

Invariants preserving schemes based on explicit Runge–Kutta methods

H. Kojima¹

Received: 30 March 2015 / Accepted: 30 January 2016 / Published online: 12 February 2016
© Springer Science+Business Media Dordrecht 2016

Abstract Numerical integration of ordinary differential equations with some invariants is considered. For such a purpose, certain projection methods have proved its high accuracy and efficiency. Unfortunately, however, sometimes they can exhibit instability. In this paper, a new, highly efficient projection method is proposed based on explicit Runge–Kutta methods. The key there is to employ the idea of the perturbed collocation method, which gives a unified way to incorporate scheme parameters for projection. Numerical experiments confirm the stability of the proposed method.

Keywords Initial value problems · Perturbed collocation methods · Numerical geometric integration · Projection methods · Explicit Runge–Kutta methods

Mathematics Subject Classification 65L05 · 65L06

1 Introduction

In this paper, we are concerned with numerical integration of ordinary differential equations with some invariants. For example, let us consider the Hamiltonian systems:

$$\frac{dy}{dt} = J \nabla H(y(t)), \quad y(0) = y_0 \in \mathbb{R}^N, \quad (1.1)$$

Communicated by Mechthild Thalhammer.

✉ H. Kojima
h.k.psi.mot@gmail.com; hiroki_kojima@mist.i.u-tokyo.ac.jp

¹ Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

where J is the symplectic matrix and $H : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is the Hamiltonian. We also call it an energy function. From the formulation (1.1), we see that the exact solution of the Hamiltonian system preserves the energy function H :

$$\frac{d}{dt}H(y(t)) = \nabla H(y(t))^\top \frac{dy}{dt} = \nabla H(y(t))^\top J \nabla H(y(t)) = 0, \quad (1.2)$$

where we used the chain rule and the skew symmetry of J . Such ODEs with (possibly multiple) invariants appear in wide range of modern science and engineering.

For such systems, we hope that numerical solutions preserve the invariants, but generally such expectation does not come true. In fact, Runge–Kutta (RK) methods cannot preserve energy function in general [7]. Hence, in these several decades, several invariants-preserving methods have been constructed (e.g. [1, 3, 9, 18, 20]). Then it has been proved that such methods are in fact stabler than general RK methods, and often well compare with other geometric numerical integration methods, such as symplectic methods for Hamiltonian systems [13].

Energy-preserving methods for the Hamiltonian systems (1.1) (or more generally, systems with invariants) can be divided into two groups: (i) methods whose numerical solutions automatically satisfy $H(y_{n+1}) = H(y_n)$ without using H or ∇H explicitly (e.g. [9, 17, 19]), and (ii) methods that first get a temporary solution \tilde{y}_{n+1} by some method, and obtain the next solution y_{n+1} by projecting the temporary solution \tilde{y}_{n+1} to some y_{n+1} satisfying $H(y_{n+1}) = H(y_n)$ (e.g. [2, 4, 8, 11]). Although methods in the first group generate stable numerical solutions, they demands the solution of at least N dimensional nonlinear system at every time step, thus demand much computation time when N is large. Even worse, most of them have only second order accuracy, and demand extra computational cost when further accuracy is necessary [12, 21]. In contrast to this, some methods in the second group only require the solution of a single nonlinear equation (instead of N) in every time step, while keeping the same accuracy as the base integrator. Therefore, from the view of actual computation, the second group seems more promising unless we can count on some fast nonlinear solvers.

Here we review typical methods of the second group. Orthogonal projection method (OPM) [13] projects \tilde{y}_{n+1} along $\nabla H(\tilde{y}_{n+1})$ to obtain the next solution y_{n+1} satisfying $H(y_{n+1}) = H(y_n)$. Calvo–Hernández-Abreu–Montijano–Rández [4] proposed a projection method called “incremental direction technique” (IDT) that searches the next solution along the direction $\tilde{y}_{n+1} - \bar{y}_{n+1}$, where \bar{y}_{n+1} is the solution by an embedded Runge–Kutta method. In contrast to these linear methods, recently a method called EQUIP method that searches for y_1 nonlinearly has been proposed by Brugnano–Iavernaro–Trigiante [2]. The key of the method is the clever use of the W -transformation associated with collocation methods.

Next let us focus on their solvability and accuracy. The known results gave in the above mentioned studies can be reorganized in more general context as follows. Let us introduce the concept of “1-parameter projection method”: we first prepare a family of integrators $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ depending on 1-parameter α ; then, in every time step, we adjust α such that $H(\Phi_\alpha(y_0)) = H(y_0)$ holds. This concept includes the OPM, IDT and EQUIP, and forms a wider class of energy-preserving methods. By an obvious generalization of the result of [2], we can prove that “1-parameter projection method”

is solvable and has the same order of accuracy as the base integrator Φ under quite reasonable assumptions (see Theorem 2). This implies, at least *theoretically*, that all 1-parameter projection methods achieve high accuracy with low computational cost.

Although this gives a clear support for all the 1-parameter projection methods, here a natural question arises from the viewpoint of actual computation: which method among all the “1-parameter projection methods” is *actually* the best in terms of stability and computational efficiency? Moreover, how can we construct it systematically? Theoretical results mentioned above are of course mathematically quite important, but do not answer these questions, because they hold only for *sufficient small time step* h , and thus do not necessarily provide information on the behavior under practical size of h . Some new perspective is needed to answer these questions. To this end, let us again review each existing 1-parameter projection method and consider whether it can be the ideal answer for the questions. OPM and IDT introduced before may be the answers, since they can be arbitrary accurate and only require the solution of a 1-dimensional nonlinear equation, i.e., they are highly efficient. Between these methods, IDT is generally considered to be better because it preserves linear invariants; thus let us below consider IDT with more details. IDT searches the next solution linearly along the difference of two numerical solutions constructed by an embedding Runge–Kutta method with no extra function evaluation. Thus this method needs surprisingly low computational costs while keeping the original high accuracy, as mentioned above. In this sense, IDT is quite a good method. However, it has a slight drawback that when the direction $\tilde{y}_{n+1} - \bar{y}_{n+1}$ happens to be almost orthogonal to ∇H , the projection can be quite unstable, i.e., the next solution might be far away from the exact solution. In other words, the correction of the projection can be huge, and that may lead to some instability. One way of working around this is to employ some adaptive step size technique. In this case, IDT is really a good method, and practical. Nevertheless, we feel that this observation reveals the possible essential limitation of the linear search methods—they can yield fatal projection steps when they fail to suggest good directions. This can happen in all linear search methods including OPM.

Another good candidate as the best 1-parameter projection method is EQUIP method. Since it is based on the W-transformation, its parameterization does not destroy the symplecticity of the Gauss method, which was the main subject in [2]. In the present paper, however, we like to see the method from a different perspective. In [2], the authors chose to discard linear search, and turn to a completely new idea of projection, where they introduced a clever, natural parameterization of the Gauss method. There, the “search” is no longer linear, but thanks to the new view as a parameterized Gauss method, the correction by the projection is expected to remain relatively small, avoiding possible fatal projections in linear search methods. In fact, in [2], numerical examples show that EQUIP runs stably with constant time steps. However, because the Gauss method is implicit, EQUIP method also demands the solution of high dimensional nonlinear equations in every time step. Therefore, although EQUIP method can generate much stabler numerical solutions, we like to continue the journey to “the best” method, unless the computational cost can be substantially decreased by, for example, the clever use of multiple core implementations.

From the discussion above, we are encouraged to construct 1-parameter projection methods based on *explicit* Runge–Kutta methods for low computational cost. At the

same time, we hope to keep the good nature of EQUIP, i.e., the *natural* parameterization of such RK methods for their stability. If we succeed in this challenge, we expect that such methods can be the definitive answer for the questions.

