Advances in Difference Equations
a SpringerOpen Journal

RESEARCH Open Access

A new approach to estimating a numerical solution in the error embedded correction framework

Philsu Kim[1], Xiangfan Piao[2], WonKyu Jung[3] and Sunyoung Bu[4*]

*Correspondence:
syboo@hongik.ac.kr
[4]Departments of Liberal arts, Hongik University, Sejong, Korea
Full list of author information is available at the end of the article

Abstract

On the basis of the error correction method developed recently, an algorithm, so-called error embedded error correction method, is proposed for initial value problems. Two deferred equations are used to approximate the solution and the error, respectively, at each integration step. For the solution, the deferred equation, which is based on a modified Euler's polygon including the information of both the solution and its estimated error at the previous integration step, is solved with the classical fourth-order Runge–Kutta method. For the error, the deferred equation, which is based on a local Hermite cubic polynomial with three pieces of information—the solution, its estimated error at the previous step, and the constructed solution—is solved by the seventh-order Runge–Kutta–Fehlberg method. The constructed algorithm controls the error and possesses a good behavior of error bound in a long time simulation. Numerical experiments are presented to validate the proposed algorithm.

Keywords: Error correction method; Runge–Kutta method; Runge–Kutta–Fehlberg method; Long time simulation; Initial value problem

1 Introduction

There are many research topics [1, 2] in developing numerical methods for solving initial value problems (IVPs) described by

$$\frac{d\phi}{dt} = f\big(t, \phi(t)\big), \quad t \in [t_0, t_f]; \qquad \phi(t_0) = \phi_0, \tag{1}$$

where f has continuously bounded partial derivatives up to required order for the analysis of the developed numerical method. A long time simulation of the solution, which is needed in many physical problems (for example, a Hamiltonian system such as the Kepler problem, harmonic oscillator, molecular dynamics, etc.), is one of the most important topics in IVPs [3–5]. Also, such long time simulations sometimes demand very special step size selection to control the local truncation error. Most existing mechanisms of the explicit single step algorithm for solving (1) may be described by

$$\begin{cases} \phi_{m+1} = F(\phi_m), \\ e_{m+1} = G(\phi_m, \phi_{m+1}), \end{cases} \tag{2}$$

© The Author(s) 2018. This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

where F and G are functions derived from the numerical methods. Here, ϕ_{m+1} and e_{m+1} denote the approximations of the solution and the local truncation error E_{m+1}, respectively, at time t_{m+1}. For a pth order scheme, the estimated error e_{m+1} is usually approximated to fit only the coefficient of the $(p+1)$th order term in the expansion of E_{m+1} about the time step size h. The other important issue is to reduce the computational costs in a long time simulation, for which an efficient control scheme of the time step size is important (for example, Radau5). There have been several approaches related to those issues (for example, embedded Runge–Kutta formulae, adaptive time stepping, long time error estimation, etc. [6–11]).

In the existing schemes, the estimated error e_m obtained from the previous time step $[t_{m-1}, t_m]$ is mainly used only for choosing an appropriate next time step size in most algorithms. Also, the solution ϕ_{m+1} at time t_{m+1} is calculated with an initial value which is a solution ϕ_m at the previous time t_m. That is, ϕ_m is assumed to be the exact initial condition for ϕ_{m+1} despite the existence of the local truncation error e_m, which leads to accumulation of the error as the time is increasing. In order to control the accumulation error, smaller integration steps or special step size controllers are sometimes required, especially for a long time simulation or stiff systems. Nevertheless, the most existing methods cannot fully resolve the error control to get a given tolerance so that it is difficult to get reliable results at stringent tolerances (for example, see [12, 13]).

The subject of this paper is to develop a new integration scheme to control the accumulation error. As a remedy to control or minimize the accumulated error of e_m, we will embed it in the algorithm of the calculation scheme for ϕ_{m+1}. Further, we want to propose an estimating scheme for e_{m+1} which is correlated with three pieces of information ϕ_m, ϕ_{m+1}, and e_m. That is, the scheme we want to develop is an explicit single step algorithm, the so-called error embedded error correction method (EEECM), of the form

$$\begin{cases} \phi_{m+1} = F(\phi_m, e_m), \\ e_{m+1} = G(\phi_m, e_m, \phi_{m+1}). \end{cases} \tag{3}$$

To concretely describe the proposed algorithm, the classical 4th order Runge–Kutta (RK4) method and the well-known 7th order Runge–Kutta–Fehlberg (RKF7) method for calculating ϕ_{m+1} and e_{m+1}, respectively, will be used. Finally, we want to develop the EEECM having the accuracy order of 7 for the solution $\tilde{\phi}_{m+1} = \phi_{m+1} + e_{m+1}$. In particular, we will develop the efficient estimating algorithm for e_{m+1}, which fits the coefficients up to 7th order term in the expansion of the global error $E_{m+1} = \phi(t_{m+1}) - \phi_{m+1}$ about the time step size h.

An error correction method (ECM) is a widely used technique in many numerical scientific computations in general. The deferred correction methods (DCM) originally developed by Pereyra and Zadunaisky [14, 15] are the representative ECMs for solving (1). There are also extended results about the DCMs (for example, see [16–24]). These ECMs are based on the deferred equation of the form

$$\frac{d\psi(t)}{dt} = f\big(t, \psi(t) + x(t)\big) - x'(t), \tag{4}$$

where x is a local approximation of the solution defined on each integration step $[t_m, t_{m+1}]$. After solving (4), the solution $\phi(t)$ of (1) can be obtained by using the identity

$$\phi(t) = x(t) + \psi(t). \tag{5}$$

Two relations (4) and (5) enable us to develop the EEECM of the form (3) for solving (1). Practically, for the approximate solution ϕ_{m+1}, we use a local linear approximation x which uses the information of both the solution and its slope depending on the error e_m at time t_m and solve the deferred equation (4) with RK4. As mentioned above, we want to estimate the exact quantity of the error E_{m+1} up to the desired convergence order. To derive a formula for e_{m+1}, another local approximation x is constructed by a local Hermite cubic interpolation polynomial having all the information of the calculated solutions and those slopes at both time t_m and t_{m+1}. Based on the local approximation, we again solve the deferred equation (4) with the RKF7. As an appropriate step size controller, we exploit a standard step size controller to focus only on the EEECM for non-stiff problems. The constructed EEECM controls the error at each integration step, and it turns out that the proposed method possesses a good behavior of error bound in a long time simulation with a given tolerance. For an assessment of the effectiveness of the proposed algorithm, particularly its error bounds in a long time simulation, a simple harmonic oscillator problem with analytical solution and a hard error controlling problem are numerically solved. Finally, a two-body Kepler problem is also used to assess the efficiency of this algorithm. Throughout these numerical tests, it is shown that the proposed method is quite efficient compared to several existing methods.

This paper is organized as follows. In Sect. 2, we describe the methodology to formulate and control the solution and error formulas based on ECM. In Sect. 3, we give a concrete analysis of the convergence for the developed EEECM. Several numerical results are presented in Sect. 4 to give both the numerical evidences for the theoretical analysis and the numerical effectiveness of EEECM. Finally, in Sect. 5, a summary for EEECM and some discussion for further works are given.

