Numer Algor (2012) 59:147–159
DOI 10.1007/s11075-011-9481-6

ORIGINAL PAPER

Computational uncertainty and the application of a high-performance multiple precision scheme to obtaining the correct reference solution of Lorenz equations

Pengfei Wang · Jianping Li · Qian Li

Received: 19 January 2011 / Accepted: 6 June 2011 /
Published online: 1 July 2011

Abstract The computational uncertainty principle (CUP) is applied to explain the experimental formulae of the critical time of decoupling for Lorenz equations (LEs). We apply the multiple precision (MP) library in obtaining the long-time solution of LEs, and based on the classic Taylor scheme, we developed a high-performance parallel Taylor solver to do the computation. The new solver is several hundreds times faster than the reported solvers developed in MATHEMATICA software, and it has the ability to yield longer solutions of LEs, up to $t \sim 10^4$ LTU (Lorenz time unit). Further, we notice that the two computation processes with different precisions or orders will produce the reliable correct reference solutions before they have a significant difference. According to this property we propose an approach for maintaining the correct numerical solution. The new solver and the solution validation approach are used to identify and correct an erroneous solution reported in a previous study.

P. Wang (✉) · J. Li
State Key Laboratory of Atmospheric Sciences and Geophysical Fluid Dynamics (LASG), Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing, China
e-mail: wpf@mail.iap.ac.cn

J. Li
e-mail: ljp@lasg.iap.ac.cn

P. Wang
Graduate University of Chinese Academy of Sciences, Beijing, China

Q. Li
Key Laboratory of Middle Atmosphere and Global Environment Observation (LAGEO), Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing, China
e-mail: qian.li@mail.iap.ac.cn

Keywords Multiple precision · Computational uncertainty · Taylor scheme · Parallel · Lorenz equations

1 Introduction

Recent studies have documented the time-step sensitivity of numerical solutions for Lorenz equations [1] (LEs) and other simple nonlinear atmospheric models. Teixeira et al. ([2]; hereafter, TRJ07) discussed time-step sensitivity for nonlinear systems and a quasigeostrophic (QG) model. Yao and Hughes ([3]; hereafter, YH08) considered it an important contribution to future studies, although, in contrast to TRJ07, they considered that the correct trajectory of LEs remains unknown. Both TRJ07 and YH08 emphasized the importance of time-step sensitivity in terms of numerical nonlinear dynamical systems. Liao ([4]; hereafter, L09) reported that many factors (e.g., computational chaos and computational periodicity) may cause the numerical results of a nonlinear dynamic system to deviate markedly from the correct solution. This finding highlights the importance of studying prediction uncertainty in numerical computations.

YH08 noted that, 'The sensitivity of computed results for chaos or turbulence to the size of integration time steps is not completely unknown in the numerical analysis community.' Indeed, Li et al. ([5, 6]; hereafter LZC00 and LZC01, respectively) and Li [7] had already carried out systematic investigations on the sensitivity of numerical solutions of nonlinear ordinary differential equations (ODEs), employing both numerical experiments and theoretical analysis. In particular, they presented the computational uncertainty principle (CUP) in numerically solving ODEs. The works of LZC00 and LZC01 are important because their results can help us to understand and explain most of the phenomena reported by TRJ07 and L09. Based on LZC00 and LZC01 Hu and Chou [8] further proposed a new concept of globally convergence of numerical pattern to study uncertainty of the numerical solution of a nonlinear system's long-term behavior. A review on CUP could be seen in Li and Chou [9] and Li and Wang [10].

The sensitivity of computed results for chaos systems is important. For example, LZC00, TRJ07, and L09 reported that the effective computation time for double precision is approximately 35 LTU (Lorenz time unit) for LEs. Consequently, many studies that use a step size of 0.01 and that integrate for more than 3500 steps before analyzing the result of LEs would suffer from a degree of inaccuracy. In such cases, the validity of the experimental results is doubtful if only the original double-precision environment is employed. Therefore, it is even more important to obtain a suitable difference solver to provide the correct solution for 100 and 1000+ LTU, rather than to simply present arguments regarding the time-step sensitivity and other sensitivities.

