Journal of Computational Physics 351 (2017) 376–391

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An embedded formula of the Chebyshev collocation method for stiff problems

Xiangfan Piao [a], Sunyoung Bu [b], Dojin Kim [c], Philsu Kim [c,*]

[a] Department of Mathematics, Hannam University, Daejeon, 34430, Republic of Korea
[b] Department of Liberal Arts, Hongik University, Sejong, 30016, Republic of Korea
[c] Department of Mathematics, Kyungpook National University, Daegu, 41566, Republic of Korea

ARTICLE INFO

Article history:
Received 4 March 2017
Received in revised form 16 September 2017
Accepted 23 September 2017
Available online 28 September 2017

Keywords:
Generalized Chebyshev polynomial
Collocation method
Embedded formula
Stiff initial value problem

ABSTRACT

In this study, we have developed an embedded formula of the Chebyshev collocation method for stiff problems, based on the zeros of the generalized Chebyshev polynomials. A new strategy for the embedded formula, using a pair of methods to estimate the local truncation error, as performed in traditional embedded Runge–Kutta schemes, is proposed. The method is performed in such a way that not only the stability region of the embedded formula can be widened, but by allowing the usage of larger time step sizes, the total computational costs can also be reduced. In terms of concrete convergence and stability analysis, the constructed algorithm turns out to have an 8th order convergence and it exhibits A-stability. Through several numerical experimental results, we have demonstrated that the proposed method is numerically more efficient, compared to several existing implicit methods.

1. Introduction

The target problem we are concerned with is an initial value problem (IVP) of the form,

$$\phi'(t) = f(t, \phi(t)), \quad t \in (t_0, t_f]; \quad \phi(t_0) = \phi_0, \tag{1.1}$$

where $\phi(t) := [\phi_1(t), \cdots, \phi_d(t)]^{\mathrm{T}}$, and $f(t, \phi(t)) := [f_1(t, \phi(t)), \cdots, f_d(t, \phi(t))]^{\mathrm{T}}$ are assumed to be sufficiently smooth for the simplicity of the analysis. Here, the superscript T and index d denote the transpose of a vector and the dimension of the given system, respectively. Initial value problems are mathematical models used to describe a diverse range of natural phenomena, such as those in computational fluid dynamics, chemical reactions, electrical engineering, medical science, etc. Therefore, the effort to develop a numerical scheme capable to solve IVPs and applicable to a broad variety of scientific problems, is of special interest. In particular, understanding the stiffness of an IVP is a critical issue in this regard, and research conducted by mathematicians and other scientists on this problem has intensified in recent years. Indeed, the prolonged effort to seek ever more efficient and stable numerical methods has continued for several decades. Stiff IVPs are mainly solved using implicit schemes [2,3,5,11,20,22,23], as explicit schemes require very small step sizes due to the issue of their instability.

* Corresponding author.
E-mail addresses: piaoxf76@hanmail.net (X. Piao), syboo@hongik.ac.kr (S. Bu), kimdojin@knu.ac.kr (D. Kim), kimps@knu.ac.kr (P. Kim).

https://doi.org/10.1016/j.jcp.2017.09.046
0021-9991/

This study particularly focuses on the collocation method, which is one of several widespread implicit Runge–Kutta (RK) schemes and uniquely defined by the choice of collocation points. That is, for given pairwise distinct collocation points c_j^N $(0 \le c_j^N \le 1)$, $j = 0, \cdots, N$, if we let $t_{mj} = t_m + c_j^N h$, with a time step size h, the collocation method corresponding to c_j^N is represented by the collocation polynomial $p_N(t)$ of degree $N+1$ satisfying

$$p_N'(t_{mj}) = f\left(t_{mj}, \boldsymbol{\alpha}_j^N\right), \quad j = 0, 1, 2, \cdots, N; \; p_N(t_m) = y_m, \tag{1.2}$$

where y_m is the approximation of $\phi(t_m)$ and $\boldsymbol{\alpha}_j^N := p_N(t_{mj})$ is an approximation of $\phi(t_{mj}) = \phi(t_m + c_j^N h)$. Let us define a matrix $\mathcal{B}_N := \left(a_{ij}^N\right)_{i,j=1}^N$ and a vector $\boldsymbol{\gamma}_N := \left[a_{10}^N, \cdots, a_{N0}^N\right]^{\mathrm{T}}$ with

$$a_{ij}^N := \int_0^{c_i^N} l_j^N(\eta) d\eta, \quad l_j^N(\eta) := \prod_{k \ne j} \frac{\eta - c_k^N}{c_j^N - c_k^N}, \quad i, j = 0, 1, \cdots, N.$$

Using this matrix and vector, if the first collocation point satisfies $c_0^N = 0$, the solution $\boldsymbol{\alpha}_j^N$ of (1.2) can be obtained systematically from the solution of the nonlinear equation

$$\boldsymbol{\alpha}^N = h\left(\mathcal{B}_N \otimes \mathcal{I}_d\right) \boldsymbol{F}(\boldsymbol{\alpha}^N) + \mathbf{1}_N \otimes y_m + h\boldsymbol{\gamma}_N \otimes f(t_m, y_m), \tag{1.3}$$

which is known as the equivalent RK form of (1.2) [7]. Here,

$$\boldsymbol{\alpha}^N := \left[\left(\boldsymbol{\alpha}_1^N\right)^{\mathrm{T}}, \cdots, \left(\boldsymbol{\alpha}_N^N\right)^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \boldsymbol{F}(\boldsymbol{\alpha}^N) := \left[\left(f\left(t_{m1}, \boldsymbol{\alpha}_1^N\right)\right)^{\mathrm{T}}, \cdots, \left(f\left(t_{mN}, \boldsymbol{\alpha}_N^N\right)\right)^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \mathbf{1}_N := [1, \cdots, 1]^{\mathrm{T}},$$

$\mathcal{I}_d$ and $\otimes$ denote the identity matrix of size d, and the Kronecker product, respectively. Notice that the nonlinear system (1.3) requires a Newton-like iterative method for solving a $(Nd) \times (Nd)$ linear system expressed by the full Jacobian matrix for $\boldsymbol{F}$, at each iteration, which is computationally expensive. In this study, we will employ the simplified Newton's iteration (SNI), which uses a fixed Jacobian in each integration step to overcome this difficulty (see Section 2).

For controlling the size of the time step, a good estimation of the local truncation error is highly important in implicit RK methods. Traditionally, two methods are used simultaneously for the estimation of the local truncation error: the difference between the numerical solution and the extra lower order solution. One of the main challenges of implicit methods is the question of the reduction of computational cost of the calculation of the extra solution. For example, the widely used embedded Radau method [7] calculates the lower order solution from a linear combination of the intra-step values of the numerical solution to reduce the total computational cost. Generally, a collocation polynomial $p_{\tilde{N}}(s)$ of degree $\tilde{N} < N$ provides an expensive algorithm for solving the nonlinear system (1.3), when the lower order solution is calculated from (1.3) directly. Therefore, studying the accurate and efficient estimation of the local truncation error for collocation methods it is a very important research direction.

The purpose of this study is to provide an efficient embedded formula for the collocation method. In the early 1980s, Hasegawa et al. [9] introduced an elegant algorithm for the generalized Chebyshev interpolation procedure, that increased the number of sample points moderately rather than simply doubling them, and in such a manner that all zeros of the lower degree Chebyshev interpolation were contained by those of the higher degree Chebyshev interpolation. Following the publication of the algorithm, it was extensively used in interpolation theory, quadrature rules, and collocation methods for solving IVPs (e.g., see [9,10,12–18]).

The main contribution of this study is to achieve an efficient embedded formula for the collocation method, using this idea for the zeros of the Chebyshev polynomials. More precisely, the collocation points are taken as the zeros of the generalized Chebyshev polynomials, consisting the Chebyshev–Gauss–Lobatto (CGL) points $\{\eta_j\}_{j=0}^4$, together with two extra points, $\{\nu_k\}_{k=0}^1$, given by

$$\begin{aligned} \eta_j &:= \frac{1}{2}\left(\cos\left(\frac{4-j}{4}\pi\right) + 1\right), \quad j = 0, 1, \cdots, 4, \\ \nu_k &:= \frac{1}{2}\left(\cos\left(\left(\frac{k}{4} + \frac{3}{8}\right)\pi\right) + 1\right), \quad k = 0, 1, \end{aligned} \tag{1.4}$$

where η_j and ν_k are the zeros of the Chebyshev polynomials $T_5^*(s) - T_3^*(s)$ and $T_2^*(s) - \cos(3\pi/4)$, respectively. Here, $T_k^*(s)$ denotes the shifted Chebyshev polynomial of the first kind of degree k. Now, let $\{\tau_j\}$ be a set of the points, η_j, together with ν_k, and arranged so that

$$\tau_j = \begin{cases} \eta_j, & 0 \le j \le 4, \\ \nu_{j-5}, & 4 < j \le 6. \end{cases} \tag{1.5}$$

From here on, N is taken a fixed number of a value of either $N=4$ or $N=6$. Depending on the choice of N, a pair of c_j^N $(0 \le j \le N)$ is introduced, given by

$$c_j^N := \begin{cases} \eta_j, & N=4, \\ \tau_j, & N=6, \end{cases} \tag{1.6}$$

where η_j and τ_j are defined by (1.4) and (1.5), respectively. Since $c_4^N = 1$, the next approximation y_{m+1} can be obtained by

$$y_{m+1} \approx \boldsymbol{\alpha}_4^N$$

where $\boldsymbol{\alpha}^N$ is the solution of (1.3). Also, the collocation method based on the collocation points of (1.6) has the following convergence property.

