

MATHEMATICAL ENGINEERING TECHNICAL REPORTS

Invariants Preserving Schemes Based on Explicit Runge–Kutta Methods

Hiroki KOJIMA

(Communicated by Takayasu MATSUO)

METR 2015–11

March 2015

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

WWW page: <http://www.keisu.t.u-tokyo.ac.jp/research/techrep/index.html>

The METR technical reports are published as a means to ensure timely dissemination of scholarly and technical work on a non-commercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

Invariants Preserving Schemes Based on Explicit Runge–Kutta Methods

Hiroki KOJIMA[†]

Department of Mathematical Informatics
Graduate School of Information Science and Technology
The University of Tokyo

[†]`hiroki.kojima@mist.t.u-tokyo.ac.jp`

March 2015

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, but 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. Numerical experiments confirm the stability of the proposed method.

1 Introduction

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

$$\frac{dy}{dt} = J\nabla H(y(t)), \quad y(0) = y_0 \in \mathbb{R}^N, \quad (1)$$

where J is the symplectic matrix and $H : \mathbb{R}^N \rightarrow \mathbb{R}$ is called Hamiltonian, or in more general context, an energy function. From the formulation (1), we see that the exact solution of the Hamiltonian system preserves the energy function H :

$$\frac{d}{dt}H(y(t)) = \nabla H(y(t))^\top \frac{dy}{dt} = \nabla H(y(t))^\top J\nabla H(y(t)) = 0, \quad (2)$$

where we used the chain rule and the skew symmetry of J . Such ODEs with (possibly multiple) invariants appear in many modern science and engineering. In particular, by appropriately discretizing some partial differential equations (PDEs) only in space, we obtain such ODEs (see, for example, [5]).

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 [4]. Hence, in these several decades, several invariants-preserving methods have been constructed and analyzed (e.g. [13, 16, 18]), and it has been proved 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 [11].

Energy-preserving methods for the Hamiltonian systems (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. [7, 15, 17]), and (ii) methods that first gets 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. [1, 2, 6, 9]). Although methods in the first group generate stable numerical solutions, they need solving at least an N dimensional nonlinear equations in every time step, thus demand much computation time when N is so large (e.g. for ODEs derived by the spatial discretization of PDEs). Even worse, most of them have only second order accuracy, and demand extra computational cost when further accuracy is necessary [10, 19]. In contrast to this, some methods in the second group only need solving a single nonlinear equation (in stead 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) [11] 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 [2] 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–Trigante [1]. 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. Each author gave some theoretical results on the solvability and accuracy, which can be reorganized in more general context as follows. Let us introduce a 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 the result of [1], 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). Therefore we see *theoretically* that all 1-parameter projection methods is highly accurate demanding only low computational cost.

Although this gives a clear support for all the 1-parameter projection methods, a natural question can arise in the aspect of actual computation: which method among 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 do not necessarily provide informa-

tion on the behavior under practical size of h . Therefore a 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 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 parametrization does not destroy the symplecticity of the Gauss method, which was the main subject in [1]. In the present paper, however, we like to see the method from a different perspective. In [1], the authors chose to discard linear search, and turn to a completely new idea of projection, where they introduced a clever, natural parametrization of the Gauss method. There, the “search” is no longer linear, but thanks to the new view as a parametrized Gauss method, the correction by the projection is expected to remain relatively small, avoiding possible fatal projections in linear search methods. In fact, in [1], numerical examples show that EQUIP runs stably with constant time steps. However, because the Gauss method is implicit, EQUIP method also needs solving high dimensional nonlinear equation 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* parametrization 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 parametrization 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 [14]. 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 parametrize 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 parametrization only in higher order perturbation terms. This sounds natural for achieving as small correction as possible. This task is, however, not so straightforward as simply imagined. Since as described above the perturbed collocation method can express both explicit and implicit RK methods, inappropriate parametrization of explicit RK methods should destroy the explicitness and result in *implicit* schemes. This is not what we hope in view 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 [14]. 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 Section 2, we introduce perturbed collocation methods and the general theory for 1-parameter projection methods. In Section 3, we propose a new method via perturbed collocation methods. In Section 4, we test the proposed method and show its stability and efficiency comparing with existing methods. In Section 5, we give conclusions of this paper.

2 Preliminaries

2.1 Perturbed Collocation Methods

Here we give a brief review of perturbed collocation methods. First, we introduce the perturbation operator.