To overcome the limitation of the effective computation time that arises from low floating-point precision, Wang et al. [11] and L09 presented multiple precision utilities to be applied in solving LEs. These works proved that

increased precision is valuable in terms of obtain reliable numerical solution. L09 extended the reliable numerical solution of LEs to approximately 1200 LTU; however, additional work is required in this area. For example, L09 did not validate the result obtained at 1200 LTU. Furthermore, a high-precision solver is computationally expensive: L09 required 461 h to obtain the result at $t = 1200$. The long computation time means that the scheme is seldom used.

The aim of the present study is to propose an operational method which can obtain the LEs' long time correct numerical solution, and then we improve the scheme performance to reduce the computation time. The organizations of this article are as follows. In Section 2 the CUP is introduced and applied to explain one of the experimental formulae proposed by TRJ07. In Section 3, the theory and method to maintain the correct numerical solution are proposed. The Sections 4 and 5 focus on the time cost analysis of reliable computation with Taylor scheme, and then develop a high performance parallel version Taylor scheme to solve LEs. The summary and some discussions are put in Section 6.

2 Computational uncertainty principle and an explanation of results reported by TRJ07

By employing the four classes (explicit one-step methods, explicit multistep methods, implicit methods, and modified predictor corrector methods) of 29 standard numerical integration methods, LZC00 and LZC01 reported the step size sensitivity of numerical nonlinear models, as also documented by TRJ07. Figure 2 of TRJ07 essentially shows the same phenomena as those in Fig. 1 of LZC00. Furthermore, LZC00 and LZC01 studied the sensitivity to computation precision (taking into account machine precision, such as round-off errors) of numerical solutions for chaotic systems. In fact, step size sensitivity is just one of multiple sensitivities to computational precision. Unlike TRJ07, who used time steps to study the error evolution, LZC00 identified the optimal step size (OS) and maximally effective computation time (MECT) by using an optimal searching method. LZC00 and LZC01's experiments and theoretical analyses are relatively complicated in seeking to understand the global error evolution of numerical solutions to chaos systems.

TRJ07 sought to explain the time step sensitivity and numerical convergence (as outlined above) in terms of truncation error growth. In light of the findings presented by LZC01, we consider that truncation error, when considered alone, does not explain the sensitivity or numerical convergence. In fact, to explore this issue, the total numerical error (i.e., the sum of the truncation error e and round-off error r) growth should be taken into account. LZC00's experiments showed that the precision of floating-point operations has an important effect on the long-time numerical integration of nonlinear systems (TRJ07 did not discuss the influence of machine precision on the results).

Hence, to address this issue, LZC01 introduced three types of convergences: theoretical, numerical, and actual.

To ensure that the results of LZC00 and TRJ07 are applicable to more general cases, it is necessary to improve the classical results and to obtain a unified error estimate for the general multistep method, in particular to obtain the normal accumulated growth of the round-off error for a floating-point machine. By introducing a new type of recurrent inequality in LZC01, not only are the classical error bounds improved, but a unified estimate is obtained for the total error of the general method. Specifically, probabilistic theory is used to obtain the normal accumulated growth of round-off error for a floating-point machine. The detailed proof can be found in LZC01.[1] Here, we provide only the main results.

If the initial value is accurate, the numerical error is given by the sum of the truncation error e and the round-off error r, yielding a total error of $E = e + r$. Based on the statistical theory of local round-off errors (Henrici [12]), LZC01 derived the following unified estimate for the total error of the general k-step size method for ODEs:

$$\|E(t;h)\| = \|e(t;h)\| + \|r(t;h)\| \le C(t)\,\tilde{E}(h,n) = C(t)\left(Ch^p + \frac{\sigma}{\tilde{C}\sqrt{h}}\right), \quad (1)$$

where h is the step size, $C(t) = e^{C_L \tilde{\Gamma}(t-t_0)}/\sqrt{C_L}$, $\tilde{E}(h,n) = \tilde{e} + \tilde{r}$ is the core function ($\tilde{e} = Ch^p, \tilde{r} = \sigma/\tilde{C}\sqrt{h}$), C is a constant that depends on the employed numerical method, C_L is a constant that depends on the ODEs, $\tilde{C} = \sqrt{2C_L}$, p is the order of the numerical method, $\sigma = 10^{-n}M_0$, $M_0 = \max_{\tau\in[t_0,t]} \|y(\tau)\|/2\sqrt{3}$, and n is the number of significant digits of the floating-point operation.

