

REVIEW

Open Access



Some robust integrators for large time dynamics

Dina Razafindralandy^{1*} , Vladimir Salnikov², Aziz Hamdouni¹ and Ahmad Deeb¹

*Correspondence:

dina.razafindralandy@univ-lr.fr

¹Laboratoire des Sciences de l'Ingénieur pour l'Environnement (LaSIE), UMR 7356 CNRS, Université de La Rochelle, Avenue Michel Crépeau, 17042 La Rochelle Cedex, France
Full list of author information is available at the end of the article

Abstract

This article reviews some integrators particularly suitable for the numerical resolution of differential equations on a large time interval. Symplectic integrators are presented. Their stability on exponentially large time is shown through numerical examples. Next, Dirac integrators for constrained systems are exposed. An application on chaotic dynamics is presented. Lastly, for systems having no exploitable geometric structure, the Borel–Laplace integrator is presented. Numerical experiments on Hamiltonian and non-Hamiltonian systems are carried out, as well as on a partial differential equation.

Keywords: Symplectic integrators, Dirac integrators, Long-time stability, Borel summation, Divergent series

Introduction

In many domains of mechanics, simulations over a large time interval are crucial. This is, for instance, the case in molecular dynamics, in weather forecast or in astronomy. While many time integrators are available in literature, only few of them are suitable for large time simulations. Indeed, many numerical schemes fail to correctly predict the expected physical phenomena such as energy preservation, as the simulation time grows.

For equations having an underlying geometric structure (Hamiltonian systems, variational problems, Lie symmetry group, Dirac structure, . . .), geometric integrators appear to be very robust for large time simulation. These integrators mimic the geometric structure of the equation at the discrete scale.

The aim of this paper is to make a review of some time integrators which are suitable for large time simulations. We consider not only equations having a geometric structure but also more general equations. We first present symplectic integrators for Hamiltonian systems. We show in “Symplectic integrators” section their ability in preserving the Hamiltonian function and some other integrals of motion. Applications will be on a periodic Toda lattice and on n -body problems. To simplify, the presentation is done in canonical coordinates.

In “Dirac integrators” section, we show how to fit a constrained problem into a Dirac structure. We then detail how to construct a geometric integrator respecting the Dirac structure. The presentation will be simplified, and the (although very interesting) theoretical geometry is skipped. References will be given for more in-depth understanding. A

numerical experiment, showing the good long time behaviour of Dirac integrators, will be carried out.

In “Borel–Laplace integrator” section, we present the Borel–Padé–Laplace integrator (BPL). BPL is a general-purpose time integrator, based on a time series decomposition of the solution, followed by a resummation to enlarge the validity of the series and then reducing the numerical cost on a large time simulation. Finally, the long time behaviour will be investigated through numerical experiments on Hamiltonian and non-Hamiltonian system, as well as on a partial differential equation. Numerical cost will be examined when relevant.

Symplectic integrators

We first make some reminder on Hamiltonian systems and their flows in canonical coordinates. Some examples of symplectic integrators are given afterwards and numerical experiments are presented.

Hamiltonian system

A Hamiltonian system in $\mathbb{R}^d \times \mathbb{R}^d$ is a system of differential equations which can be written as follows:

$$\begin{cases} \frac{d\mathbf{q}}{dt} = \frac{\partial H}{\partial \mathbf{p}}, \\ \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}}, \end{cases} \tag{1}$$

the Hamiltonian H being a function of time and of the unknown vectors $\mathbf{q} = (q_1, \dots, q_d)^\top$ and $\mathbf{p} = (p_1, \dots, p_d)^\top$. Equation (1) can be written in a more compact way as follows:

$$\frac{d\mathbf{u}}{dt} = \mathbb{J} \nabla H \tag{2}$$

where $\mathbf{u} = (\mathbf{q}, \mathbf{p})^\top$, $\nabla H = \frac{\partial H}{\partial \mathbf{u}}$ and \mathbb{J} is the skew-symmetric matrix

$$\mathbb{J} = \begin{pmatrix} 0 & \mathbb{I}_d \\ -\mathbb{I}_d & 0 \end{pmatrix},$$

\mathbb{I}_d being the identity matrix of \mathbb{R}^d .

The flow of the Hamiltonian system (2) at time t is the function Φ_t which, to an initial condition \mathbf{u}^0 associates the solution $\mathbf{u}(t)$ of the system. More precisely, Φ_t is defined as:

$$\begin{aligned} \mathbb{R}^d \times \mathbb{R}^d &\longrightarrow \mathbb{R}^d \times \mathbb{R}^d \\ \Phi_t : \\ \mathbf{u}^0 = (\mathbf{q}^0, \mathbf{p}^0)^\top &\longmapsto \mathbf{u}(t) = (\mathbf{q}(t), \mathbf{p}(t))^\top. \end{aligned} \tag{3}$$

The property that $\mathbb{J}^{-1} = \mathbb{J}^\top = -\mathbb{J}$ leads to the symplecticity property of Φ_t :

$$(\nabla \Phi_t)^\top \mathbb{J} (\nabla \Phi_t) = \mathbb{J}. \tag{4}$$

Note that \mathbb{J} can be seen as an area form, in the following sense. If \mathbf{v} and \mathbf{w} are two vectors of $\mathbb{R}^d \times \mathbb{R}^d$, with components

$$\mathbf{v} = (v_{q_1}, \dots, v_{q_d}, v_{p_1}, \dots, v_{p_d})^\top, \quad \mathbf{w} = (w_{q_1}, \dots, w_{q_d}, w_{p_1}, \dots, w_{p_d})^\top$$

then

$$\mathbf{v}^\top \mathbb{J} \mathbf{w} = \sum_{i=1}^d (v_{q_i} w_{p_i} - v_{p_i} w_{q_i}).$$

In words, $\mathbf{v}^\top \mathbb{J} \mathbf{w}$ is the sum of the areas formed by \mathbf{v} and \mathbf{w} in the planes (q_i, p_i) . The symplecticity property (4) then means that the flow of a Hamiltonian system is area preserving.

In the sequel, H is assumed autonomous in time. It can then be shown that H is preserved along trajectories.

Flow of a numerical scheme

Consider a numerical scheme which computes an approximation \mathbf{u}^n of the solution $\mathbf{u}(t^n)$ of Eq. (2) at time t^n . The flow of this scheme is defined as

$$\varphi_{t^n} : \mathbf{u}^0 = (\mathbf{q}^0, \mathbf{p}^0)^\top \longmapsto \mathbf{u}^n = (\mathbf{q}^n, \mathbf{p}^n)^\top \tag{5}$$

For a one-step integrator, with a time step $\Delta t = t^{n+1} - t^n$ (which may depend on n), it is more convenient to work with the one-step flow

$$\varphi_{\Delta t}^n : \mathbf{u}^n \longmapsto \mathbf{u}^{n+1} \tag{6}$$

As an example, consider the explicit Euler integration scheme

$$\begin{cases} \mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t \frac{\partial H}{\partial \mathbf{p}}(\mathbf{q}^n, \mathbf{p}^n) \\ \mathbf{p}^{n+1} = \mathbf{p}^n - \Delta t \frac{\partial H}{\partial \mathbf{q}}(\mathbf{q}^n, \mathbf{p}^n). \end{cases}$$