The key idea for this challenge in the present paper is as follows. Recall that the natural parameterization in EQUIP was done by viewing the Gauss method as collocation method. But exactly this very trick implied the implicitness of the resulting schemes. Then, how can we do a similar thing based on *explicit* RK methods? The idea here is that we interpret explicit RK methods as perturbed collocation methods as proposed by Nørsett–Wanner [16]. With the aid of perturbation operators, perturbed collocation methods can express larger class of RK methods, including both explicit and implicit ones. Then we basically follow the strategy in EQUIP; we first consider a RK method and its perturbed collocation method representation. Then we try to find a natural way to parameterize the representation, so that the modification (which will result in the correction of the projection) remains as small as possible. The latter step is quite different from that of EQUIP. In EQUIP method, the small correction was made in terms of the W -transformed expression of the Gauss method. In the perturbed collocation method, however, no corresponding expression exists. Instead, in the present paper, we propose to introduce parameterization only in higher order perturbation terms. This sounds natural for achieving as small correction as possible. This task is, however, not straightforward as one would simply imagine. Since as described above the perturbed collocation method can express both explicit and implicit RK methods, inappropriate parameterization of explicit RK methods should destroy the explicitness and result in *implicit* schemes. This is not what we hope in terms of computational cost. In the present paper, we show that this difficulty can be worked around by using the characterization of explicitness obtained by Nørsett–Wanner [16]. We also would like to point out that the idea of perturbed collocation methods has not been intensively utilized recently, as far as the author understands—in this sense, the present work might be interesting for related researchers in that it breathes new life into perturbed collocation methods themselves.

This paper is organized as follows. In Sect. 2, we introduce perturbed collocation methods and the general theory for 1-parameter projection methods. In Sect. 3, we propose a new method via perturbed collocation methods. In Sect. 4, we test the proposed method and show its stability and efficiency comparing with existing methods. In Sect. 5, we give conclusions of this paper.

2 Preliminaries

2.1 Perturbed collocation methods

Here we give a brief review of perturbed collocation methods [16]. These methods are proposed mainly to expand the relation between the collocation methods and some implicit RK methods. In fact, perturbed collocation methods can express a larger class of RK methods, including some explicit RK methods.

First, we introduce the perturbation operator.

Definition 1 (*Perturbation operator [16]*) Let Π_s be the linear space of s -degree polynomials. Then the perturbation operator $P_{t_0,h} : \Pi_s \rightarrow \Pi_s$ is defined as below:

$$(P_{t_0,h}u)(t) := u(t) + \sum_{j=1}^s N_j \left(\frac{t-t_0}{h} \right) u^{(j)}(t_0)h^j, \tag{2.1}$$

where $N_j(t)$ ($j = 1, \dots, s$) are s -degree polynomials defined as

$$N_j(t) = \frac{1}{j!} \sum_{i=0}^s (p_{ij} - \delta_{ij}) t^i, \tag{2.2}$$

$$p_{ij} \in \mathbb{R} \quad (i = 0, 1, \dots, s; j = 1, 2, \dots, s), \tag{2.3}$$

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

For convenience, t_0 and h will be omitted when they are fixed. Using this perturbation operator, we define the perturbed collocation method.

Definition 2 (*Perturbed collocation methods [16]*) Let s be a positive integer and P a perturbation operator. There is an s -degree polynomial $u(t)$ satisfying the following conditions:

$$u(t_0) = y_0, \tag{2.5}$$

$$\dot{u}(t_0 + c_i h) = f(Pu(t_0 + c_i h)) \quad (1 \leq i \leq s). \tag{2.6}$$

Then the numerical solution of perturbed collocation methods is defined by $y_1 := u(t_0 + h)$.

From these definitions, we see perturbed collocation methods are an extension of collocation methods in the sense that they reduce to collocation methods when P is the identity operator.

Next we review the relation between perturbed collocation methods and RK methods.

Proposition 1 [16] *The perturbed collocation method defined by (2.5) and (2.6) is equivalent to a Runge–Kutta method whose coefficients are the entries of*

$$A = V_c P J V^{-1}, \tag{2.7}$$

$$b = (1, \dots, 1)^\top J V^{-1}, \tag{2.8}$$

where

$$V_c = \begin{pmatrix} 1 & c_1 & \dots & c_1^s \\ \vdots & & & \vdots \\ 1 & c_s & \dots & c_s^s \end{pmatrix}, \quad P = \begin{pmatrix} 1 & p_{01} & \dots & p_{0s} \\ 0 & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & p_{s1} & \dots & p_{ss} \end{pmatrix}, \tag{2.9}$$

$$J = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 0 & 1/2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1/s \end{pmatrix}, \quad V = \begin{pmatrix} 1 & c_1 & \dots & c_1^{s-1} \\ \vdots & & & \vdots \\ 1 & c_s & \dots & c_s^{s-1} \end{pmatrix}. \tag{2.10}$$

Conversely, a Runge–Kutta method that satisfies (2.8) and has distinct $\{c_i\}_{1 \leq i \leq s}$ can be expressed by a perturbed collocation method.

Remark 1 It is not unique how to express a RK method by perturbed collocation methods. Let us consider polynomial $M(t) = \prod_{i=1}^s (t - c_i)$. Adding this M to N_j (which compose the perturbation operator P) does not change the perturbed collocation method (see [16, Prop. 4]). From this observation, we see that we can always express a RK method by a perturbed collocation method whose perturbation operator is $P : \Pi_s \rightarrow \Pi_{s-1}$; in other words, $p_{sj} = 0$ ($j = 1, \dots, s$). Hence we assume $Pu \in \Pi_{s-1}$ in what follows.

This proposition shows that perturbed collocation methods form a subclass of RK methods. Moreover, in [16], a characterization for a perturbed collocation method being an explicit RK method was given. This characterization will play a key role in the clever parameterization of explicit RK methods in Sect. 3.

Proposition 2 [16] *We consider a perturbed collocation method that satisfies $Pu \in \Pi_{s-1}$ and define its RK coefficients triple by $(A, \mathbf{b}, \mathbf{c})$. Then the entry a_{ij} is zero if and only if the vector*

$$\left((N_1 + t)(c_i), \dots, (s - 1)! \left(N_s + \frac{t^s}{s!} \right) (c_i) \right)^\top \tag{2.11}$$

is a linear combination of

$$(1, \dots, c_k^{s-1})^\top \quad (k = 1, \dots, j - 1, j + 1, \dots, s). \tag{2.12}$$

Moreover, a perturbed collocation method which satisfies $Pu \in \Pi_{s-1}$ corresponds with an explicit Runge–Kutta method if and only if the perturbation operator P satisfies (2.12) for $1 \leq i \leq j \leq s$.

2.2 A general theory for 1-parameter projection methods

As we mentioned in the introduction, we introduce a general theory for 1-parameter projection methods. This theory is easily obtained by extending the results of Brugnano–Iavernaro–Trigiane [2] (similar results can be found in Calvo–Hernández–Abreu–Montijano–Rández [4]). First we define a family of integrators.

Definition 3 We call $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ a family of integrators with order reduction r when each integrator Φ_α ($\alpha \neq 0$) is well-defined and has order $p - r$, where p is the order of the “base integrator” with $\alpha = 0$, i.e., Φ_0 . We also define the solution curve $y_1(\alpha, h) := \Phi_\alpha(y_0)$.

In order to implement 1-parameter projection methods, we need to solve the following nonlinear equation in terms of α with h fixed:

$$g(\alpha, h) = H(y_1(\alpha, h)) - H(y_0). \tag{2.13}$$

Under this general setting, we can prove the existence of solutions of (2.13) and the recovery of the order. To prove them, we suppose that the following assumptions \mathcal{B}_1 and \mathcal{B}_2 hold for $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ for a fixed state y_0 .

