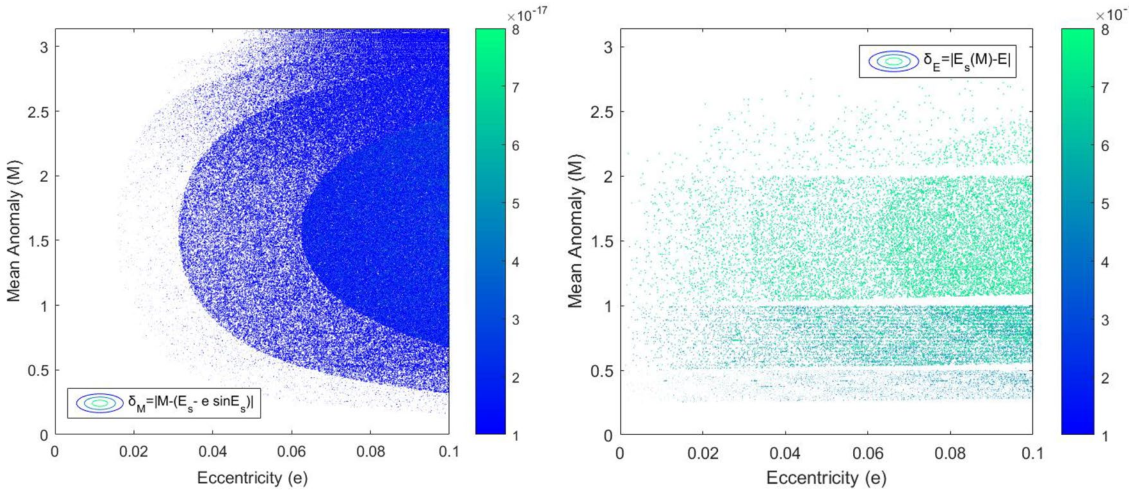

Figure 7. Contour plots of the errors $\delta_M = \left|M - \left(E_{symbol} - e\sin\left(E_{symbol}\right)\right)\right|$ (left plot) and $\delta_E = \left|E_{symbol}(M) - E\right|$ (right plot) over the domain $(M, e) \in [0, \pi] \times [0, 0.1]$.

After $E_{symbol}(M)$ has been solved, we recomputed M_{re} by $M_{re} = E_{symbol} - e\sin\left(E_{symbol}\right)$. Then, the contour plot of errors $\delta_M = |M - M_{re}| = \left|M - \left(E_{symbol} - e\sin\left(E_{symbol}\right)\right)\right|$ (left) (are shown in Fig. 7 over $(M, e) \in [0, \pi] \times [0, 0.1]$. It can be found from Fig. 7 (left) that as M becomes closer to $\frac{\pi}{2}$, the contour lines become more and more dense, which means the error δ_M increases; with the increase of e, the color of contour lines changes from dark blue to light blue, which means the error δ_M increases slightly. That is matched perfectly with Fig. 5. As can be seen from Fig. 7 (right), the error value δ_E is symmetrically distributed with $M \approx \frac{\pi}{2}$; the more inclined it is to $\frac{\pi}{2}$, the greater the error δ_E is. With the increase of e, the error also increases slightly. However, the errors are almost all under 10^{-17}.

Numerical errors (shown in Figs. 5, 6 and 7) almost match the analytic error distribution (shown in Fig. 4), with 99.93% of all numerical errors computed lower than machine precision ϵ, and all $\delta_E = \left|E_{symbol}(M) - E\right| \leq 4.4409 \times 10^{-16}$ over the entire interval.

Discussion of truncation error. The deviations δ_E and δ_M become larger in the corner $\left(e \to 0.1, M \to \frac{\pi}{2}\right)$ of the plane (M, e). That is because the Eq. (6) is expanded by Taylor series expansion at $e = 0$, also called Maclaurin series expansion. The principle of Maclaurin series expansion is that the equation is only available in the neighborhood of the expansion point $e = 0$, which is why when e goes far away from 0, the errors become larger. Figure 8 shows the contour plot of the analytic errors $\delta_{analysis} = \left|E_{true} - f_2(E_{true} - e\sin E_{true})\right|$ (as shown in Eq. (17)) on the plane $(e, E) \in [0, 0.2] \times [0, \pi]$. In Fig. 8, the error $\delta_{analysis}$ increases significantly when $e = 0.2$ (strong elliptic), but it performs perfectly when $e \in [0, 0.1]$, meeting most needs of geoscience applications (most Earth satellites' orbits are nearly-circular at $e = 10^{-2} \sim 10^{-3}$).