2 Derivation of algorithm

In this section, we present the algorithm of EEECM based on the deferred equations. Let us assume that the approximated solution ϕ_m and the estimated error e_m for the solution $\phi(t_m)$ and the error E_m, respectively, at time t_m are already calculated. Then, as a local approximation of the solution $\phi(t)$ on the integration step $[t_m, t_{m+1}]$, one may consider the modified Euler's polygon $y(t)$ defined by

$$y(t) := \phi_m + (t - t_m) f(t_m, \phi_m + e_m), \quad t \in [t_m, t_{m+1}]. \tag{6}$$

Let $\psi(t)$ be the difference between $\phi(t)$ and $y(t)$ such that

$$\psi(t) := \phi(t) - y(t), \quad t \in [t_m, t_{m+1}]. \tag{7}$$

Differentiating both sides of (7) and combining the result with (1) and (6), one can see that the difference $\psi(t)$ satisfies the following deferred differential equation:

$$\begin{cases} \psi'(t) = g_1(t, \psi(t)), \quad t \in (t_m, t_{m+1}], \\ \psi(t_m) = E_m, \end{cases} \tag{8}$$

where g_1 is defined by

$$g_1\big(t, \psi(t)\big) := f\big(t, \psi(t) + y(t)\big) - y'(t) = f\big(t, \psi(t) + y(t)\big) - f(t_m, \phi_m + e_m). \tag{9}$$

Observe that the initial condition $\psi(t_m)$ of (8) is given by the unknown actual error E_m at time t_m, and hence problem (8) cannot be solved directly. Since e_m is assumed to be an estimated error of $E_m = \psi(t_m)$, instead of solving (8), it is natural to consider the following IVP:

$$\begin{cases} \theta'(t) = g_1(t, \theta(t)), \quad t \in (t_m, t_{m+1}], \\ \theta(t_m) = e_m \end{cases} \tag{10}$$

for an approximation of $\psi(t_{m+1})$. One may check that applying RK4 to (10) leads to

$$\begin{aligned} &\theta(t_{m+1}) \approx e_m + \frac{h}{6}[-5\nu_1 + 2\nu_2 + 2\nu_3 + \nu_4], \\ &\nu_1 = f(t_m, \tilde{\phi}_m), \qquad \nu_2 = f\left(t_m + \frac{h}{2}, \tilde{\phi}_m + \frac{h}{2}\nu_1\right), \\ &\nu_3 = f\left(t_m + \frac{h}{2}, \tilde{\phi}_m + \frac{h}{2}\nu_2\right), \qquad \nu_4 = f(t_m + h, \tilde{\phi}_m + h\nu_3), \end{aligned} \tag{11}$$

where $\tilde{\phi}_m := \phi_m + e_m$. Combining approximation (11) with (6) and (7), one may get an approximation formula for $\phi(t_{m+1})$ as follows.

$$\phi_{m+1} := \tilde{\phi}_m + \frac{h}{6}[\nu_1 + 2\nu_2 + 2\nu_3 + \nu_4], \tag{12}$$

where the intermediate values ν_i are defined by (11). Note that the classical RK4 uses only the approximate value ϕ_m at time t_m to calculate ϕ_{m+1}, whereas algorithm (12) uses the value $\tilde{\phi}_m := \phi_m + e_m$ instead of ϕ_m, which is a remarkable difference compared to the RK4.

Since the estimated error e_m at time t_m is embedded in algorithm (12), a recursive relation for a sequence $\{e_m\}$ is needed to complete the algorithm. We try to derive this relation using another deferred equation together with an appropriate local approximation. Recall that after the calculation of (12), one can use the information of both the approximate solutions and those slopes at time t_m and t_{m+1}. Hence, as the local approximation, it is natural to use a local Hermite cubic interpolation such that

$$x(t) = a_0 + a_1(t - t_m) + a_2(t - t_m)^2 + a_3(t - t_m)^2(t - t_{m+1}) \tag{13}$$

satisfying $x(t_m) = \tilde{\phi}_m$, $x'(t_m) = f(t_m, \tilde{\phi}_m)$, $x(t_{m+1}) = \phi_{m+1}$, and $x'(t_{m+1}) = f(t_{m+1}, \phi_{m+1})$. Then it solves [25]

$$\begin{aligned} x(t) = {} & x(t_m) + x'(t_m)(t - t_m) + \frac{x(t_{m+1}) - x(t_m) - x'(t_m)h}{h^2}(t - t_m)^2 \\ & + \frac{(x'(t_{m+1}) + x'(t_m))h - 2(x(t_{m+1}) - x(t_m))}{h^3}(t - t_m)^2(t - t_{m+1}). \end{aligned} \tag{14}$$

Let $\psi(t)$ be the difference between $\phi(t)$ and $x(t)$ such that

$$\psi(t) := \phi(t) - x(t), \quad t \in [t_m, t_{m+1}]. \tag{15}$$

As the derivation of (8), one can see that the difference $\psi(t)$ defined by (15) satisfies the following deferred differential equation:

$$\begin{cases} \psi'(t) = g_2(t, \psi(t)), \quad t \in (t_m, t_{m+1}], \\ \psi(t_m) = E_m - e_m, \end{cases} \tag{16}$$

where g_2 is defined by

$$g_2\big(t, \psi(t)\big) := f\big(t, \psi(t) + x(t)\big) - x'(t). \tag{17}$$

Observe that the initial condition $\psi(t_m)$ of (16) contains the unknown value E_m and hence problem (16) cannot be solved directly. Since e_m is the estimated error of E_m, if one assumes that it is well approximated, then the initial value $\psi(t_m)$ becomes quite small. Hence, instead of solving (16), it is natural to consider the following IVP:

$$\begin{cases} \theta'(t) = g_2(t, \theta(t)), \quad t \in (t_m, t_{m+1}], \\ \theta(t_m) = 0 \end{cases} \tag{18}$$

for an approximation of $\psi(t_{m+1})$. To solve (18), we consider the well-known RKF7 with Butcher array [26]

$$\begin{array}{c|c} \mathbf{c} & \mathcal{A} \\ \hline & \mathbf{b}^T \end{array}, \tag{19}$$

where

$$\begin{aligned} \mathbf{c} &= [c_1, c_2, \ldots, c_{11}]^T := \left[0, \frac{2}{27}, \frac{1}{9}, \frac{1}{6}, \frac{5}{12}, \frac{1}{2}, \frac{5}{6}, \frac{1}{6}, \frac{2}{3}, \frac{1}{3}, 1\right]^T, \\ \mathbf{b} &= [b_1, b_2, \ldots, b_{11}]^T := \left[\frac{41}{840}, 0, 0, 0, 0, \frac{34}{105}, \frac{9}{35}, \frac{9}{35}, \frac{9}{280}, \frac{9}{280}, \frac{41}{840}\right]^T, \end{aligned} \tag{20}$$

$$\mathcal{A} = (\alpha_{i,j}) := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{2}{27} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{36} & \frac{1}{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{24} & 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{5}{12} & 0 & -\frac{25}{16} & \frac{25}{16} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{20} & 0 & 0 & \frac{1}{4} & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 \\ -\frac{25}{108} & 0 & 0 & \frac{125}{108} & -\frac{65}{27} & \frac{125}{54} & 0 & 0 & 0 & 0 \\ \frac{31}{300} & 0 & 0 & 0 & \frac{61}{225} & -\frac{2}{9} & \frac{13}{900} & 0 & 0 & 0 \\ 2 & 0 & 0 & -\frac{53}{6} & \frac{704}{45} & -\frac{107}{9} & \frac{67}{90} & 3 & 0 & 0 \\ -\frac{91}{108} & 0 & 0 & \frac{23}{108} & -\frac{976}{135} & \frac{311}{54} & -\frac{19}{60} & \frac{17}{6} & -\frac{1}{12} & 0 \\ \frac{2383}{4100} & 0 & 0 & -\frac{341}{164} & \frac{4496}{1025} & -\frac{301}{82} & \frac{2133}{4100} & \frac{45}{82} & \frac{45}{164} & \frac{18}{41} \end{bmatrix}.$$

Since $\theta(t_m) = 0$, by applying the RKF7 to problem (18) and using (17), $\theta(t_{m+1})$ can be approximated as

$$\begin{aligned} &\theta(t_{m+1}) \approx h \sum_{i=2}^{11} b_i K_i, \\ &K_i = g_2\left(t_m + c_i h, h \sum_{j=2}^{i-1} \alpha_{i,j} K_j\right) \\ &\quad = f\left(t_m + c_i h, h \sum_{j=2}^{i-1} \alpha_{i,j} K_j + x(t_m + c_i h)\right) - x'(t_m + c_i h), \quad i = 2, \dots, 11. \end{aligned} \tag{21}$$

From the definition of the Hermite interpolation x defined by (14), one may see that $\psi(t_{m+1}) = \phi(t_{m+1}) - \phi_{m+1} := E_{m+1}$. Also, we recall that system (18) is a perturbed system from IVP (16). Thus, one may take the approximation of $\theta(t_{m+1})$ given in (21) as an estimated error e_{m+1} for the actual error E_{m+1}. That is, we define

$$e_{m+1} = h \sum_{i=2}^{11} b_i K_i, \tag{22}$$

where K_i are defined by (21).