\mathcal{B}_1 : $g(\alpha, h)$ is analytic near the origin.

\mathcal{B}_2 : Let d be the order of the error in the energy function H when the base method is applied:

$$g(0, h) = H(y_1(0, h)) - H(y_0) = c_0 h^d + \mathcal{O}(h^{d+1}), \tag{2.14}$$

where $c_0 \neq 0$. Then, for any fixed $\alpha \in \mathbb{R}$ ($\alpha \neq 0$),

$$g(\alpha, h) = c(\alpha)h^{d-r} + \mathcal{O}(h^{d-r+1}) \tag{2.15}$$

holds with $c'(0) \neq 0$.

The above setting basically follows [2]. Here we rephrase the underlying idea. From the fact that the base integrator Φ has order p , it is obvious d is generally $p + 1$. But it can happen that $d > p + 1$ holds coincidentally, and we do not hope to exclude such cases. Thus we define d as the lowest integer with nonzero c_0 and $c'(0)$. The order $p - r$ for $g(\alpha, h)$ comes from the fact that an integrator $\Phi \in \{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ with a fixed α has order $p - r$.

Under these assumptions we prove the existence of the solution for (2.14). This is essentially by Brugnano–Iavernaro–Trigiane [2], although there they focused on EQUIP method.

Theorem 1 *Let us assume \mathcal{B}_1 and \mathcal{B}_2 . Then there exists a positive number h_0^* and a function $\alpha^*(h)$ s.t.*

- (1) $g(\alpha^*(h), h) = 0, \quad h \in (-h_0^*, h_0^*),$
- (2) $\alpha^*(h) = \text{const} \cdot h^r + \mathcal{O}(h^{r+1}).$

Proof The following proof is a straightforward extension of the proof in [2]. From the assumptions, $g(\alpha, h)$ can be expanded as

$$g(\alpha, h) = \sum_{j=d}^{\infty} \frac{1}{j!} \frac{\partial^j g}{\partial h^j}(0, 0)h^j + \sum_{i=1}^{\infty} \sum_{j=d-r}^{\infty} \frac{1}{i!j!} \frac{\partial^{i+j} g}{\partial \alpha^i \partial h^j}(0, 0)\alpha^i h^j. \tag{2.16}$$

The first term comes from (2.14), and the second from (2.15). Then we hope to apply the implicit function theorem to search for a solution in the form of $\alpha(h) = \eta(h)h^r$, where $\eta(h)$ is a real-valued function of h . To this end, we consider the change of variable $\alpha = \eta h^r$ and insert it into (2.16) to get

$$\begin{aligned} g(\alpha, h) &= \frac{1}{d!} \frac{\partial^d g}{\partial h^d}(0, 0)h^d + \frac{1}{(d+1)!} \frac{\partial^{d+1} g}{\partial h^{d+1}}(0, 0)h^{d+1} \\ &+ \frac{1}{(d-r)!} \frac{\partial^{d-r+1} g}{\partial \alpha \partial h^{d-r}}(0, 0)h^d \eta + \frac{1}{(d-r+1)!} \frac{\partial^{d-r+2} g}{\partial \alpha \partial h^{d-r+1}}(0, 0)h^{d+1} \eta \\ &+ \mathcal{O}(h^{d+2}). \end{aligned} \tag{2.17}$$

Hence, for $h \neq 0$, $g(\alpha, h) = 0$ if and only if

$$\begin{aligned} \tilde{g}(\eta, h) &= \frac{1}{(d-r+1) \cdots (d-1)d} \frac{\partial^d g}{\partial h^d}(0, 0) + \frac{1}{(d-r+1) \cdots d(d+1)} \frac{\partial^{d+1} g}{\partial h^{d+1}}(0, 0)h \\ &+ \frac{\partial^{d-r+1} g}{\partial \alpha \partial h^{d-r}}(0, 0)\eta + \frac{1}{d-r+1} \frac{\partial^{d-r+2} g}{\partial \alpha \partial h^{d-r+1}}(0, 0)h\eta \\ &+ \mathcal{O}(h^2), \end{aligned} \tag{2.18}$$

is zero. By the assumption \mathcal{B}_2 , $\frac{\partial^{d-r+1} g}{\partial \alpha \partial h^{d-r}}(0, 0)$ is not zero, and we can apply the implicit function theorem which ensures that there is a real-valued function $\eta(h)$ satisfying $\tilde{g}(\eta(h), h) = 0$. In addition, from (2.18), $\eta(h)$ is calculated as follows. For sufficiently small h ,

$$\begin{aligned} \eta(h) &= -\frac{\frac{1}{(d-r+1) \cdots (d-1)d} \frac{\partial^d g}{\partial h^d}(0, 0) + \frac{1}{(d-r+1) \cdots d(d+1)} \frac{\partial^{d+1} g}{\partial h^{d+1}}(0, 0)h + \mathcal{O}(h^2)}{\frac{\partial^{d-r+1} g}{\partial \alpha \partial h^{d-r}}(0, 0) + \frac{1}{d-r+1} \frac{\partial^{d-r+2} g}{\partial \alpha \partial h^{d-r+1}}(0, 0)h} \\ &= -\frac{1}{(d-r+1) \cdots (d-1)d} \frac{\frac{\partial^d g}{\partial h^d}(0, 0)}{\frac{\partial^{d-r+1} g}{\partial \alpha \partial h^{d-r}}(0, 0)} + \mathcal{O}(h). \end{aligned} \tag{2.19}$$

Note that $\frac{\partial^d g}{\partial h^d}(0, 0)$ is not zero from \mathcal{B}_2 . This reveals that the solution of $g(\alpha, h) = 0$ for α takes the form

$$\alpha^*(h) = \eta(h)h^r = -\frac{1}{(d-r+1) \cdots (d-1)d} \frac{\frac{\partial^d g}{\partial h^d}(0, 0)}{\frac{\partial^{d-r+1} g}{\partial \alpha \partial h^{d-r}}(0, 0)} h^r + \mathcal{O}(h^{r+1}). \tag{2.20}$$

This completes the proof.

From this theorem 1-parameter projection method recovers its order to the same order as the base integrator.

Theorem 2 *Let us consider 1-parameter projection methods defined in Definition 3. These methods have the same order as the base integrator under the mild assumptions \mathcal{B}_1 and \mathcal{B}_2 .*

Proof The following proof is also an obvious replication of the proof of [2]. Then we apply the mean value theorem to obtain

$$y_1(\alpha, h) = y_1(0, h) + \alpha \int_0^1 \frac{\partial y_1}{\partial \alpha}(\tau\alpha, h) d\tau. \tag{2.21}$$

When α is fixed, the solution $y_1(\alpha, h)$ has the accuracy $\mathcal{O}(h^{p+1-r})$ because of Definition 3. On the other hand, the solution $y_1(0, h)$ provided by the base integrator is $\mathcal{O}(h^{p+1})$. Therefore we see

$$\int_0^1 \frac{\partial y_1}{\partial \alpha}(\tau\alpha, h) d\tau = \mathcal{O}(h^{p+1-r}). \tag{2.22}$$

If we define α^* as the solution of (2.14), then from Theorem 1, we have $\alpha^* = \mathcal{O}(h^r)$ thus

$$y_1(\alpha^*, h) - y(t_0 + h) = y_1(0, h) - y(t_0 + h) + \alpha^* \int_0^1 \frac{\partial y_1}{\partial \alpha}(\tau\alpha, h) d\tau = \mathcal{O}(h^{p+1}). \tag{2.23}$$

This completes the proof.