To analyze the error distribution, we rearrange Eq. (6) as Eq. (20). The coefficients of $e^i (i = 1, 2, \ldots, 18)$ are expressed as $b_i(M)$. Table 1 represents the extremum value Max(b_i) and the corresponding extremum point Max(M) of the coefficient $b_i(M)$ over the domain $M \in [0, \pi]$. Apparently, the coefficients of the e^n are expressed in the polynomial function forms of $\sin iM$. Consequently, the larger the b_i is, the larger the coefficients of e are, and the greater the errors of e accumulate, resulting in the larger final deviations. Therefore, when $M \to \frac{\pi}{2}$ and $e \to 1$, the error accumulated. The analytic error distribution $\delta_{analysis}$ is shown in Fig. 8. In spite of that, the errors shown in Fig. 8 are almost under 5×10^{-16}.

$$\begin{aligned} E(e, M) = M + b_1 e + b_2 e^2 + b_3 e^3 + b_4 e^4 + b_5 e^5 + b_6 e^6 + b_7 e^7 + b_8 e^8 + b_9 e^9 + b_{10} e^{10} \\ + b_{11} e^{11} + b_{12} e^{12} + b_{13} e^{13} + b_{14} e^{14} + b_{15} e^{15} + b_{16} e^{16} + b_{17} e^{17} + b_{18} e^{18} \end{aligned} \tag{20}$$

The plot of truncation driving errors are manifested in Fig. 9. The dashed-blue line pictures the errors $\delta_{analysis}$ when the series ceases at $k_{max} = 16$, which is lower than the machine precision ϵ. With an additional term ($k_{max} = 17$), the errors $\delta_{analysis}$ are cut down to almost one order of magnitude lower than the ones at $k_{max} = 16$, portrayed by solid line. Then setting the termination at 18 ($k_{max} = 18$), represented by the dashed-green line, the errors $\delta_{analysis}$ are almost identical to the ones at $k_{max} = 17$, much lower than the standard double float accuracy, where we can get the tolerance of 2×10^{-17}. Therefore, we use the Eq. (6) at $k_{max} = 17$ to run the numerical computation in the next section.

Comparisons with traditional numerical iterative methods.

In this section, error comparisons are conducted between our SICAA method and two traditional Newton iterative methods to deal with the Kepler's equation. All methods were executed in the Python programming language, counting on numerical and algebraic libraries in the system. The details of the comparisons among our SICAA method, the standard Newton–Raphson method and the Halley's method are as follows.