Equation 1 can explain the variation in numerical error with step size. With decreasing step size, the total error initially decreases with decreasing discretization error (also know as truncation error), because the total error is primarily generated by the discretization error during this period. However, when the step size is smaller than a certain critical value, the round-off error caused by computation becomes dominant, meaning that the total error increases with increasing round-off error. Invariably, there exists a step size H for which the total error is minimized. This step size H is the OS. Using the above theoretical formula, LZC01 found the OS (i.e., H) to be

$$H = \left(\frac{10^{-n}}{2pCD}\right)^{1/(p+0.5)}, \quad (2)$$

where $D = \tilde{C}/M_0$.

[1] PDF files of the papers LZC00 and LZC01 can be download from http://159.226.119.57/wpf/CUP/2000-ljp.pdf and http://159.226.119.57/wpf/CUP/2001-ljp.pdf, respectively.

An important contribution by TRJ07 is the approximation formulation of the critical time of decoupling, as follows:

$$T_c(h) \approx T_c(1) - 5.5 \log_{10}(h), \tag{3}$$

where $h = \Delta t$ is the step size. This relationship establishes the limit of predictability for Lorenz equations with step size h. However, (3) is only suitable for the second-order numerical scheme used by Lorenz [1], for Lorenz equations, and in cases for which the truncation error is much larger than the round-off error. In fact, (3) is a special case of the work presented by LZC01, which tells us the general relation between the critical time of decoupling and the order of the numerical scheme, ODEs, step size, and machine precision. Given the error tolerance δ, machine precision with a significant digit n and a scheme of order p, (3) can be obtained from (1). Following (1) one has

$$\frac{C(T_c(h_1))}{C(T_c(h))} = \frac{h^p + \sigma/C\tilde{C}\sqrt{h}}{h_1^p + \sigma/C\tilde{C}\sqrt{h_1}},$$

i.e.,

$$T_c(h) = T_c(h_1) - \frac{1}{C_L\tilde{\Gamma}} \ln \frac{h^p + \sigma/C\tilde{C}\sqrt{h}}{h_1^p + \sigma/C\tilde{C}\sqrt{h_1}}. \tag{4}$$

This is the general relation of the critical time of decoupling between two step sizes h and h_1, which depend on the ODE, the order of the scheme, and machine precision. In the case of a large truncation error, (4) is approximately simplified as

$$T_c(h) \approx T_c(h_1) - \frac{p}{C_L\tilde{\Gamma}} (\ln h - \ln h_1), \tag{5}$$

where h and $h_1 > H$. When $p = 2$ and $h_1 > 1$, we have

$$T_c(h) \approx T_c(1) - \frac{2}{C_L\tilde{\Gamma}} \ln h \approx T_c(1) - \frac{4.6}{C_L\tilde{\Gamma}} \log_{10} h. \tag{6}$$

Equation 6 contains (3).

The formula derived from LZC01 can also be used to explain three other experiment-based relations obtained by L09 (for details, see Cao and Wang [13]).

3 Utility of multiple precision and determination of the correct numerical solution

Here, we focus on obtaining the correct long-time reference solution of ODEs. We use the Lorenz equation as an example because it has been intensively studied and because the solutions are sensitive to the initial conditions, due to chaos.

From Section 2, we know that MECT exists in a computation when the floating-point precision is fixed. LZC00 reported that MECT values for single and double precision are 16.8 and 35.4, respectively. Therefore, if we want to obtain the solution of $t = 1000$, single- and double-precision machines are insufficient.

Fortunately, a suite of software named 'MP' (meaning 'multiple precision'; Oyanarte [14]) has been developed for computation. The MP library provides high floating-point precision for numerical computation. The similar reliable mathematic librarys are support by scientific software such as MATHEMATICA, MAPLE, and MATLAB. MP is well known for its successful application in computing π to millions of digits. Wang et al. [11] described how to apply MP in solving LEs. The precision in MP is bits precision, which is different from significant digits. For example, single and double precision correspond to 7.22 and 15.95 significant digits, respectively, while they correspond to 24- and 53-bit precision in MP, respectively.

Using MP, it is possible to choose a sufficiently high precision with a certain step size h to maintain round-off errors that are negligible compared with the truncation error. In such a case, the total computation errors are derived solely from truncation errors. Although we apply very high precision in the computation process, for convenience of analysis, we only output (to file or to the screen) the computed result to the first 16 digits. Thus, the result with a relevant error less than 10^{-16} will appear the same as the correct result in the case of 16-digit output. Hereafter, we regard $E_0 = 10^{-15}$ as a visible error for 16-digit output.