The one-step flow of this scheme is

$$\varphi_{\Delta t}^n(\mathbf{u}^n) = \mathbf{u}^n + \Delta t \mathbb{J} \nabla H(\mathbf{u}^n). \tag{7}$$

The one-step flow of a fourth order Runge–Kutta scheme is

$$\varphi_{\Delta t}^n(\mathbf{u}^n) = \mathbf{u}^n + \Delta t \frac{f_1 + 2f_2 + 2f_3 + f_4}{6} \tag{8}$$

where

$$\begin{aligned} f_1 &= \mathbb{J} \nabla H(\mathbf{u}^n), & f_2 &= \mathbb{J} \nabla H(\mathbf{u}^n + \frac{\Delta t}{2} f_1), \\ f_3 &= \mathbb{J} \nabla H(\mathbf{u}^n + \frac{\Delta t}{2} f_2), & f_4 &= \mathbb{J} \nabla H(\mathbf{u}^n + \Delta t f_3). \end{aligned}$$

Some symplectic integrators

A time integrator is called symplectic if its flow is symplectic, meaning that

$$(\nabla \varphi_{t^n})^\top \mathbb{J} (\nabla \varphi_{t^n}) = \mathbb{J}. \tag{9}$$

Geometrically, a symplectic integrator is then a time scheme which preserves the area form. For a one-step scheme, this property is equivalent to

$$(\nabla \varphi_{\Delta t}^n)^\top \mathbb{J} (\nabla \varphi_{\Delta t}^n) = \mathbb{J} \tag{10}$$

at each iteration n .

When $d = 1$, it can easily be shown that, for an explicit Euler scheme,

$$(\nabla \varphi_{\Delta t}^n)^\top \mathbb{J} (\nabla \varphi_{\Delta t}^n) = \left(1 + \Delta t^2 \left(\frac{\partial^2 H}{\partial q^2} \frac{\partial^2 H}{\partial p^2} - \frac{\partial^2 H}{\partial q \partial p} \right) \right) \mathbb{J}, \tag{11}$$

meaning that the explicit Euler scheme is not symplectic. Neither the implicit Euler scheme is symplectic. By contrast, by mixing the explicit and the implicit Euler scheme, we get a symplectic scheme, called symplectic Euler scheme, defined as follows

$$\begin{cases} \mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t \frac{\partial H}{\partial \mathbf{p}}(\mathbf{q}^n, \mathbf{p}^{n+1}), \\ \mathbf{p}^{n+1} = \mathbf{p}^n - \Delta t \frac{\partial H}{\partial \mathbf{q}}(\mathbf{q}^n, \mathbf{p}^{n+1}). \end{cases} \tag{12}$$

Note that in (12), one can take $(\mathbf{q}^{n+1}, \mathbf{p}^n)$ in the right-hand-side instead of $(\mathbf{q}^n, \mathbf{p}^{n+1})$. This leads to the other symplectic Euler scheme.

Both symplectic Euler schemes are first order. A way to get a second order scheme is to compose two symplectic Euler schemes with time steps $\Delta t/2$. One then obtains the Störmer–Verlet schemes [13]. Another way is to take the mid-points in the right-hand side of an Euler scheme [15], yielding the mid-point scheme:

$$\begin{cases} \mathbf{q}^{n+1} = \mathbf{q}^n + h \nabla_{\mathbf{p}} H \left(\mathbf{q}^{n+\frac{1}{2}}, \mathbf{p}^{n+\frac{1}{2}} \right), \\ \mathbf{p}^{n+1} = \mathbf{p}^n - h \nabla_{\mathbf{q}} H \left(\mathbf{q}^{n+\frac{1}{2}}, \mathbf{p}^{n+\frac{1}{2}} \right), \end{cases} \tag{13}$$

where

$$\mathbf{q}^{n+\frac{1}{2}} = \frac{\mathbf{q}^n + \mathbf{q}^{n+1}}{2}, \quad \mathbf{p}^{n+\frac{1}{2}} = \frac{\mathbf{p}^n + \mathbf{p}^{n+1}}{2}.$$

Symplectic Runge–Kutta schemes of higher order can be built as follows. An s -stage Runge–Kutta integrator of Eq. (1) is defined as [14,29]:

$$\begin{aligned} \mathbf{U}_i &= \mathbf{u}^n + \Delta t \sum_{j=1}^s \alpha_{ij} \mathbb{J} \nabla H(\mathbf{U}_j), \quad i = 1, \dots, s, \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \Delta t \sum_{i=1}^s \beta_i \mathbb{J} \nabla H(\mathbf{U}_i), \end{aligned} \tag{14}$$

for some real numbers $\alpha_{ij}, \beta_i, i, j = 1, \dots, s$. Scheme (14) is symplectic if and only if the coefficients verify the relation [18,28]:

$$\beta_i \beta_j = \beta_i \alpha_{ij} + \beta_j \alpha_{ji}, \quad i, j = 1, \dots, s. \tag{15}$$

An example of symplectic Runge–Kutta scheme is the 4th order, 3-stage scheme defined by the coefficients

$$\alpha = \begin{pmatrix} \frac{b}{2} & 0 & 0 \\ b & \frac{1}{2} - b & 0 \\ b & 1 - 2b & \frac{b}{2} \end{pmatrix}, \quad \beta = (b \quad 1 - 2b \quad b), \tag{16}$$

where

$$b = \frac{2 + 2^{1/2} + 2^{-1/3}}{3}.$$

Many other variants of symplectic Runge–Kutta methods can be found in the literature (see for example [10]).

Symplectic integrators do not preserve exactly the Hamiltonian in general. However, the symplecticity condition seems to be strong enough that, experimentally, symplectic integrators exhibit a good behaviour toward the preservation property. In fact, we have the following error estimation on the Hamiltonian [1, 12]:

$$|H(p^n, q^n) - H(p^0, q^0)| = O(\Delta t^r) \quad \text{for } n\Delta t \leq e^{\frac{\gamma}{2\Delta t}} \quad (17)$$

for some constant $\gamma > 0$, r being the order of the symplectic scheme. This relation states that the error is bounded over an exponentially long discrete time. Moreover, it was shown in [5] that symplectic Runge–Kutta methods preserve exactly quadratic invariants.

In the next subsection, some interesting numerical properties of symplectic schemes are highlighted through some model problems.