Theorem 1.1. *The numerical solution of the collocation method (1.2) corresponding to the collocation points c_j^N defined by (1.6) has a convergence order $N+2$ and a stage order $N+1$ $(N=4,6)$.*

Proof. Let us define a function $M(t)$ by $M(t) = \prod_{j=0}^{N} \left(t - c_j^N\right)$. Then, using the change of variable $s = 2t-1$ and the definition of c_j^N given by (1.6), it can be seen that

$$\begin{aligned} \int_0^1 M(t) t^{q-1} dt &= \frac{1}{2} \int_{-1}^{1} \left(\frac{s+1}{2}\right)^{q-1} M\left(\frac{s+1}{2}\right) ds \\ &= \frac{1}{2^{N+1+q}} \begin{cases} \displaystyle\int_{-1}^{1} (s+1)^{q-1} s\left(s^2-1\right)\left(s^2-\frac{1}{2}\right) ds, & N=4, \\ \displaystyle\int_{-1}^{1} (s+1)^{q-1} s\left(s^2-1\right)\left(s^2-\frac{1}{2}\right)\left(s^2-\frac{2-\sqrt{2}}{4}\right) ds, & N=6. \end{cases} \end{aligned}$$

Since the two integrands, $s\left(s^2-1\right)\left(s^2-\frac{1}{2}\right)$ and $s\left(s^2-1\right)\left(s^2-\frac{1}{2}\right)\left(s^2-\frac{2-\sqrt{2}}{4}\right)$, are odd functions over the integration interval, it can be shown that the above integral is zero only when $q=1$. Thus, by using Theorem 7.9 in [6], the proof of the convergence order can be completed. In addition, since Chebyshev–Gauss–Lobatto points are used as collocation points, the stage order is $N+1$. For a detailed proof, see the reference [8]. □

As mentioned above, the SNI of (1.3) for the lower order solution $\boldsymbol{\alpha}_4^4$ corresponding to $c_4^4 = 1$ requires massive computational costs with each iteration and integration step. To overcome this complexity, instead of using the matrix, $\mathcal{B}_4^{-1}$ directly, its eigenvalues are replaced with those of $\mathcal{B}_6^{-1}$ in such a way that not only the stability region of the embedded formula can be widened, but also both LU decompositions from $\mathcal{B}_6^{-1}$, and the function values from $\boldsymbol{\alpha}^6$, can be used to find the solution $\boldsymbol{\alpha}^4$. This approach yields a dramatic reduction of the computational cost (see Section 3 for further details). In terms of the concrete convergence and stability analysis, the constructed algorithm turns out to be A-stable and have an 8th order convergence (see Sections 2 and 3, respectively).

Recently, we have developed E^2CCM4(6) [18] using the Chebyshev interpolation for the solution ϕ, based on the same collocation points defined by (1.6). The method has only the 6th order of convergence, but is almost L-stable. Despite this good stability, E^2CCM4(6) lacks the efficient embedded structure compared to the proposed scheme, which we have confirmed numerically by running several numerical tests. Finally, in Section 4, the results of several experiments are presented to validate our analysis.

2. Implementation for (1.3)

At the beginning of this section, we give a review of the SNI we implemented to solve the nonlinear system (1.3), which can be found, for example, in Ref. [7]. After it, we discuss the eigenvalue decomposition for the resulting linear systems, which will be used in Section 3 to develop an embedded formula for estimating the local truncation error. To discuss the SNI, let us combine the new variable

$$\boldsymbol{w}^N := \left[\left(\boldsymbol{w}_1^N\right)^{\mathrm{T}}, \cdots, \left(\boldsymbol{w}_N^N\right)^{\mathrm{T}}\right]^{\mathrm{T}} = \boldsymbol{\alpha}^N - \mathbf{1}_N \otimes y_m \tag{2.1}$$

with the system (1.3) to obtain

$$\boldsymbol{G}(\boldsymbol{w}^N) := -\boldsymbol{w}^N + \mathbf{d}_N + h\Big(\mathcal{B}_N \otimes \mathcal{I}_d\Big)\boldsymbol{F}\Big(\mathbf{1}_N \otimes y_m + \boldsymbol{w}^N\Big) = \mathbf{0}, \tag{2.2}$$

where $\mathbf{d}_N := h\boldsymbol{\gamma}_N \otimes f(t_m, y_m)$. Then, the SNI for the $(k+1)$th approximation $\boldsymbol{w}^{(N,k+1)}$ to the solution $\boldsymbol{w}^N$ is described by

$$\begin{cases} \Big(\mathcal{I}_{Nd} - h(\mathcal{B}_N \otimes \mathcal{J})\Big)\Delta\boldsymbol{w}^{(N,k)} = \boldsymbol{G}(\boldsymbol{w}^{(N,k)}), \\ \boldsymbol{w}^{(N,k+1)} = \boldsymbol{w}^{(N,k)} + \Delta\boldsymbol{w}^{(N,k)}, \end{cases} \tag{2.3}$$

where $\boldsymbol{w}^{(N,0)}$ is a given initial guess and $\mathcal{J}$ is a fixed Jacobian matrix of the function f given by

$$\mathcal{J} := f_\phi(t_m, y_m).$$

By setting $N = 6$, the next approximate solution y_{m+1} can be obtained for $\phi(t_{m+1})$, using the SNI given by (2.3). We assume that the iteration of (2.3) stops at the k^*th step, and satisfies

$$\|\Delta\boldsymbol{w}^{(6,k^*)}\|_2 = \|\boldsymbol{w}^{(6,k^*+1)} - \boldsymbol{w}^{(6,k^*)}\|_2 < \epsilon \tag{2.4}$$

for a given tolerance ϵ. For a practical choice of the tolerance ϵ, our algorithm adopts a traditional procedure introduced in [7, p. 121], which is implicitly defined by

$$\epsilon = \frac{1-\theta_{k^*}}{\theta_{k^*}}\kappa\Big(\mathrm{Atol} + \|y_m\|_2\,\mathrm{Rtol}\Big), \tag{2.5}$$

where Atol and Rtol are given absolute and relative tolerances, respectively, and

$$\theta_{k^*} = \frac{\|\Delta\omega^{(6,k^*)}\|_2}{\|\Delta\omega^{(6,k^*-1)}\|_2}, \quad \kappa = \max\Big(2.22\times 10^{-15}/\,\mathrm{Rtol}, \min(0.03, \mathrm{Rtol}^{1/3})\Big).$$

Since $c_4^6 = 1$, the approximation y_{m+1} for $\phi(t_{m+1})$ is then defined by

$$y_{m+1} = y_m + \boldsymbol{w}_4^{(6,k^*+1)}. \tag{2.6}$$

The choice of the initial guess $\boldsymbol{w}^{(6,0)}$ strongly influences the convergence of the SNI (2.3). In this paper, we are using the technique of polynomial extrapolation for the initial guess as follows. We first introduce a notation

$${}_{(m)}\boldsymbol{w}^{(6,k)} := \Big[\Big({}_{(m)}\boldsymbol{w}_1^{(6,k)}\Big)^{\mathrm{T}}, \cdots, \Big({}_{(m)}\boldsymbol{w}_6^{(6,k)}\Big)^{\mathrm{T}}\Big]^{\mathrm{T}}$$

for the kth iterated solution obtained by (2.3) in the mth integration step $[t_m, t_{m+1}]$, where each component ${}_{(m)}\boldsymbol{w}_j^{(6,k)}$ denotes the vector of intra-step values at the intra-time $t_{mj} = t_m + hc_j^6$. Now, we use an extrapolation polynomial β of degree 6 satisfying the interpolation conditions

$$\beta(0) = 0, \quad \beta(c_j^6) = {}_{(m-1)}\boldsymbol{w}_j^{(6,k^*+1)}, \quad j = 1, 2, \cdots, 6,$$

where ${}_{(m-1)}\boldsymbol{w}_j^{(6,k^*+1)}$ denotes the intra-step values in the previous step $[t_{m-1}, t_m]$. Then, each component ${}_{(m)}\boldsymbol{w}_j^{(6,0)}$ $(1 \le j \le 6)$ of the initial guess is chosen in the mth integration step $[t_m, t_{m+1}]$ by evaluating β at time $1 + s_m c_j^6$ as follows

$$\begin{cases} {}_{(0)}\boldsymbol{w}_j^{(6,0)} = 0, \\ {}_{(m)}\boldsymbol{w}_j^{(6,0)} = \beta(1 + s_m c_j^6) + y_{m-1} - y_m, \quad m \ge 1, \quad j = 1, 2, \cdots, 6, \end{cases}$$

where $s_m = \frac{t_{m+1}-t_m}{t_m - t_{m-1}}$.

In the following lemma, an approximation property is shown for the intra-step values $y_m + \boldsymbol{w}_j^{(6,k^*)}$ obtained from the k^*th iteration step of the SNI (2.3), which will be used in the construction of the embedded formula.

Lemma 2.1. *Assume that $\Delta\boldsymbol{w}_j^{(6,k^*)}$ $(1 \le j \le 6)$ are obtained at the k^*th iteration step from (2.3), and they also satisfy the inequality given by (2.4), for a given tolerance ϵ. Then, for a sufficiently small h, it is valid that*

$$\Big\|\phi(t_{mj}) - \Big(y_m + \boldsymbol{w}_j^{(6,k^*)}\Big)\Big\|_2 = \mathcal{O}\Big(h^7 + \epsilon\Big). \tag{2.7}$$

Proof. Using the definition (2.1), Theorem 1.1, and the triangle inequality, we obtain the following:

$$\begin{aligned}\left\|\phi(t_{mj})-\left(y_m+\boldsymbol{w}_j^{(6,k^*)}\right)\right\|_2 &= \left\|\phi(t_{mj})-\boldsymbol{\alpha}_j^6+\left(\boldsymbol{\alpha}_j^6-y_m-\boldsymbol{w}_j^{(6,k^*)}\right)\right\|_2 \\ &= \mathcal{O}\left(h^7+\|\boldsymbol{w}_j^6-\boldsymbol{w}_j^{(6,k^*)}\|_2\right).\end{aligned} \tag{2.8}$$

For the estimation of $\|\boldsymbol{w}_j^6-\boldsymbol{w}_j^{(6,k^*)}\|_2$, the Taylor expansion of (2.2) is applied about $\boldsymbol{w}^{(6,k^*)}$ given by

$$\tilde{\mathcal{J}}\boldsymbol{X}=\boldsymbol{G}(\boldsymbol{w}^{(6,k^*)})+\mathcal{O}\left(\|\boldsymbol{X}\|_2^2\right),\quad \boldsymbol{X}:=\boldsymbol{w}^6-\boldsymbol{w}^{(6,k^*)}, \tag{2.9}$$

where $\tilde{\mathcal{J}}$ is the Jacobian defined by

$$\tilde{\mathcal{J}}:=\mathcal{I}_{6d}-h\left(\widehat{\mathcal{J}}_{ij}\right),\quad \widehat{\mathcal{J}}_{ij}:=a_{ij}^6 f_\phi\left(t_{mj},y_m+\boldsymbol{w}_j^{(6,k^*)}\right).$$