The truncation error for each computation step is small; however, after thousands of computation steps, the accumulation of truncation errors means that the total computation error becomes a significant error E_s, $E_s >> E_{t1}$ where E_{t1} is the truncation error derived from a single computation step. For example, for the fourth-order Runge–Kutta (hereafter, RK4) method, using $h = 10^{-6}$, $p = 4$, and $E_{t1} \approx h^4 \approx 10^{-24}$, a significant error arises in the case of $E_s = 10^{-8} >> 10^{-24}$ (we regard $E_s = 10^{-8}$ as a significant error in this study). The error evolution for E_s to the critical error E_c can be approximately described by the initial error rule from Eq. 60 in LZC01:

$$\|e\,(t;h)\| \leq N_1\,(\eta)\,E_s e^{kL\Gamma^*(t-t_0)}, \tag{7}$$

where k is the number of steps in the multistep method, and the definition of $N_1(\eta)$, L, Γ^* are given in LZC01.

Given the error tolerance δ, the initial error vector is $(\Delta x_0, 0, 0)$, meaning that $E_s = \sqrt{\Delta x_0^2} = \Delta x_0$; i.e., $N_1(\eta)\Delta x_0 e^{kL\Gamma^* t} = \delta$.

Taking the base 10 logarithm of both sides of the above equation, we have

$$T_c = -\frac{2.3}{kL\Gamma^*}\log_{10}\Delta x_0 + \frac{2.3}{kL\Gamma^*}\log_{10}\frac{\delta}{N_1\,(\eta)}. \tag{8}$$

Table 1 Last three lines from Table 1 in L09

t	x	y	z
1150	−14.378782424952437	−11.819346602645444	37.319351169225996
1175	−11.794511899005188	−13.181679857519981	29.65720151904728
1200	2.4537546196402595	4.124943247158509	19.349201739150004

This formula implies a general linear relationship between T_c and the logarithm of the inaccuracy of the initial conditions Δx_0.

For certain initial conditions, (8) becomes the formula proposed by L09:

$$T_c \approx -2.5 \log_{10} (\Delta x_0) . \tag{9}$$

If we make substitution of $E_s = 10^{-8}$ into (9), we have $T_c \approx 20$. This means that if the computation error reaches the error $E_s = 10^{-8}$, it will reach E_c within 20 LTU. Therefore, if we start two computation processes with different precisions or orders, the resulting outputs are the correct reference solutions before they develop a significant error E_s. This is an important and operational determination to obtain the correct reference solutions. Using the program developed in Section 5, we assessed the result reported by L09 with an initial value of (−15.8, −17.48, 35.64). The numerical solution listed in Table 1 of L09 is correct before t reaches 1150, but after this point the solution contains visible errors (The last three lines are listed in Table 1). For $t = 1175$, the error becomes the significant error (approximately $\Delta x \approx 10^{-6}$); the result at $t = 1200$ is already incorrect. Table 2 lists the correct numerical solutions for time values of 1150, 1175, and 1200 (The reliable computation which gives the results up to 2500 LTU same as the experiments in Table 4 can tell us that the values in Table 2 is correct.). Even so, Liao's result at $t = 1100$ LTU is correct and confirmed by the experiments, and to our knowledge that Liao [4] was the first one to gain the accurate numerical result of Lorenz equation in such a long time.

The incorrect nature of the original solution example indicates that the experimental formula employed by L09 is not exact, especially the relation of Tc to K, and of Tc to M. However, Liao's formula yields the approximately correct result and his work is very valuable to the subsequently research. This minor error indicates the importantance on developing a suitable method for assessing the accuracy of numerical results, to avoid providing inaccuracy results.

Table 2 Corrected results for the last three lines from Table 1 in L09

t	x	y	z
1150	−14.378782424952439	−11.819346602645446	37.319351169225996
1175	−11.794510477754745	−13.181680560399085	29.657196905685254
1200	−7.360729071096472	−11.937734008420378	16.803323148067371

4 Time cost analysis to obtain the correct reference solution for LEs

The operational way to obtain the long-time numerical solution for chaos dynamics systems does not only depend on the step size and precision of the difference method: it is also limited by the real-world time cost of the solution process. For example, in the computation of LEs by RK4, we can choose a very small step size such as $h = 10^{-170}$ and a very high precision $K = 10000$ to obtain a correct solution for $t = 1000$; however, the total number of computation loops is approximately

$$\frac{t}{h} \approx 10^{173}. \tag{10}$$

In this case, if one loop of computation costs 10^{-4} seconds, reflecting the CPU speed of the machine, it would take approximately 3.1×10^{160} years to finish the computation. Therefore, the above approach to obtaining the solution of $t = 1000$ is unrealistic given the current state of technology.