Numerical experiments

Periodic Toda lattice

The evolution of a periodic Toda lattice with d particles can be described with the Hamiltonian function

$$H = \sum_{k=1}^d \left(\frac{1}{2} p_k^2 + e^{q_k - q_{k+1}} \right)$$

where q_k is the (one-dimensional) position of the k -th particle, $q_{d+1} = q_1$ and p_k is its momentum. A periodic Toda lattice is completely integrable and it is known that the eigenvalues of the following matrix L are first integrals of the system:

$$L = \begin{pmatrix} a_1 & b_1 & 0 & 0 & 0 & \cdots & 0 & b_d \\ b_1 & a_2 & b_2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & b_2 & a_3 & b_3 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & & & \\ 0 & 0 & 0 & \cdots & 0 & b_{d-2} & a_{d-1} & b_{d-1} \\ b_d & 0 & 0 & \cdots & 0 & 0 & b_{d-1} & a_d \end{pmatrix} \quad (18)$$

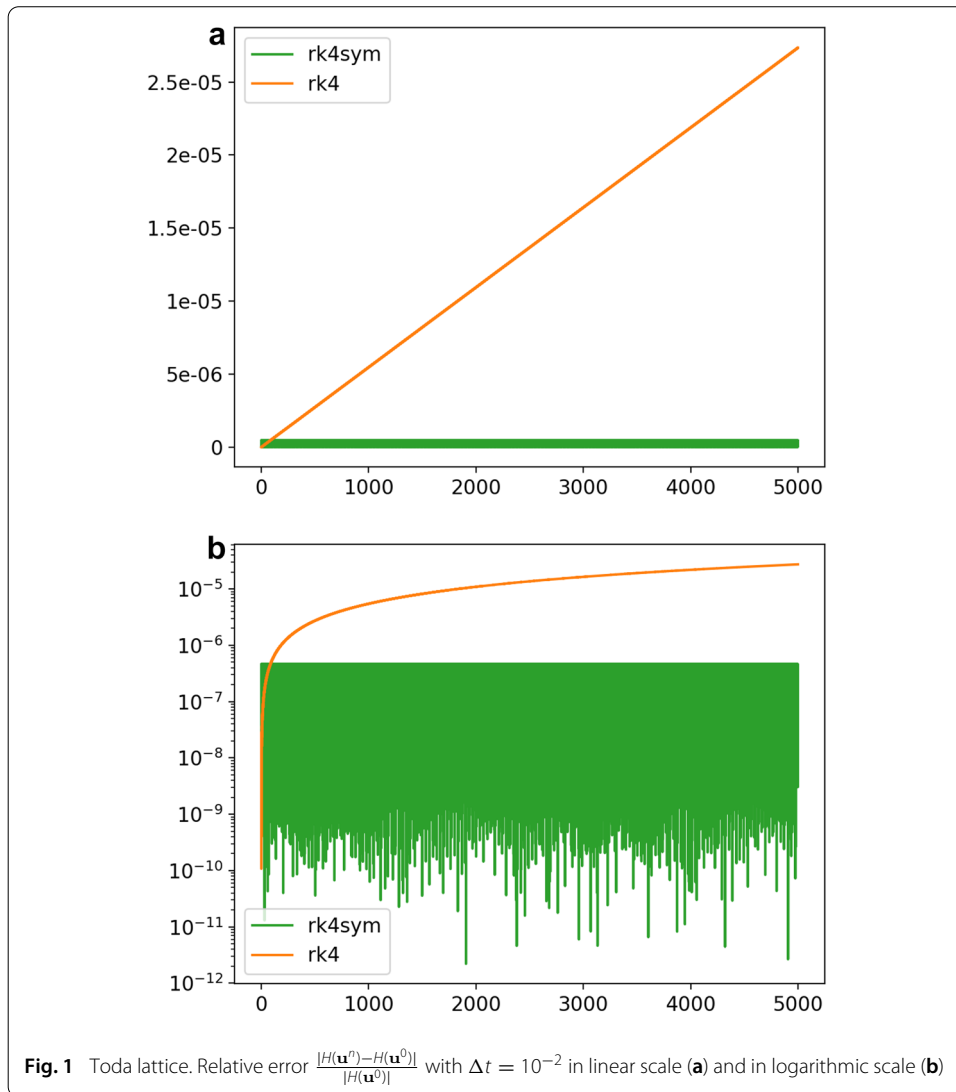
where

$$a_k = -\frac{1}{2} p_k, \quad b_k = \frac{1}{2} e^{\frac{1}{2}(q_k - q_{k+1})}.$$

In the numerical test, we consider $d = 3$ particles, positioned initially at $q_1 = 0$, $q_2 = 2$ and $q_3 = 3$. The initial momenta are $p_1 = 0.5$, $p_2 = -1.5$ and $p_3 = 1$.

First, we choose a time step $\Delta t = 10^{-2}$. The Hamiltonian equation is solved with the classical 4-th order Runge–Kutta scheme (RK4) and the symplectic version (RK4sym) defined by (16) up to $t = 5000$. The relative error on the Hamiltonian is presented in Fig. 1. As can be seen, the RK4 error oscillates and increases globally linearly. It remains acceptable for $t < 5000$ since it does not exceed $2.735 \cdot 10^{-5}$. The RK4sym error also oscillates but is much closer to zero. It is bounded by $4.625 \cdot 10^{-7}$, that is an order of 10^{-2} below RK4 error at $t = 5000$, as can be observed in Fig. 1b. Figure 2 shows that both schemes globally preserve the three eigenvalues of the matrix L .

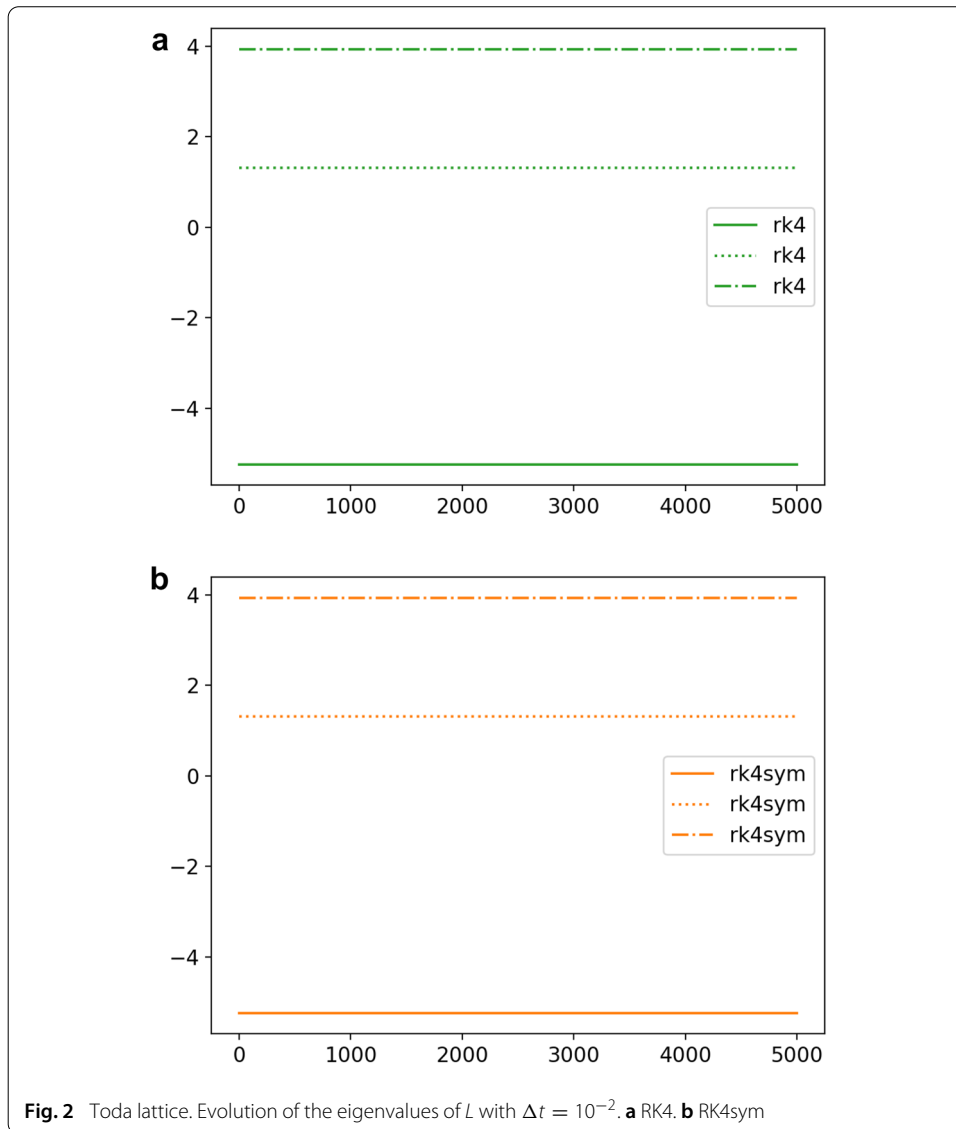
To analyze the robustness of the schemes, Δt is now set to 10^{-1} . With this time step, the error of the classical Runge–Kutta scheme increases quickly from the first iterations, as can be observed in Fig. 3. It reaches 50% around $t = 4.2 \cdot 10^3$. As for it, the RK4sym