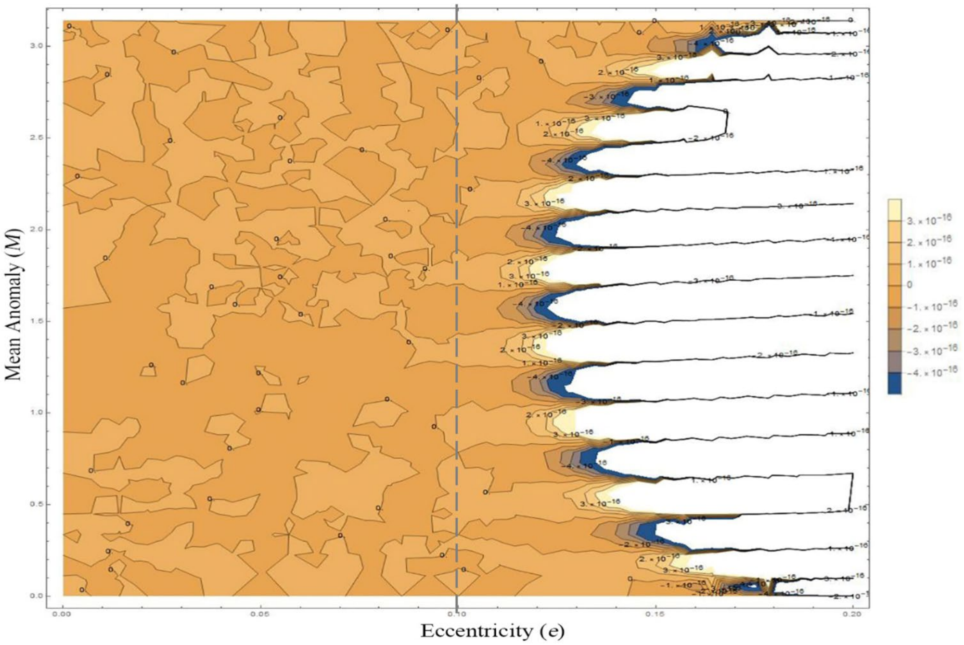

Figure 8. Contour plot of the errors $\delta_{analysis} = \left|E_{true} - f\left(E_{true} - e\sin E_{true}\right)\right|$ on the plane $(e, E) \in [0, 0.2] \times [0, \pi]$. For visibility, the gray line shows the case $e = 0.1$.

b_i	b_1	b_2	B_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}	b_{17}
Max(b_i)	1	0.5	0.5	0.46	0.5	0.59	0.7	0.89	1.18	1.47	2.0	2.56	3.54	4.66	6.51	7.25	12.30
Max(M)	$\frac{\pi}{2}$	0.8	$\frac{\pi}{2}$	1.14	$\frac{\pi}{2}$	1.27	$\frac{\pi}{2}$	1.34	$\frac{\pi}{2}$	1.75	$\frac{\pi}{2}$	1.42	$\frac{\pi}{2}$	1.44	$\frac{\pi}{2}$	1.23	$\frac{\pi}{2}$

Table 1. The Extremum values and the corresponding extremum point of the coefficients.

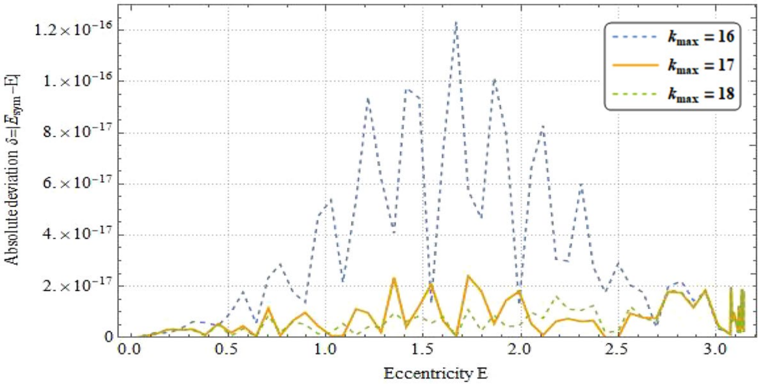

Figure 9. Errors $\delta_{analysis} = \left|E_{true} - f\left(E_{true} - e\sin E_{true}\right)\right|$ for the cases if the series is terminated at $k_{max} = 16$ (dashed-blue), $k_{max} = 17$ (solid), and $k_{max} = 18$ (dashed-green), respectively.

Hardware: Comparisons conducted on a modest laptop computer (Intel i7-8550U x64 CPU @ 1.8GHz 2GHz, RAM 8GB, and with the Windows operating system with 10.0.18362.1139 kernel).

Data preparation: Forward-calculating $M_{i,j} = E_i - e_j \sin E_i$, using $1001 \times 1001 \left(E_i, e_j\right)$ pairs uniformly sampled by the equation $E_i = \frac{i}{1000}\pi$ and $e_j = 0.1 \times \frac{j}{1000}$, $(i, j = 0, 1, \ldots, 1000)$. Then, the $M_{i,j}$ and e_j are the input of our algorithm, outputting the inverse function $\widehat{E_{i,j}}\left(M_{i,j}, e_j\right)$.