The truncation error is about $E_t \sim Ch^p$ and the computation time is mainly affected by $\frac{t}{h}$; therefore, if we need $E_t \sim C_1 h^4$ for the RK4 method, we can use a higher-order method with a fixed step size (such as 0.01) to reach the same $E_t = C_2 0.01^p$. The higher-order time cost is on the magnitude of

$$\frac{t}{0.01} \approx 10^5 \tag{11}$$

loops, and one loop of the higher-order method may cost 10^1 seconds, which is larger than the cost in RK4. The total computation cost is about 10^6 seconds.

The above analysis reveals that the use of a suitable higher-order method in obtaining the correct reference solution of LEs can yield an exponential reduction in computation time. When using MP and higher-order methods, the search for the optimal step size is not as important as that in the single- or double-precision floating-point environments because the increased order of the scheme is more efficient in the computation of this type of differential equation.

5 Multiple precision Taylor method and its parallel performance

The classical LEs introduced by Lorenz [1] are as follows:

$$\begin{cases} \dfrac{dx}{dt} = -\sigma x + \sigma y \\ \dfrac{dy}{dt} = Rx - y - xz \\ \dfrac{dz}{dt} = xy - b z \end{cases}, \tag{12}$$

where σ, R, and b are nondimensional constants, and t is nondimensional time ($R = 28.0, \sigma = 10.0, b = 8/3$).

A classical higher-order method to solve ODEs is the Taylor method. Accounts of the application of the Taylor scheme to LEs can be found in LZC01, Lorenz [15], and L09. The p-order truncated Taylor scheme with step size h is

$$\begin{cases} x_{n+1} = x_n + \sum_{k=1}^{p} \alpha_k h^k \\ y_{n+1} = y_n + \sum_{k=1}^{p} \beta_k h^k \\ z_{n+1} = z_n + \sum_{k=1}^{p} \gamma_k h^k \end{cases}, \tag{13}$$

where $\alpha_k = \frac{1}{k!}\frac{\mathrm{d}^k x(t_n)}{\mathrm{d}t^k}$, $\beta_k = \frac{1}{k!}\frac{\mathrm{d}^k y(t_n)}{\mathrm{d}t^k}$, and $\gamma_k = \frac{1}{k!}\frac{\mathrm{d}^k z(t_n)}{\mathrm{d}t^k}$; i.e.,

$$\begin{cases} x_{n+1} = x_n + \sum_{k=1}^{p} \alpha_k h^k = x_n + \sum_{k=1}^{p} \frac{h^k}{k!} x^{(k)} \\ y_{n+1} = y_n + \sum_{k=1}^{p} \beta_k h^k = y_n + \sum_{k=1}^{p} \frac{h^k}{k!} y^{(k)} \\ z_{n+1} = z_n + \sum_{k=1}^{p} \gamma_k h^k = z_n + \sum_{k=1}^{p} \frac{h^k}{k!} z^{(k)} \end{cases},$$

where $x^{(k)} = \frac{\mathrm{d}^k x(t_n)}{\mathrm{d}t^k}$, $y^{(k)} = \frac{\mathrm{d}^k y(t_n)}{\mathrm{d}t^k}$, and $z^{(k)} = \frac{\mathrm{d}^k z(t_n)}{\mathrm{d}t^k}$.

The core process of the Taylor method is computation of the coefficients $x^{(k)}$, $y^{(k)}$, and $z^{(k)}$. For (12),

$$\begin{cases} x^{(1)} = -\sigma x_0 + \sigma y_0 \\ y^{(1)} = R x_0 - y_0 - x_0 z_0 \\ z^{(1)} = x_0 y_0 - b z_0 \end{cases}$$

and

$$\begin{cases} x^{(k+1)} = -\sigma x^{(k)} + \sigma y^{(k)} \\ y^{(k+1)} = R x^{(k)} - y^{(k)} - \sum_{i=0}^{k} C_k^i x^{(k-i)} z^{(i)} \\ z^{(k+1)} = \sum_{i=0}^{k} C_k^i x^{(k-i)} y^{(i)} - b z^{(k)} \end{cases}, \tag{14}$$

where $C_k^i = \frac{k!}{i!(k-i)!}$.