error oscillates around $2.26 \cdot 10^{-3}$ but does not present any increasing global tendency. Its highest value is about $3.27 \cdot 10^{-3}$, as can be checked in the figure.

A comparison between RK4 and the symplectic Euler scheme defined in (12) is also given in Fig. 3b. It shows that the error of the Euler scheme oscillates around $0.83 \cdot \Delta t$ and does not exceed $2.42 \Delta t$. So, for t greater than 820, even the first order symplectic Euler scheme produces an error smaller than the 4-th order non-symplectic Runge–Kutta scheme on the Hamiltonian.

The evolution of the eigenvalues of the matrix L is presented in Fig. 4. It clearly shows that the eigenvalues are not preserved by the classical Runge–Kutta scheme. For example, the computed smallest eigenvalue at $t = 5000$ is about -3.50 whereas its initial value is -5.24 . With the symplectic Runge–Kutta scheme, the error on the smallest eigenvalue oscillates around $4.85 \cdot 10^{-3}$ but does not present an increasing tendency. With the first order symplectic Euler scheme, the oscillations are much more pronounced, as can be observed in Fig. 5, but as with RK4sym, there is no increasing trend. It becomes smaller than the non-symplectic RK4 error when the simulation time increases.



It is clear from these experiments that symplectic schemes are particularly stable for large time simulations, where the user wishes a time step as large as possible to reduce the computation cost. In some situation, even a symplectic scheme with a smaller order gives better results over a long time than a classical integrator.

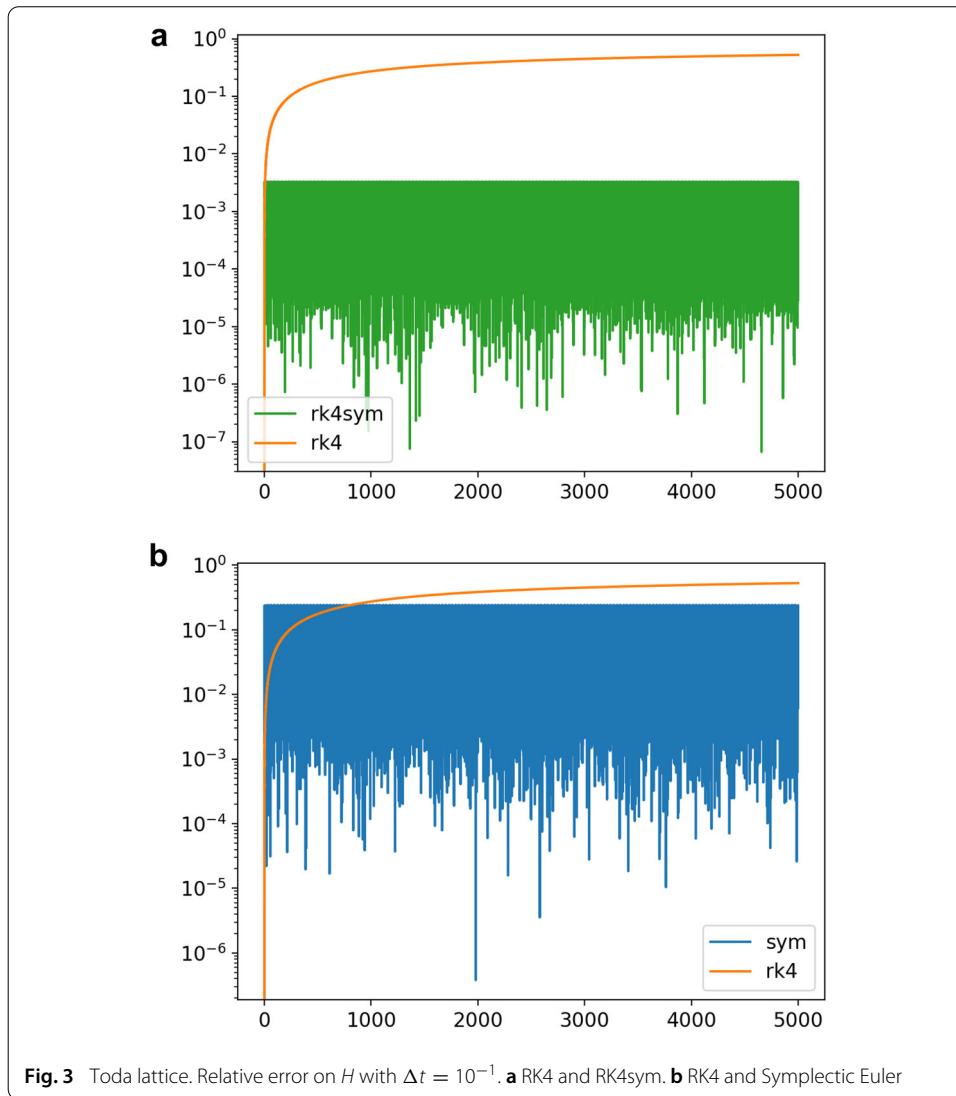
***n*-body problem**

As a second example, consider the system of d bodies subjected to mutual gravitational forces. The evolution of the system is described by the Hamiltonian function

$$H = \sum_{k=1}^d \frac{1}{2} \frac{\|\mathbf{p}_k\|^2}{m_k} - \sum_{k=1}^d \sum_{l=k+1}^d \frac{Gm_k m_l}{\|\mathbf{q}_l - \mathbf{q}_k\|}$$

In this expression, \mathbf{q}_k and \mathbf{p}_k are the position vector and the momentum of the k -th body, m_k is its mass and G is the gravitation constant.

For the simulation, we take $d = 3$. We consider the initial configuration corresponding to the choreographic figure-eight in [4]. The solution is periodic, with period



$T \simeq 6.32591398$. The common trajectory and the initial position are presented in Fig. 6a. The simulation is run up to $t = 2200T$, with a time step $\Delta t = 0.02T$. In these configurations, the classical Runge–Kutta scheme provides a fairly good result but the symplectic scheme is much more accurate regarding the preservation properties. The relative error of RK4 on the Hamiltonian increases linearly and is about $1.87 \cdot 10^{-5}$ at the final time. It can be viewed in Fig. 7a in logarithmic scale. The relative error of RK4sym on H oscillates but never exceeds $9.99 \cdot 10^{-8}$. The errors on the angular momentum are plotted in Fig. 7b. It shows that the error of RK4 is bounded by $1.4 \cdot 10^{-9}$ whereas the error of RK4sym is remains under the machine precision (around $2 \cdot 10^{-12}$).