Definition 1 (Perturbation operator [14]). *Let Π_s be the linear space of s -degree polynomials. Then the perturbed operator $P_{t_0, h} : \Pi_s \rightarrow \Pi_s$ is defined as below:*

$$(P_{t_0, h}u)(t) := u(t) + \sum_{j=1}^s N_j \left(\frac{t-t_0}{h} \right) u^{(j)}(t_0) h^j, \quad (3)$$

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

$$N_j(t) = \frac{1}{j!} \sum_{i=0}^s (p_{ij} - \delta_{ij}) t^i, \quad (4)$$

$$p_{ij} \in \mathbb{R} \quad (i = 0, 1, \dots, s; \quad j = 1, 2, \dots, s), \quad (5)$$

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

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

$$u(t_0) = y_0, \quad (7)$$

$$\dot{u}(t_0 + c_i h) = f(Pu(t_0 + c_i h)) \quad (1 \leq i \leq s). \quad (8)$$

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 ([14]). *The perturbed collocation method defined by (7) and (8) is equivalent to a Runge–Kutta method whose coefficients are the entries of*

$$A = V_c P J V^{-1}, \quad (9)$$

$$\mathbf{b} = (1, \dots, 1)^\top J V^{-1}, \quad (10)$$

where

$$V_c = \begin{pmatrix} 1 & c_1 & \dots & c_1^s \\ \vdots & & & \vdots \\ 1 & c_s & \dots & c_s^s \end{pmatrix}, \quad P = \begin{pmatrix} 1 & p_{01} & \dots & p_{0s} \\ 0 & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & p_{s1} & \dots & p_{ss} \end{pmatrix}, \quad (11)$$

$$J = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 0 & 1/2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1/s \end{pmatrix}, \quad V = \begin{pmatrix} 1 & c_1 & \dots & c_1^{s-1} \\ \vdots & & & \vdots \\ 1 & c_s & \dots & c_s^{s-1} \end{pmatrix}. \quad (12)$$

Conversely, a Runge–Kutta method that satisfies (10) 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 [14, 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 [14], a characterization for a perturbed collocation method being an explicit RK method was given. This characterization will play a key role when we parametrize explicit RK methods appropriately in Section 3.

Proposition 2 ([14]). *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*

$$\left((N_1 + t)(c_i), \dots, (s-1)!(N_s + \frac{t^s}{s!})(c_i) \right)^\top \quad (13)$$

is a linear combination of

$$(1, \dots, c_k^{s-1})^\top \quad (k = 1, \dots, j-1, j+1, \dots, s). \quad (14)$$

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 (14) 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 [1] (similar results can be found in Calvo–Hernández–Abreu–Montijano–Rández [2]). 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 .*

1-parameter projection methods need solving the following nonlinear equation about α with h fixed:

$$g(\alpha, h) = H(\Phi_\alpha(y_0)) - H(y_0). \quad (15)$$

Under this general setting, we can prove the existence of solutions of (15) 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:

$$g(0, h) = H(y_1(0)) - H(y_0) = c_0 h^d + \mathcal{O}(h^{d+1}), \quad (16)$$

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

$$g(\alpha, h) = c(\alpha) h^{d-r} + \mathcal{O}(h^{d-r+1}) \quad (17)$$

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

The above setting basically follows [1]. 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 $c(\alpha)$. 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 (16). This is essentially by Brugnano–Iavernaro–Trigiane [1], 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.*

$$(1) \quad g(\alpha^*(h), h) = 0, \quad h \in (-h_0^*, h_0^*),$$

$$(2) \quad \alpha^*(h) = \text{const} \cdot h^r + \mathcal{O}(h^{r+1}).$$

proof. The following proof is a straightforward extension of the proof in [1]. From the assumptions, $g(\alpha, h)$ can be expanded as

$$g(\alpha, h) = \sum_{j=d}^{\infty} \frac{1}{j!} \frac{\partial^j g}{\partial h^j}(0, 0) h^j + \sum_{i=1}^{\infty} \sum_{j=d-r}^{\infty} \frac{1}{i!j!} \frac{\partial^{i+j} g}{\partial \alpha^i \partial h^j}(0, 0) \alpha^i h^j. \quad (18)$$

The first term comes from (16), and the second from (17). 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 (18) to get

$$\begin{aligned} g(\alpha, h) &= \frac{1}{d!} \frac{\partial^d g}{\partial h^d}(0, 0) h^d + \frac{1}{(d+1)!} \frac{\partial^{d+1} g}{\partial h^{d+1}}(0, 0) h^{d+1} \\ &\quad + \frac{1}{(d-r)!} \frac{\partial^{d-r+1} g}{\partial \alpha \partial h^{d-r}}(0, 0) h^d \eta + \frac{1}{(d-r+1)!} \frac{\partial^{d-r+2} g}{\partial \alpha \partial h^{d-r+1}}(0, 0) h^{d+1} \eta \\ &\quad + \mathcal{O}(h^{d+2}). \end{aligned} \quad (19)$$