Equation 14 indicates that each $x^{(k)}$, $y^{(k)}$, and $z^{(k)}$ can be computed from the previous $x^{(i)}$, $y^{(i)}$, and $z^{(i)}$ without computation of the complex higher-order derivative of (12).

Table 3 Performance of the paralleled Taylor scheme for LEs ($t = 1200$, $K = 800$, $p = 400$)

CPU	Time (h)
1	32.91
5	7.52
10	4.13
20	2.46
50	1.48

To save computation time, $a_k = \frac{h^k}{k!}$ and $C_k^i = \frac{k!}{i!(k-i)!}$ are computed previously and stored in memory.

After computation of each $x^{(k)}$, $y^{(k)}$, and $z^{(k)}$, we use $\begin{cases} x_{n+1} = x_n + \sum_{k=1}^{p} a_k x^{(k)} \\ y_{n+1} = y_n + \sum_{k=1}^{p} a_k y^{(k)} \\ z_{n+1} = z_n + \sum_{k=1}^{p} a_k z^{(k)} \end{cases}$ to obtain the values at the next step.

The time required for the Taylor scheme is mainly the computation cost of $x^{(k)}$, $y^{(k)}$, and $z^{(k)}$. At each step, $x^{(k)}$ needs $3p$ times of computation, while $y^{(k)}$ needs $2p + \frac{p(p+1)}{2}$ times and $z^{(k)}$ needs $p + \frac{p(p+1)}{2}$ times, yielding a total

Table 4 Reference solutions to $t = 2500$, for initial values of $(-15.8, -17.48, 35.64)$, $p = 1000$, $K \approx 2100$, and $h = 0.01$

t	x	y	z
100	−10.510118721506247	−12.172542813682252	27.476265630374758
200	−6.697233173381983	−11.911020483539128	13.036826414358320
300	10.197534991661733	3.906517722362926	35.337427092404411
400	−1.889247649804987	−3.565788040897466	20.299639635504597
500	−5.305099631571520	−9.425991029211517	12.302184230689779
600	−0.863505382597614	0.499057856286716	21.581438144249073
700	10.884963668216702	16.329893792467036	22.247458859587208
800	1.396334715413953	2.408771267581340	14.590441270059282
900	−6.449367823985297	−10.984642417532422	14.647974468278282
1000	13.881997000862393	19.918303160406392	26.901943308376104
1100	2.297459271183663	2.299710874996515	19.617779431769037
1200	−7.360729071096472	−11.937734008420378	16.803323148067371
1300	−5.888408663778747	−10.303983599962610	13.202322915916419
1400	−2.956453871180797	−1.638437228803989	23.191811743846468
1500	−10.139807686977655	−7.626371611693746	31.858410078297187
1600	−0.298335828888970	−0.429115365008562	12.939623163098718
1700	0.925156507499731	1.760513513099816	16.621563705172555
1800	−3.657244075577691	1.995602340686063	29.343722751687984
1900	8.268904400223825	2.769812726031063	32.709871058174137
2000	−6.873883693205019	−1.484834827669842	31.349521074674275
2100	−16.125339444926961	−10.712941885247025	42.104638675048292
2200	0.156180911104446	−2.183461941521876	23.296902764781983
2300	−12.769575216010759	−15.170846915761850	29.884784041691550
2400	10.167229763725251	2.538185738525101	36.359275696105307
2500	2.759152441189508	0.476284454767803	24.641050496931861

of approximately $6p + p(p+1)$ times. When the order of p is large, the computation time is $T_w \propto p^2$.

This Taylor scheme, written in the language C with the MP library running on an Intel Xeon 3.0 GHz CPU, can finish the computation for $t = 1200$ within 50 h. Compared with the time result reported by L09 (461 h), the present approach is 9–10 times faster; however, this is still a long simulation time, especially if we use higher-order and higher-precision computation to validate the correction of the solution. Although our computer has many nodes, a serial version of the Taylor scheme is only able to use one core; therefore, we developed a parallel version of this scheme.