In a second simulation, the initial position and momentum of the body in the middle in Fig. 6 are changed into their values at $t = T/80$ in the figure-eight solution. The initial configuration of the bodies at the two ends are kept as in the previous simulation. In this case, the figure-eight is broken. A part of the trajectory of each body and their initial positions are presented in Fig. 6b. The evolution of the error on the Hamiltonian and on the angular momentum is plotted in Fig. 8. As can be noticed, the error on the Hamiltonian increases quickly with the classical RK4. It reaches 50 percent at $t \simeq 2166T \simeq 13702$. At

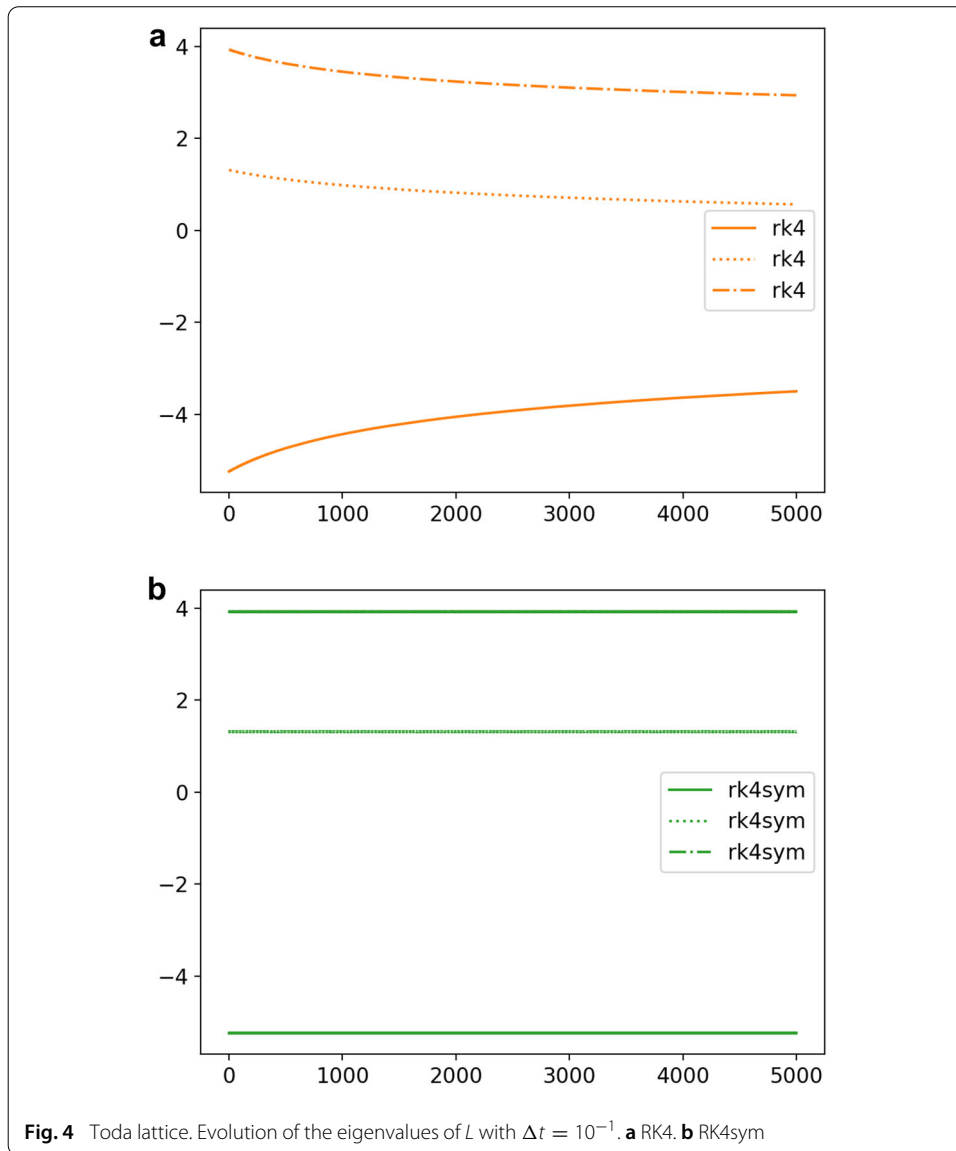


Fig. 4 Toda lattice. Evolution of the eigenvalues of L with $\Delta t = 10^{-1}$. **a** RK4. **b** RK4sym

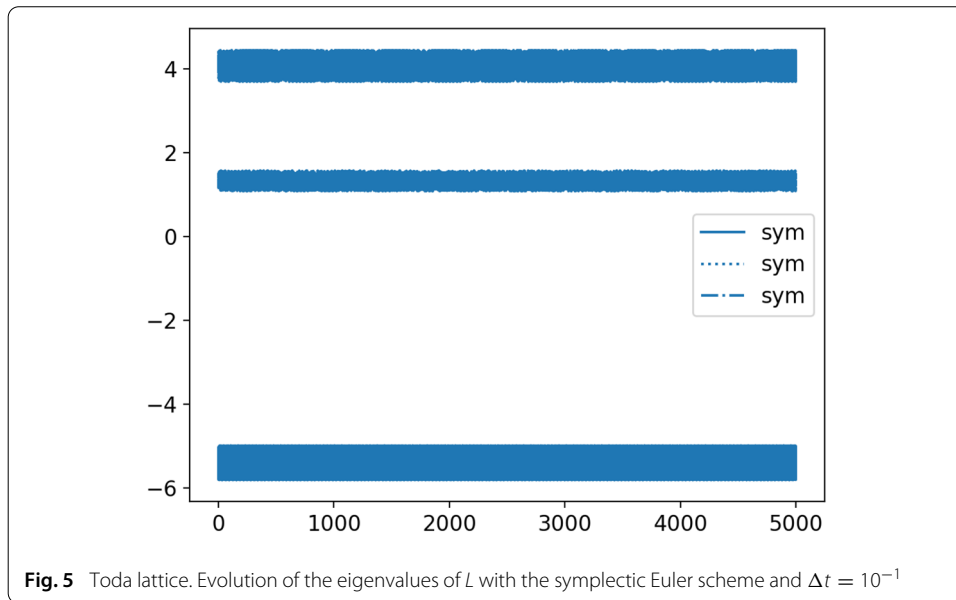
this time value, the error on the Hamiltonian with the symplectic RK4 is about $7.75 \cdot 10^{-4}$. It reaches 1 percent much later, around $t \simeq 6315 \cdot T \simeq 39950$ (not seen on the graphic).

These numerical experiments show again that symplectic schemes are more robust than classical ones for long time dynamics simulation. They have a good behaviour regarding the preservation of the Hamiltonian and some other first integrals despite the perturbation introduced in the initial configuration. Similar results have been obtained in a previous work [24] on an harmonic oscillator, on Kepler’s problem and on vortex dynamics.

Obviously, not all mechanical systems fit into the Hamiltonian formalism, hence there are other geometric constructions that are worth being considered in the context of structure-preserving integrators.

Dirac integrators

In this section, we give an overview of the so-called Dirac structures and describe a class of mechanical systems where those appear naturally, namely systems with constraints.