Since

$$\tilde{\mathcal{J}}=\mathcal{I}_{6d}-h(\mathcal{B}_6\otimes\mathcal{J})+\mathcal{O}(h^2),$$

equation (2.9) reads

$$\left(\mathcal{I}_{6d}-h(\mathcal{B}_6\otimes\mathcal{J})\right)\boldsymbol{X}=\boldsymbol{G}(\boldsymbol{w}^{(6,k^*)})+\mathcal{O}\left(\|\boldsymbol{X}\|_2^2+h^2\|\boldsymbol{X}\|_2\right). \tag{2.10}$$

For a sufficiently small h and a smooth function f, the matrix $\mathcal{I}_{6d}-h(\mathcal{B}_6\otimes\mathcal{J})$ is bounded. Hence, combining (2.3) with (2.4) leads to

$$\boldsymbol{G}\left(\boldsymbol{w}^{(6,k^*)}\right)=\mathcal{O}(\epsilon). \tag{2.11}$$

Note that for a sufficiently small h,

$$\left\|\left(\mathcal{I}_{6d}-h\left(\mathcal{B}_6\otimes\mathcal{J}\right)\right)^{-1}\right\|\le\frac{1}{1-h\|\mathcal{B}_6\otimes\mathcal{J}\|}\le C,$$

where $\|\cdot\|$ denotes a matrix norm and C is a constant independent of h. Therefore, the equation (2.10) gives

$$\|\boldsymbol{X}\|_2\le C_1\left\|\boldsymbol{G}\left(\boldsymbol{w}^{(6,k^*)}\right)\right\|_2+C_2\|\boldsymbol{X}\|_2^2+C_3h^2\|\boldsymbol{X}\|_2,$$

where C_i $(i=1,2,3)$ are constants independent of h. Since $\boldsymbol{w}^{(6,k^*)}$ is a numerical approximation of $\boldsymbol{w}^6$, without loss of generality, it can be assumed that the quantity $\|\boldsymbol{X}\|_2$ is also sufficiently small. Then, the above quadratic inequality for $\|\boldsymbol{X}\|_2$, can be solved by

$$a\le\frac{\kappa-\sqrt{\kappa^2-4C_1C_2b}}{2C_2}=\frac{4C_1C_2b}{2C_2(\kappa+\sqrt{\kappa^2-4C_1C_2b})}\le Cb,\quad \kappa:=1-C_3h^2 \tag{2.12}$$

for some constant C independent of the step size h, where $a=\|\boldsymbol{X}\|_2$ and $b=\left\|\boldsymbol{G}\left(\boldsymbol{w}^{(6,k^*)}\right)\right\|_2$. Combining (2.8) with (2.11) and substituting the result into (2.12) completes the proof. □

We now review the eigenvalue decomposition used to reduce the computational cost of the SNI (2.3). Using the complex eigenvalues $\lambda_{6,j}\pm i\mu_{6,j}$ of $\mathcal{B}_6^{-1}$, and their corresponding eigenvectors $\mathbf{u}_{6,j}\pm i\mathbf{v}_{6,j}$, the matrix $\mathcal{B}_6^{-1}$ can be decomposed by

$$\begin{aligned}&\mathcal{B}_6^{-1}=T_6\Lambda_6T_6^{-1},\\ &\Lambda_6=\mathrm{diag}(\Lambda_{6,1},\Lambda_{6,2},\Lambda_{6,3}),\quad T_6=\left[\mathbf{u}_{6,1},-\mathbf{v}_{6,1},\mathbf{u}_{6,2},-\mathbf{v}_{6,2},\mathbf{u}_{6,3},-\mathbf{v}_{6,3}\right],\end{aligned}$$

where $\Lambda_{6,j}=\begin{bmatrix}\lambda_{6,j} & -\mu_{6,j}\\ \mu_{6,j} & \lambda_{6,j}\end{bmatrix}$. Multiplying the first equation in (2.3) by $h^{-1}T_6^{-1}\mathcal{B}_6^{-1}\otimes\mathcal{I}_d$ yields

$$\left(h^{-1}\Lambda_6\otimes\mathcal{I}_d-\mathcal{I}_6\otimes\mathcal{J}\right)(T_6^{-1}\otimes\mathcal{I}_d)\Delta w^{(6,k)}=(h^{-1}T_6^{-1}\mathcal{B}_6^{-1}\otimes\mathcal{I}_d)\boldsymbol{G}(\boldsymbol{w}^{(6,k)}). \tag{2.13}$$

Instead of solving the matrix system (2.13) directly, by introducing new variables given by

$$\begin{aligned}&\boldsymbol{x}^{(6,k)}:=\left[\left(\boldsymbol{x}_1^{(6,k)}\right)^{\mathrm{T}},\cdots,\left(\boldsymbol{x}_6^{(6,k)}\right)^{\mathrm{T}}\right]^{\mathrm{T}}:=(T_6^{-1}\otimes\mathcal{I}_d)\Delta w^{(6,k)},\\ &\boldsymbol{q}:=(\boldsymbol{q}_1^{\mathrm{T}},\cdots,\boldsymbol{q}_6^{\mathrm{T}})^{\mathrm{T}}:=(h^{-1}T_6^{-1}\mathcal{B}_6^{-1}\otimes\mathcal{I}_d)\boldsymbol{G}(\boldsymbol{w}^{(6,k)}),\end{aligned} \tag{2.14}$$

its solution can be obtained by using three $d \times d$ complex linear systems given by

$$\begin{aligned}
&\left(h^{-1}(\lambda_{6,1}+i\mu_{6,1})\mathcal{I}_d-\mathcal{J}\right)\left(\boldsymbol{x}_1^{(6,k)}+i\boldsymbol{x}_2^{(6,k)}\right)=\boldsymbol{q}_1+i\boldsymbol{q}_2,\\
&\left(h^{-1}(\lambda_{6,2}+i\mu_{6,2})\mathcal{I}_d-\mathcal{J}\right)\left(\boldsymbol{x}_3^{(6,k)}+i\boldsymbol{x}_4^{(6,k)}\right)=\boldsymbol{q}_3+i\boldsymbol{q}_4,\\
&\left(h^{-1}(\lambda_{6,3}+i\mu_{6,3})\mathcal{I}_d-\mathcal{J}\right)\left(\boldsymbol{x}_5^{(6,k)}+i\boldsymbol{x}_6^{(6,k)}\right)=\boldsymbol{q}_5+i\boldsymbol{q}_6.
\end{aligned} \tag{2.15}$$

In this study, we use the LU factorization as a linear solver for the linear systems of (2.15). It should be noted that if the dimension d of the system (1.1) is large, one can use a parallel computation for (2.15), after calculating the vector $\boldsymbol{q}$ defined by (2.14).

This section is ended by a demonstration of the stability of the collocation method (1.3). For this purpose, we test the method with Dahlquist's problem $\phi'(t)=\lambda\phi(t)$. Since $f(t,\phi)=\lambda\phi$ for Dahlquist's problem, the right-hand side of (1.3) becomes

$$h\Big(\mathcal{B}_N\otimes\mathcal{I}_d\Big)\boldsymbol{F}(\boldsymbol{\alpha}^N)+\mathbf{1}_N\otimes y_m+h\boldsymbol{\gamma}_N\otimes f(t_m,y_m)=\lambda h\mathcal{B}_N\boldsymbol{\alpha}^N+y_m\Big(\mathbf{1}_N+h\lambda\boldsymbol{\gamma}_N\Big).$$

Hence, the Chebyshev collocation system (1.3) applied to the Dahlquist's problem leads to

$$\Big(\mathcal{I}_N-\lambda h\mathcal{B}_N\Big)\boldsymbol{\alpha}=y_m\Big(\mathbf{1}_N+\lambda h\boldsymbol{\gamma}_N\Big) \tag{2.16}$$

with the unknown $\boldsymbol{\alpha}=[\alpha_1,\cdots,\alpha_N]^{\mathrm{T}}$. Thus, by solving the linear system (2.16) with $N=6$, the proposed method becomes

$$y_{m+1}=\mathcal{S}(\lambda h)y_m, \tag{2.17}$$

where the stability function $\mathcal{S}(z)$ is the 4th component of the vector $\Big(\mathcal{I}_6-z\mathcal{B}_6\Big)^{-1}\Big(\mathbf{1}_6+z\boldsymbol{\gamma}_6\Big)$. From Cramer's rule, it can be shown that

$$\mathcal{S}(z)=\frac{\det\begin{pmatrix}\mathcal{I}_6-z\mathcal{B}_6 & \mathbf{1}_6+z\boldsymbol{\gamma}_6\\ -z\mathbf{a}_4^{\mathrm{T}} & 1+za_{40}\end{pmatrix}}{\det\Big(\mathcal{I}_6-z\mathcal{B}_6\Big)}, \tag{2.18}$$

where $\mathbf{a}_4$ and a_{40} denote the 4th column of the matrix $\mathcal{B}_6^{\mathrm{T}}$ and the vector $\boldsymbol{\gamma}_6^{\mathrm{T}}$, respectively. A detailed proof can be found in Ref. [14, Proposition 5.2]. For the stability function $\mathcal{S}(z)$, the stability region of the method is defined by (see [7, p. 16])

$$\Gamma:=\{z\in\mathbb{C}:|\mathcal{S}(z)|<1\}. \tag{2.19}$$

When the left-half complex plane is contained in Γ, the method is called A-stable [7, p. 42]. Applying a symbolic calculation by Mathematica to the formula (2.18), it can be shown that the explicit form of $\mathcal{S}(z)$ is given by

$$\mathcal{S}(z)=\frac{Q(z)}{Q(-z)}:=\prod_{k=1}^{6}\frac{z+z_k}{z-z_k},\quad Q(z):=\sum_{k=0}^{6}\omega_k z^k,$$

where z_k are the roots of $Q(-z)$ and

$$(\omega_0,\cdots,\omega_6):=\left(1,\frac{1}{2},\frac{76+\sqrt{2}}{672},\frac{20+\sqrt{2}}{1344},\frac{130+17\sqrt{2}}{107520},\frac{38+11\sqrt{2}}{645120},\frac{2+\sqrt{2}}{1290240}\right).$$

From the Routh tableau for $Q(z)$ given in Table 1, it can be seen that the conditions of the Routh theorem are satisfied, and hence the real parts of $-z_k$ are all negative (see Theorem 13.4 in [6]). Thus, for every complex number z whose real part is negative, it can be concluded that

$$\left|\frac{z+z_k}{z-z_k}\right|\le 1,\quad k=1,2,\cdots,6.$$

This means that the collocation method (1.3) is indeed A-stable.