Precision: Average eccentric anomaly errors $\bar{\delta}_E\left(e_j\right)$ are computed by the Eq. (21), where $\delta_{Ei,j}\left(M_{i,j}, e_j\right)$ denotes the error obtained in one computation, as shown in Eq. (22). The average CPU time (units: second) is computed by the Eq. (23), where $t_{i,j}\left(M_{i,j}, e_j\right)$ denotes the CPU time cost in one computation. Starters: $E_0 = M_{i,j}$, as mentioned in "Symbolic iteration method based on computer algebra analysis (SICAA)" section.

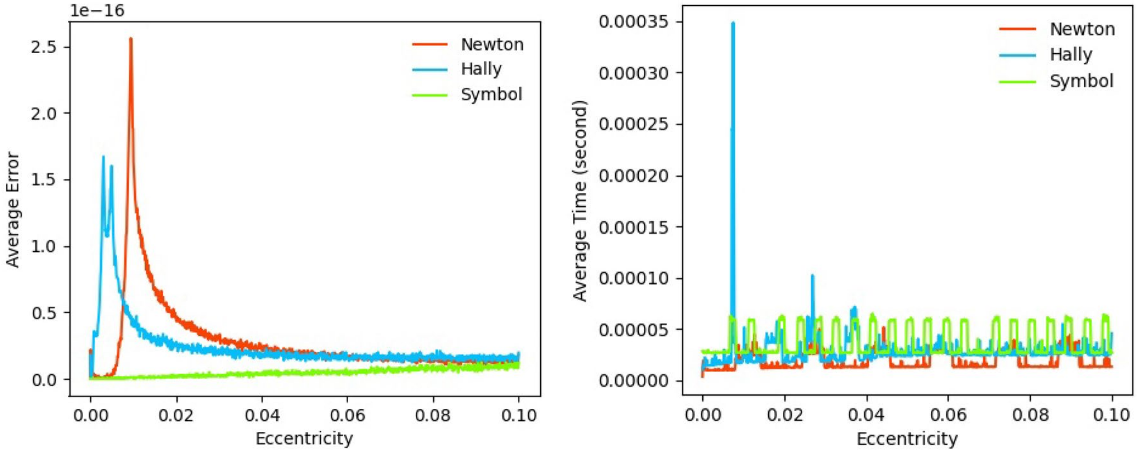

Figure 10. Numerical comparisons among our SICAA method, the standard Newton–Raphson method and the Halley's method to solve the Kepler's equation. Errors (left) and CPU times (right, units: second) are averaged with respect to $M \in [0, \pi]$.

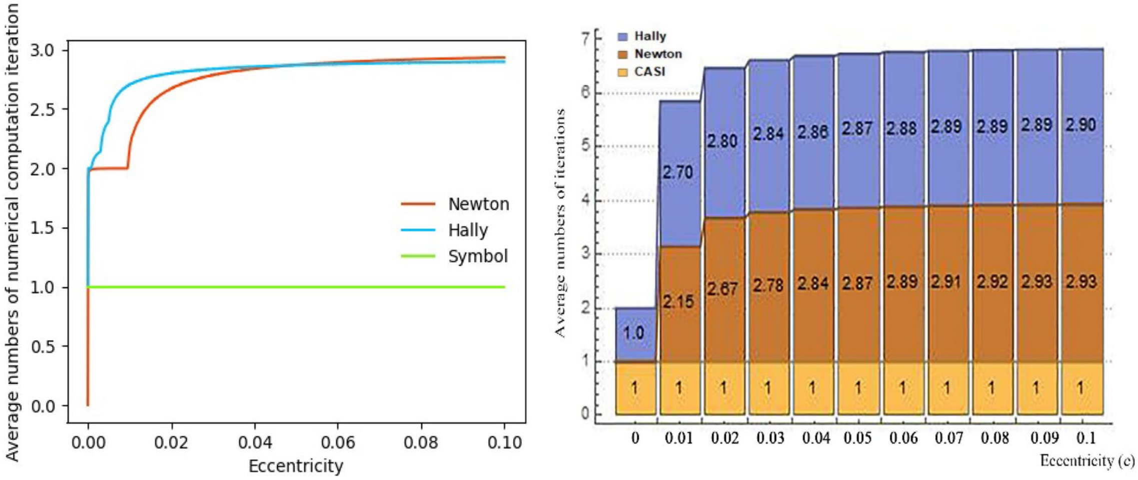

Figure 11. Compared with SICAA analytic solution, average numbers of iteration of standard Newton–Raphson method and Halley's method, which are averaged with respect to $M \in [0, \pi]$, respectively.