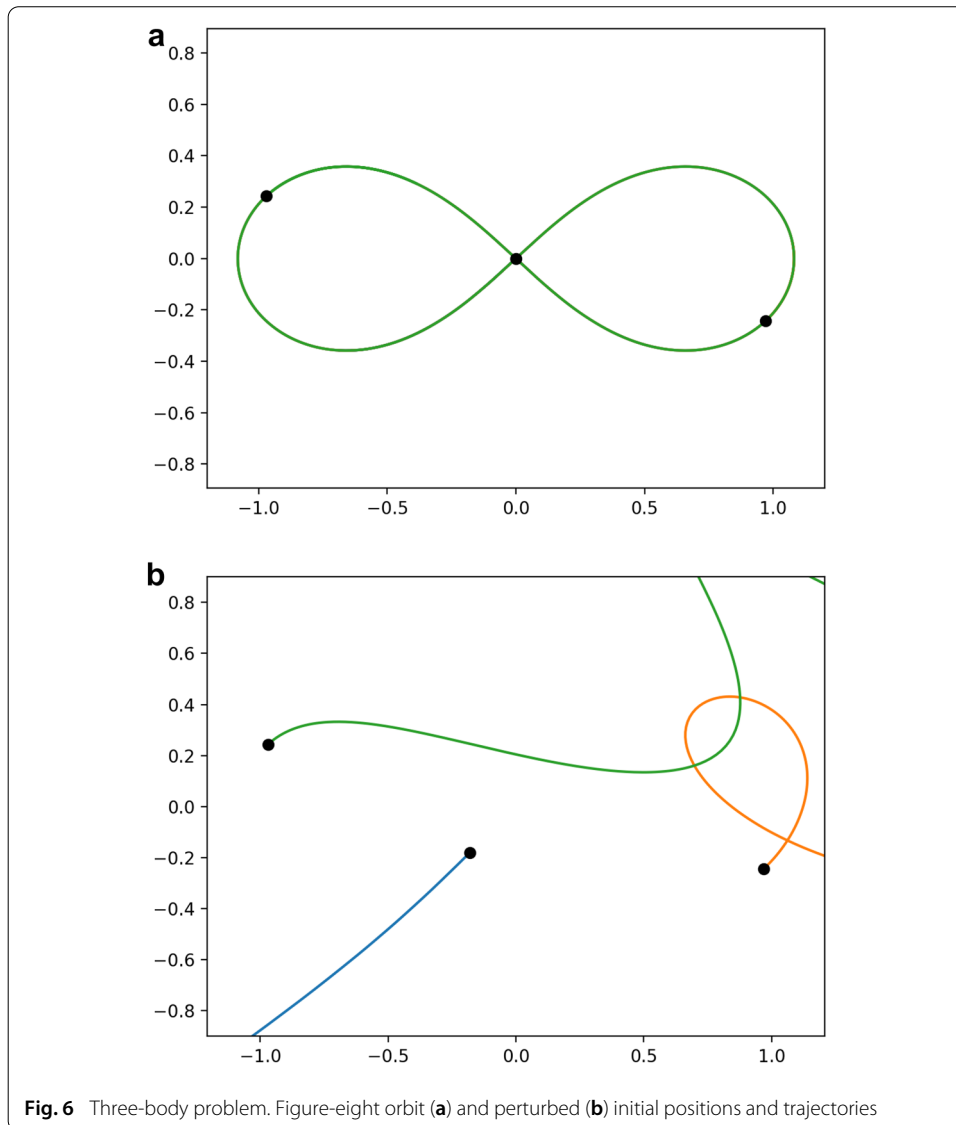
Originally, Dirac structures appear in the work of Courant [6]. The initial motivation was coming from mechanics. As is known, for mechanical systems one can choose between Lagrangian and Hamiltonian formalisms, both being equivalent in finite dimension. The rough idea behind Dirac structures is to consider both formalisms simultaneously, i.e. working with velocities *and* momenta, however not forgetting that those are dependent variables. Geometrically, this means that instead of choosing between the tangent and cotangent bundles TM or T^*M for the phase space, we consider their direct sum $E = TM \oplus T^*M$ and a subbundle of it, subject to some compatibility conditions. Somehow, the original work did not have direct applications to mechanics, since the geometry of the problem turned out to be rather intricate, and gave rise to a lot of development in *higher structures* and in *theoretical physics*. In the last decade, however, it was revived with the introduction of so-called port–Hamiltonian [32] and implicit Lagrangian systems [33,34].

Geometric construction

Not to overload the presentation here with geometric details, let us talk about spaces instead of bundles. To recover the geometric picture, a motivated reader is invited to read the original paper [6] of Courant or the overview of relevant results in [27]. The object of study will be \mathbb{R}^{2d} (in the same notations as in the beginning of the previous section), and some natural construction around it. We are going to view it as $\mathbb{R}^{2d} = \mathbb{R}^d \times V^*$, that is the trivial bundle over \mathbb{R}^d with a fiber being V^* —the dual of some d -dimensional vector space V . Morally, V is the space of velocities \mathbf{v} at each point \mathbf{q} and V^* corresponds to the space of momenta \mathbf{p} . In coordinates:

$$\mathbf{q} = (q_1, \dots, q_d)^T \in \mathbb{R}^d, \quad \mathbf{v} = (v_1, \dots, v_d)^T \in V, \quad \mathbf{p} = (p_1, \dots, p_d)^T \in V^*.$$

In this setting, imposing constraints on the system means defining some restriction on couples (\mathbf{q}, \mathbf{v}) , i.e. not all the points \mathbf{q} are permitted, and at each point \mathbf{q} , \mathbf{v} is not arbitrary, but belongs to a subspace of V . Under some regularity conditions, one can say, that at each velocity \mathbf{v} belongs to a set $\Delta_{\mathbf{q}} \subset V$ which is the kernel of a set of linear forms α^a .



And since everything depends on the point \mathbf{q} , globally, the permitted vector fields $\mathbf{v}(\mathbf{q})$ live in the kernel $\Delta \subset \mathbb{R}^d \times V$ of m differential 1-forms $\alpha^a(\mathbf{q})$, $a = 1, \dots, m$.

To transfer this construction from $\mathbb{R}^d \times V$ to $\mathbb{R}^d \times V^*$, one needs to consider double vector bundles [31]. In our simplified setting this means that the space of interest is $\mathcal{V} = \mathbb{R}^{4d}$, where each component has some geometric interpretation. Namely, we consider \mathcal{V} as the tangent to $\mathbb{R}^d \times V^*$. Naturally, $\mathbb{R}^d \times V$ is embedded in \mathcal{V} (recall that V is tangent to \mathbb{R}^d). The constraint set is then a subset $\tilde{\Delta} \subset \mathcal{V}$, and the differential forms $\alpha^a(\mathbf{q})$ generate its annihilator Δ_0 that naturally belongs to \mathcal{V}^* . Note that, since $\mathbb{R}^d \times V^*$ is a symplectic space, it is equipped with a bilinear antisymmetric non-degenerate closed form Ω (this form Ω is the generalisation of the matrix \mathbb{J} of “Symplectic integrators” section in non-canonical coordinates). One can then construct a symplectic mapping $\Omega^b: \mathcal{V} \rightarrow \mathcal{V}^*$.