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

$$\begin{aligned} \tilde{g}(\eta, h) &= \frac{1}{(d-r+1) \cdots (d-1)d} \frac{\partial^d g}{\partial h^d}(0, 0) + \frac{1}{(d-r+1) \cdots d(d+1)} \frac{\partial^{d+1} g}{\partial h^{d+1}}(0, 0) h \\ &\quad + \frac{\partial^{d-r+1} g}{\partial \alpha \partial h^{d-r}}(0, 0) \eta + \frac{1}{d-r+1} \frac{\partial^{d-r+2} g}{\partial \alpha \partial h^{d-r+1}}(0, 0) h \eta \\ &\quad + \mathcal{O}(h^2), \end{aligned} \quad (20)$$

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 (20), $\eta(h)$ is calculated as follows. For sufficiently small h ,

$$\begin{aligned} \eta(h) &= - \frac{\frac{1}{(d-r+1) \cdots (d-1)d} \frac{\partial^d g}{\partial h^d}(0, 0) + \frac{1}{(d-r+1) \cdots d(d+1)} \frac{\partial^{d+1} g}{\partial h^{d+1}}(0, 0) h + \mathcal{O}(h^2)}{\frac{\partial^{d-r+1} g}{\partial \alpha \partial h^{d-r}}(0, 0) + \frac{1}{d-r+1} \frac{\partial^{d-r+2} g}{\partial \alpha \partial h^{d-r+1}}(0, 0) h} \\ &= - \frac{1}{(d-r+1) \cdots (d-1)d} \frac{\frac{\partial^d g}{\partial h^d}(0, 0)}{\frac{\partial^{d-r+1} g}{\partial \alpha \partial h^{d-r}}(0, 0)} + \mathcal{O}(h). \end{aligned} \quad (21)$$

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

$$\alpha^*(h) = \eta(h)h^r = -\frac{1}{(d-r+1)\cdots(d-1)d}\frac{\frac{\partial^d g}{\partial h^d}(0,0)}{\frac{\partial^{d-r+1}g}{\partial \alpha \partial h^{d-r}}(0,0)}h^r + \mathcal{O}(h^{r+1}). \quad (22)$$

This completes the proof. \square

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

Theorem 2. *1-parameter projection 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 [1]. First we define $y_1(\alpha, h) := \Phi_\alpha(y_0)$. Then we apply the mean value theorem to obtain

$$y_1(\alpha, h) = y_1(0, h) + \alpha \int_0^1 \frac{\partial y_1}{\partial \alpha}(t\alpha, h) dt. \quad (23)$$

By Definition 3 we see

$$\int_0^1 \frac{\partial y_1}{\partial \alpha}(t\alpha, h) dt = \mathcal{O}(h^{p+1-r}). \quad (24)$$

If we define α^* as the solution of (16), then from Theorem 1, we have $\alpha^* = \mathcal{O}(h^r)$ thus

$$y_1(\alpha^*, h) - y(t_0 + h) = y_1(0, h) - y(t_0 + h) + \alpha^* \int_0^1 \frac{\partial y_1}{\partial \alpha}(t\alpha, h) dt = \mathcal{O}(h^{p+1}). \quad (25)$$

This completes the proof. \square

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 Section 2, by perturbing classical collocation methods, we can express larger class of RK methods. Recall that EQUIP method aimed at a natural parametrization 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 this project, a special care is needed to keep a family of explicit RK methods.

Now we propose an algorithm to construct a family $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ of *explicit* RK methods. This construction needs 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 from the elementary row operations of the following matrix

$$\begin{pmatrix} 1 & c_1 & \dots & \dots & c_1^{s-1} \\ \vdots & & & & \\ 1 & c_{s-1} & \dots & \dots & c_{s-1}^{s-1} \end{pmatrix} \rightarrow \begin{pmatrix} 1 & c_1 & \dots & \dots & c_1^{s-1} \\ \vdots & & & & \\ 0 & \dots & 0 & \hat{d}_{s-1} & \hat{d}_s \end{pmatrix}. \quad (26)$$

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

$$d_{s-1} = \frac{1}{(s-2)!} \hat{d}_{s-1}, \quad d_s = \frac{1}{(s-1)!} \hat{d}_s. \quad (27)$$

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

$$\tilde{N}_j = N_j \quad (j = 1, \dots, s-2), \quad (28)$$

$$\tilde{N}_j = N_j + \alpha d_j \prod_{k=1}^{s-1} (t - c_k) \quad (j = s-1, s). \quad (29)$$

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 $(A, \mathbf{b}, \mathbf{c})$ displayed in Table 1. This method can be expressed by the perturbed collocation method

Table 1: Butcher Tableau of 3/8 formula.