From (14), we have a time cost of p^2 when we compute $y^{(k)}$ and $z^{(k)}$ for the item $\sum_{i=0}^{k} C_k^i x^{(k-i)} z^{(i)}$. Because each item of $C_k^i x^{(k-i)} z^{(i)}$ is independent, we can separate it to N CPUs for computation, and then collect the summed result. The time cost of $\sum_{i=0}^{k} C_k^i x^{(k-i)} z^{(i)}$ is about $\frac{p(p+1)}{2N}$ theoretically, and the total computation time is $T_w \propto p^2/N$. Because the communication latency of each CPU is limited by InfiniBand hardware, the parallel efficiency is unable

Table 5 The values of last 25 steps of $t = 2500$ for variables x, the initial values are $(-15.8, -17.48, 35.64)$, $p = 1000$, $K \approx 2100$, and $h = 0.01, 0.008, 0.005$ respectively

t	$x(h = 0,01)$	t	$x(h = 0.008)$	t	$x(h = 0.005)$
2499.75	13.940269824026972	2499.8	13.044056525462182	2499.875	8.496147161034672
2499.76	13.963489999277806	2499.808	12.683151682861965	2499.88	8.165165980200449
2499.77	13.884116751049143	2499.816	12.274972388527750	2499.885	7.838897386867657
2499.78	13.701701156725459	2499.824	11.826386104294077	2499.89	7.518343312004366
2499.79	13.419448335960793	2499.832	11.344716937583200	2499.895	7.204392528717058
2499.8	13.044056525462182	2499.84	10.837475942758564	2499.9	6.897823051341781
2499.81	12.585280973458492	2499.848	10.312111383610313	2499.905	6.599305751905954
2499.82	12.055279844395985	2499.856	9.775791388535488	2499.91	6.309408959870233
2499.83	11.467825738980608	2499.864	9.235226897870936	2499.915	6.028603826509313
2499.84	10.837475942758564	2499.872	8.696538380967786	2499.92	5.757270254234712
2499.85	10.178787316218246	2499.88	8.165165980200449	2499.925	5.495703212187169
2499.86	9.505642076995475	2499.888	7.645819805245442	2499.93	5.244119281353541
2499.87	8.830724903233399	2499.896	7.142465144558241	2499.935	5.002663294353930
2499.88	8.165165980200449	2499.904	6.658336322461971	2499.94	4.771414956183620
2499.89	7.518343312004366	2499.912	6.195972655458072	2499.945	4.550395352067108
2499.9	6.897823051341781	2499.92	5.757270254234712	2499.95	4.339573266847069
2499.91	6.309408959870233	2499.928	5.343544081341681	2499.955	4.138871256791963
2499.92	5.757270254234712	2499.936	4.955595536648457	2499.96	3.948171429279260
2499.93	5.244119281353541	2499.944	4.593781769143467	2499.965	3.767320898501717
2499.94	4.771414956183620	2499.952	4.258083809963834	2499.97	3.596136896220903
2499.95	4.339573266847069	2499.96	3.948171429279260	2499.975	3.434411525768145
2499.96	3.948171429279260	2499.968	3.663463309076453	2499.98	3.281916155109350
2499.97	3.596136896220903	2499.976	3.403181686177955	2499.985	3.138405451002044
2499.98	3.281916155109350	2499.984	3.166401059863488	2499.99	3.003621061240996
2499.99	3.003621061240996	2499.992	2.952090889017366	2499.995	2.877294955870724
2500	2.759152441189508	2500	2.759152441189508	2500	2.759152441189508

to reach 100%. Table 3 lists the performance of the paralleled Taylor scheme for LEs.

Using the parallel program, it is possible to compute the correct reference solution for $t = 1200$ within 1.5 h, which is 300 times faster than that reported by L09. Furthermore, we obtain the solution for $t = 2500$ within 30 h (with $p = 1000$ and precision = 7000 bits; i.e., $K \approx 2100$) and the result is validated by computation ($p = 1200$, precision = 7000 bits; Table 4).

Table 5 gives the result of three different step-size near $t = 2500$. We can find that even for this long computation time, the reliable computation ($p = 1000$, $K \approx 2100$) can still give step insensitivity result.

6 Summary and discussion

We introduced a theoretical analysis of truncation and round-off error in the numerical integration of ODEs. The computational uncertainty principle was applied to explain one of the experimental formulae proposed by TRJ07. With a focus on the computation of long-time solutions of LEs, the MP library was applied in this numerical computation. We also proposed a method to maintain the correct numerical solution.