We can now define the *Dirac structure*¹ associated to the system with constraints:

¹The correct terminology is *almost Dirac structure*. For details see [27].

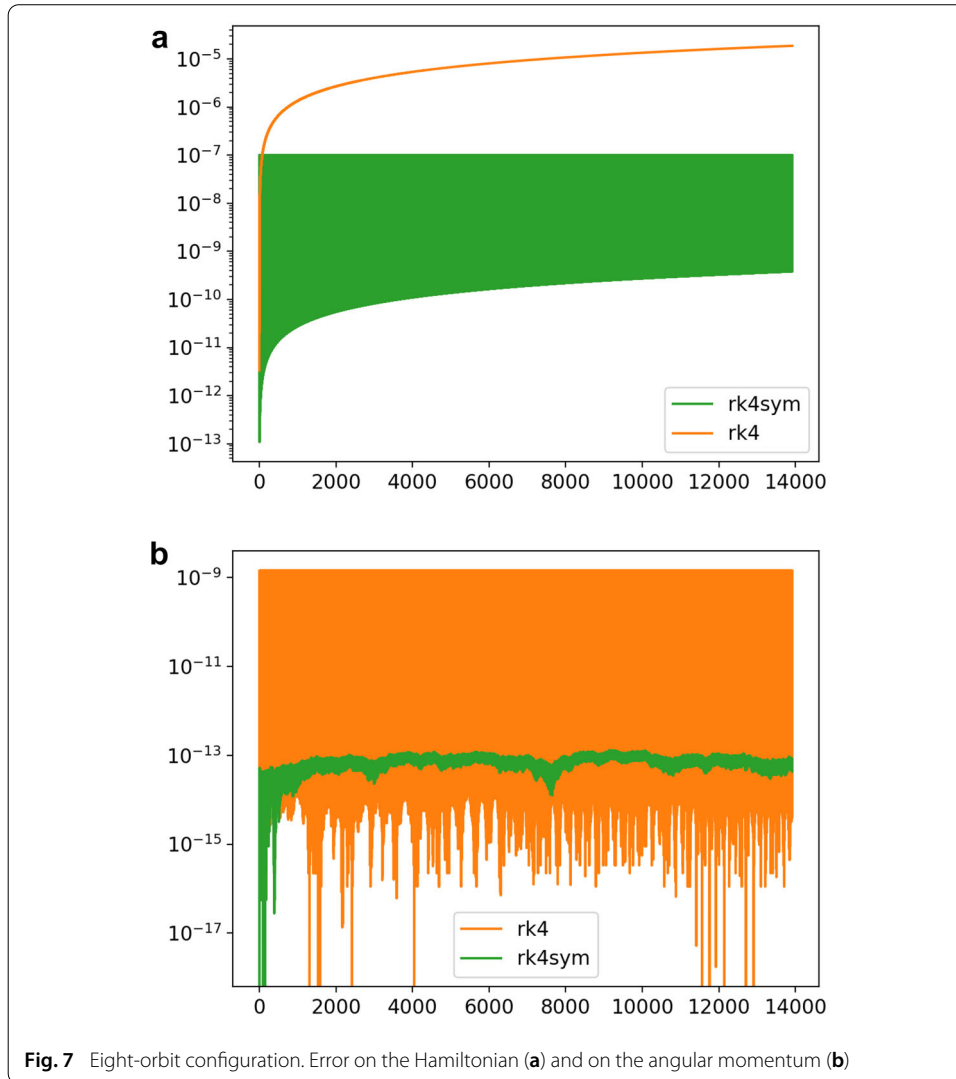


Fig. 7 Eight-orbit configuration. Error on the Hamiltonian (a) and on the angular momentum (b)

$$\mathbb{D}_\Delta = \{(w, \beta) \in \mathcal{V} \times \mathcal{V}^* \mid w \in \tilde{\Delta}, \beta - \Omega^b w \in \Delta_0\}.$$

To define the dynamics of the system, we introduce two more objects. First, the Lagrangian, which as usual is a mapping $L: \mathbb{R}^d \times V \rightarrow \mathbb{R}$, induces its differential which is a mapping dL from $\mathbb{R}^d \times V$ to its cotangent. And again by post-composing it with a symplectomorphism of the appropriate double bundles, one constructs the *Dirac differential* which locally reads

$$\begin{aligned} \mathbb{R}^d \times V &\longrightarrow \mathcal{V}^* \\ DL: & \\ (\mathbf{q}, \mathbf{v}) &\longmapsto \left(\mathbf{q}, \frac{\partial L}{\partial \mathbf{v}}, -\frac{\partial L}{\partial \mathbf{q}}, \mathbf{v} \right) \end{aligned}$$

Second, the evolution of the system will be described by a so-called *partial vector field* X , i.e. a mapping

$$X : \Delta \oplus \text{Leg}(\Delta) \subset (\mathbb{R}^d \times V) \oplus (\mathbb{R}^d \times V^*) \longrightarrow \mathcal{V},$$