		0	0	0	0	0
		1/3	1/3	0	0	0
		2/3	-1/3	1	0	0
		1	1	-1	1	0
\mathbf{c}	\mathbf{b}		1/8	3/8	3/8	1/8

$(\{N_j\}_{1 \leq j \leq s}, \{c_i\}_{1 \leq i \leq s})$:

$$N_1(t) = 0, \quad (30)$$

$$N_2(t) = -\frac{7}{6}t + 4t^2 - 3t^3, \quad (31)$$

$$N_3(t) = -\frac{1}{12}t + \frac{1}{4}t^2 - \frac{1}{6}t^3, \quad (32)$$

$$N_4(t) = \frac{5}{324}t - \frac{1}{12}t^2 + \frac{1}{9}t^3 - \frac{1}{24}t^4, \quad (33)$$

$$c_1 = 0, \quad c_2 = 1/3, \quad c_3 = 2/3, \quad c_4 = 1. \quad (34)$$

These can be calculated from the relation (9). 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.

$$\begin{pmatrix} 1 & c_1 & c_1^2 & c_1^3 \\ 1 & c_2 & c_2^2 & c_2^3 \\ 1 & c_3 & c_3^2 & c_3^3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1/3 & 1/9 & 1/27 \\ 1 & 2/3 & 4/9 & 8/27 \end{pmatrix} \xrightarrow{\text{row operations}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/3 & 1/9 & 1/27 \\ 0 & 0 & 2/9 & 2/9 \end{pmatrix} \quad (35)$$

Then we obtain d_3, d_4 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 $(\{\tilde{N}_j\}_{1 \leq j \leq s}, \{c_i\}_{1 \leq i \leq s})$:

$$\tilde{N}_1(t) = 0, \quad (36)$$

$$\tilde{N}_2(t) = -\frac{7}{6}t + 4t^2 - 3t^3, \quad (37)$$

$$\tilde{N}_3(t) = -\frac{1}{12}t + \frac{1}{4}t^2 - \frac{1}{6}t^3 + \frac{1}{9}\alpha t(t - \frac{1}{3})(t - \frac{2}{3}), \quad (38)$$

$$\tilde{N}_4(t) = \frac{5}{324}t - \frac{1}{12}t^2 + \frac{1}{9}t^3 - \frac{1}{24}t^4 + \frac{1}{27}\alpha t(t - \frac{1}{3})(t - \frac{2}{3}), \quad (39)$$

$$c_1 = 0, \quad c_2 = 1/3, \quad c_3 = 2/3, \quad c_4 = 1. \quad (40)$$

This family has the coefficients in Table 2 in terms of RK methods.

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

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

Some comments on each step are in order.

Comment for Step 1. Recall the claim that RK methods satisfying (10) 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.

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 (4) 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 $(0, \dots, 0, *, *)^\top$. \square

Comment for Step 3. This step constructs a family of the perturbed collocation methods, i.e., a family of the corresponding explicit RK methods; in other words, it 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 parametrize 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 above two goals.

For (i), observe (28) and (29). One might think that parameterizing only N_s is most desirable, but, such a parametrization is not feasible as we can see in the proof of Theorem 3 and the characterization Proposition 2. The matrix (26) 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 (14). In other words, parameterizing only N_s leads to a family of implicit RK methods in general. Therefore parameterization in Definition 4 seems most appropriate for (i).

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

Theorem 3. *The family of the perturbed collocation methods constructed in Definition 4 is actually the 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 $(\{\tilde{N}_j\}_{1 \leq j \leq s}, \{c_i\}_{1 \leq i \leq s})$ 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.

$$\mathbb{V}_k := \{\mathbf{v} \in \mathbb{R}^s : \mathbf{v} \text{ is a linear combination of } (1, c_l, \dots, c_l^{s-1})^\top \text{ where } l = 1, \dots, k-1, k+1, \dots, s\}. \quad (41)$$

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

$$\left((\tilde{N}_1 + t)(c_i), \dots, (s-1)!(\tilde{N}_s + \frac{t^s}{s!})(c_i) \right)^\top \in \mathbb{V}_j. \quad (42)$$

Recall, by (28) and (29), this vector is equivalent to

$$\begin{aligned} & \left((N_1 + t)(c_i), \dots, (s-1)!(N_s + \frac{t^s}{s!})(c_i) \right)^\top \\ & + \alpha \left(\underbrace{0, \dots, 0}_{s-2}, (s-2)!d_{s-1} \prod_{k=1}^{s-1} (c_i - c_k), (s-1)!d_s \prod_{k=1}^{s-1} (c_i - c_k) \right)^\top. \end{aligned} \quad (43)$$