3. Embedded formula

Since the collocation method (1.3) corresponding to c_j^6 defined by (1.6) is a 7-stage implicit RK method, the local truncation error can be estimated by using the well-known results of RK theory. Instead of this method, here we propose a new embedded formula which uses modified eigenvalues of $\mathcal{B}_4^{-1}$, and has a wide stability region, without the extra function

Table 1
Routh tableau for $Q(z)$.

	$j=0$	$j=1$	$j=2$	$j=3$
$i=0$	$\frac{2+\sqrt{2}}{1290240}$	$-\frac{130+17\sqrt{2}}{107520}$	$\frac{76+\sqrt{2}}{672}$	-1
$i=1$	$\frac{38+11\sqrt{2}}{645120}$	$-\frac{20+\sqrt{2}}{1344}$	$\frac{1}{2}$	
$i=2$	$\frac{23(2430+119\sqrt{2})}{64619520}$	$\frac{-41140+743\sqrt{2}}{403872}$	1	
$i=3$	$\frac{57866-5739\sqrt{2}}{7084161}$	$\frac{16(-9140+347\sqrt{2})}{337341}$		
$i=4$	$\frac{17761948-2250053\sqrt{2}}{322284480}$	-1		
$i=5$	$\frac{2352(384720-27977\sqrt{2})}{3183566377}$			
$i=6$	1			

evaluations featuring in the usual RK method. For the estimation of the local truncation error, we first solve the linear system

$$\begin{aligned}\left(\mathcal{I}_{4d}-h\left(\mathcal{B}_4\otimes\mathcal{J}\right)\right)\left(\boldsymbol{w}^{(4,1)}-\boldsymbol{w}^{(4,0)}\right)&=\boldsymbol{G}\left(\boldsymbol{w}^{(4,0)}\right),\\ \boldsymbol{w}^{(4,0)}&=\left[\left(\boldsymbol{w}_1^{(6,k^*)}\right)^{\mathrm{T}},\cdots,\left(\boldsymbol{w}_4^{(6,k^*)}\right)^{\mathrm{T}}\right]^{\mathrm{T}},\end{aligned}\tag{3.1}$$

and consider a lower order solution $\tilde{y}_{m+1}$ defined by

$$\tilde{y}_{m+1}=y_m+\boldsymbol{w}_4^{(4,1)}.\tag{3.2}$$

Here, $\boldsymbol{w}^{(6,k^*)}$ is the solution obtained from (2.3) and satisfying (2.4). The linear system (3.1) can be induced by a similar application of the SNI introduced in (2.3) for the collocation method of the case $N=4$ (see (1.3) and (1.6)). In the subsequent lemma and theorem, we want to show that the solution $\tilde{y}_{m+1}$ of (3.2) can be used as a lower order solution in an embedded formula.

For this purpose, we begin with an estimation of the solution of (3.1) by using the results of Theorem 1.1 and Lemma 2.1.

Lemma 3.1. *For the function $\boldsymbol{G}$ defined by (2.2) and the variable $\boldsymbol{w}^{(4,0)}$ defined by (3.1), we have*

$$\|\boldsymbol{w}^{(4,1)}-\boldsymbol{w}^{(4,0)}\|_2=\mathcal{O}\left(h^5+\epsilon\right).\tag{3.3}$$

Proof. For the solution $\boldsymbol{w}^4$ of the nonlinear equation (2.2) with $N=4$, applying the Taylor expansion about $\boldsymbol{w}^4$ for $\boldsymbol{G}(\boldsymbol{w}^{(4,0)})$ yields

$$\boldsymbol{G}\left(\boldsymbol{w}^{(4,0)}\right)=\boldsymbol{G}\left(\boldsymbol{w}^4\right)+\mathcal{J}_G\left(\boldsymbol{\zeta}\right)\left(\boldsymbol{w}^{(4,0)}-\boldsymbol{w}^4\right)=\mathcal{J}_G\left(\boldsymbol{\zeta}\right)\left(\boldsymbol{w}^{(4,0)}-\boldsymbol{w}^4\right),$$

where $\mathcal{J}_G$ denotes the Jacobian of $\boldsymbol{G}$ and $\boldsymbol{\zeta}$ is a vector between $\boldsymbol{w}^{(4,0)}$ and $\boldsymbol{w}^4$. Note that the assumption that the function f is smooth leads to a uniform bound for the Jacobian $\mathcal{J}_G$. Thus, by taking the l_2-norm of both sides of the above equation and applying the triangle inequality together with Lemma 2.1 and Theorem 1.1, one finds

$$\begin{aligned}\left\|\boldsymbol{G}(\boldsymbol{w}^{(4,0)})\right\|_2&\le C_1\|\boldsymbol{w}^{(4,0)}-\boldsymbol{w}^4\|_2\\ &\le C_1\left(\left\|\boldsymbol{w}^{(6,k^*)}-\boldsymbol{w}^6\right\|_2+\sqrt{\sum_{j=1}^{4}\left\|\boldsymbol{w}_j^6-\boldsymbol{w}_j^4\right\|_2^2}\right)\\ &=C_1\left(\left\|\boldsymbol{w}^{(6,k^*)}-\boldsymbol{w}^6\right\|_2+\sqrt{\sum_{j=1}^{4}\left\|\boldsymbol{\alpha}_j^6-\boldsymbol{\alpha}_j^4\right\|_2^2}\right)\\ &\le C_1\left(\left\|\boldsymbol{w}^{(6,k^*)}-\boldsymbol{w}^6\right\|_2+\sqrt{\sum_{j=1}^{4}\left\|\boldsymbol{\alpha}_j^6-\phi(t_{mj})\right\|_2^2+\left\|\phi(t_{mj})-\boldsymbol{\alpha}_j^4\right\|_2^2}\right)\\ &\le C_2(h^5+\epsilon),\end{aligned}\tag{3.4}$$

where C_1 and C_2 are some constants. For a sufficiently small h and a smooth function f, we have

$$\left\| \left(\mathcal{I}_{4d} - h\left(\mathcal{B}_4 \otimes \mathcal{J}\right) \right)^{-1} \right\| \le \frac{1}{1 - h\|\mathcal{B}_4 \otimes \mathcal{J}\|} \le C,$$

where C is a constant that is independent of h. Then, by combining (3.4) with (3.1), the proof can be completed. □

Based on the above lemma, the estimation for the lower order solution defined by (3.2) can be made as follows.

Theorem 3.2. *Let $\tilde{y}_{m+1} := y_m + \boldsymbol{w}_4^{(4,1)}$ be the lower order solution at time $t = t_{m+1}$ defined by (3.2). Then, for a given tolerance ϵ, it is valid that*

$$\left\| \phi(t_{m+1}) - \tilde{y}_{m+1} \right\|_2 = \mathcal{O}\left(h^5 + \epsilon\right).$$

Proof. Lemma 2.1, Lemma 3.1 and the triangle inequality lead to

$$\begin{aligned} \left\| \phi(t_{m+1}) - \tilde{y}_{m+1} \right\|_2 &= \left\| \phi(t_{m+1}) - (y_m + \boldsymbol{w}_4^{(4,1)}) \right\|_2 \\ &\le \left\| \phi(t_{m+1}) - (y_m + \boldsymbol{w}_4^{(4,0)}) \right\|_2 + \left\| \boldsymbol{w}_4^{(4,0)} - \boldsymbol{w}_4^{(4,1)} \right\|_2 \\ &= \mathcal{O}\Big(h^5 + \epsilon\Big), \end{aligned}$$

which completes the proof. □

Let us assume that $\left(\lambda_{4,j} \pm i\mu_{4,j}, \mathbf{u}_{4,j} \pm i\mathbf{v}_{4,j}\right)$ is a pair of the eigenvalues and eigenvectors of $\mathcal{B}_4^{-1}$. Then, the matrix $\mathcal{B}_4^{-1}$ can be decomposed by

$$\mathcal{B}_4^{-1} = T_4 \Lambda_4 T_4^{-1}, \quad \Lambda_4 = \mathrm{diag}(\Lambda_{4,1}, \Lambda_{4,2}), \quad T_4 = \Big[\mathbf{u}_{4,1}, -\mathbf{v}_{4,1}, \mathbf{u}_{4,2}, -\mathbf{v}_{4,2}\Big],$$

where $\Lambda_{4,j} = \begin{bmatrix} \lambda_{4,j} & -\mu_{4,j} \\ \mu_{4,j} & \lambda_{4,j} \end{bmatrix}$. Using this eigenvalue decomposition and the process for obtaining (2.13), the linear system (3.1) can be written as

$$\Big(h^{-1}\Lambda_4 \otimes \mathcal{I}_d - \mathcal{I}_4 \otimes \mathcal{J}\Big)(T_4^{-1} \otimes \mathcal{I}_d)\Big(\boldsymbol{w}^{(4,1)} - \boldsymbol{w}^{(4,0)}\Big) = (h^{-1}T_4^{-1}\mathcal{B}_4^{-1} \otimes \mathcal{I}_d)\boldsymbol{G}(\boldsymbol{w}^{(4,0)}). \tag{3.5}$$

Thus, as can be seen in (2.15), two $d \times d$ complex linear systems have to be solved to obtain the lower order solution for (3.5), which is computationally inefficient compared to the existing RK methods. Hence, we would like to construct an efficient algorithm to calculate $\boldsymbol{w}^{(4,1)}$ by modifying the eigenvalues of $\mathcal{B}_4^{-1}$. To achieve this aim, we take the perturbed eigenvalues $\widehat{\lambda}_{4,k} + i\widehat{\mu}_{4,k}$ of $\lambda_{4,k} + i\mu_{4,k}$ such that

$$\left| \widehat{\lambda}_{4,k} + i\widehat{\mu}_{4,k} - (\lambda_{4,k} + i\mu_{4,k}) \right| \le c \tag{3.6}$$

with a fixed constant c and define matrices

$$\widehat{\mathcal{B}}_4^{-1} = T_4 \widehat{\Lambda}_4 T_4^{-1}, \quad \widehat{\Lambda}_4 = \begin{bmatrix} \widehat{\Lambda}_{4,1} & 0 \\ 0 & \widehat{\Lambda}_{4,2} \end{bmatrix}, \qquad \widehat{\Lambda}_{4,j} = \begin{bmatrix} \widehat{\lambda}_{4,j} & -\widehat{\mu}_{4,j} \\ \widehat{\mu}_{4,j} & \widehat{\lambda}_{4,j} \end{bmatrix}.$$

Then, instead of solving the linear system (3.5) for the lower order solution, we solve

$$\Big(h^{-1}\widehat{\Lambda}_4 \otimes \mathcal{I}_d - \mathcal{I}_4 \otimes \mathcal{J}\Big)(T_4^{-1} \otimes \mathcal{I}_d)\Big(\widehat{\boldsymbol{w}}^{(4,1)} - \boldsymbol{w}^{(4,0)}\Big) = (h^{-1}T_4^{-1}\mathcal{B}_4^{-1} \otimes \mathcal{I}_d)\boldsymbol{G}(\boldsymbol{w}^{(4,0)}), \tag{3.7}$$

and define the modified lower order solution $\widehat{y}_{m+1}$ by

$$\widehat{y}_{m+1} := y_m + \widehat{\boldsymbol{w}}_4^{(4,1)}. \tag{3.8}$$

The following theorem shows that the modified lower order solution $\widehat{y}_{m+1}$ of (3.8) can be used as a lower order solution in the embedded formula.