3 Construction of new schemes

The proposed method belongs to 1-parameter projection methods. Thus we begin with the construction of a family of integrators $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$.

3.1 Deriving a family of 1-parameter explicit Runge–Kutta methods

In what follows, we will propose how to construct a family $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ of explicit RK methods. As we saw in Sect. 2, by perturbing classical collocation methods, we can express larger class of RK methods. Recall that EQUIP method aimed at a natural parameterization via collocation method which led to small correction. In the present explicit RK method context, we parameterize higher order terms of perturbation operators, hoping it leads to small correction in a similar manner. In this project, a special care is needed to keep the explicitness.

Our construction of a family $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ of *explicit* RK methods consists of three steps.

Definition 4 We derive a family of explicit Runge–Kutta methods in the following three steps.

- **Step 1.** Take an explicit RK method which can be expressed by a perturbed collocation method with $\{N_j\}_{1 \leq j \leq s}$ and c_i ($i = 1, \dots, s$).
- **Step 2.** Calculate d_{s-1}, d_s by Gaussian elimination of the following matrix

$$\begin{pmatrix} 1 & c_1 & \dots & \dots & c_1^{s-1} \\ \vdots & & & & \\ 1 & c_{s-1} & \dots & \dots & c_{s-1}^{s-1} \end{pmatrix} \rightarrow \begin{pmatrix} 1 & c_1 & \dots & \dots & c_1^{s-1} \\ \vdots & & & & \\ 0 & \dots & 0 & \widehat{d}_{s-1} & \widehat{d}_s \end{pmatrix}. \tag{3.1}$$

Then we take d_{s-1}, d_s as

$$d_{s-1} = \frac{1}{(s-2)!} \widehat{d}_{s-1}, \quad d_s = \frac{1}{(s-1)!} \widehat{d}_s. \tag{3.2}$$

- **Step 3.** Make a family $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ of perturbed collocation methods whose expressions are $\{\widetilde{N}_j\}_{1 \leq j \leq s}$ and c_i ($i = 1, \dots, s$):

$$\widetilde{N}_j = N_j \quad (j = 1, \dots, s-2), \tag{3.3}$$

$$\widetilde{N}_j = N_j + \alpha d_j \prod_{k=1}^{s-1} (t - c_k) \quad (j = s-1, s). \tag{3.4}$$

This family $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ is in fact a family of explicit RK methods.

Here we illustrate these steps taking an example.

Example 1 (Based on the 3/8 formula) We consider the 3/8 formula which is a 4-stage explicit RK method.

Step 1. Take the 3/8 formula whose coefficient triple is $(A, \mathbf{b}, \mathbf{c})$ shown in Table 1.

Table 1 Butcher Tableau of the 3/8 formula

c	A	$=$	0	0	0	0	0
	1/3		1/3	0	0	0	0
	2/3		-1/3	1	0	0	0
	1		1	-1	1	0	0
	b		1/8	3/8	3/8	1/8	

This method can be expressed as the perturbed collocation method with $(\{N_j\}_{1 \leq j \leq s}, \{c_i\}_{1 \leq i \leq s})$:

$$N_1(t) = 0, \tag{3.5}$$

$$N_2(t) = -\frac{7}{6}t + 4t^2 - 3t^3, \tag{3.6}$$

$$N_3(t) = -\frac{1}{12}t + \frac{1}{4}t^2 - \frac{1}{6}t^3, \tag{3.7}$$

$$N_4(t) = \frac{5}{324}t - \frac{1}{12}t^2 + \frac{1}{9}t^3 - \frac{1}{24}t^4, \tag{3.8}$$

$$c_1 = 0, \quad c_2 = 1/3, \quad c_3 = 2/3, \quad c_4 = 1. \tag{3.9}$$

These can be calculated from the relation (2.7). Note that this expression satisfies $Pu \in \Pi_{s-1}$ ($p_{sj} = 0$ ($j = 1, \dots, s$)).

Step 2. To calculate d_3, d_4 , the matrix is transformed into rank normal form.

$$\begin{aligned} \begin{pmatrix} 1 & c_1 & c_1^2 & c_1^3 \\ 1 & c_2 & c_2^2 & c_2^3 \\ 1 & c_3 & c_3^2 & c_3^3 \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1/3 & 1/9 & 1/27 \\ 1 & 2/3 & 4/9 & 8/27 \end{pmatrix} \\ \xrightarrow{\text{Gaussian elimination}} &\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/3 & 1/9 & 1/27 \\ 0 & 0 & 2/9 & 2/9 \end{pmatrix} \end{aligned} \tag{3.10}$$

Then we obtain d_3, d_4 as those satisfying $(2!d_3, 3!d_4) = (2/9, 2/9) \Leftrightarrow (d_3, d_4) = (1/9, 1/27)$.

Step 3. Construct a family $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ whose expressions are $(\{\tilde{N}_j\}_{1 \leq j \leq s}, \{c_i\}_{1 \leq i \leq s})$:

$$\tilde{N}_1(t) = 0, \tag{3.11}$$

$$\tilde{N}_2(t) = -\frac{7}{6}t + 4t^2 - 3t^3, \tag{3.12}$$

$$\tilde{N}_3(t) = -\frac{1}{12}t + \frac{1}{4}t^2 - \frac{1}{6}t^3 + \frac{1}{9}\alpha t \left(t - \frac{1}{3}\right) \left(t - \frac{2}{3}\right), \tag{3.13}$$

$$\tilde{N}_4(t) = \frac{5}{324}t - \frac{1}{12}t^2 + \frac{1}{9}t^3 - \frac{1}{24}t^4 + \frac{1}{27}\alpha t \left(t - \frac{1}{3}\right) \left(t - \frac{2}{3}\right), \tag{3.14}$$

$$c_1 = 0, \quad c_2 = 1/3, \quad c_3 = 2/3, \quad c_4 = 1. \tag{3.15}$$

This family has the coefficients in Table 2 as RK methods. It is easy to see that the method has order 3 for all values of α and has order 4 for $\alpha = 0$.

Some comments on each step are in order.

Comment for Step 1. Recall the claim that RK methods satisfying (2.8) and having distinct c_i 's are equivalent to perturbed collocation methods. In addition, s must be greater than 3 for Step 1 to be well-defined.

Table 2 The family of the proposed explicit RK method based on the 3/8 formula

		0	0	0	0	0
c	$A(\alpha)$	1/3	1/3	0	0	0
		2/3	-1/3	1	0	0
		1	$1 + \alpha$	$-1 - 2\alpha$	$1 + \alpha$	0
		b	1/8	3/8	3/8	1/8

Comment for Step 2. A crucial observation here is that this step is always feasible.

Lemma 1 *Step 2 in Definition 4 is always feasible.*

Proof The base integrator chosen in Step 1 can be expressed as a perturbed collocation method, and thus c_i 's are distinct and a $(s - 1) \times (s - 1)$ sub-matrix of the left hand side matrix in (2.2) is a Vandermonde matrix. In other words, its rank is $s - 1$, and we can always transform the matrix into a ladder matrix whose bottom row has the form $(0, \dots, 0, *, *)^\top$.

Comment for Step 3. This step does not destroy the characterization of explicit RK method. This is ensured by Theorem 3 below. Before its proof, we mention the idea of the parameterization. As we mentioned in the beginning of this section, we aim (i) to parameterize higher order terms of a perturbation operator and (ii) to construct a family $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ of explicit RK methods. The construction in Definition 4 is designed to achieve the above two goals.