A time cost analysis revealed that an increase in the order of the scheme is more valuable than a decrease in the size of the time step. Based on the classic Taylor scheme, we developed a high-performance parallel Taylor solver with which to compute LEs. This solver was used to identify and correct an erroneous solution proposed in a previous study. In addition, the new solver has a computation speed that is 100 times faster than that of existing solvers; it can also help to obtain a longer solution of LEs. The algorithms of the parallel Taylor solver are also suitable for other well-known nonlinear chaos ODEs such as the Rossler attractor and Chen attractor. The solver is widely applicable in obtaining the correct long-time reference solutions for these strange attractors. The high-performance Taylor scheme, which yields the long-time solution for LEs, aims to inspire the study of chaotic dynamic systems in the future as well as today.

We consider that sets of correct reference solutions of LEs are required to validate correction of the relevant program and the accuracy of the solution before beginning an analysis, especially in terms of the long-time predictability or behavior of LEs. We obtained three sets of reference solutions corresponding to the three experimental parameters for initial values proposed by Lorenz [1] (0,1,0), LZC00 (5,5,10), and L09 (−15.8, −17.48, 35.64). These solutions can be downloaded from http://159.226.119.57/wpf/prog/Lorenz_Eq/.

It remains a challenging task to obtain the long-time reference solution of LEs. Our solver can extend t to the magnitude of 10^4 LTU; however, the time cost shows a marked increase, making it unrealistic to obtain the solution for $t > 10^{4\sim5}$. It remains a challenge for all researchers working in this field to develop more efficient and enhanced schemes or methods to overcome this problem.

Acknowledgements The authors gratefully acknowledge Prof. Liao for his suggestions that helped improve this manuscript. We also thank the reviewers for the useful examination and suggestions of our paper. This research was jointly supported by National Basic Research Program of China (2011CB309704) and the National Science Foundation of China (40730952).

References

1. Lorenz, E.N.: Deterministic nonperiodic flow. J. Atmos. Sci. **20**, 130–141 (1963)
2. Teixeira, J., Reynolds, C.A., Judd, K.: Time step sensitivity of nonlinear ntmospheric models: numerical convergence, truncation error growth, and ensemble design. J. Atmos. Sci. **64**, 175–189 (2007)
3. Yao, L.S., Hughes, D.: Comments on "time step sensitivity of nonlinear atmospheric models: numerical convergence, truncation error growth, and ensemble design." J. Atmos. Sci. **65**, 681–682 (2008)
4. Liao, S.J.: On the reliability of computed chaotic solutions of non-linear differential equations. Tellus **61A**, 550–564 (2009)
5. Li, J.P., Zeng, Q.C., Chou, J.F.: Computational uncertainty principle in nonlinear ordinary differential equations—I. Numerical results. Sci. China (Series E) **43**, 449–461 (2000)
6. Li, J.P., Zeng, Q.C., Chou, J.F.: Computational uncertainty principle in nonlinear ordinary differential equations—II. Theoretical analysis. Sci. China (Series E) **44**, 55–74 (2001)
7. Liz, J.P.: Computational uncertainty principle: meaning and implication. Bull. Chin. Acad. Sci. **15**, 428–430 (2000)
8. Hu, S.J., Chou, J.F.: Uncertainty of the numerical solution of a nonlinear system's long-term behavior and global convergence of the numerical pattern. Adv. Atmos. Sci. **21**, 767–774 (2004)
9. Li, J.P., Chou, J.F.: The global analysis theory of climate system and its applications. Chin. Sci. Bull. **48**, 1034–1039 (2003)
10. Li, J.P., Wang, S.: Some mathematical and numerical issues in geophysical fluid dynamics and climate dynamics. Commun. Comput. Phys. **3**, 759–793 (2008)
11. Wang, P.F., Huang, G., Wang, Z.Z.: Analysis and application of multiple precision computation and round-off error for nonlinear dynamical systems. Adv. Atmos. Sci. **23**, 758–766 (2006)
12. Henrici, P.: Error Propagation for Difference Methods. Wiley, New York (1963)
13. Cao, J., Wang, P.F.: Study on the effective computation time for numerical nonlinear ordinary differential equations. (2011, in press)
14. Oyanarte, P.: MP—a multiple precision package. Comput. Phys. Commun. **59**, 345–358 (1990)
15. Lorenz, E.N.: Computational periodicity as observed in a simple system. Tellus **58A**, 549–557 (2006)