Theorem 3.3. *The modified lower order solution $\widehat{y}_{m+1}$ defined by (3.8) satisfies*

$$\left\| \phi(t_{m+1}) - \widehat{y}_{m+1} \right\|_2 = \mathcal{O}\left(h^5 + \epsilon\right)$$

for a given tolerance ϵ.

Proof. For a sufficiently small h and a smooth function f, we first note that

$$\left\| \left(\mathcal{B}_4 \widehat{\mathcal{B}}_4^{-1} \otimes \mathcal{I}_d - h\,(\mathcal{B}_4 \otimes \mathcal{J}) \right)^{-1} \right\| \le \frac{\|\widehat{\mathcal{B}}_4 \mathcal{B}_4^{-1}\|}{1 - h\|\widehat{\mathcal{B}}_4 \otimes \mathcal{J}\|} \le C,$$

where C is a constant independent of h. Thus, by using the inequality (3.4), the solution of (3.7) can be estimated by

$$\|\widehat{\boldsymbol{w}}^{(4,1)} - \boldsymbol{w}^{(4,0)}\|_2 = \mathcal{O}\Big(\|\boldsymbol{G}(\boldsymbol{w}^{(4,0)}\|_2\Big) = \mathcal{O}\Big(h^5 + \epsilon\Big).$$

Therefore, Lemma 2.1 and the triangle inequality lead to

$$\begin{aligned} \|\phi(t_{m+1}) - \widehat{y}_{m+1}\|_2 &= \left\| \phi(t_{m+1}) - (y_m + \widehat{\boldsymbol{w}}_4^{(4,1)}) \right\|_2 \\ &\le \left\| \phi(t_{m+1}) - (y_m + \boldsymbol{w}_4^{(4,0)}) \right\|_2 + \left\| \boldsymbol{w}_4^{(4,0)} - \widehat{\boldsymbol{w}}_4^{(4,1)} \right\|_2 \\ &= \mathcal{O}\Big(h^5 + \epsilon\Big), \end{aligned}$$

which completes the proof. □

From now on, as mentioned in Introduction, we demonstrate a way to choose perturbed eigenvalues $\widehat{\lambda}_{4,k} + i\widehat{\mu}_{4,k}$ so that not only the stability region of the embedded formula can be widened, but also the total amount of computational cost can be reduced. For this purpose, using the property

$$\max_{j,k} \left| \lambda_{6,j} + i\mu_{6,j} - (\lambda_{4,k} + i\mu_{4,k}) \right| \le 10, \tag{3.9}$$

which can be obtained by the aid of Mathematica, we take the perturbed eigenvalue $\widehat{\lambda}_{4,k} + i\widehat{\mu}_{4,k}$ as

$$\widehat{\lambda}_{4,k} + i\widehat{\mu}_{4,k} = \underset{\lambda_{6,j} + i\mu_{6,j}}{\arg\min} \left| \Big(\lambda_{6,j} + i\mu_{6,j}\Big) - \Big(\lambda_{4,k} + i\mu_{4,k}\Big) \right|, \quad k = 1, 2, \quad j = 1, 2, 3.$$

In this case, the constant c in (3.6) becomes less than 10, so the above discussions with these perturbed eigenvalues are valid. Thus, by a similar procedure of (2.13)–(2.15), after solving

$$\begin{aligned} &\big(h^{-1}(\widehat{\lambda}_{4,1} + i\widehat{\mu}_{4,1})\mathcal{I}_d - \mathcal{J}\big)\big(\boldsymbol{y}_1 + i\boldsymbol{y}_2\big) = \boldsymbol{q}_1 + i\boldsymbol{q}_2, \\ &\big(h^{-1}(\widehat{\lambda}_{4,2} + i\widehat{\mu}_{4,2})\mathcal{I}_d - \mathcal{J}\big)\big(\boldsymbol{y}_3 + i\boldsymbol{y}_4\big) = \boldsymbol{q}_3 + i\boldsymbol{q}_4, \\ &\boldsymbol{q} := (\boldsymbol{q}_1^{\mathrm{T}}, \cdots, \boldsymbol{q}_4^{\mathrm{T}})^{\mathrm{T}} := (h^{-1}T_4^{-1}\mathcal{B}_4^{-1} \otimes I_d)\boldsymbol{G}(\boldsymbol{w}^{(4,0)}), \end{aligned} \tag{3.10}$$

the solution $\widehat{\boldsymbol{w}}^{(4,1)} - \boldsymbol{w}^{(4,0)}$ of the linear system (3.7) can be obtained by

$$\widehat{\boldsymbol{w}}^{(4,1)} - \boldsymbol{w}^{(4,0)} = \Big(T_4 \otimes \mathcal{I}_d\Big)\Big[\boldsymbol{y}_1^{\mathrm{T}}, \cdots, \boldsymbol{y}_4^{\mathrm{T}}\Big]^{\mathrm{T}}.$$

Since the two $d \times d$ complex matrices of (3.10) are the same as those in the linear systems of (2.15), the same LU decompositions in (2.15) can be used.

Finally, this section is concluded by studying the stability of the solution $\widehat{\boldsymbol{w}}^{(4,1)}$, obtained from the linear systems (3.7). For this, we apply the Dahlquist's problem with $\phi' = f(t, \phi) = \lambda\phi$ to the system (3.7). From (2.2), we have

$$\boldsymbol{G}(\boldsymbol{w}^{(4,0)}) = -\boldsymbol{w}^{(4,0)} + z\boldsymbol{\gamma}_4 y_m + z\mathcal{B}_4(\mathbf{1}_4 y_m + \boldsymbol{w}^{(4,0)}), \quad z := \lambda h.$$

Thus, the equation (3.7) gives the linear system

$$\Big(\mathcal{B}_4\widehat{\mathcal{B}}_4^{-1} - z\mathcal{B}_4\Big)(\widehat{\boldsymbol{w}}^{(4,1)} - \boldsymbol{w}^{(4,0)}) = -\boldsymbol{w}^{(4,0)} + z\boldsymbol{\gamma}_4 y_m + z\mathcal{B}_4(\mathbf{1}_4 y_m + \boldsymbol{w}^{(4,0)}),$$

or equivalently

$$\Big(\mathcal{I}_4 - z\widehat{\mathcal{B}}_4\Big)\widehat{\boldsymbol{w}}^{(4,1)} = \Big(\mathcal{I}_4 - \widehat{\mathcal{B}}_4\mathcal{B}_4^{-1}\Big)\boldsymbol{w}^{(4,0)} + z\widehat{\mathcal{B}}_4\mathcal{B}_4^{-1}\boldsymbol{\gamma}_4 y_m + z\widehat{\mathcal{B}}_4\mathbf{1}_4 y_m. \tag{3.11}$$

By the definition of $\boldsymbol{w}^{(4,0)}$ in (3.1) and the equation (2.16), each component $\boldsymbol{w}_j^{(6,k^*)}$ of $\boldsymbol{w}^{(4,0)}$ can be calculated by

$$\boldsymbol{w}_j^{(6,k^*)} = \Big(\mathcal{S}_j(z) - 1\Big)y_m, \quad \mathcal{S}_j(z) = \frac{\det\begin{pmatrix} \mathcal{I}_6 - z\mathcal{B}_6 & \mathbf{1}_6 + z\boldsymbol{\gamma}_6 \\ -z\mathbf{a}_j^{\mathrm{T}} & 1 + za_{j0} \end{pmatrix}}{\det\Big(\mathcal{I}_6 - z\mathcal{B}_6\Big)},$$

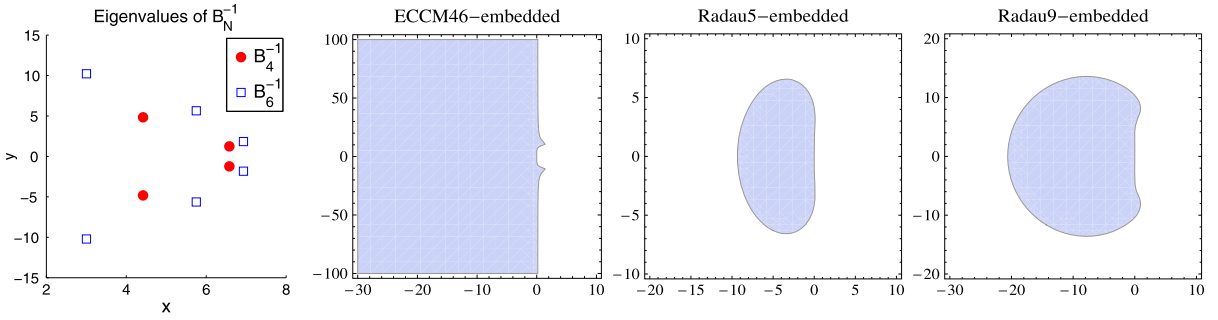

Fig. 1. The stability regions for the lower order solution obtained from our proposed method $\widehat{\mathcal{S}}_4(z)$ and the Radau schemes ($\widetilde{\mathcal{S}}_s(z)$, $s=3,5$).

where $\mathbf{a}_j$ and a_{j0} denote the jth column of the matrix $\mathcal{B}_6^{\mathrm{T}}$ and the vector $\boldsymbol{\gamma}_6^{\mathrm{T}}$, respectively. Thus, if we define a vector $\boldsymbol{\theta}$ by

$$\boldsymbol{\theta} := [\mathcal{S}_1(z)-1, \cdots, \mathcal{S}_4(z)-1]^{\mathrm{T}},$$

then the equation (3.11) can be simplified by

$$\left(\mathcal{I}_4 - z\widehat{\mathcal{B}}_4\right)\widehat{\boldsymbol{w}}^{(4,1)} = \boldsymbol{\eta}(z)y_m, \quad \boldsymbol{\eta}(z) := \left(\mathcal{I}_4 - \widehat{\mathcal{B}}_4\mathcal{B}_4^{-1}\right)\boldsymbol{\theta} + z\widehat{\mathcal{B}}_4\mathcal{B}_4^{-1}\boldsymbol{\gamma}_4 + z\widehat{\mathcal{B}}_4\mathbf{1}_4. \tag{3.12}$$