For (i), recall (3.3) and (3.4). An important observation here is that parameterizing only N_s is not feasible as we can see in the proof of Theorem 3 and the characterization Proposition 2. The matrix (3.1) in Step 2 has $s - 1$ rank, and thus, in general, vectors of the forms $(0, 0, \dots, 0, *)^\top$ cannot be composed by linear combinations of (2.12). In other words, parameterizing only N_s leads to a family of implicit RK methods in general. This encourages us to employ the parameterization in Definition 4.

For (ii), the key idea is to add $\prod_{k=1}^{s-1} (t - c_k)$ to N_j ($j = s - 1, s$). Leaving its detail to the proof of Theorem 3, here we emphasize that the polynomial $\prod_{k=1}^{s-1} (t - c_k)$ vanishes for $t = c_j$ ($j = 1, \dots, s - 1$), which realizes (ii).

Theorem 3 *The family of the perturbed collocation methods constructed in Definition 4 is actually a family of explicit RK methods.*

Proof Let $(A, \mathbf{b}, \mathbf{c})$ be a coefficients triple of an explicit RK method in Step 1 and $(\{N_j\}_{1 \leq j \leq s}, \{c_i\}_{1 \leq i \leq s})$ are the corresponding expression in terms of the perturbed collocation method. Also we define P as a perturbation operator composed by $\{N_j\}_{1 \leq j \leq s}$, and assume P satisfies $Pu \in \Pi_{s-1}$. And let $(\tilde{A}, \tilde{\mathbf{b}}, \tilde{\mathbf{c}})$ be the RK family which is constructed in Step 3 and $(\{\tilde{N}_j\}_{1 \leq j \leq s}, \{\tilde{c}_i\}_{1 \leq i \leq s})$ be the corresponding expressions of the family. Also we define \tilde{P} as a perturbation operator composed by $\{\tilde{N}_j\}_{1 \leq j \leq s}$. Note that both $\tilde{A} = (\tilde{a}_{ij})$ and $\{\tilde{N}_j\}_{1 \leq j \leq s}$ depend on α linearly and the claim of this proposition is that $\tilde{a}_{ij} = 0$ ($1 \leq i \leq j \leq s$) hold.

To check $\tilde{a}_{ij} = 0$ ($1 \leq i \leq j \leq s$), it is sufficient to show that

$(\{\tilde{N}_j\}_{1 \leq j \leq s}, \{c_i\}_{1 \leq i \leq s})$ satisfies the assumption of Proposition 2. Here we define a vector space \mathbb{V}_k ($k = 1, \dots, s$) to simplify notation.

$$\mathbb{V}_k := \{v \in \mathbb{R}^s : v \text{ is a linear combination of } (1, c_l, \dots, c_l^{s-1})^\top \text{ where } l = 1, \dots, k-1, k+1, \dots, s\}. \tag{3.16}$$

We first note that, by construction, the perturbation operator \tilde{P} satisfies $\tilde{P}u \in \Pi_{s-1}$. From Proposition 2, for (i, j) in $1 \leq i \leq j \leq s$, $\tilde{a}_{ij} = 0$ ($1 \leq i \leq j \leq s$) if and only if

$$\left((\tilde{N}_1 + t)(c_i), \dots, (s-1)! \left(\tilde{N}_s + \frac{t^s}{s!} \right) (c_i) \right)^\top \in \mathbb{V}_j. \tag{3.17}$$

Recall, by (3.3) and (3.4), this vector is equivalent to

$$\begin{aligned} & \left((N_1 + t)(c_i), \dots, (s-1)! \left(N_s + \frac{t^s}{s!} \right) (c_i) \right)^\top \\ & + \alpha \left(\underbrace{0, \dots, 0}_{s-2}, (s-2)!d_{s-1} \prod_{k=1}^{s-1} (c_i - c_k), (s-1)!d_s \prod_{k=1}^{s-1} (c_i - c_k) \right)^\top. \end{aligned} \tag{3.18}$$

The first term of this vector belongs to \mathbb{V}_j for all i, j in $1 \leq i \leq j \leq s$, since the base integrator is an explicit RK method. Note that the perturbation operator P satisfies $Pu \in \Pi_{s-1}$.

For $i \leq s-1$, the second term of (3.18) trivially vanishes, and thus in total (3.17) holds. This good nature comes from the fact that the modification to N_j ($j = s-1, s$) takes the form of the polynomial $\prod_{k=1}^{s-1} (t - c_k)$. For $i = s$, the second term of (3.18) does not vanish, but from (3.1), (3.2), it is equivalent to

$$\alpha \prod_{k=1}^{s-1} (c_s - c_k) (0, \dots, 0, \hat{d}_{s-1}, \hat{d}_s)^\top, \tag{3.19}$$

which belongs to \mathbb{V}_s by construction. This completes the proof.

Remark 2 Note that, from the proof in [16], we see that the upper $s-1$ rows of \tilde{A} coincide with those of A . Therefore, if an approximation Φ_α is constructed for some α , we just need an additional evaluation of f for the computation of $\Phi_{\hat{\alpha}}$ ($\hat{\alpha} \neq \alpha$). Thus the proposed method does not need many evaluations f in solving $H(\Phi_\alpha(y_0)) = H(y_0)$. Note that IDT needs *no* evaluation f for the construction of another integrator $\Phi_{\hat{\alpha}}$ ($\hat{\alpha} \neq \alpha$). Therefore the proposed method demands a bit longer computation time than IDT.

Remark 3 From an easy calculation, for the family constructed in Definition 4, we see that we can take r to $p-3$ in Definition 3, if the base integrator has order p . We can apply an adaptive step size control technique (see e.g. [14]), based on the two

numerical integrators with order p and 3. This implementation requires one additional evaluation of f . A natural question here would be whether there exists a family whose r is independent of p or not. The present author has a pessimistic view on this point. Perturbed collocation methods were well investigated theoretically, in particular, in terms of their order of accuracy by Nørsett–Wanner [16]. The analysis was based on the nonlinear Variation-Of-Constants formula. However, it is difficult to use the formula for explicit RK methods because the derivative of the perturbed collocation polynomial (u in Definition 2) may diverge as Nørsett–Wanner [16] have already pointed out.

Remark 4 Another, simple way to parameterize explicit RK methods would be to consider parameterizations for each explicit RK method directly, carefully observing order conditions. Of course this approach makes sense, but in the present paper we did not employ it, since order conditions are generally quite complicated, and it seems difficult to discuss which conditions are important or safe to violate. Furthermore, even if we can do that for a specific explicit RK method, we should do the whole thing all over again from scratch when we move on to another RK method. The parameterization technique proposed in this paper is free from this difficulty. Because parameterizing RK coefficients matrix A is equivalent to adding parameterized polynomials to N_j ($1 \leq j \leq s$) which compose the perturbation operator, we can intuitively say that a nice parameterization should be the one that changes the perturbation operator to a minimum degree. One way to realize this is to only allow the changes in N_j with higher indices. The following numerical experiments support this view.

3.2 Preserving multiple invariants

We can generalize the proposed method to preserve several invariants. We realize this by constructing a family of RK methods depending on a set of parameters. Namely we parameterize them with the same number of parameters as the number of the targeted invariants. For example, if two invariants are targeted, we should construct a family with two parameters $\{\Phi_{(\alpha, \beta)}\}_{(\alpha, \beta) \in \mathbb{R}^2}$.