$$\bar{\delta}_E(e_j) = \frac{\sum_{i=0}^{1000} \delta_{Ei,j}(M_{i,j}, e_j)}{1001} \tag{21}$$

$$\delta_{Ei,j}(M_{i,j}, e_j) = \left| f_2(M_{i,j}, e_j) - \widehat{E_{i,j}} \right| \tag{22}$$

$$\bar{t}_E(e_j) = \frac{\sum_{i=0}^{1000} t_{i,j}(M_{i,j}, e_j)}{1001} \tag{23}$$

Figure 10 shows numerical comparisons among our SICAA method, the standard Newton–Raphson method and the Halley's method. The SICAA method (Green line) has a higher precision than the traditional iterative methods when $e \ll 1$, with all errors under $4.5 \times 10 -{}^{16}$, less than machine precision ϵ. As for CPU time cost, the SICAA method' s time cost is close to the Halley's method (cubic-Newton method, the blue line) at 10^{-5}order of magnitude (units: second). Therefore, compared with traditional iterative methods, the SICAA method is available to solve the Kepler' equation for nearly-circular orbits, by one order of magnitude more in precision with time cost at around 0.00005 s. The precision of our method is stable.

Compared with our analytic solution, Fig. 11 shows average numbers of iteration of two traditional Newton iterative methods. Our method only needs to solve one analytic formula to get the final results. With the increase of e, the iteration numbers of traditional methods increase rapidly, and the total computation numbers of transcendental functions will increase exponentially as iteration numbers grow. Meanwhile, iteration numbers are greatly affected by the iterative starters. When the quality of starters is poor, the performance of traditional iterative methods will be significantly degraded. Many studies[18] focus on the starting algorithms of the traditional iterative methods to solve the Kepler's equation. As for nearly-circular orbits, the starter $E_0 = M$ is a simple form of starter with relatively small computational cost. However, compared with our method, there is still a certain gap in accuracy, mainly due to the limits of truncation caused by machine precision ϵ in each iterative step.

As can be seen from Fig. 6, 99.93% of the all errors computed by our SICAA method are lower than machine precision $\epsilon = 2.220446049250313 \times 10^{-16}$. The maximum errors are 4.4409×10^{-16} accounting for 0.07%, while minimum errors are 0 accounting for 96.42%, over the entire interval, showing that the accuracy is almost one order of magnitude higher than that of traditional Newton iterative methods (double precision). In this case, our method, to some extent, has a better accuracy with acceptable time consumption to solve the Kepler's equation for nearly-circular motions than the traditional Newton iterative methods.

Comparisons with traditional non-iterative methods. There are also many classic non-iterative solutions. For example, the method proposed by American scholar Nijenhuis[1] in 1991 combines the Mikkola's idea and higher-order Newton's method. Besides, the method proposed by NASA researcher Markley[12] in 1995 is based on higher-order processing of cubic algebraic equations. At that time, those methods could effectively solve the Kepler's equation, by reducing the calculation numbers of the transitional functions and providing non-iterative solutions, which optimizes the algorithms, compared with the Newton iterative methods. However, due to the limitations of computer science at that time, these methods could not balance the accuracy with CPU time consumption. With the rapid development of computer science, although their ideas still shine with wisdom, the algorithms themselves are difficult to meet the current requirements of accuracy.