It is easy to check that the coefficients (20) in Butcher array (19) have the following identities:

$$\begin{aligned} &\sum_{j=1}^{i-1} \alpha_{i,j} = c_i, \quad i = 1, \dots, 11, \\ &\sum_{j=1}^{i-1} \alpha_{i,j} c_j = \frac{c_i^2}{2}, \qquad \sum_{j=1}^{i-1} \alpha_{i,j} c_j^2 = \frac{c_i^3}{3}, \quad i = 3, \dots, 11, \\ &\sum_{j=1}^{11} b_j = 1, \qquad \sum_{j=1}^{11} b_j c_j = \frac{1}{2}, \qquad \sum_{j=1}^{11} b_j c_j^2 = \frac{1}{3}. \end{aligned} \tag{23}$$

Using these identities, one may prove the following lemma.

Lemma 1 *The algorithm for e_{m+1} defined by* (22) *can be simplified by*

$$e_{m+1} = \phi_m + e_m - \phi_{m+1} + h\sum_{i=1}^{11} b_i V_i, \tag{24}$$

where the intermediate values V_i are defined by

$$\begin{aligned} &V_1 := v_1, \qquad V_2 := f\big(t_m + c_2 h, x(t_m + c_2 h)\big), \\ &V_i := f\left(t_m + c_i h, \phi_m + e_m + h\sum_{j=1}^{i-1} \alpha_{i,j} V_j\right), \quad i = 3, \dots, 11, \end{aligned} \tag{25}$$

where v_1 is defined by (11).

Proof For the quantity K_i defined by (21), we let

$$\Gamma_i := K_i + x'(t_m + c_i h), \quad i = 2, 3, \dots, 11.$$

Then algorithm (22) can be written as

$$\begin{aligned} e_{m+1} &= \gamma + h\sum_{i=2}^{11} b_i \Gamma_i, \\ \Gamma_2 &= f\big(t_m + c_2 h, x(t_m + c_2 h)\big), \\ \Gamma_i &= f\left(t_m + c_i h, h\sum_{j=2}^{i-1} \alpha_{i,j}\Gamma_j + \beta_i\right), \quad i = 3, \dots, 11, \end{aligned} \tag{26}$$

where γ and β_i are defined by

$$\begin{aligned} \gamma &:= -h\sum_{i=2}^{11} b_i x'(t_m + c_i h), \\ \beta_i &:= -h\sum_{j=2}^{i-1} \alpha_{i,j} x'(t_m + c_j h) + x(t_m + c_i h). \end{aligned} \tag{27}$$

For a simplification of γ and β_i defined in (27), we consider Taylor's expansion of x about $t = t_m$ given by

$$x(t) = x(t_m) + (t - t_m)x'(t_m) + \frac{(t - t_m)^2}{2}x''(t_m) + \frac{(t - t_m)^3}{6}x^{(3)}(t_m). \tag{28}$$

By substituting (28) into the formula of γ given in (27) and combining the result with (23) and (28), one may check that

$$\begin{aligned} \gamma &= hb_1 x'(t_m) - hx'(t_m) - \frac{h^2}{2}x''(t_m) - \frac{h^3}{6}x^{(3)}(t_m) \\ &= hb_1 x'(t_m) + x(t_m) - x(t_{m+1}) \end{aligned} \tag{29}$$

and

$$\beta_i = x(t_m) + h\alpha_{i,1}x'(t_m), \quad i = 3, \dots, 11. \tag{30}$$

Hence, substituting (29) and (30) into (26) and considering the definition of V_i defined by (25), one can complete the proof. □

Remark 1 Remark that 16 evaluations of the Hermite interpolation and its derivatives are required for algorithm (22). However, by introducing Lemma 1, only one evaluation of the Hermite interpolation is required. It is remarkable.

For summarizing the algorithm we discussed, we consider the Butcher array of RK4 given by

$$\frac{\mathbf{n}\big|\mathcal{S}}{\big|\mathbf{k}}, \tag{31}$$

where

$$\begin{aligned} &\mathbf{n} = [n_1, n_2, n_3, n_4] := \left[0, \frac{1}{2}, \frac{1}{2}, 1\right], \qquad \mathbf{k} = [k_1, k_2, k_3, k_4] := \left[\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6}\right], \\ &\mathcal{S} = (s_{i,j}) := \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \end{aligned} \tag{32}$$

Also, if we define

$$V_0 := f(t_{m+1}, \phi_{m+1}),$$

then, from the definition of x given in (14), we have

$$\begin{aligned} x(t_m + c_2h) = {} & \phi_m + e_m + (\phi_{m+1} - \phi_m - e_m)c_2^2(3 - 2c_2) \\ & + c_2(1 - c_2)h\big((1 - c_2)V_1 - c_2V_0\big). \end{aligned}$$

Thus, by combining it with (24) and (12), one can derive the algorithm EEECM given by

$$\begin{cases} \phi_{m+1} = \phi_m + e_m + h\sum_{i=1}^{4} k_i\nu_i, \\ e_{m+1} = \phi_m + e_m - \phi_{m+1} + h\sum_{i=1}^{11} b_iV_i, \end{cases} \quad m \geq 0, \tag{33}$$

where the intermediate values ν_i and V_i are defined by

$$\begin{aligned} &\nu_1 = f(t_m, \phi_m + e_m), \qquad \nu_i = f(t_m + n_ih, \phi_m + e_m + hs_{i,i-1}\nu_{i-1}), \quad i = 2, 3, 4, \\ &V_0 = f(t_{m+1}, \phi_{m+1}), \qquad V_1 = \nu_1, \\ &V_2 = f\big(t_m + c_2h, \phi_m + e_m + c_2^2(3 - 2c_2)(\phi_{m+1} - \phi_m - e_m)\big) \end{aligned} \tag{34}$$

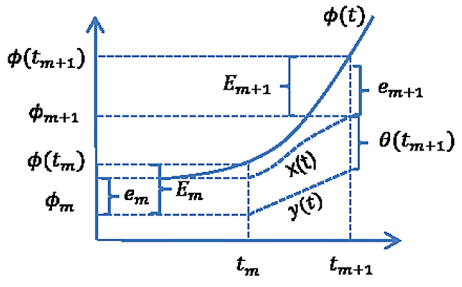


Figure 1 Geometric meaning of the error embedded error correction methods

$$+ c_2(1-c_2)h\big((1-c_2)V_1 - c_2V_0\big)\big),$$
$$V_i = f\left(t_m + c_ih, \phi_m + e_m + h\sum_{j=1}^{i-1}\alpha_{i,j}V_j\right), \quad i = 3,\dots,11.$$

Also, if we let $\tilde{\phi}_m := \phi_m + e_m$, then one may get a better approximation $\{\tilde{\phi}_m\}$ than the approximation $\{\phi_m\}$, and it satisfies the following recurrence relation:

$$\tilde{\phi}_{m+1} = \tilde{\phi}_m + h\sum_{i=1}^{11} b_iV_i, \quad m \geq 0, \tag{35}$$

where the intermediate values V_i are calculated by

$$\begin{aligned}
&v_1 = f(t_m, \tilde{\phi}_m), \qquad v_i = f(t_m + n_ih, \tilde{\phi}_m + hs_{i,i-1}v_{i-1}), \quad i = 2,3,4,\\
&V_0 = f\left(t_{m+1}, \tilde{\phi}_m + h\sum_{i=1}^{4} k_iv_i\right), \qquad V_1 = v_1,\\
&V_2 = f\left(t_m + c_2h, \tilde{\phi}_m + c_2^2(3-2c_2)h\sum_{i=1}^{4} k_iv_i + c_2(1-c_2)h\big((1-c_2)V_1 - c_2V_0\big)\right),\\
&V_i = f\left(t_m + c_ih, \tilde{\phi}_m + h\sum_{j=1}^{i-1}\alpha_{i,j}V_j\right), \quad i = 3,\dots,11.
\end{aligned} \tag{36}$$

Remark 2 The algorithm of EEECM (33) (or (35)) needs 15 function evaluations in each time step, which is two more function evaluations than those of RKF78. Unlike RKF78, not only is the estimated error sequence $\{e_m\}$ embedded to calculate the solution ϕ_m, but also it will be used to control the time step size. This is the reason why we call the proposed algorithm an error embedded error correction method. We also remark that scheme (33) (or (35)) is also applicable to a system of ODEs of the form

$$\Phi'(t) = \mathfrak{F}\big(t, \Phi(t)\big), \quad t \in (t_0, t_f]; \qquad \Phi(t_0) = \Phi_0,$$

where $\Phi := [\phi_1(t),\dots,\phi_d(t)]^T$ and $\mathfrak{F} := [f_1(t,\Phi(t)),\dots,f_d(t,\Phi(t))]^T$.