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 (43) trivially vanishes, and thus in total (42) holds. This is a good consequence that the polynomial $\prod_{k=1}^{s-1} (t - c_k)$ is added to N_j ($j = s-1, s$). For $i = s$, the second term of (43) does not vanish, but from (26), (27), it is equivalent to

$$\alpha \prod_{k=1}^{s-1} (c_s - c_k) \left(0, \dots, 0, \hat{d}_{s-1}, \hat{d}_s \right)^\top, \quad (44)$$

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

Remark 2. Note that, from the proof in [14], we see that the upper $s-1$ rows of \tilde{A} coincide with those of A . Therefore, if an integrator Φ_α with some α is once constructed, we need just one time of evaluation f for the construction of another integrator $\Phi_{\hat{\alpha}}$ ($\hat{\alpha} \neq \alpha$). 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 can take r as $p-3$ in Definition 3, if the base integrator has order p . Therefore we can apply adaptive step size controls (see e.g. [12]) to the proposed method because two numerical integrators which have order p and 3 are obtained by additionally evaluating the function f only once. A natural question arise here whether a way to construct a family whose r is independent of p exists. We think this is a difficult question to answer. Perturbed collocation methods were well investigated theoretically, in particular, in terms of their order of accuracy by Nørsett–Wanner [14]. 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 [14] have already pointed out.

Remark 4. Another way to parametrize explicit RK methods is to consider parameterizations for each explicit RK method, 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 parametrization technique proposed in this paper is free from

this difficulty. Because parameterizing RK coefficients matrix A is equivalent to adding parametrized polynomials to N_j ($1 \leq j \leq s$) which compose the perturbation operator, we can intuitively say that a nice parametrization 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 parametrize them with the same numbers of parameters as the numbers 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

$$\begin{pmatrix} 1 & c_1 & \dots & c_1^{s-1} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ 1 & c_s & \dots & c_s^{s-1} \end{pmatrix} \rightarrow \begin{pmatrix} 1 & c_1 & \dots & c_1^{s-1} \\ \vdots & & & \vdots \\ 0 & \dots & \hat{d}_{s-2}^{(1)} & \hat{d}_{s-1}^{(1)} & \hat{d}_s^{(1)} \\ 0 & \dots & 0 & \hat{d}_{s-1}^{(2)} & \hat{d}_s^{(2)} \end{pmatrix}. \quad (45)$$

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

$$d_{s-2}^{(1)} = \frac{1}{(s-3)!} \hat{d}_{s-2}^{(1)}, \quad d_{s-1}^{(1)} = \frac{1}{(s-2)!} \hat{d}_{s-1}^{(1)}, \quad d_s^{(1)} = \frac{1}{(s-1)!} \hat{d}_s^{(1)}, \quad (46)$$

$$d_{s-1}^{(2)} = \frac{1}{(s-2)!} \hat{d}_{s-1}^{(2)}, \quad d_s^{(2)} = \frac{1}{(s-1)!} \hat{d}_s^{(2)}. \quad (47)$$

- **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 $\{c_i\}_{1 \leq i \leq s}$:

$$\tilde{N}_j = N_j \quad (j = 1, \dots, s-3), \quad (48)$$

$$\tilde{N}_{s-2} = N_{s-2} + \alpha d_{s-2}^{(1)} \prod_{k=1}^{s-1} (t - c_k), \quad (49)$$

$$\tilde{N}_{s-1} = N_{s-1} + (\alpha d_{s-1}^{(1)} + \beta d_{s-1}^{(2)}) \prod_{k=1}^{s-1} (t - c_k), \quad (50)$$

$$\tilde{N}_s = N_s + (\alpha d_s^{(1)} + \beta d_s^{(2)}) \prod_{k=1}^{s-1} (t - c_k). \quad (51)$$

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 showing 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:

$$\frac{dy}{dt} = J\nabla H(y), \quad y \in \mathbb{R}^4, \quad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \in \mathbb{R}^{4 \times 4}, \quad (52)$$

$$H(y) = H(q_1, q_2, p_1, p_2) = \frac{1}{2}(p_1^2 + p_2^2) + U(q_1, q_2), \quad (53)$$

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 for describing stellar motion, followed for a very long time, inside the (gravitational) potential $U(y_0)$ where y_0 denotes the initial state (see e.g. [11]). In our experiments we set the potential to

$$U(q_1, q_2) = \frac{1}{2}(q_1^2 + q_2^2) + q_1^2 q_2 - \frac{1}{3} q_2^3. \quad (54)$$

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 RK4 with a sufficiently small step size $h = 2/300$. Fig. 1 also displays the contours of the potential. 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 [12] is taken as the embedded RK method. 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, 1-parameter projection methods are all the same due to Theorem 2; 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 differences among 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 utilize adaptive step size control (see Remark 4); however, the proposed method is not so appropriate as IDT because the proposed method needs one extra evaluation of f , thus the proposed method is a bit worse in this sense.