Discussions and applications

Mathematics essence of our method and relationship with traditional Fourier–bessel series. The traditional series expansion of the eccentric anomaly is provided by scientists Colwell[13] and Battin[9], as shown in Eqs. (24) and (25).

$$E = M + 2\sum_{k=1}^{\infty} \frac{1}{k} J_k(ke) \sin kM \tag{24}$$

$$J_k(ke) = \sum_{j=0}^{\infty} (-1)^j \frac{\left(\frac{ke}{2}\right)^{k+2j}}{j!\,(k+j)!} \tag{25}$$

The Eq. (24) contains two infinite series, which cannot be directly solved by numerical methods. The scientists have studied the truncation methods of infinite Fourier–Bessel representations, for example, the method based on the Lambert W function[17] (referred to as the Lambert W method), as shown in Eqs. (26) and (27). Moreover, the kth-order Bessel function also need to be truncated, as shown in Eq. (28), where $j_{max} = s$ is obtained by the Eqs. (29) and (30).

$$E \approx M + 2\sum_{k=1}^{k_{max}} \frac{1}{k} J_k^s(ke) \sin kM \tag{26}$$

$$k_{max} = k^* - 1 = \left(\frac{c_2}{c_1}\right) \mathcal{W}\left(\exp\left(\frac{c_3}{c_2}\right)\frac{c_1}{c_2}\right) - 1 \tag{27}$$

$$J_k^s(ke) = \sum_{j=0}^{s} (-1)^j \frac{\left(\frac{ke}{2}\right)^{k+2j}}{j!\,(k+j)!} \tag{28}$$

$$s = \left(\frac{k_{max}}{2}\right) + 2\frac{g_1\left(\frac{k_{max}}{2}\right)}{g_2\left(\frac{k_{max}}{2}\right)} \tag{29}$$

$$\begin{cases} g_1(s^*) = \frac{d}{ds^*}\left(\frac{1}{g(s^*)}\right) \\ g_2(s^*) = \frac{d^2}{ds^{*2}}\left(\frac{1}{g(s^*)}\right) \end{cases} \tag{30}$$

In Eq. (27), c_1, c_2, c_3 are parameters related to the basic inequality of kth-order Bessel function[24] and the Lambert W function[17].

As our method can reach the accuracy of 10^{-17} (given in "Analysis and comparison" section), we compare the truncation of infinite series expansion based on the Lambert W function (the Lambert W method) at the same accuracy of 10^{-17}. For $e = 0.1$ and $10^{-N_{tol}} = 10^{-17}$, the max order obtained through the Eq. (27) [17] is $k_{max} = 17$. To reach the accuracy of 10^{-17} for a second-order Newton–Raphson solution, we use $k = k_{max} = 17$, and $e = 0.1$ in the Eq. (28) and the Eq. (29); the order needed in $J_{17}(17e)$ is $s = 1$. Figure 12 shows numerical comparisons