Remark 3 (Geometric interpretation) A geometric meaning of EEECM is interpreted in Fig. 1, which consists of two steps: the first step calculates the approximated solution ϕ_{m+1}

at t_{m+1} based on the deferred equation constructed by the Euler polygon $y(t)$, for which two pieces of information ϕ_m and e_m calculated at time t_m are used. To complete the algorithm, a scheme embedding the sequence of the estimated error e_m into the algorithm itself is required. Therefore, in the second step, the local truncation error E_{m+1} is estimated with another deferred equation based on higher order local approximation $x(t)$, for which all pieces of information ϕ_m, ϕ_{m+1}, and e_m are used.

3 Convergence analysis

The aim of this section is to give a concrete convergence analysis for algorithm (35). For the simplicity of the analysis, we assume that IVP (1) is an autonomous problem. That is, we assume $f(t,\phi(t)) := f(\phi(t))$. For a simplification, we introduce the operator D^k defined by

$$D^k f(y) := f\frac{\partial}{\partial y}\big(D^{k-1}f(y)\big), \quad k \geq 1, \tag{37}$$

where $D^0 f(y) := f(y)$. Let $F(y,h;f)$ be a function of the form

$$\begin{aligned}
&F(y,h;f) := \sum_{i=1}^{11} b_i V_i = b_1 V_1 + \sum_{i=6}^{11} b_i V_i,\\
&v_1 = f(y), \qquad v_i = f(y + hs_{i,i-1}v_{i-1}), \quad i = 2,3,4,\\
&V_0 = f\left(y + h\sum_{i=1}^{4} k_i v_i\right), \qquad V_1 = v_1,\\
&V_2 = f\left(y + h\left[c_2^2(3-2c_2)\sum_{i=1}^{4} k_i v_i + c_2(1-c_2)\big((1-c_2)V_1 - c_2 V_0\big)\right]\right),\\
&V_i = f\left(y + h\sum_{j=1}^{i-1}\alpha_{i,j}V_j\right), \quad i = 3,\dots,11,
\end{aligned} \tag{38}$$

where k_i and $(s_{i,j})$ are defined in (32) and c_i and $(\alpha_{i,j})$ are defined in (20). Note that, by Taylor's expansion of $f(y + hv)$ about y, one may rewrite V_i of (38) by

$$V_i = \sum_{k=0}^{\infty}\frac{f^{(k)}(y)}{k!}h^k X_i^k, \quad i = 3,\dots,11, \tag{39}$$

where

$$X_i := \sum_{j=1}^{i-1}\alpha_{i,j}V_j, \quad i = 3,\dots,11. \tag{40}$$

The above two relations (39) and (40) give a simple expansion of X_i as follows.

Lemma 2 *For the quantities X_i ($i \geq 4$) defined by* (40), *we have*

$$X_i = \sum_{k=0}^{5}\frac{h^k}{k!}D^k f\big(\mathcal{A}\mathbf{c}^k\big)_i + f'\sum_{k=4}^{5}\frac{h^k}{(k-1)!}D^{k-1}f\left(\mathcal{A}^2\mathbf{c}^{k-1} - \frac{1}{k}\mathcal{A}\mathbf{c}^k\right)_i$$

$$+\frac{h^5}{3!}D^3f\left(f''f\left(\mathcal{A}\left(\mathbf{c}\cdot\mathcal{A}\mathbf{c}^3\right)-\frac{1}{4}\mathcal{A}\mathbf{c}^5\right)+\left(f'\right)^2\left(\mathcal{A}^3\mathbf{c}^3-\frac{1}{4}\mathcal{A}^2\mathbf{c}^4\right)\right)_i+\mathcal{O}\left(h^6\right), \quad (41)$$

where all the functions on the right-hand side are evaluated at the value y and $(\mathbf{a})_i$ *denotes the ith component of a vector* $\mathbf{a}$. *Here,* $\mathbf{c}^0 := [1,1,\dots,1]^T$ *and also a multiplication between two vectors* $\mathbf{a} := [a_1,\dots,a_{11}]^T$ *and* $\mathbf{b} := [b_1,\dots,b_{11}]^T$ *is defined by* $\mathbf{a}\cdot\mathbf{b} := [a_1b_1,\dots,a_{11}b_{11}]^T$ *and* $\mathbf{a}^k := \mathbf{a}\cdot\mathbf{a}^{k-1}$.

Proof For the value X_i defined on (40), let us define a vector $\mathbf{X}$ by $\mathbf{X} := [X_1,\dots,X_{11}]^T$ with $X_1 = X_2 = 0$. Then, from the definition of the matrix $\mathcal{A}$ of (20), combining (40) with (39) and the identity $\sum_{j=1}^{i-1}\alpha_{i,j} = c_i$ of (23) yields

$$\begin{aligned} X_i &= f\alpha_{i,1}+\alpha_{i,2}V_2+f\sum_{j=3}^{i-1}\alpha_{i,j}+\sum_{k=1}^{5}\frac{f^{(k)}}{k!}h^k\sum_{j=3}^{i-1}\alpha_{i,j}X_j^k+\mathcal{O}\left(h^6\right)\\ &= f\mathbf{c}_i+\sum_{k=1}^{5}\frac{f^{(k)}}{k!}h^k\left(\mathcal{A}\mathbf{X}^k\right)_i+\mathcal{O}\left(h^6\right), \quad i\geq 4, \end{aligned} \quad (42)$$

where $\alpha_{i,2} = 0$ $(i \geq 4)$ is used in the above second equality and the power $\mathbf{X}^k$ is obtained by the above vector multiplication. To obtain a series expansion of X_i in terms of h, we let

$$\mathbf{X} := \sum_{k=0}^{5}h^k\mathbf{a}_k+\mathcal{O}\left(h^6\right) \quad (43)$$

and substitute it into (42). Here, we may assume $(\mathbf{a}_k)_i = 0$ $(i = 1, 2)$ and $(\mathbf{a}_k)_3$ are determined by Taylor's expansion of X_3 in terms of h. Further, we expand the resulted equation in ascending order of h. Then one may check that for $i \geq 4$,

$$\begin{aligned} X_i &= fc_i+hf'(\mathcal{A}\mathbf{a}_0)_i+h^2\left(f'\mathcal{A}\mathbf{a}_1+\frac{f''}{2}\mathcal{A}\mathbf{a}_0^2\right)_i+h^3\left(f'\mathcal{A}\mathbf{a}_2+f''\mathcal{A}(\mathbf{a}_0\cdot\mathbf{a}_1)+\frac{f^{(3)}}{3!}\mathcal{A}\mathbf{a}_0^3\right)_i\\ &\quad+h^4\left(f'\mathcal{A}\mathbf{a}_3+\frac{f''}{2}\mathcal{A}\left(\mathbf{a}_1^2+2\mathbf{a}_0\cdot\mathbf{a}_2\right)+\frac{f^{(3)}}{2}\mathcal{A}\left(\mathbf{a}_0^2\cdot\mathbf{a}_1\right)+\frac{f^{(4)}}{4!}\mathcal{A}\mathbf{a}_0^4\right)_i\\ &\quad+h^5\Bigg(f'\mathcal{A}\mathbf{a}_4+f''\mathcal{A}(\mathbf{a}_1\cdot\mathbf{a}_2+\mathbf{a}_0\cdot\mathbf{a}_3)\\ &\quad+\frac{f^{(3)}}{2}\mathcal{A}\left(\mathbf{a}_0\cdot\mathbf{a}_1^2+\mathbf{a}_0^2\cdot\mathbf{a}_2\right)+\frac{f^{(4)}}{3!}\mathcal{A}\left(\mathbf{a}_0^3\cdot\mathbf{a}_1\right)+\frac{f^{(5)}}{5!}\mathcal{A}\mathbf{a}_0^5\Bigg)_i+\mathcal{O}\left(h^6\right). \end{aligned} \quad (44)$$