Applying Cramer's rule to (3.12), the 4th component $\widehat{\boldsymbol{w}}_4^{(4,1)}$ of $\widehat{\boldsymbol{w}}^{(4,1)}$ can be obtained by

$$\widehat{\boldsymbol{w}}_4^{(4,1)} = \frac{\det\begin{pmatrix} \mathcal{I}_4 - z\widehat{\mathcal{B}}_4 & \boldsymbol{\eta}(z) \\ -z\widehat{\mathbf{a}}_4^{\mathrm{T}} & \widehat{a}_{40} \end{pmatrix}}{\det\left(\mathcal{I}_4 - z\widehat{\mathcal{B}}_4\right)} y_m,$$

where $\widehat{\mathbf{a}}_4$ and $\widehat{a}_{40}$ denote the 4th column of the matrix $\widehat{\mathcal{B}}_4^{\mathrm{T}}$ and the vector $\boldsymbol{\eta}(z)^{\mathrm{T}}$, respectively. Finally, the lower order solution $\widehat{y}_{m+1}$ at time $t=t_{m+1}$ for the Dahlquist's problem becomes

$$\widehat{y}_{m+1} = y_m + \widehat{\boldsymbol{w}}_4^{(4,1)} = \widehat{\mathcal{S}}_4(z)y_m, \quad \widehat{\mathcal{S}}_4(z) = 1 + \frac{\det\begin{pmatrix} \mathcal{I}_4 - z\widehat{\mathcal{B}}_4 & \boldsymbol{\eta}(z) \\ -z\widehat{\mathbf{a}}_4^{\mathrm{T}} & \widehat{a}_{40} \end{pmatrix}}{\det\left(\mathcal{I}_4 - z\widehat{\mathcal{B}}_4\right)}. \tag{3.13}$$

The lower order solution of the Dahlquist's problem by the Radau scheme with s stages is stated as follows

$$\begin{aligned} \widetilde{y}_{m+1} &= y_m + h\widetilde{\lambda}^{-1} f(t_m, y_m) + h\sum_{j=1}^{s} \widetilde{b}_j f(t_m + c_j^s h, g_j) \\ &= \widetilde{\mathcal{S}}_s(z)y_m, \end{aligned}$$

where $\widetilde{\lambda}$ is the real eigenvalue of the matrix $\mathcal{B}_s^{-1}$, g_j $(j=1,2,\cdots,s)$ are the values obtained from the Radau method, and $\widetilde{b}_j$ are to be determined such that $\widetilde{y}_{m+1} - y_{m+1} = \mathcal{O}(h^{s+1})$. A detailed formula for $\widetilde{\mathcal{S}}_s(z)$ can be seen in [7, p. 123]. For comparison, we present the stability region, Γ defined in (2.19) with $\widehat{\mathcal{S}}_4(z)$, of the proposed method in Fig. 1 (second) and those of the Radau schemes in Fig. 1 (third, fourth). The first panel of Fig. 1 shows the distribution of the eigenvalues for $\mathcal{B}_N^{-1}$. It can be seen that the stability region of the proposed scheme for the lower order solution has a better stability region than those of the Radau schemes.

Remark 3.4.

(1) For the pair of solutions $\boldsymbol{\alpha}_4^N$ and $\widehat{y}_{m+1}$ of the Dahlquist's problem obtained from the systems (2.16) and (3.13), respectively, an asymptotic behavior of the local truncation errors $\widehat{err}(z) = |\boldsymbol{\alpha}_4^6 - \widehat{y}_{m+1}|$ and $err(z) = |\boldsymbol{\alpha}_4^6 - \boldsymbol{\alpha}_4^4|$ can be seen in Fig. 2. It can be seen that not only have both errors, $\widehat{err}(z)$ and $err(z)$, the asymptotic property that $\widehat{err}(z), err(z) \to 0$ as $z \to \infty$, but also both their magnitudes are less than 1.

(2) It is well known that the estimation $\widehat{err}(z)$ for the local truncation error in the Radau method goes to -1 as $z \to \infty$; hence, it is not completely free from the so-called "hump" phenomenon. To overcome this problem, Hairer and Wanner [7] suggested an adjusted error estimation recalculated by one additional function evaluation for an adjusted initial value. In contrast, the proposed scheme has a well-behaved asymptotic behavior without any extra function evaluations, which is remarkable.

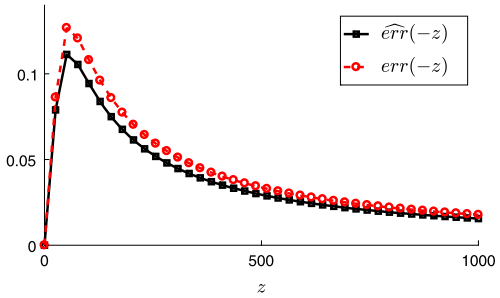

Fig. 2. The estimate errors for $y' = \lambda y$ with different error estimation schemes versus $z = \lambda h$.

Finally, based on (2.6), (2.15), (3.8), (3.10), and the discussion above, we introduce the pseudo code for our algorithm ECCM46.

ALGORITHM ECCM46(f, $[t_0, t_{end}]$, y_0, Rtol, Atol)
Remark: The algorithm ECCM46 is the Chebyshev collocation method based on zeros of the generalized Chebyshev polynomials defined in (1.6). It calculates the approximate solution of (1.1). Here, $[t_0, t_{end}]$ is the required integration interval, and y_0 is a given initial value. Furthermore, the relative and absolute tolerances are given as Rtol and Atol, respectively.

1. Initialize h, $m = 0, t_m := t_0$, $y_m := y_0$.
2. If $t_m \geq t_{end}$, then exit.
3. If $t_m + h > t_{end}$, then $h = t_{end} - t_m$.
4. Perform the iteration scheme (2.14) and (2.15), and then calculate y_{m+1} of (2.6) with the tolerance ϵ defined by (2.5).
5. Solve the linear system (3.10) to obtain $\widehat{\boldsymbol{w}}^{(4,1)}$ and then calculate $\widehat{y}_{m+1}$ by using (3.8).
6. Compute the error $err := \sqrt{\frac{1}{d}\sum_{j=1}^{d}\left(\frac{y_{m+1,j}-\widehat{y}_{m+1,j}}{\text{Atol}+max(|y_{m,j}|,|y_{m+1,j}|)\,\text{Rtol}}\right)^2}$.
7. If $err < 1$, then save $t_m := t_m + h$, $y_m := y_{m+1}$. After setting the next time step size h with the standard step size controller (see [7,18]), go to step 2.
8. If $err \geq 1$, then resize h using the traditional step size controller (see [7,18]) and go to step 4.

4. Numerical results

In this section, in order to demonstrate the efficiency of our method (ECCM46), it is tested by five widely used physical problems. For the numerical comparison, three other implicit methods were used: Radau5, Radau9 [7], and E^2CCM46 [18]. The efficiency of each numerical algorithm was measured in terms of the computational time (cputime), the number of function evaluations (nfeval), and the number of acceptance time steps (naccept) required to solve each problem, by varying the relative tolerances (Rtol) and absolute tolerances (Atol). As shown in Fig. 3 and Fig. 5, the marked points in all numerical comparisons, going from left to right, correspond to the errors as the tolerances are reduced in size. In each numerical experiment, the cputime was determined by taking the average cputime of 200 executions of the numerical schemes for a small scale problem, or the average of 10 executions for a large scale problem, in order to get an accurate measurement. All numerical tests were executed using the Visual Studio 2010 software, written in C++, and in a Windows 7 operating system.

4.1. Prothero–Robinson equation

In this example, the order of numerical convergence was evaluated by using a well-known Prothero–Robinson equation [19] given by

$$\phi'(t) = \nu(\phi(t) - g(t)) + g'(t), \qquad t \in (0, 20]; \quad \phi(0) = 0,$$

where ν is a negative constant and $g(t) = \sin(t)$. Note that the exact solution of this problem is $\phi(t) = \sin(t)$, which does not exhibit stiffness, but the magnitude of the eigenvalue ν constitutes the stiffness of the problem. The maximum error $Err(h) = \max_m\{|\phi(t_m) - y_m|\}$ and the rate of convergence $Rate(h) = \log(Err(h)/Err(h/2))/\log 2$ were calculated for a given step size h, with different eigenvalues $\nu = -10^k$, $k = 0, 6$; the results for a selection of step sizes h are shown in Table 2. The table shows that the method has 8th order of numerical convergence for the non-stiff case ($\nu = -1$), which guarantees the order of theoretical convergence discussed in Section 3. Moreover, it has the 6th order of numerical convergence for

Table 2
Order of convergence.

n	$\nu = -10^0$ (non-stiff)		$\nu = -10^6$ (stiff)	
	$Err(h)$	$Rate(h)$	$Err(h)$	$Rate(h)$
−2	2.3599×10^{-4}	–	5.1828×10^{-9}	–
−1	8.2026×10^{-7}	8.19	4.7815×10^{-11}	6.76
0	3.4361×10^{-9}	7.90	6.8093×10^{-13}	6.13
1	1.3599×10^{-11}	7.98	1.0464×10^{-14}	6.02
2	5.8842×10^{-14}	7.85	6.8001×10^{-16}	3.94

Table 3
Comparisons of ECCM46 and ECCM46O for Burgers' equation with $\nu = 0.01$.

Method	Rtol = 100 Atol	Error	naccept	nfeval	njac	ndec	cputime
ECCM46	1.0e−4	6.0362e−6	12	204	6	12	18.285
	1.0e−5	2.3134e−6	14	236	7	13	19.758
ECCM46O	1.0e−4	4.6160e−6	11	215	7	12	27.946
	1.0e−5	1.5295e−7	12	240	6	12	28.497

Table 4
Comparisons of ECCM46 and ECCM46O for Burgers' equation with $\nu = 0.001$.