In Fig. 2 and Fig. 3 we show numerical solutions by IDT and the proposed method respectively. Both 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.

Comparing the numerical solutions, we find that the solution by the proposed method are closer to the reference solutions than that by IDT. In particular, we

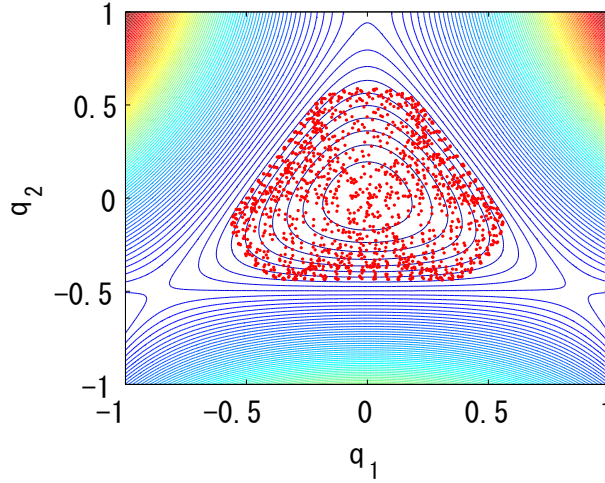


Fig. 1: A reference solution (represented as points) by RK4 with sufficiently small step size $h = 2/300$. The solid lines are the contours of the potential of the Hénon-Heiles equation.

see that the solution by the proposed method distributes appropriately against the potential contours.

To observe consequences of the proposed parametrization, 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 IDT are often large, while the values of the proposed method are constantly small. The reason IDT often takes large parameter seems that in such occasions the search direction is almost perpendicular to ∇H and ceases to work with the constant step size h . Accordingly, there the corrections by projection also get large values. This can cause a certain instability, unless some remedy such as adaptive time stepping is utilized, as mentioned before. In contrast, the proposed method does not exhibit such a behavior even with this large time step $h = 2/3$, and consequently, the correction remains small. Therefore, we feel we can say that these figures support numerically our expectation that the proposed parametrization is natural and yields small corrections.

The naturalness of the proposed parametrization 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. 4, we show lines which denote zeros of $g(\alpha, h) = H(\Phi_\alpha(y_0)) - H(y_0)$, $(h, \alpha) \in [-0.9, 0.9] \times [-9, 9]$ where y_0 is a state vector which IDT takes large α ($t = 8/3$). The top of Fig. 4 is the contour of IDT 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 IDT sharply goes