Thus, by comparing the coefficients of two equations (43) and (44), one may have the following recurrence relations for $\mathbf{a}_i$:

$$\begin{aligned} &(\mathbf{a}_0)_i = f(\mathbf{c})_i, \qquad (\mathbf{a}_1)_i = f'(\mathcal{A}\mathbf{a}_0)_i, \qquad (\mathbf{a}_2)_i = \left(f'\mathcal{A}\mathbf{a}_1+\frac{f''}{2}\mathcal{A}\mathbf{a}_0^2\right)_i,\\ &(\mathbf{a}_3)_i = \left(f'\mathcal{A}\mathbf{a}_2+f''\mathcal{A}(\mathbf{a}_0\cdot\mathbf{a}_1)+\frac{f^{(3)}}{3!}\mathcal{A}\mathbf{a}_0^3\right)_i,\\ &(\mathbf{a}_4)_i = \left(f'\mathcal{A}\mathbf{a}_3+\frac{f''}{2}\mathcal{A}\left(\mathbf{a}_1^2+2\mathbf{a}_0\cdot\mathbf{a}_2\right)+\frac{f^{(3)}}{2}\mathcal{A}\left(\mathbf{a}_0^2\cdot\mathbf{a}_1\right)+\frac{f^{(4)}}{4!}\mathcal{A}\mathbf{a}_0^4\right)_i, \end{aligned} \quad (45)$$

$$(\mathbf{a}_5)_i = \left(f'\mathcal{A}\mathbf{a}_4 + f''\mathcal{A}(\mathbf{a}_1 \cdot \mathbf{a}_2 + \mathbf{a}_0 \cdot \mathbf{a}_3) + \frac{f^{(3)}}{2}\mathcal{A}\left(\mathbf{a}_0 \cdot \mathbf{a}_1^2 + \mathbf{a}_0^2 \cdot \mathbf{a}_2\right) + \frac{f^{(4)}}{3!}\mathcal{A}\left(\mathbf{a}_0^3 \cdot \mathbf{a}_1\right) + \frac{f^{(5)}}{5!}\mathcal{A}\mathbf{a}_0^5 \right)_i, \quad i \geq 4.$$

Finally, we solve the recurrence relations (45) with the aid of the relations in (23) and (37). Then one may get the required identity in (41). □

From equation (39) together with (41) in the above lemma, we have the following corollary.

Corollary 1 *For the intermediate values* V_i $(i \geq 6)$ *defined in* (38), *we have*

$$\begin{aligned} V_i = {} & \sum_{k=0}^{6} \frac{h^k}{k!} D^k f(y)\mathbf{c}_i^k + \frac{h^5}{4!} f' D^4 f(y)\left(\mathcal{A}\mathbf{c}^4 - \frac{\mathbf{c}^5}{5}\right)_i \\ & + h^6\left(\left(f'(y)\right)^2 D^4 f(y)\left(\mathcal{A}^2\mathbf{c}^4 - \frac{\mathcal{A}\mathbf{c}^5}{5}\right)_i + \frac{f'(y)}{5!} D^5 f(y)\left(\mathcal{A}\mathbf{c}^5 - \frac{\mathbf{c}^6}{6}\right)_i \right. \\ & \left. + \frac{f''(y)f(y)}{4!} D^4 f(y)\left(\mathbf{c} \cdot \mathcal{A}\mathbf{c}^4 - \frac{\mathbf{c}^6}{5}\right)_i + \mathcal{O}\left(h^7\right). \right. \end{aligned} \tag{46}$$

Proof By directly substituting (41) into (39) and expanding the resulted equation in ascending order of h with the aid of the identity $(\mathcal{A}\mathbf{c}^3)_i = \frac{\mathbf{c}_i^4}{4}$, $i \geq 6$, one may get the required equation (46). □

Substituting expansion (46) into the sum of F defined by (38) leads to the following theorem.

Theorem 1 *Let us assume that the slope function* f *is sufficiently smooth. Then the function* F *defined by* (38) *satisfies*

$$F(y,h;f) = \sum_{k=0}^{6} \frac{h^k}{(k+1)!} D^k f(y) + \mathcal{O}\left(h^7\right). \tag{47}$$

Proof By substituting the expansion of V_i in (46) into the sum of F in (38) and simplifying the result, one may get

$$\begin{aligned} F(y,h;f) = {} & f(y)\sum_{i=1}^{11} b_i + \sum_{k=1}^{6} \frac{D^k f(y)}{k!} h^k \left(\sum_{i=6}^{11} b_i c_i^k\right) \\ & + \frac{h^5}{4!} f'(y) D^4 f(y) \sum_{i=6}^{11} b_i \left(\mathcal{A}\mathbf{c}^4 - \frac{1}{5}c_i^5\right)_i \\ & + h^6 \left[\left(f'(y)\right)^2 D^4 f(y) \sum_{i=6}^{11} b_i \left(\mathcal{A}^2\mathbf{c}^4 - \frac{1}{5}\mathcal{A}\mathbf{c}^5\right)_i \right. \\ & \left. + \frac{f'(y)}{5!} D^5 f(y) \sum_{i=6}^{11} b_i \left(\mathcal{A}\mathbf{c}^5 - \frac{1}{6}c^6\right)_i \right. \end{aligned}$$

$$+\frac{f''f}{4!}D^4f(y)\sum_{i=6}^{11}b_i\left(c_i\left(\mathcal{A}\mathbf{c}^4\right)_i-\frac{1}{5}c_i^6\right)\Bigg]+\mathcal{O}\left(h^7\right). \tag{48}$$

From the Butcher array (19) with the coefficients in (20), one may check that

$$\begin{aligned}
&\sum_{i=6}^{11}b_i\left(\mathcal{A}\mathbf{c}^4\right)_i=\frac{1}{5}\sum_{i=6}^{11}b_ic_i^5, \qquad \sum_{i=6}^{11}b_i\left(\mathcal{A}^2\mathbf{c}^4\right)_i=\frac{1}{5}\sum_{i=6}^{11}b_i\left(\mathcal{A}\mathbf{c}^5\right)_i,\\
&\sum_{i=6}^{11}b_i\left(\mathcal{A}\mathbf{c}^5\right)_i=\frac{1}{6}\sum_{i=6}^{11}b_ic_i^6, \qquad \sum_{i=6}^{11}b_ic_i\left(\mathcal{A}\mathbf{c}^4\right)_i=\frac{1}{5}\sum_{i=6}^{11}b_ic_i^6,\\
&\sum_{i=6}^{11}b_ic_i^k=\frac{1}{k+1}, \qquad \sum_{i=1}^{11}b_i=1.
\end{aligned} \tag{49}$$

Combining the relations in (49) with equation (48) yields the required equation (47). □

For a concrete convergence analysis of scheme (35), similar to the methodology in [25], we now define the truncation error by

$$T_m(\phi):=\phi(t_{m+1})-\phi(t_m)-hF\left(\phi(t_m),h;f\right), \quad m\geq 0, \tag{50}$$

and define $\tau_m(\phi)$ implicitly by

$$T_m(\phi)=h\tau_m(\phi). \tag{51}$$

Then two equations (50) and (51) give

$$\phi(t_{m+1})=\phi(t_m)+hF\left(\phi(t_m),h;f\right)+h\tau_m(\phi), \quad m\geq 0. \tag{52}$$

Thus, subtract (35) from (52) together with (38) to obtain

$$\tilde{E}_{m+1}=\tilde{E}_m+h\left[F\left(\phi(t_m),h;f\right)-F(\tilde{\phi}_m,h;f)\right]+h\tau_m(\phi), \tag{53}$$

in which $\tilde{E}_m:=\phi(t_m)-\tilde{\phi}_m$. For the simplicity of the convergence analysis, we now assume that the function F satisfies a Lipschitz condition

$$\left|F(y,h;f)-F(z,h;f)\right|\leq L|y-z| \tag{54}$$

for all $-\infty<y,z<\infty$ and all small $h>0$. This condition can be usually obtained by using the Lipschitz condition on f and its derivatives. Applying the Lipschitz condition (54) into (53) leads to

$$|\tilde{E}_{m+1}|\leq(1+Lh)|\tilde{E}_m|+h\tau(h), \quad m\geq 0, \tag{55}$$

where $\tau(h)$ is defined by

$$\tau(h)=\max_{m\geq 0}\left|\tau_m(\phi)\right|. \tag{56}$$

On the other hand, from Taylor's expansion of $\phi(t_{m+1})$ about t_m and two equations (47) and (50), one may get

$$\tau_m(\phi) = \mathcal{O}\big(h^7\big), \quad m \geq 0, \tag{57}$$

for a sufficiently smooth function f. Hence, from the above three relations (55), (56), and (57), one can get the following convergence theorem for algorithm (35).