- **Step 1.** Take an explicit RK method which can be expressed by a perturbed collocation method with $\{N_j\}_{1 \leq j \leq s}$ and $\{c_i\}_{1 \leq i \leq s}$.
- **Step 2.** Calculate $d_{s-2}^{(1)}, d_{s-1}^{(1)}, d_s^{(1)}$ and $d_{s-1}^{(2)}, d_s^{(2)}$ from the elementary row operations of the following matrix

$$\begin{pmatrix} 1 & c_1 & \dots & c_1^{s-1} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ 1 & c_s & \dots & c_s^{s-1} \end{pmatrix} \rightarrow \begin{pmatrix} 1 & c_1 & \dots & c_1^{s-1} \\ \vdots & & & \vdots \\ 0 & \dots & \hat{d}_{s-2}^{(1)} & \hat{d}_{s-1}^{(1)} & \hat{d}_s^{(1)} \\ 0 & \dots & 0 & \hat{d}_{s-1}^{(2)} & \hat{d}_s^{(2)} \end{pmatrix}. \tag{3.20}$$

Then we take $d_{s-2}^{(1)}, d_{s-1}^{(1)}, d_s^{(1)}$ and $d_{s-1}^{(2)}, d_s^{(2)}$ as

$$d_{s-2}^{(1)} = \frac{1}{(s-3)!} \hat{d}_{s-2}^{(1)}, \quad d_{s-1}^{(1)} = \frac{1}{(s-2)!} \hat{d}_{s-1}^{(1)}, \quad d_s^{(1)} = \frac{1}{(s-1)!} \hat{d}_s^{(1)}, \quad (3.21)$$

$$d_{s-1}^{(2)} = \frac{1}{(s-2)!} \hat{d}_{s-1}^{(2)}, \quad d_s^{(2)} = \frac{1}{(s-1)!} \hat{d}_s^{(2)}. \quad (3.22)$$

– **Step 3.** Make a family $\{\Phi_{(\alpha,\beta)}\}_{(\alpha,\beta) \in \mathbb{R}^2}$ of the perturbed collocation methods whose expressions are $\{\tilde{N}_j\}_{1 \leq j \leq s}$ and $\{c_i\}_{1 \leq i \leq s}$:

$$\tilde{N}_j = N_j \quad (j = 1, \dots, s-3), \quad (3.23)$$

$$\tilde{N}_{s-2} = N_{s-2} + \alpha d_{s-2}^{(1)} \prod_{k=1}^{s-1} (t - c_k), \quad (3.24)$$

$$\tilde{N}_{s-1} = N_{s-1} + (\alpha d_{s-1}^{(1)} + \beta d_{s-1}^{(2)}) \prod_{k=1}^{s-1} (t - c_k), \quad (3.25)$$

$$\tilde{N}_s = N_s + (\alpha d_s^{(1)} + \beta d_s^{(2)}) \prod_{k=1}^{s-1} (t - c_k). \quad (3.26)$$

This family $\{\Phi_{(\alpha,\beta)}\}_{(\alpha,\beta) \in \mathbb{R}^2}$ is in fact a family of explicit RK methods.

In a similar manner as above, we are able to construct a family of explicit RK methods with $s - 1$ parameters. The proof of the explicitness is similar to Theorem 3, and hence we omit it here.

4 Numerical tests

In this section we show numerical tests that confirm the efficiency of the proposed method. In particular, we focus on the consequences of the natural parameterization of our approach.

We consider the Hénon–Heiles equation:

$$\frac{dy}{dt} = J \nabla H(y), \quad y \in \mathbb{R}^4, \quad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \in \mathbb{R}^{4 \times 4}, \quad (4.1)$$

$$H(y) = H(q_1, q_2, p_1, p_2) = \frac{1}{2}(p_1^2 + p_2^2) + U(q_1, q_2), \quad (4.2)$$

where $U(q_1, q_2)$ is called potential (also we denote it as $U(y)$ to simplify notation). The Hénon–Heiles equation was derived to describe the stellar motion inside the (gravitational) potential $U(y_0)$ for a very long time. The constant y_0 denotes the initial state. See, e.g., [13]. In our experiment we set the potential to

$$U(q_1, q_2) = \frac{1}{2}(q_1^2 + q_2^2) + q_1^2 q_2 - \frac{1}{3} q_2^3. \quad (4.3)$$

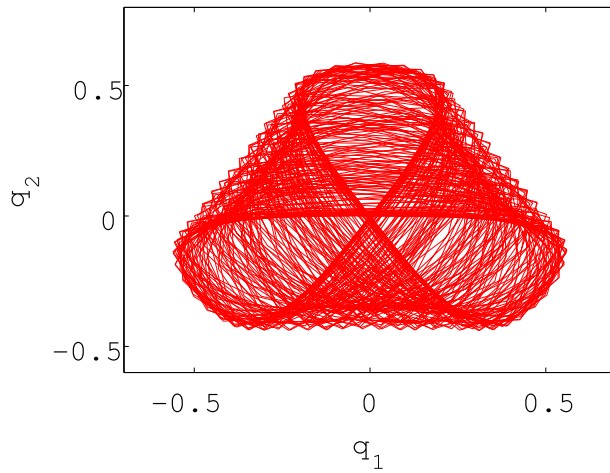


Fig. 1 A reference solution (represented as a *red line*) by RK4 with sufficiently small step size $h = 2/300$ (color figure online)

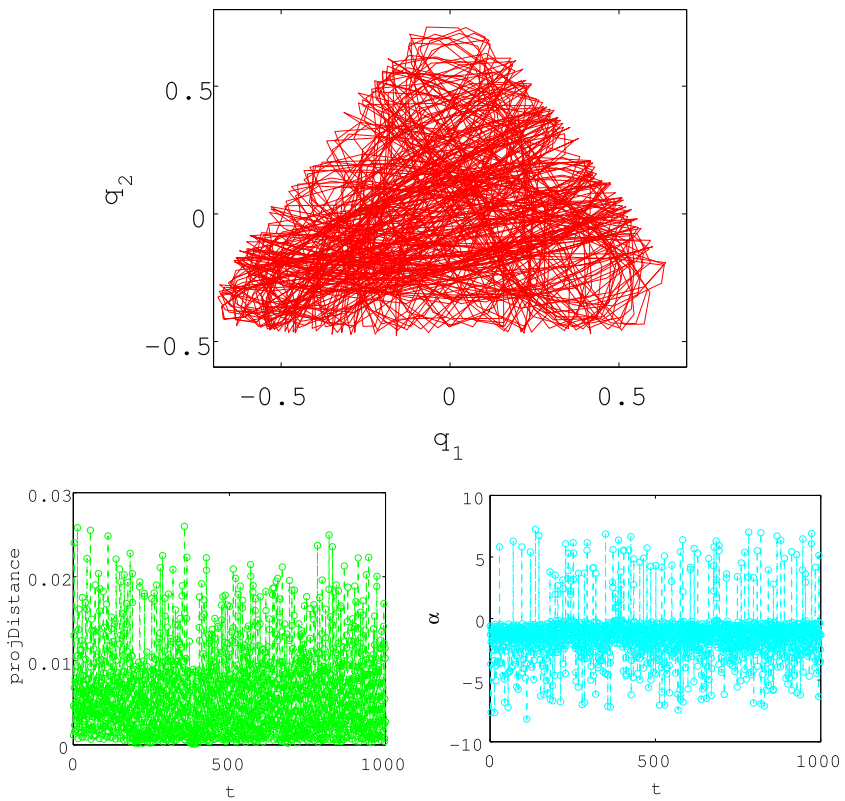


Fig. 2 IDT34: *top* the numerical solution with $h = 2/3$, *bottom left* value of the corrections by projection (its label “projDistance” means the corrections by projection), *bottom right* value of the parameter