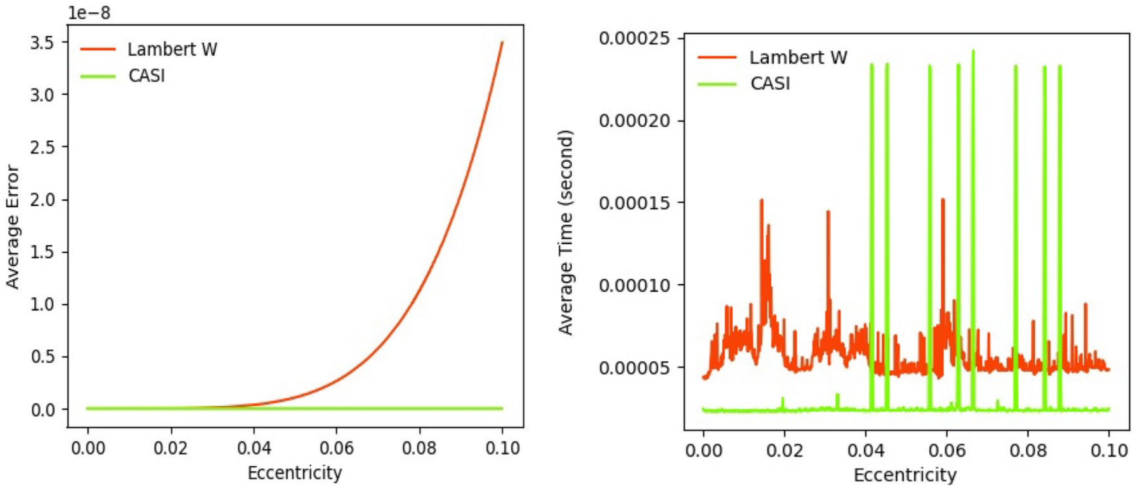

Figure 12. Numerical comparisons among the SICAA method and the Lambert W method to solve the Kepler's equation. Errors (left) and CPU times (right, units: second) are averaged with respect to $M \in [0, \pi]$ over the domain $e \in [0, 0.1]$.

between the SICAA method and the Lambert W method. The SICAA method (Green line) has a higher precision than the Lambert W method. As for CPU time cost, the SICAA method's time cost is shorter than that of the Lambert W method except for a few outliers. Therefore, compared with the traditional truncation of series expansion method, our method presents better truncation combinations of the two infinite series expansion for solving the Kepler's equation when the orbit is nearly-circular.

In conclusion, based on the Eq. (9) and the Eq. (10), the mathematics essence of our method is a new effective and accurate truncation method of Fourier–Bessel representations for the Kepler's equation, which could achieve the accuracy of E at 10^{-17}, with higher accuracy than traditional truncation method of Lambert W's.

The performance and the application of the SICAA method. At present, most of the solutions for the Kepler's equation focus on refining the iterative methods. Relying on high-speed computer hardware, by optimizing the iterative algorithms and their starters, iterative methods[25] are often more efficient to solve transcendental functions than complex analytic solutions, not to mention that there are still many transcendental functions that could not be solved analytically. For example, in 2019, the fast switch and spline scheme[21] proposed by Spanish scholar Daniele Tomasini is used to solve the Kepler's equation with a faster speed than that of the standard Newton–Raphson method. The CORDIC-like iterative method[20] proposed by German scholar Zechmeister can be applied for both elliptic orbits and hyperbolic orbits, and consumes less CPU time than the Newton iterative methods. However, If only nearly-circle orbits are considered, our method could get accuracy at 10^{-17}, the accuracy of which is higher than CORDIC-like iterative method.

Conclusion

This paper presents a new efficient and accurate truncation method of the infinite Fourier–Bessel representation, symbolic iteration method based on computer algebra analysis, taking Kepler's equation as an example. Based on the computer algebra system, it eventually provides an analytical formula to compute the eccentric anomaly without complex iterative computation at run-time. Compared with traditional truncation method (Lambert W), our method has higher accuracy and shorter CPU consumption; compared with commonly used iterative methods (the standard Newton–Raphson method and the Halley's method), our method has higher accuracy and stronger robustness; compared with high-performance iterative algorithms proposed in the new century, on the basis of retaining the accuracy of the algorithm, our method takes into account the advantages of analytical methods and the simplicity of codes. Further, the simple codes make our method well-suited for various satellite orbit determination algorithms, semi-analytic satellite orbit propagators form low fidelity to high fidelity, and other geoscience fields.

Code availability

Name of code: "CASI Mthod.py" and "numerical CASI.py"; Developer: Ruichen Zhang; Year first available: 2020.11; Hardware required: Win 10; Software required: Mathematica 11.2; Python 3.8; Program language: Mathematica, python; Program size: 4 KB, 1 KB; The source code are available for download at GitHub: https://github.com/chur-614/CSI_chur.

Received: 26 August 2021; Accepted: 11 February 2022
Published online: 22 February 2022

References

1. Nijenhuis, A. Solving Kepler's equation with high efficiency and accuracy. *Celest. Mech. Dyn. Astron.* **51**(4), 319–330 (1991).