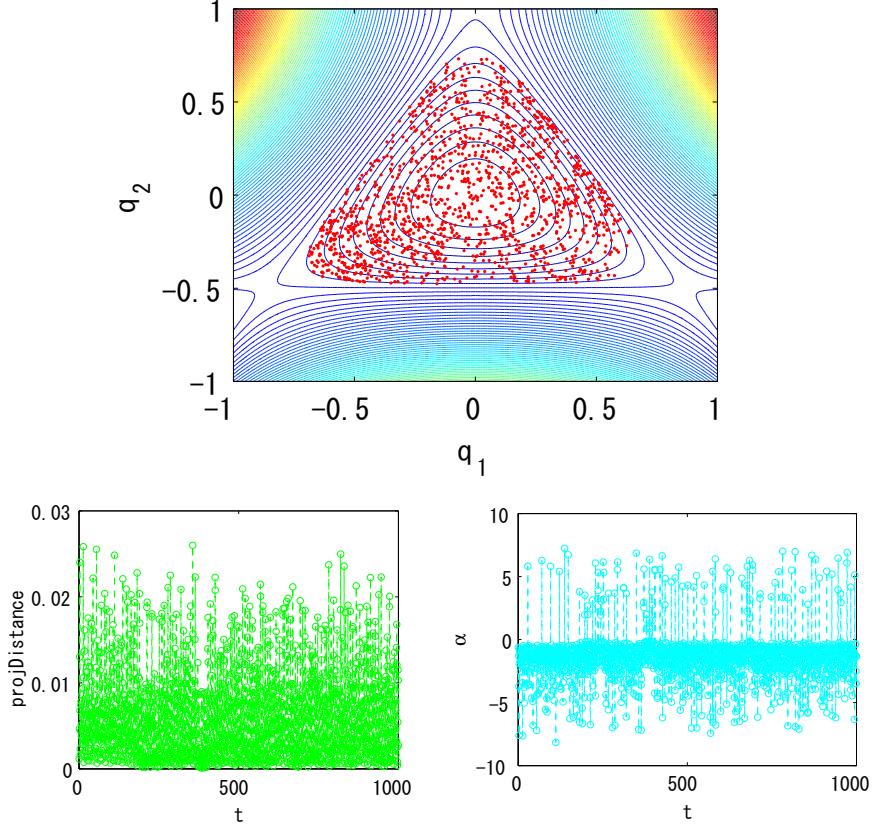


Fig. 2: IDT: top = the numerical solution with $h = 2/3$, bottom left = value of the corrections by projection (its label “projDistance” means the corrections by projection), bottom right = value of the parameter.

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. Therefore we see the naturalness of the proposed parameterization. But again we would like to emphasize that this happens only occasionally (recall Fig. 2; in most time steps the corrections are small), and even if it can happen, that can be easily avoided by adaptive time stepping. 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. There 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

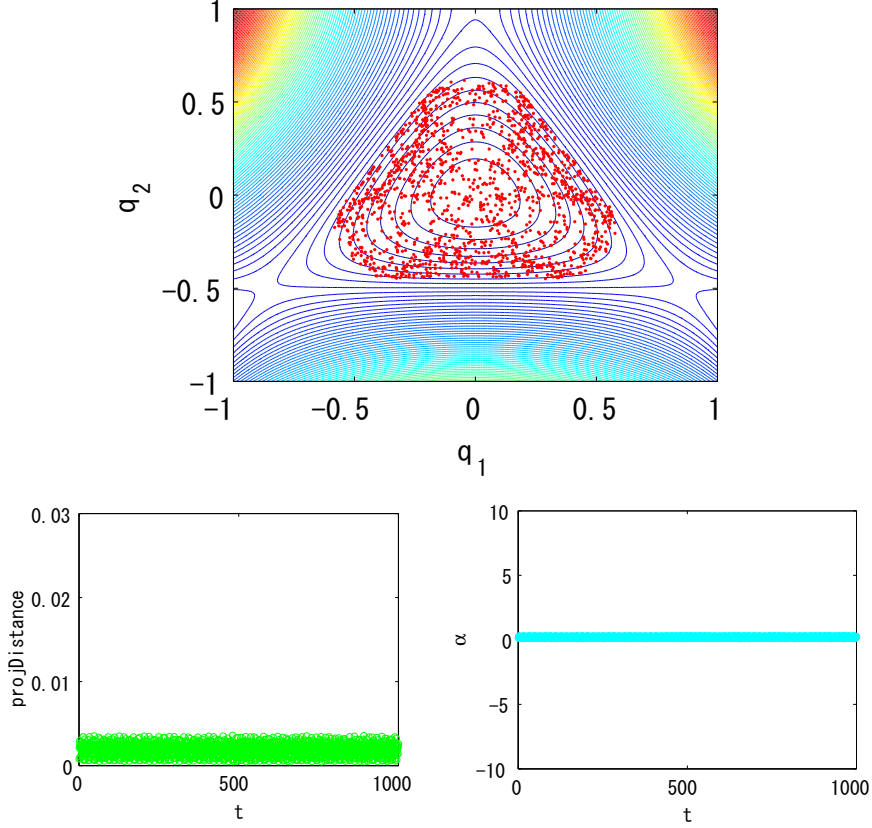


Fig. 3: The proposed method: top = the numerical solution with $h = 2/3$, bottom left = value of the corrections by projection (its label “projDistance” means the corrections by projection), bottom right = value of the parameter. The maximum value of α is about 0.3.

the proposed strategy in fact result in small corrections robustly.

Here we comment some further remarks. First, we can 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 [3, 8]. 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 is 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 circumstances. 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.

A possible future work is to develop a rigorous theoretical explanation on

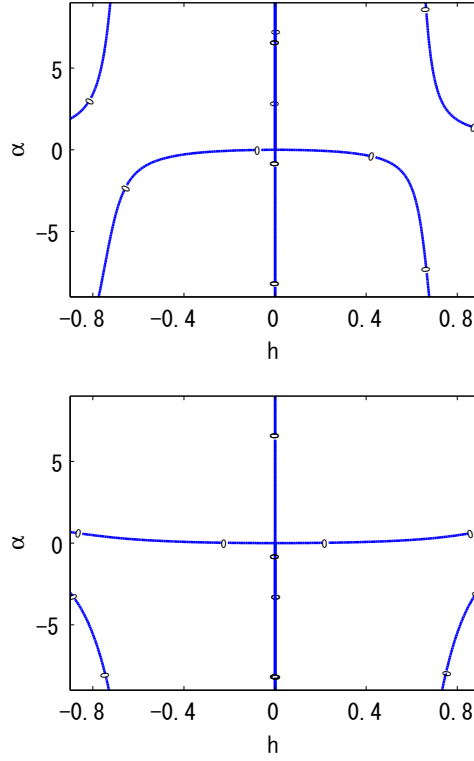


Fig. 4: Lines which display $g(\alpha, h) = 0$, $(h, \alpha) \in [-0.9, 0.9] \times [-9, 9]$. Top shows lines of IDT. Bottom shows lines of the proposed method.