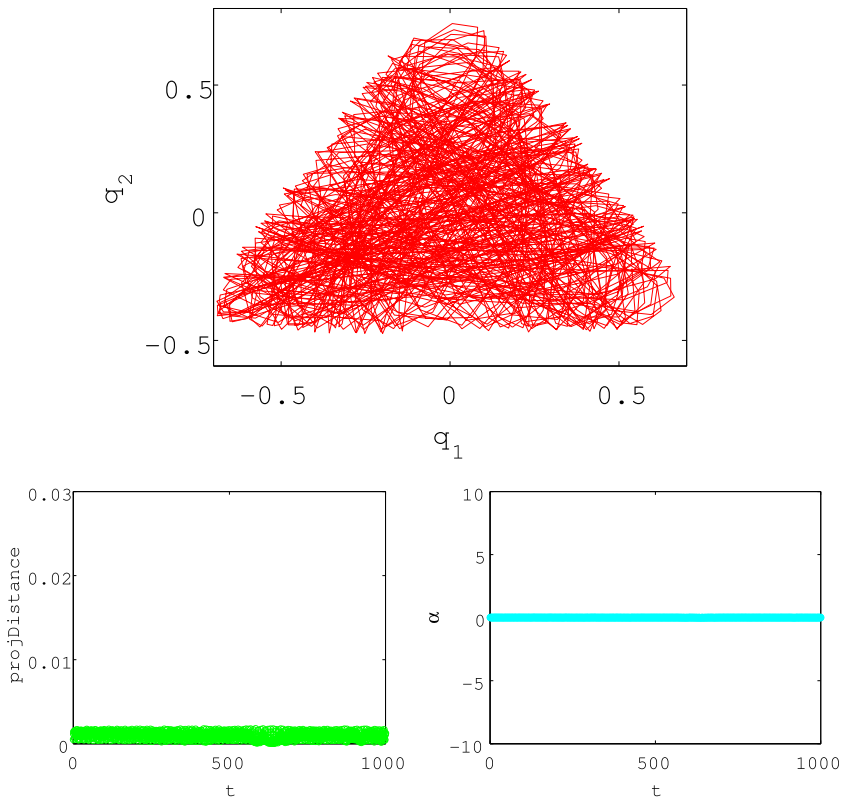


Fig. 3 IDT14: *top* the numerical solution with $h = 2/3$, *bottom left* value of the corrections by projection (its label “projDistance” means the corrections by projection). The maximum value of projDistance is about 1.7×10^{-3} , *bottom right* value of the parameter. The maximum value of α is about 0.01

We will compare the proposed method with IDT to show the naturalness and effectiveness of the proposed parameterization. We set the final time to $T = 1000$ and the initial state to $y_0 = (0, 0, \sqrt{\frac{3}{10}}, 0)^\top$. For comparison, in Fig. 1 we show a reference solution constructed by the 3/8 RK4 with a sufficiently small step size $h = 2/300$. For the proposed method and IDT, we take RK introduced in Example 1 as the base integrator. In addition, for IDT, the third order method introduced in [14] and the first order method (the Euler method) are taken as the embedded RK method. When we need to distinguish these two methods, we call the former IDT34 and the latter IDT14, clarifying the order of the embedded methods. Here IDT14 is considered in view of [4]. In what follows we set the step size h to $2/3$. The reason we take such a large time step is that we aim to see how robust the methods are. Recall that, for sufficiently small step size, Theorem 2 can be applied to all 1-parameter projection methods; but that does not necessarily reflect actual behaviors. In addition we do not apply adaptive step size controls in our experiments because this technique may conceal the difference among the methods, although from a practical point of view, we do not doubt the effectiveness of the controls. Note that both the proposed method and IDT can

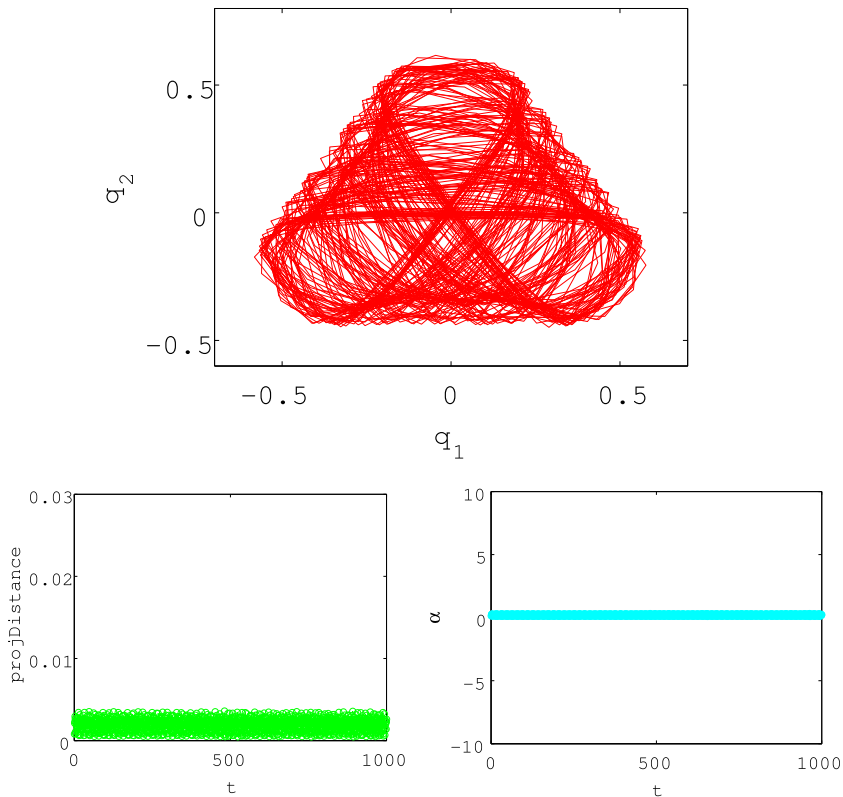


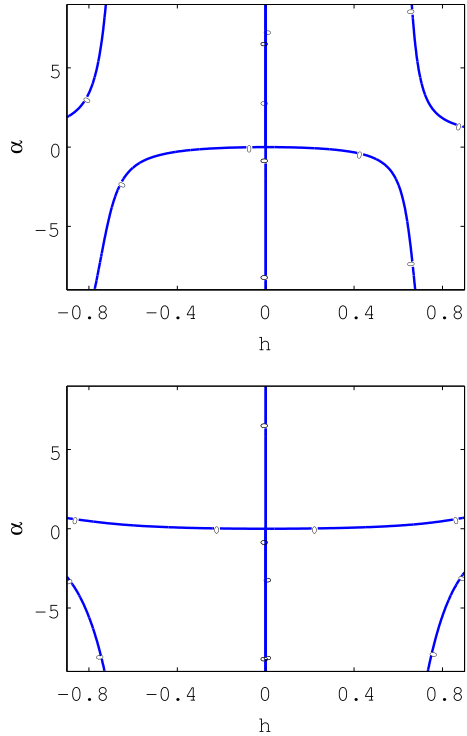
Fig. 4 The proposed method: *top* the numerical solution with $h = 2/3$, *bottom left* value of the corrections by projection (its label “projDistance” means the corrections by projection). The maximum value of projDistance is about 3.5×10^{-3} , *bottom right* value of the parameter. The maximum value of α is about 0.3

utilize adaptive step size control (see Remark 4); however, the proposed method falls slightly behind IDT, because the proposed method needs one extra evaluation of f , unless some sophisticated technique such as FSAL is employed (see e.g. [13]).

In Figs. 2, 3 and 4 we show numerical solutions by IDT34, IDT14 and the proposed method respectively. All figures also include values of the parameter α and the corrections by projection. Here we take the 2-norm of the differences between reference solutions and the numerical solutions as the values of the corrections.