2. Ng, E. W. A general algorithm for the solution of Kepler's equation for elliptic orbits. *Celest. Mech. Dyn. Astron.* **20**(3), 243–249 (1979).
3. Brouwer, D. & Clemence, G. M. *Methods of Celestial Mechanics* (Elsevier, 1961).
4. Danby, J. M. & Teichmann, T. Fundamentals of celestial mechanics. *Phys. Today.* **16**, 63 (1963).
5. Kaula, W. *Theory of Satellite Geodesy* (Blaisdell Publishing Company, 1966).
6. Xu, G. & Xu, J. *Orbits—2nd Order Singularity-Free Solutions* (Springer, 1964).
7. Ashby, N. & Spilker, J. J. Introduction to relativisitic effects on the global positioning system. In *Global Positioning System: Theory and Applicaions* Vol. I, Chapter 18 (eds Parkinson, B. W. & Spiller, J. J.) (AIAA, 1996).
8. Aarseth, S. J. *Gravitational N-Body Simulations Tools and Algorithms* (Cambridge University Press, 2003).
9. Battin, R. H. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Version* (AIAA, 1999).
10. Karslioglu, M. O. An interactive program for GPS-based dynamic orbit determination of small satellites. *Comput. Geosci.* **31**(3), 309–317 (2005).
11. Tenzer, R. *et al.* Spectral harmonic analysis and synthesis of Earth's crust gravity field. *Comput. Geosci.* **16**, 193–207 (2012).
12. Markley, F. L. Kepler equation solver. *Celest. Mech. Dyn. Astron.* **63**(1), 101–111 (1995).
13. Colwell, E. *Solving Kepler's Equation Over Three Centuries* (Willman-Bell, 1993).
14. Taff, L. G. & Brennan, T. A. On solving Kepler's equation. *Celest. Mech. Dyn. Astron.* **46**, 163–176 (1989).
15. Odell, A. W. & Gooding, R. H. Procedures for solving Kepler's equation. *Celest. Mech.* **38**, 307–334 (1986).
16. Mikkola, S. A cubic approximation for Kepler's equation. *Celest. Mech.* **40**, 329–334 (1987).
17. Sengupta, P. The Lambert W function and solutions to Kepler's equation. *Celest. Mech. Dyn. Astron.* **99**(1), 13–22 (2007).
18. Calvo, M. *et al.* Optimal starters for solving the elliptic Kepler's equation. *Celest. Mech. Dyn. Astron.* **115**(2), 143–160 (2013).
19. Oltrogge, D. L. Efficient solutions of Kepler's equation via hybrid and digital approaches. *J. Astronaut. Sci.* **62**(4), 271–297 (2015).
20. Zechmeister, M. CORDIC-like method for solving Kepler's equation. *Astron. Astrophys.* **619**, A128 (2018).
21. Tommasini, D. & Olivieri, D. N. Fast switch and spline scheme for accurate inversion of nonlinear functions: The new first choice solution to Kepler's equation. *Appl. Math. Comput.* **364**, 124677 (2020).
22. Traub, J. *Iterative Methods for the Solution of Equations* (Prentice-Hall, 1964).
23. Fukushima, T. A method solving Kepler's equation without transcendental function evaluations. *Celest. Mech. Dyn. Astron.* **66**(3), 309–319 (1997).
24. Watson, G. N. *A Treatise on the Theory of Bessel Functions* 2nd edn. (Cambridge University Press, 1966).
25. Evensen, G. Analysis of iterative ensemble smoothers for solving inverse problems. *Comput. Geosci.* **22**, 885–908 (2018).

Author contributions

Author Contributions: Conceptualization, R.Z. and S.B.; methodology, R.Z.; software, R.Z.; validation, R.Z., S.B. and H.L.; formal analysis, R.Z.; investigation, R.Z.; resources, S.B.; data curation, H.L.; writing—original draft preparation, R.Z.; writing—review and ed-iting, R.Z.; visualization, R.Z.; supervision, S.B.; project administration, S.B.; funding acquisition, S.B. and H.L.. All authors have read and agreed to the published version of the manuscript.

Funding

[National Natural Science Foundation of China] Grant Number [42074010, 41971416], [National Science Foundation for Outstanding Young Scholars] Grant Number [42122025] and [Natural Science Foundation for Distinguished Young Scholars of Hubei Province of China] Grant Number [2019CFA086].

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-07050-5.

Correspondence and requests for materials should be addressed to R.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.