Theorem 2 *Assume that the present method* (35) *satisfies the Lipschitz condition* (54) *and the slope function f is sufficiently smooth. Then, for the IVP* (1), *algorithm* (35) *has the rate of convergence $\mathcal{O}(h^7)$.*

Remark 4 Theorem 2 shows that the estimated error e_{m+1} in algorithm (33) exactly estimates the coefficients of Taylor's expansion about h of the error $E_{m+1} := \phi(t_{m+1}) - \phi_{m+1}$ up to the 7th order term, whereas the embedded RKF78 exactly estimates the 8th order term only. Also, unlike the existing embedded schemes, the estimated error e_m is embedded in the algorithm EEECM itself by considering as an initial value at each time interval. It turns out that the proposed algorithm (33) is more efficient in a long time simulation, which is shown throughout several numerical results (see Sect. 4).

4 Numerical results

In this section, we show several numerical results and compare the efficiency of the proposed method to those of other existing methods such as BV78, RKF78, Radau5, and Matlab built-in routines—ode113 and ode45 [2, 26, 27]. As a time step control for the proposed method, we use a standard step size selection algorithm (for example, [1, 28]) which is given by

$$h_{m+1} = \left(\frac{tol}{\|e_m\|_\infty}\right)^{\frac{1}{5}} h_m, \tag{58}$$

where tol is a given tolerance and e_m is the estimated local truncation error at time t_m calculated by (33). Also, the initial time step size is chosen by $h_0 = \frac{1}{4}(tol)^{\frac{1}{5}}$, since RK4 is used to approximate the solution. In each test problem, we calculate both errors $E_m = \phi(t_m) - \phi_m$ and $\tilde{E}_m = \phi(t_m) - \phi_m - e_m$ denoted by EEECM and EEECM(e), respectively.

4.1 Simple problems

In this subsection, we will show the efficiency of EEECM with two simple IVPs. One is a well-known simple harmonic oscillator. The other is knowing that the global error control is quite difficult [12]. Details of each problem will be explained in each subsection.

Example 1 Consider a harmonic oscillator described by

$$\begin{cases} y_1'(t) = -y_2(t), \\ y_2'(t) = y_1(t), \end{cases} \tag{59}$$

whose analytic solutions are given by $[y_1(t), y_2(t)] = [\cos(t), \sin(t)]$.

Table 1 Convergence order of EEECM for solving a simple harmonic oscillator

Step-size	Error	Rate
0.5	2.7007e−006	
0.25	1.8878e−008	7.160437
0.125	1.3484e−010	7.129298
0.0625	9.9618e−013	7.080680
0.03125	7.0429e−015	7.144086

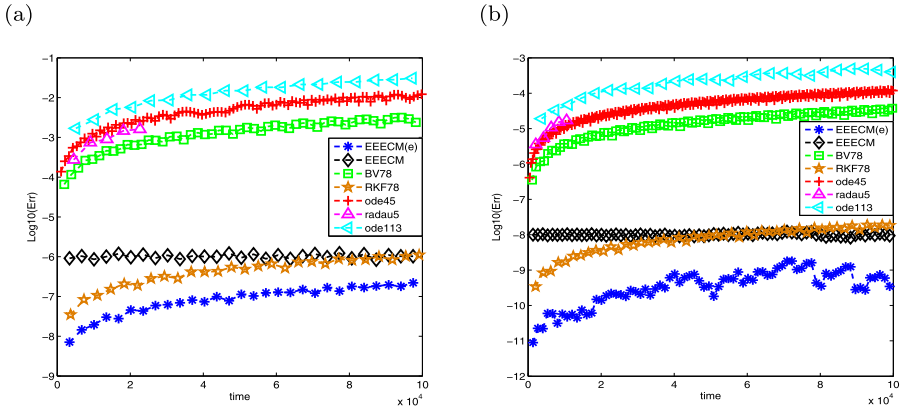


Figure 2 Comparison of the error with different tolerances (**a**) *tol* = 1e−6 (**b**) *tol* = 1e−8

To validate the theoretical convergence analysis in Theorem 2, the problem is solved on the interval $[0, 500]$ with different step sizes and the results are reported in Table 1. The first column shows time step sizes, the second does the errors measured by sup norm at the final time, and the last gives the rates between the errors generated by using the previous and the current step sizes. The results show that the numerical convergence order is 7, which validates theoretical convergence order.

For a demonstration of a long time simulation of EEECM, we solve the problem on the interval $[0, 10^5]$ and show how solutions and errors are well calculated. In Fig. 2, we plot the absolute errors in a log scale calculated by various numerical schemes with two different tolerances (a) 1e−6 and (b) 1e−8. It can be seen that all existing methods have the exponential growth of the error in the sense that the errors over time are increasing linearly up in a log scale. On the other hand, the figures of EEECM have uniform-like error bound during the whole time interval under the given tolerances. Furthermore, the results of EEECM(e) are superior to those of the existing methods. These remarkable results may contribute to many other fields which stood in needs of long-term simulations.

Finally, we calculate the time cost required to obtain a desired accuracy by varying tolerances from *tol* = 1e−5 to *tol* = 1e−10. In Fig. 3, we plot the numerical absolute errors at the final time (y-axis) corresponding to the given tolerances versus the demanded CPU time (x-axis). The numerical results show that the proposed scheme obtains the most accurate solution for each fixed CPU time. In particular, one can see that the proposed method achieves the required accuracies within the given tolerances, whereas all existing methods except for RKF78 fail to achieve this requirement. Also, EEECM(e) is comparable to RKF78 in the sense of the CPU time and accuracy for given tolerances. We therefore

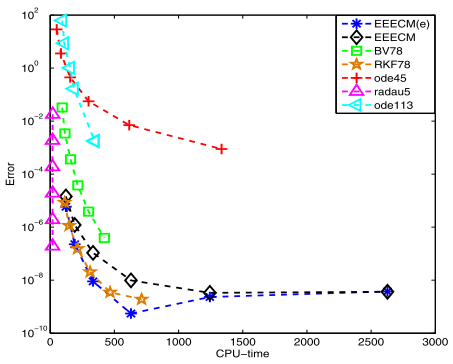


Figure 3 Comparison of errors versus CPU-times for given tolerances from 1e–5 to 1e–10

conclude that the proposed method is the most efficient scheme in view of the above discussion, restricted to this harmonic oscillator problem.

Example 2 In this example, we test a system that the global error control task becomes more difficult [12] as the time goes on. The system consists of four equations given by

$$\begin{cases} y_1' = 2ty_2{}^{1/5}y_4, \\ y_2' = 10t\exp(5(y_3-1))y_4, \\ y_3' = 2ty_4, \\ y_4' = -2t\log(y_1) \end{cases} \tag{60}$$

defined on the interval $[0,20]$ with the initial condition $y(0) = [1,1,1,1]^T$. Its analytic solutions are given by

$$\begin{aligned} &y_1(t) = \exp\bigl(\sin\bigl(t^2\bigr)\bigr), \qquad y_2(t) = \exp\bigl(5\sin\bigl(t^2\bigr)\bigr), \\ &y_3(t) = \sin\bigl(t^2\bigr) + 1, \qquad y_4(t) = \cos\bigl(t^2\bigr). \end{aligned} \tag{61}$$

The derivatives of the system show that their oscillation frequencies grow up rapidly when the time goes on. This is the reason why the global error control task [12] is difficult. We solve the problem with a fixed tolerance $tol = 1e{-}8$ and plot the absolute error in a log scale in Fig. 4(a). One can see that the proposed method achieves the required accuracy within the given tolerance on the interval $[0,20]$, whereas all other methods fail to meet the given tolerances and some results at the final time are significantly contaminated by the errors.