Method	Rtol = 100 Atol	Error	naccept	nfeval	njac	ndec	cputime
ECCM46	1.0e−4	6.4094e−4	16	370	12	18	28.739
	1.0e−5	6.4260e−4	18	426	12	18	29.184
ECCM46O	1.0e−4	6.4442e−4	18	426	18	24	54.084
	1.0e−5	6.4215e−4	17	485	13	19	46.120

the stiff case ($\nu = -10^6$), which is not surprising as our method is similar to the Lobatto IIIA method ([7, p. 226]), and the global error with a uniform time step size h can be expressed as

$$\phi(t_{m+1}) - y_{m+1} = \mathcal{S}(z)(\phi(t_m) - y_m) + O(z^{-1}h^7), \quad \text{when } h \to 0 \text{ and } z = h\nu \to \infty,$$

where $\mathcal{S}(z)$ is the stability function.

4.2. Burgers' equation

In this example, the effectiveness of the eigenvalue modification for calculating the lower order solution from the systems (3.10) was examined. We denote the method without the eigenvalue modification as ECCM46O. As a test problem, the following one-dimensional Burgers' equation was used:

$$u_t + uu_x = \nu u_{xx}, \quad u(0,x) = -\sin(\pi x),\ u(t,-1) = u(t,1) = 0,\ t \in (0,1], \quad x \in [-1,1].$$

For the spatial discretization, the fourth-order, compact finite difference scheme was used for both u_x, and u_{xx}, with a spatial grid size $\Delta x = \frac{2}{1024}$, and the method of lines by varying the viscosity $\nu = 0.01, 0.001$ was applied. The results are listed in Tables 3 and 4. In the tables, njac and ndec denote the total number of Jacobians and LU decompositions, respectively, needed to simulate the problem. It can be seen in Table 3 that the cputime and the computational costs of ECCM46 from nfeval and ndec, are better than those of ECCM46O, which supports the proposed theoretical modification. However, from the magnitude of the numerical errors listed in the third column of Table 3, it can be seen that some truncation errors were introduced by the modification of the eigenvalues, and hence the results of ECCM46O are slightly better than those of ECCM46. Nevertheless, it can be seen in Table 4, that for the relatively small viscosity $\nu = 0.001$ the computational costs of ECCM46 are much better than those of ECCM46O. In contrast, the numerical errors from both methods have similar behavior, which may be caused by the strong stiffness of Burgers' equation.

4.3. Oregonator

For the third example, the Oregonator was investigated, which is the simplest model of the chemical dynamics of the oscillatory Belousov–Zhabotinsky reaction. The problem is a stiff system consisting of three non-linear ODEs given by

$$\frac{d\phi}{dt} = f(\phi), \quad t \in [0, 360]; \quad \phi(0) = [1, 2, 3]^{\mathrm{T}},$$

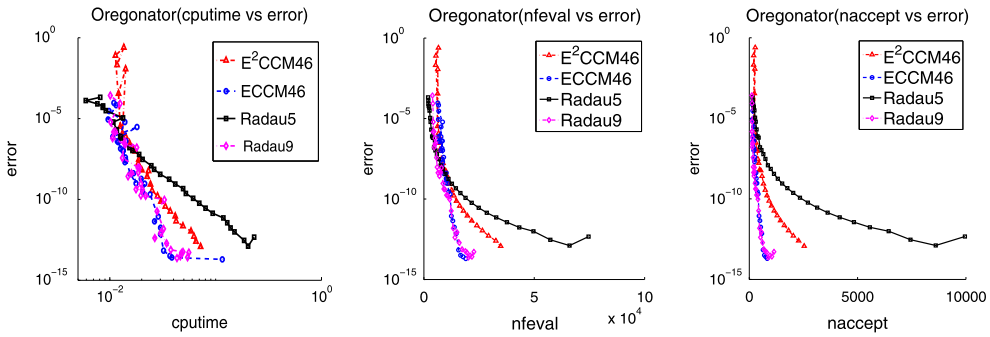

Fig. 3. Comparisons of errors with function of cputime (left), function of nfeval (center) and function of naccept (right) with tolerances $(\text{Rtol}, \text{Atol}) = (10^{-2-n/4}, 10^{-4-n/4})$, $n = 0, 1, \cdots, 32$ for the Oregonator problem.

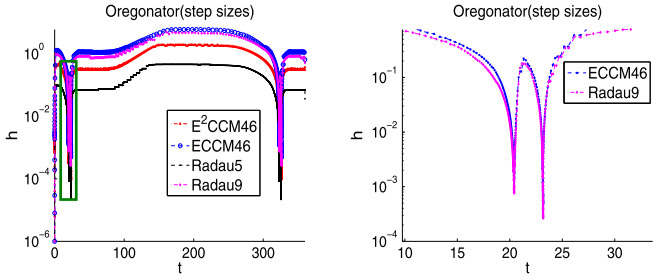

Fig. 4. Time step variation (left) in the whole time interval and the detailed figures (right) near the stiff region with $(\text{Rtol}, \text{Atol}) = (10^{-10}, 10^{-12})$.

where

$$\phi = [\phi_1, \phi_2, \phi_3]^{\mathrm{T}} \quad \text{and} \quad f(\phi) = \begin{bmatrix} 77.27(\phi_2 - \phi_1\phi_2 + \phi_1 - 8.375 \times 10^{-6}\phi_1^2) \\ (-\phi_2 - \phi_1\phi_2 + \phi_3)/77.27 \\ 0.161(\phi_1 - \phi_3) \end{bmatrix}.$$

In the absence of an analytic solution, the following reference solution was used, generated by Radau5 with $\text{Rtol} = \text{Atol} = 1.1 \times 10^{-18}$, at the end time $t = 360$,

$$\phi_1 = 0.1000814870318523 \times 10, \phi_2 = 0.1228178521549917 \times 10^4, \phi_3 = 0.1320554942846706 \times 10^3.$$

The efficiency of the numerical algorithms were examined by three measurements, cputime, nfeval, and naccept with varying tolerances $(\text{Rtol}, \text{Atol}) = (10^{-2-n/4}, 10^{-4-n/4})$, $n = 0, 1, \cdots, 32$, required to obtain the numerical solution at the given end time. In Fig. 3, the relative l_2-norm errors of the numerical solutions corresponding to each tolerance are shown as a function of the measurements, on a logarithmic scale. The numerical results of E^2CCM46 corresponding to the tolerances $(\text{Rtol}, \text{Atol}) = (10^{-2-n/4}, 10^{-4-n/4})$, $n = 0, 1, 2, 3$ are not included since the method does not work for the cases, which may come from the usage of the Chebyshev interpolation polynomial $p_N(t)$ with the interpolation conditions $p_N(t_{mj}) = \phi(t_{mj})$ for the solution ϕ.

It can be seen that the error of ECCM46 shows far better performance in terms of cputime, nfeval, and naccept than the other methods Radau9, E^2CCM46, and Radau5, as the tolerance is reduced. For example, ECCM46 exhibits an accuracy of 13 digits with about 0.025 s cputime, 17000 nfeval and 500 naccept together, while Radau9, E^2CCM46, and Radau5 need about 0.028 s, 0.07 s, 0.15 s cputime, 20000, 35000, 65000 nfeval, and 500, 2400, 8000 naccept, respectively. To investigate the efficiency of the developed embedded formula, the change of step size for each method was compared, and the results with $(\text{Rtol}, \text{Atol}) = (10^{-10}, 10^{-12})$ are shown in Fig. 4 (left). With the fixed tolerance, it can be seen that the time step sizes generated with ECCM46 are larger than those of the other methods, which means the ECCM46 uses consistently larger time step sizes in the execution of the algorithm than other methods. Furthermore, the enlarged figures for ECCM46 and Radau9 near a stiff region (Fig. 4 (right)), show how the time step sizes of ECCM46 are changed near the stiff region, compared to those of Radau9. Again, the behavior we have seen is that ECCM46 has larger time step sizes, which is a significant result when we recall that the convergence order of ECCM46 is lower than that of Radau9. It is further evidence of the efficiency of the proposed embedded scheme.

4.4. Plate problem

In this example, the proposed method was used to determine the movement of a rectangular plate under the load of a car passing over it, which is known as the plate problem [7], and is described by

Table 5
Comparison of ECCM46 and Radau9 for the plate problems.

Method	Rtol = Atol	Error	nsteps	naccept	nfeval	newt
ECCM46	10^{-8}	6.2926e−010	66	56	530	1.1970
	10^{-9}	4.2344e−010	84	74	710	1.2619
	10^{-10}	6.7468e−012	116	104	1034	1.3362
	10^{-11}	3.0856e−013	127	119	1337	1.5984
	10^{-12}	5.9130e−014	188	179	1913	1.5372
Radau9	10^{-8}	6.7808e−009	54	52	532	1.7778
	10^{-9}	1.0996e−009	75	73	768	1.8533
	10^{-10}	1.0254e−010	108	106	1116	1.8704
	10^{-11}	2.6386e−011	161	159	1669	1.8758
	10^{-12}	7.3572e−013	242	241	2526	1.8884

$$\frac{\partial^2 u}{\partial t^2} + \omega \frac{\partial u}{\partial t} + \sigma \Delta^2 u = f(x, y, t), \quad \Omega := [0, 2] \times [0, \frac{4}{3}]$$

with $\omega = 1000$, $\sigma = 100$ and the initial and boundary conditions

$$u|_{\partial\Omega} = 0, \quad \Delta u|_{\partial\Omega} = 0, \quad u(x, y, 0) = 0, \quad \frac{\partial u}{\partial t}(x, y, 0) = 0.$$

Here, the load $f(x, y, t)$ was idealized by the sum of two Gaussian curves which move in the x-direction and reside on "four wheels"

$$f(x, y, t) = \begin{cases} 200\left(e^{-5(t-x-2)^2} + e^{-5(t-x-5)^2}\right) & \text{if } y = \frac{4}{9} \text{ or } \frac{8}{9}, \\ 0 & \text{for all other } y. \end{cases}$$

As a spatial discretization for the plate problem, using a uniform grid template for the plate Ω with 40 interior grid points $x_i = i\Delta x$, $y_j = j\Delta x$, where $\Delta x = 2/9$ is the uniform grid size, the second order, central finite difference scheme was applied and a linear, and non-autonomous IVP was obtained with a medium stiffness. It is known that the Jacobian matrices for the IVP have a distributed spectrum, for which a higher stage order method can have an advantage over other methods [21]. To show this, the problem was solved on the integration interval $0 \le t \le 7$ and the numerical results from ECCM46 were compared with those from Radau9. The results are summarized in Table 5, and show that the numerical error of ECCM46 is far superior to that of Radau9. Also, ECCM46 uses a smaller number of integration trial steps (nsteps), and nfeval than Radau9. The increased efficiency results from the higher stage order of ECCM46 compared to Radau9 which has stage order 5.