how the proposed parametrization, employed in higher order perturbations, result in the small corrections. And also through such theoretical investigations, “the best 1-parameter projection method” should be looked for. The proposed strategy via perturbed collocation methods serves as a working stage of such challenges, and the proposed parametrization is just a single example of good parametrization.

Acknowledgments

The author would like to thank Prof. Takayasu Matsuo for helpful comments and discussion.

References

- [1] Brugnano, L., Iavernaro, F., and Trigiante, D., Energy- and quadratic invariant-preserving integrators based upon Gauss collocation formulae, *SIAM J. Numer. Anal.*, **50** (2012), 6, 2897–2916.

- [2] Calvo, M., Hernández-Abreu, D., Montijano, J. I., and Rández, L., On the preservation of invariants by explicit Runge–Kutta methods, *SIAM J. Sci. Comput.*, **28** (2006), 868–885.
- [3] Calvo, M., Laburta, M. P., and Montijano, J. I., Projection methods preserving Lyapunov functions, *BIT Numer. Math.*, **50** (2010), 223–241.
- [4] Celledoni, E., McLachlan, R. I., McLaren, D. I., Orwen, B., Quispel, G. R. W., and Wright, G. R., Energy preserving Runge–Kutta methods, *M2AN Math. Model. Numer. Anal.*, **43** (2009), 645–649.
- [5] Celledoni, E., Grimm, V., McLachlan, R. I., McLaren, D. I., O’Neale, D., Orwen, B., and Quispel, G. R. W., Preserving energy resp. dissipation in numerical PDEs using the “Average Vector Field” method, *J. Comput. Phys.*, **231** (2012), 6770–6789.
- [6] Dahlby, M., Owren, B., and Yaguchi, T., Preserving multiple first integrals by discrete gradients, *J. Phys A*, **44** (2011), 305205.
- [7] Gonzalez, O., Time integration and discrete Hamiltonian systems, *J. Non-linear Sci.*, **6** (1996), 449–467.
- [8] Grimm, V. and Quispel, G.R.W., Geometric integration methods that preserve Lyapunov functions, *BIT Numer. Math.*, **45** (2005), 709–723.
- [9] Hairer, E., Symmetric projection methods for differential equations on manifolds, *BIT*, **40** (2000), 726–734.
- [10] Hairer, E., Energy-preserving variant of collocation methods, *J. Numer. Anal. Ind. Appl. Math.*, **5** (2010), 73–84.
- [11] Hairer, E., Lubich, C., and Wanner, G., *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, Springer-Verlag, Berlin, 1993.
- [12] Hairer, E., Nørsett, S. P., and Wanner, G., *Solving ordinary differential equations I. Nonstiff problems*, Springer Series in Computational Mathematics 8. Springer, Berlin, 2nd edition, 1993.
- [13] Iserles, A. and Zanna A., Preserving algebraic invariants with Runge–Kutta methods, *J. Comput. Appl. Math.* **125** (2000), 69–81.
- [14] Nørsett, S. P., and Wanner, G., Perturbed collocation and Runge–Kutta methods, *Numer. Math.*, **38** (1981), 193–208.
- [15] McLachlan, R. I., Quispel, G. R. W., and Robidoux, N., Geometric integration using discrete gradients, *R. Soc. Lond. Phil. Trans. Ser. A Math. Phys. Eng. Sci.*, **357** (1999) 1021–1045.
- [16] Quispel, G. R. W. and McLaren, D. I., Integral-preserving integrators, *J Phys. A*, **37** (2004), L489–L495.
- [17] Quispel, G. R. W. and McLaren, D. I., A new class of energy-preserving numerical integration methods, *J. Phys. A*, **41** (2008) 045206.

- [18] Shampine, L. F., Conservation laws and the numerical solutions of ODEs, *Comput. Math. Appl.* **12** (1986), 1287–1296.
- [19] Yoshida, H., Construction of higher order symplectic integrators, *Phys. Lett. A*, **150** (1990), 262–268.