As shown in the first example, we also calculate the time cost required to obtain the desired accuracy by varying tolerances from $tol = 1e{-}5$ to $tol = 1e{-}10$ and plot the numerical results in Fig. 4(b). In this example, the numerical results show that the proposed scheme obtains the most accurate solution for each fixed CPU time. In particular, one can see that the proposed method achieves the required accuracies within the given tolerances, whereas all existing methods fail to achieve this requirement. Even the absolute errors of the other methods at the final time achieve about only half order for the desired accuracy even though the required CPU time is small compared to our method. That is, one may claim that our method well controls the global error within the given tolerances for this complicated system.

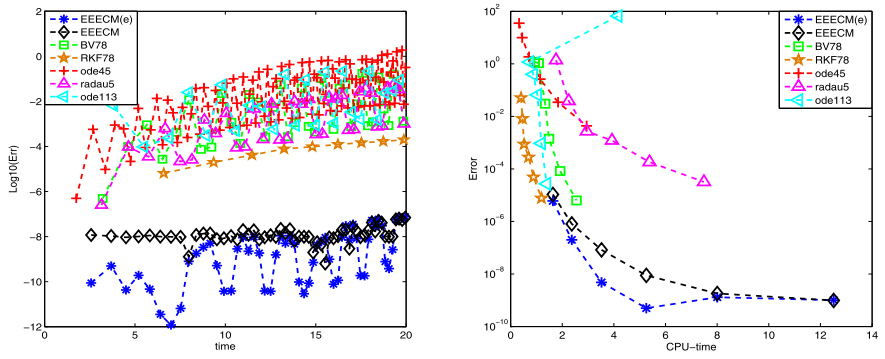


Figure 4 Comparison of (**a**) the error with a given tolerances *tol* = 1e−8 and (**b**) errors versus CPU-times for given tolerances from 1e−5 to 1e−10

4.2 Hamiltonian system

Formally, a Hamiltonian system is a dynamical system completely described by the scalar function H, the Hamiltonian. Firstly, we solve a simple pendulum problem to show how well EEECM can conserve the total energy H. Secondly, we test a two-body Kepler problem to confirm that the proposed method is well fit for the Hamiltonian system.

4.2.1 Pendulum problem

In this example, we solve the equation for the period of swing of a simple gravity pendulum depending only on its length and the local strength of gravity. The total energy of the pendulum is given by

$$H(p,q) = \frac{1}{2}p^2 - \cos(q), \tag{62}$$

whose components p and q satisfy

$$\begin{cases} p'(t) = \sin(q), \\ q'(t) = p. \end{cases} \tag{63}$$

We solve the system on the interval $[0, 500]$ with the initial conditions $p(0) = 1$ and $q(0) = \frac{\pi}{2}$ together with the given tolerance 1e−8. We examine the conservation property of the total energy H described by $|H(p(0), q(0)) - H(p_m, q_m)| = |\frac{1}{2} - H(p_m, q_m)|$, where p_m and q_m are the approximate solutions at time t_m. The numerical results are reported in Fig. 5 and show that only three methods, ode113, RKF78, and EEECM, achieve the invariance of H within the given tolerances. In particular, the numerical result of EEECM(e) has an outstanding conservation property compared to other numerical results. Hence, one may claim that the proposed method is superior to other existing methods.

4.2.2 Kepler problem

In astronomy problems, such as the Kepler problem, a long-term simulation is an indispensable factor. Hence, we solve a two-body Kepler problem subject to Newton's law of gravitation revolving around their center of mass, placed at the origin, in elliptic orbits in

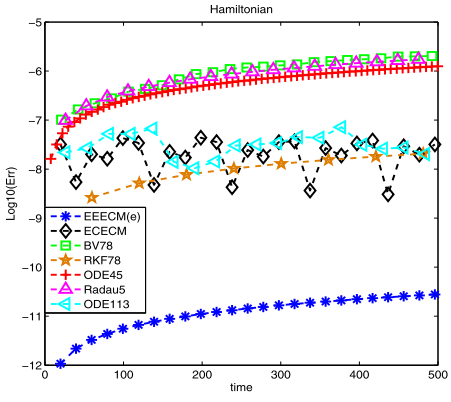


Figure 5 Comparison of invariance of the total energy H for solving the pendulum problem with tolerance 1e–8

the (q_1, q_2)-plane [29]. Assuming unitary masses and gravitational constant, the dynamics is described by the Hamiltonian function H given by

$$H(p_1, p_2, q_1, q_2) = \frac{1}{2}\left(p_1^2 + p_2^2\right) - \frac{1}{\sqrt{q_1^2 + q_2^2}} \tag{64}$$

together with the angular momentum L, which is another invariant of the system, described by

$$L(p_1, p_2, q_1, q_2) = q_1 p_2 - q_2 p_1, \tag{65}$$

whose components p_i, q_i $(i = 1, 2)$ satisfy the following IVP:

$$\begin{cases} p_1'(t) = -q_1({q_1}^2 + {q_2}^2)^{(-3/2)}, \\ p_2'(t) = -q_2({q_1}^2 + {q_2}^2)^{(-3/2)}, \\ q_1'(t) = p_1, \\ q_2'(t) = p_2. \end{cases} \tag{66}$$

We solve system (66) with the initial conditions $p_1(0) = 0$, $p_2(0) = 2$, $q_1(0) = 0.4$, $q_2(0) = 0$ on the interval $[0, 1000\pi]$ together with a fixed tolerance 1e–8. It is well known that the true solution is periodic with periodicity 2π [30]. As the previous example, we examine the conservation properties of the total energy H (Fig. 6(a)) as well as the angular momentum L (Fig. 6(b)). From the two figures, one can see that the numerical results EEECM(e) are the most accurate.

In Fig. 7, we examine the numerical periodicity with several methods and calculate the error between the starting point $(q_1(0), q_2(0)) = (0.4, 0)$ and the numerical solution at time $2k\pi$ $(k = 1, \ldots, 500)$ by using the Matlab built-in function for the cubic spline interpolation. After that, the only 16 points among the 500 calculated errors by selecting one after every 30 points are plotted in Fig. 7. The figures show that the proposed method generates the most accurate results in the sense of periodicity. One can summarize that the proposed method gives the most efficient numerical results in respect of both conservation and periodicity.

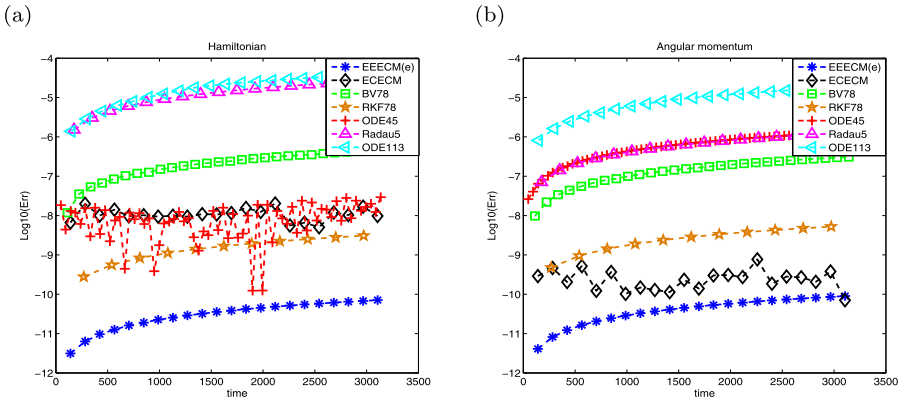
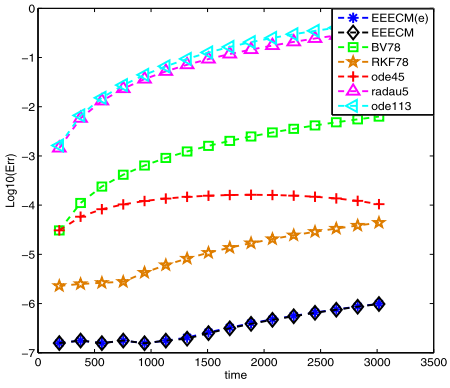