To observe the consequences of the proposed parameterization, let us pay attention to the figures of the values of the parameter and the corrections by projection. From these figures, we find that the values of IDT34 are often large, while the values of the proposed method and IDT14 remain constantly small. At the occasions where IDT34 takes large parameter it seems that the search direction is almost perpendicular to ∇H and ceases to work with the constant step size h . Accordingly, there the corrections made by projection also get large values. In contrast, the proposed method and IDT14 do not exhibit such a behavior even with this large time step $h = 2/3$, and consequently, the correction remains small. For IDT14, the search directions seem to be stabler than

Fig. 5 Lines which display $g(\alpha, h) = 0$, $(h, \alpha) \in [-0.9, 0.9] \times [-9, 9]$. *Top* shows lines of IDT34. *Bottom* shows lines of the proposed method



those of IDT34. However, these figures, in particular Fig. 2, indicate that linear search methods can yield fatal projection steps without the special technique argued in [6].

Comparing the numerical solutions, we find that the solution by the proposed method is closer to the reference solution than those by IDT34 and IDT14. In particular, we see that the solution by the proposed method generates an appropriate trajectory. This can be considered as a good consequence of the natural parameterization of the proposed method.

The naturalness of the proposed parameterization can be seen from another perspective on robustness; i.e., how large the step size h can be safely. Theorem 1 does not answer this because it is just a theoretical result in the limit of $h \rightarrow 0$. However, in actual computation, it is expected that the theorem can hold for large step h . In Fig. 5, we show lines which denote zeros of $g(\alpha, h) = H(y_1(\alpha, h)) - H(y_0)$, $(h, \alpha) \in [-0.9, 0.9] \times [-9, 9]$ where y_0 is a state vector which IDT34 takes large α ($t = 8/3$). The top of Fig. 5 is the contour of IDT34 and the bottom is that of the proposed method. The straight line on the axis $h = 0$ is obvious (since no evolution occurs), and thus should be ignored. Our attention should be paid to the horizontal curves that touch the origin. From these figures, we see the proposed method can provide appropriate parameters even for large h ; in other words, the implicit function theorem employed in Theorem 1 keeps working. On the contrary, the curve of IDT34 sharply goes down near $h = 0.6$. This shows that Theorem 1 does not hold here, and in such a case, the parameter α can be quite large, and the projection becomes fatal. There-

fore we see the naturalness of the proposed parameterization. But again we would like to emphasize that this issue usually happens only occasionally (recall Fig. 2; in most time steps the corrections are small), and can be easily avoided by adaptive time stepping if necessary. What we like to state here is the better stability of the proposed method.

5 Conclusion

In this paper we gave a new strategy to parameterize explicit RK methods via the perturbed collocation method. One of the advantages of this strategy is that we can systematically parameterize explicit RK methods. Another advantage is that in the strategy, we can demand the parameterization is done only in higher order terms of the perturbation operators, in the hope of small projection correction. We showed that such a parameterization is always feasible without destroying the explicitness of the base explicit RK method. In addition we confirmed by numerical examples that the proposed strategy in fact results in small corrections in a robust manner.

We conclude this paper with some remarks. First, we can also apply the proposed method to other classes of ODEs, in particular, dissipative ODEs with a known Lyapunov function. This is an obvious extension of the results [5, 10]. Second, in this paper, we mainly investigated the robustness of the parameterization; however, the proposed method needs a bit longer computation time than IDT because the proposed method needs more evaluations of f in solving $H(\Phi_\alpha(y_0)) = H(y_0)$. This is a trade-off between the robustness and stability. When we can take sufficiently small step size h , IDT should beautifully work, and thus it might be sufficient. But if we hope to take larger time step sizes, for example when we are considering integration over long time, the proposed method can be a better solution since it works robustly even in such a circumstance. Finally, when the system is essentially highly unstable, projections based on explicit methods might not work well, and implicit projections such as EQUIP become necessary. Such trade-offs among methods should be more carefully investigated.

A possible future work is to develop a rigorous theoretical explanation on how the proposed parameterization, employed in higher order perturbations, results in the actual small corrections. And through such theoretical investigations, we hope the challenge for “the best 1-parameter projection method” continues. In this sense, we feel the proposed strategy via perturbed collocation methods is just a starting point of such a challenge, which reveals a certain aspect of good parameterization.

Acknowledgements The author would like to thank Takayasu Matsuo for helpful comments and discussion. He is also thankful to the anonymous reviewers who gave helpful comments to improve this paper.

References

1. Brugnano, L., Calvo, M., Montijano, J.I., Rández, L.: Energy-preserving methods for Poisson systems. *J. Comput. Appl. Math.* **236**, 3890–3904 (2012)

2. 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)
3. Brugnano, L., Iavernaro, F., Trigiante, D.: Analysis of Hamiltonian boundary value methods (HBVMs): a class of energy-preserving Runge-Kutta methods for the numerical solution of polynomial Hamiltonian systems. *Commun. Nonlinear Sci. Numer. Simul.* **20**, 650–667 (2015)
4. Calvo, M., Hernández-Abreu, D., Montijano, J.I., Rández, L.: On the preservation of invariants by explicit Runge-Kutta methods. *SIAM J. Sci. Comput.* **28**, 868–885 (2006)
5. Calvo, M., Laburta, M.P., Montijano, J.I., Rández, L.: Projection methods preserving Lyapunov functions. *BIT Numer. Math.* **50**, 223–241 (2010)
6. Calvo, M., Laburta, M.P., Montijano, J.I., Rández, L.: Runge-Kutta projection methods with low dispersion and dissipation errors. *Adv. Comput. Math.* **41**, 231–251 (2015)
7. Celledoni, E., McLachlan, R.I., McLaren, D.I., Owren, B., Quispel, G.R.W., Wright, W.M.: Energy preserving Runge-Kutta methods. *M2AN Math. Model. Numer. Anal.* **43**, 645–649 (2009)
8. Dahlby, M., Owren, B., Yaguchi, T.: Preserving multiple first integrals by discrete gradients. *J. Phys. A* **44**, 305205 (2011)
9. Gonzalez, O.: Time integration and discrete Hamiltonian systems. *J. Nonlinear Sci.* **6**, 449–467 (1996)
10. Grimm, V., Quispel, G.R.W.: Geometric integration methods that preserve Lyapunov functions. *BIT Numer. Math.* **45**, 709–723 (2005)
11. Hairer, E.: Symmetric projection methods for differential equations on manifolds. *BIT Numer. Math.* **40**, 726–734 (2000)
12. Hairer, E.: Energy-preserving variant of collocation methods. *J. Numer. Anal. Ind. Appl. Math.* **5**, 73–84 (2010)
13. Hairer, E., Lubich, C., Wanner, G.: *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, Berlin (2006)
14. Hairer, E., Nørsett, S.P., Wanner, G.: *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer, Berlin (1993)
15. Iserles, A., Zanna, A.: Preserving algebraic invariants with Runge-Kutta methods. *J. Comput. Appl. Math.* **125**, 69–81 (2000)
16. Nørsett, S.P., Wanner, G.: Perturbed collocation and Runge-Kutta methods. *Numer. Math.* **38**, 193–208 (1981)
17. McLachlan, R.I., Quispel, G.R.W., Robidoux, N.: Geometric integration using discrete gradients. *Philos. Trans. R. Soc. Lond. A* **357**, 1021–1045 (1999)
18. Quispel, G.R.W., McLaren, D.I.: Integral-preserving integrators. *J. Phys. A* **37**, L489–L495 (2004)
19. Quispel, G.R.W., McLaren, D.I.: A new class of energy-preserving numerical integration methods. *J. Phys. A* **41**, 045206 (2008)
20. Shampine, L.F.: Conservation laws and the numerical solution of ODEs. *Comput. Math. Appl.* **12**, 1287–1296 (1986)
21. Yoshida, H.: Construction of higher order symplectic integrators. *Phys. Lett. A* **150**, 262–268 (1990)