4.5. Medical Akzo Nobel problem

As a final example, the problem known as the Medical Akzo Nobel problem (Medakzo) [1,4] was solved. This problem consists of coupled reaction–diffusion equations, and their spatial discretization yields a stiff ODE given by

$$\frac{dy}{dt} = f(t, y), \quad t \in [0, 20]; \quad y(0) = g \in \mathbb{R}^{2d}, \tag{4.1}$$

where the function f is defined by

$$\begin{cases} f_{2j-1} = \alpha_j \dfrac{y_{2j+1} - y_{2j-3}}{2\Delta x} + \beta_j \dfrac{y_{2j-3} - 2y_{2j-1} + y_{2j+1}}{\Delta x^2} - 100 y_{2j-1} y_{2j}, \\ f_{2j} = -100 y_{2j-1} y_{2j}, \quad j = 1, 2, \cdots, d. \end{cases}$$

Here, $\alpha_j = \frac{2(j\Delta x - 1)^3}{4^2}$, $\beta_j = \frac{(j\Delta x - 1)^4}{4^2}$, $\Delta x = \frac{1}{d}$, the initial condition g is given by $g = (0, 1, 0, 1, \cdots, 0, 1)^{\mathrm{T}}$, and the boundary conditions $y_{-1}(t)$ and $y_{2d+1}(t)$ are given by $y_{2d+1} = y_{2d-1}$, $y_{-1}(t) = \psi(t)$ with the step function $\psi(t)$ defined by

$$\psi(t) = \begin{cases} 2 & \text{for } t \in (0, 5], \\ 0 & \text{for } t \in (5, 20]. \end{cases}$$

The boundary conditions show that f undergoes a discontinuity in time at $t = 5$. Hence, we take this IVP as a benchmark problem to show that the proposed scheme works well in the case with discontinuities. The problem (4.1) was simulated with various tolerances $(\mathrm{Rtol}, \mathrm{Atol}) = (10^{-2-n/4}, 10^{-2-n/4})$, $n = 0, 1, \cdots, 32$, and $d = 1000$ was taken to achieve a high resolution of the spatial accuracy. To compare the efficiency, the reference solution obtained from Radau9 was used with $(\mathrm{Rtol}, \mathrm{Atol}) = (10^{-13}, 10^{-15})$, and three measurements, cputime, nfeval, and naccept were performed which was required to

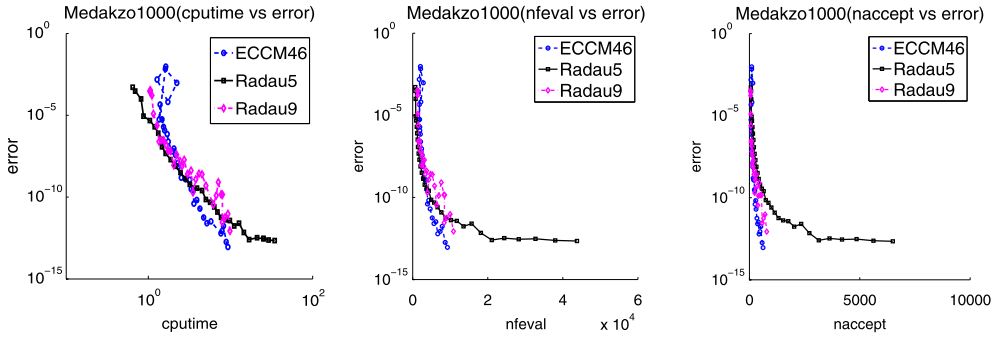

Fig. 5. Comparison of errors with function of cputime (left), function of nfeval (middle) and function of naccept (right) by varying tolerances Rtol = Atol = $10^{-2-n/4}$, $n = 0, 1, \cdots, 32$ for the Medakzo problem.

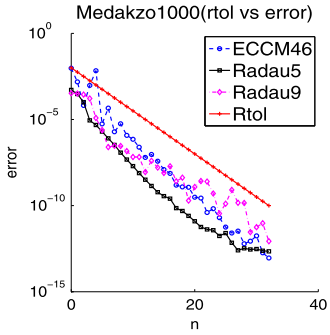

Fig. 6. Reliability of control for error behavior by varying tolerances Rtol = Atol = $10^{-2-n/4}$, $n = 0, 1, \cdots, 32$ for the Medakzo problem.

obtain the numerical solution at the end time. The relative l_2-norm errors of the numerical solutions corresponding to each tolerance are shown in Fig. 5 as a function of the measurements, on a logarithmic scale. It can be seen that ECCM46 yields to a better performance in terms of cputime, nfeval, and naccept compared to other methods. Moreover, the smaller the tolerance, the better the performance of ECCM46 was. For example, ECCM46 achieves an accuracy of 10 digits with about 2.5 s cputime, 3000 nfeval, and 200 naccept, while Radau5 and Radau9 need about 6 s and 7 s cputime, 7000 and 8000 nfeval, and 500 and 1000 naccept, respectively.

Finally, to show the reliability of the control of error estimate in our method, we demonstrate how the obtained accuracy depends on the control parameter in Fig. 6. As expected, the behavior of both higher order methods (Radau9 and ECCM46) corresponds to the given tolerances more consistently compared to the lower order method (Radau5). In addition, it can be seen that the proposed method ECCM46 behaves well without any oscillation compared to Radau9, as the tolerance goes to small.

5. Conclusion and further discussion

An embedded formula of the Chebyshev collocation method for stiff problems has been constructed and analyzed. Instead of the traditional method to estimate the local truncation error, we developed a new embedded formula to calculate the lower order solution. To achieve this, the eigenvalues of the complex linear systems, derived from the collocation method for the lower order solution, are replaced by those of the higher order solution. The proposed embedded scheme yields to a wide stability region for the Dahlquist's problem. The local truncation error, $err(z)$, of the scheme goes to 0, as z goes to ∞, which is similar to the asymptotic behavior of another widely used method, Radau5. The convergence and stability analysis shows that the scheme has 8th order of convergence and A-stability. By using several numerical experiments, we have shown that the scheme has a higher order of convergence, better stability and lower computational costs, compared to existing methods.

Acknowledgements

This research was supported by the basic science research program through the National Research Foundation of Korea (NRF), and funded by the Ministry of Education, Science and Technology (grant number 2016R1A2B2011326). The first author Piao was supported by the 2015 Hannam University Research Fund (Project No. 2015A013), supported by the R&D programs through NFRI (National Fusion Research Institute) funded by the Ministry of Science and ICT of the Republic of Korea (grant number NFRI-EN1741-3), and funded by Basic Science Research Program through the National Research Foun-

dation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (grant number NRF-2017R1C1B1002370). The second author Bu was supported by the basic science research program through the National Research Foundation of Korea (NRF), and funded by the Ministry of Education, Science and Technology (grant number 2016R1D1A1B03930734).

References

[1] https://www.dm.uniba.it/~testset/testsetivpsolvers.
[2] J.C. Butcher, Implicit Runge–Kutta processes, Math. Comput. 18 (1964) 50–64.
[3] G.J. Cooper, J.C. Butcher, An iteration scheme for implicit Runge–Kutta methods, IMA J. Numer. Anal. 3 (1983) 127–140.
[4] U.K.S. Din, F. Ismail, Z.A. Majid, R.R. Ahmad, Solving Medical Akzo Nobel problem using functional load balancing algorithm of 4(3) DIRK method, Int. J. Mod. Phys. Conf. Ser. 9 (2012) 480–487.
[5] K. Gustafsson, Control-theoretic techniques for stepsize selection in implicit Runge–Kutta methods, ACM Trans. Math. Softw. 20 (4) (1994) 496–517.
[6] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, Springer Ser. Comput. Math., Springer, 1993.
[7] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Stiff and Differential–Algebraic Problems, Springer Ser. Comput. Math., Springer, 1996.
[8] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration, Springer Ser. Comput. Math., Springer, 2002.
[9] T. Hasegawa, T. Torii, I. Ninomiya, Generalized Chebyshev interpolation and its application to automatic quadrature, Math. Comput. 41 (1983) 537–553.
[10] T. Hasegawa, T. Torii, H. Sugiura, An algorithm based on the FFT for a generalized Chebyshev interpolation, Math. Comput. 54 (189) (1990) 195–210.
[11] C. Johnson, 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) (1988) 908–926.
[12] P. Kim, A Chebyshev quadrature rule for one sided finite part integrals, J. Approx. Theory 111 (2) (2001) 196–219.
[13] P. Kim, U.J. Choi, Two trigonometric quadrature formulae for evaluating hypersingular integrals, Int. J. Numer. Methods Eng. 56 (3) (2003) 469–486.
[14] P. Kim, X. Piao, S.D. Kim, An error corrected Euler method for solving stiff problems based on Chebyshev collocation, SIAM J. Numer. Anal. 49 (6) (2011) 2211–2230.
[15] S.D. Kim, X. Piao, D.H. Kim, P. Kim, Convergence on error correction methods for solving initial value problems, J. Comput. Appl. Math. 236 (17) (2012) 4448–4461.
[16] P. Kim, S.D. Kim, E. Lee, Simple ECEM algorithms using function values only, Kyungpook Math. J. 53 (4) (2013) 573–591.
[17] S.D. Kim, J. Kwon, X. Piao, P. Kim, A Chebyshev collocation method for stiff initial value problems and its stability, Kyungpook Math. J. 51 (4) (2011) 435–456.
[18] P. Kim, J. Kim, W. Jung, S. Bu, A higher order error embedded method based on generalized Chebyshev polynomials, J. Comput. Phys. 306 (2016) 55–72.
[19] A. Prothero, A. Robinson, On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations, Math. Comput. 28 (125) (1974) 145–162.
[20] H. Ramos, J. Vigo-Aguiar, A fourth order Runge–Kutta method based on BDF-type Chebyshev approximations, J. Comput. Appl. Math. 204 (1) (2007) 124–136.
[21] L.M. Skvortsov, Singly implicit diagonally extended Runge–Kutta methods of fourth order, Comput. Math. Math. Phys. 54 (5) (2014) 755–765.
[22] J. Vigo-Aguiar, H. Ramos, A family of A-stable Runge–Kutta collocation methods of higher order for initial-value problems, IMA J. Numer. Anal. 27 (2007) 798–817.
[23] K. Wright, Chebyshev collocation methods for ordinary differential equations, Comput. J. 6 (1963) 358–363.