Figure 6 The invariances of (**a**) total energy H and (**b**) angular momentum L for a two-body Kepler problem with tolerance 1e–8

Figure 7 Comparison of periodicity for a two-body Kepler problem with tolerance 1e–8



5 Conclusion and further discussion

A new error control strategy for non-stiff problems is developed within the ECM framework. Unlike the traditional way to approximate solutions in an explicit single step method, we suggest a methodology that contains the estimated error at each integration step and enables us to control the bound of the local truncation error for a long time simulation. Throughout several numerical results, it is shown that the proposed method obtains a uniform-like error bound, which is outstanding compared with existing numerical methods. Also, it is seen that like symplectic methods, the proposed scheme preserves the invariants such as the energy and angular momentum in Hamiltonian systems.

In order to fully explore the efficiency of EEECM, several extended issues are currently being pursued. One of them is to optimize the number of function evaluations to reduce the computational cost such as the existing embedded algorithms. Another issue is to investigate strategies for selecting time integration step size, since an adaptive time stepping is necessary to find efficient solutions for a long time simulation. The proposed method is developed only for non-stiff problems, and we solved simple Hamiltonian systems. Hence, the other challenge is to extend the idea of the proposed method into stiff systems. Additionally, the generalization of the proposed idea will be applied to many physical problems expressed by partial differential equations (PDEs). Results along these directions will be reported in the future.

Acknowledgements
The authors would like to express their gratitude to the reviewers and the editor for valuable suggestions and comments.

Funding
The first author Kim was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant number 2016R1A2B2011326). Also, the corresponding author Bu was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (grant number 2016R1D1A1B03930734). The second author Piao was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (grant number 2017R1C1B1002370).

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
PK and XP provided the basic idea of this work and developed all theory needed in this manuscript. WJ simulated the numerical examples, and the corresponding author SB completed the proofs for all the theorems in this manuscript and wrote the manuscripts. All authors read and approved the final manuscript.

Author details
[1]Department of Mathematics, Kyungpook National University, Daegu, Korea. [2]Department of Mathematics, Hannam University, Daejeon, Korea. [3]Dongwoo Fine Chem, Pyeongtaek, Korea. [4]Departments of Liberal arts, Hongik University, Sejong, Korea.

Publisher's Note

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

Received: 18 September 2017 Accepted: 26 April 2018 Published online: 08 May 2018

References

1. Hairer, E., Norsett, S.P., Wanner, G.: Solving Ordinary Differential Equations. I Nonstiff. Springer Series in Computational Mathematics. Springer, Berlin (1993)
2. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations. II Stiff and Differential-Algebraic Problems. Springer Series in Computational Mathematics. Springer, Berlin (1996)
3. Calvo, M.P., Hairer, E.: Accurate long-term integration of dynamical systems. Appl. Numer. Math. **18**, 95–105 (1995)
4. Hairer, E.: Long-time integration of non-stiff and oscillatory Hamiltonian systems. AIP Conf. Proc. **1168**(1), 3–6 (2009)
5. Tiwari, S., Kumar, M.: An initial value technique to solve two-point non-linear singularly perturbed boundary value problems. Appl. Comput. Math. **14**(2), 150–157 (2015)
6. Dormand, J.R., Prince, P.J.: A family of embedded Runge–Kutta formulae. J. Comput. Appl. Math. **6**(1), 19–26 (1980)
7. Estep, D.: A posteriori error bounds and global error control for approximation of ordinary differential equations. SIAM J. Numer. Anal. **32**(1), 1–48 (1995)
8. Gustafsson, K.: Control-theoretic techniques for stepsize selection in implicit Runge–Kutta methods. ACM Trans. Math. Softw. **20**(4), 496–517 (1994)
9. Johnson, C.: Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinary differential equations. SIAM J. Numer. Anal. **25**(4), 908–926 (1988)
10. Kavetski, D., Binning, P., Sloan, S.W.: Adaptive time stepping and error control in a mass conservative numerical solution of the mixed form of Richards equation. Adv. Water Resour. **24**, 595–605 (2001)
11. Kulikov, G.Y.: Global error control in adaptive nordsieck methods. SIAM J. Sci. Comput. **34**(2), 839–860 (2012)
12. Kulikov, G.Y., Weiner, R.: Global error estimation and control in linearly-implicit parallel two-step peer W-methods. Comput. Appl. Math. **236**, 1226–1239 (2011)
13. Shampine, L.F.: Error estimation and control for ODEs. J. Sci. Comput. **25**(1), 3–16 (2005)
14. Pereyra, V.: Iterated deferred correction for nonlinear boundary value problems. Numer. Math. **11**, 111–125 (1968)
15. Zadunaisky, P.E.: On the estimation of errors propagated in the numerical integration of ordinary differential equations. Numer. Math. **27**, 21–40 (1976)
16. Bu, S., Huang, J., Minion, M.L.: Semi-implicit Krylov deferred correction methods for differential algebraic equations. Math. Comput. **81**(280), 2127–2157 (2012)
17. Bu, S., Lee, J.: An enhanced parareal algorithm based on the deferred correction methods for a stiff system. J. Comput. Appl. Math. **255**(1), 297–305 (2014)
18. Dutt, A., Greengard, L., Rokhlin, V.: Spectral deferred correction methods for ordinary differential equations. BIT Numer. Math. **40**(2), 241–266 (2000)
19. Huang, J., Jia, J., Minion, M.L.: Accelerating the convergence of spectral deferred correction methods. J. Comput. Phys. **214**(2), 633–656 (2006)
20. Huang, J., Jia, J., Minion, M.L.: Arbitrary order Krylov deferred correction methods for differential algebraic equations. J. Comput. Phys. **221**(2), 739–760 (2007)
21. Kim, P., Piao, X., Kim, S.D.: An error corrected Euler method for solving stiff problems based on Chebyshev collocation. SIAM J. Numer. Anal. **49**, 2211–2230 (2011)
22. Kim, S.D., Piao, X., Kim, D.H., Kim, P.: Convergence on error correction methods for solving initial value problems. J. Comput. Appl. Math. **236**(17), 4448–4461 (2012)
23. Kim, S.D., Kim, P.: Exponentially fitted error correction methods for solving initial value problems. Kyungpook Math. J. **52**, 167–177 (2012)

24. Kim, P., Lee, E., Kim, S.D.: Simple ECEM algorithms using function values only. Kyungpook Math. J. **53**, 573–591 (2013)
25. Atkinson, K.E.: An Introduction to Numerical Analysis. Wiley, New York (1989)
26. Fehlberg, E.: Classical fifth-, sixth-, seventh-, and eighth-order Runge–Kutta formulas with stepsize control. In: NASA; for Sale by the Clearinghouse for Federal Scientific and Technical Information, Springfield (1968)
27. Shampine, L.F.: Vectorized solution of ODEs in MATLAB. Scalable Comput.: Pract. Experience **10**, 337–345 (2010)
28. Gear, C.W.: Numerical Initial Value Problems in Ordinary Differential Equations. Prentice Hall, New York (1971)
29. Brugnano, L., Iavernaro, F., Trigiante, D.: Energy- and quadratic invariants-preserving integrators based upon Gauss collocation formulae. SIAM J. Numer. Anal. **50**, 2897–2916 (2012)
30. Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration: Structure-Preserving Algorithm for Ordinary Differenital Equations, 2nd edn. Springer Series in Computational Mathematics. Springer, Berlin (2006)

Submit your manuscript to a SpringerOpen journal and benefit from:

► Convenient online submission
► Rigorous peer review
► Open access: articles freely available online
► High visibility within the field
► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com