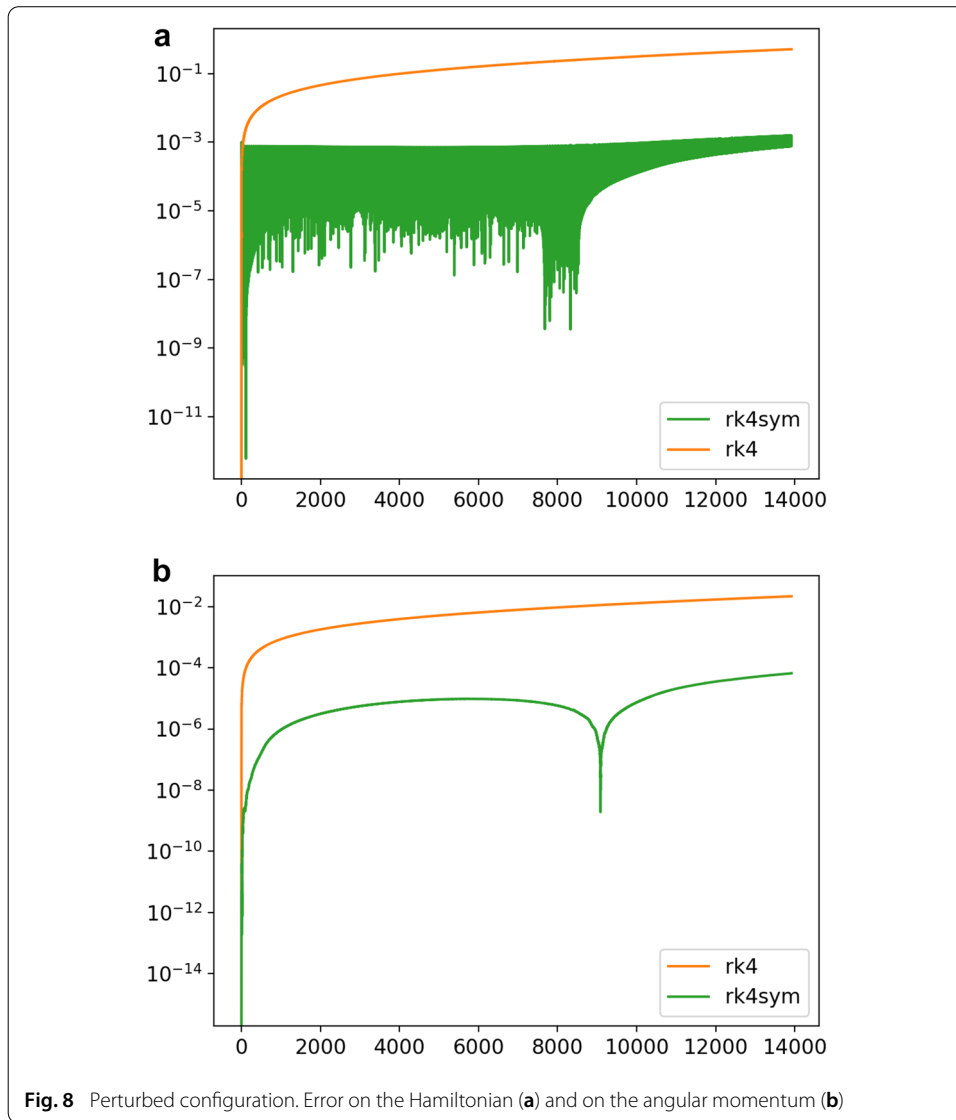


Fig. 8 Perturbed configuration. Error on the Hamiltonian (a) and on the angular momentum (b)

where $Leg(\Delta)$ is the image of Δ by the Legendre transform. X should be viewed as a vector field on $\mathbb{R}^d \times V^*$, with the momenta parametrized by the Legendre transform of the velocities compatible with the constraints.

With the above notations, the *implicit Lagrangian system* is a triple (L, Δ, X) , such that $(X, DL) \in \mathbb{D}_\Delta$. In local coordinates, this means:

$$\begin{cases} \frac{d\mathbf{q}}{dt} \in \Delta, & \mathbf{p} = \frac{\partial L}{\partial \mathbf{v}}, \\ \frac{d\mathbf{q}}{dt} = \mathbf{v}, & \frac{d\mathbf{p}}{dt} - \frac{\partial L}{\partial \mathbf{q}} \in \Delta_0. \end{cases} \quad (19)$$

One understands easily the mechanical interpretation of the first three equations above. The fourth one can be rewritten as

$$\frac{d\mathbf{p}}{dt} - \frac{\partial L}{\partial \mathbf{q}} = \sum_{a=1}^m \lambda_a \alpha^a,$$

and one recognizes immediately the Lagrange multipliers.

Discretization

It is important to note that the previous section is not just a “fancy” way of recovering the well-known theory: every step of the construction admits a discrete analog. We briefly present the recipe of this discretization and, again, refer the interested reader to [27] for details and examples.

The system is characterized by the following continuous data: the Lagrangian function $L(q, \dot{q})$ and the set of constraint 1-forms $\alpha^a, a = 1, \dots, m$. The discrete version L_d of the Lagrangian at time t^n is

$$L_d = \Delta t L(\mathbf{q}^n, \mathbf{v}^n).$$

And the constraints are rewritten as

$$\langle \alpha_{d^a}^a, \mathbf{v}^n \rangle = 0, \quad a = 1, \dots, m.$$

where $\langle \alpha_{d^a}^a, \mathbf{v}^n \rangle = \alpha_{d^a}^a(\mathbf{v}^n)$, and $\alpha_{d^a}^a$ is a discrete version of α^a . In the Equations above, \mathbf{q}^n is the value of \mathbf{q} and \mathbf{v}^n is an approximation of the velocity \mathbf{v} , both at time t^n .

To construct the numerical method out of these data, one applies the following procedure:

$$\mathbf{p}^{n+1} = \frac{1}{\Delta t} \frac{\partial L_d}{\partial \mathbf{v}^n} \tag{20}$$

$$\mathbf{p}^n - \frac{1}{\Delta t} \frac{\partial L_d}{\partial \mathbf{v}^n} + \frac{\partial L_d}{\partial \mathbf{q}^n} = \sum_{a=1}^m \lambda_a \frac{\partial \langle \alpha_{d^a}^a, \mathbf{v}^n \rangle}{\partial \mathbf{v}^n} \tag{21}$$

$$\langle \alpha_{d^a}^a, \mathbf{v}^n \rangle = 0, \quad a = 1, \dots, m. \tag{22}$$

The variables appearing explicitly are the values of \mathbf{p} at the n -th and the $(n + 1)$ -st steps, and \mathbf{v}^n should be some approximation of the velocity, thus bringing \mathbf{q}^n to the system. Here, we consider two natural options:

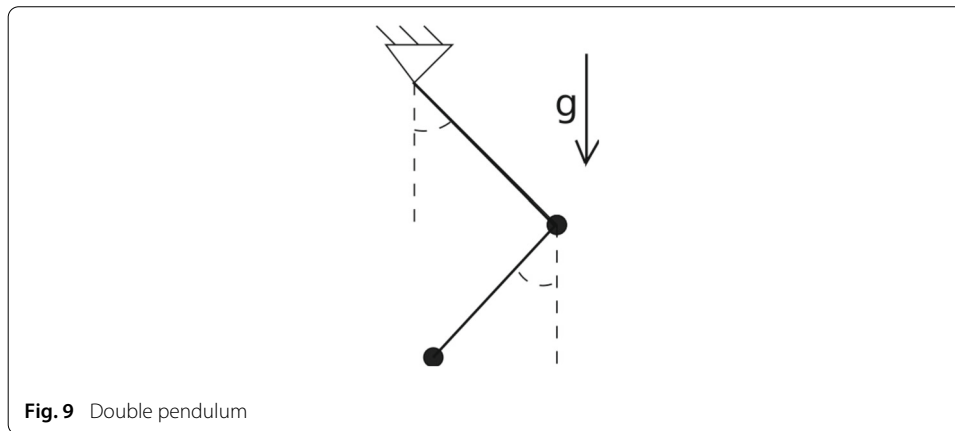
- $\mathbf{v}^n := \frac{\mathbf{q}^{n+1} - \mathbf{q}^n}{\Delta t}$, we label it Dirac-1, and
- $\mathbf{v}^n := \frac{\mathbf{q}^{n+1} - \mathbf{q}^{n-1}}{2\Delta t}$, labelled Dirac-2.

Dirac-1 permits to recover the method from [19] whereas we introduced Dirac-2 in [27]. In both cases, we obtain $2d + m$ equations: d from each of the lines (20) and (21) of the equations above, and m from the constraints (22). At the n -th step the unknowns are \mathbf{q}^{n+1} , \mathbf{p}^{n+1} and $\lambda = (\lambda_1, \dots, \lambda_m)$, so the system we obtain is complete. It is linear in λ and \mathbf{p}^{n+1} , and when the constraints are holonomic, in \mathbf{q}^{n+1} as well.

It is important to note that, in some sense, this construction generalizes the previous section. If one considers the system without constraints but still applies the procedure, (22) becomes obsolete, and in (21) the right-hand-sides vanish, so one obtains a numerical method for the dynamics of a Lagrangian system governed by L . By a straightforward computation, one checks that for a natural mechanical system with a potential U , i.e. when $L = \frac{1}{2}m\mathbf{v}^2 + U(\mathbf{q})$, Dirac-1 is symplectic. And it is also meaningful to consider a symplectic version of Dirac-2 (we do not detail it here since we would need to explain what is symplecticity for a multistep method).

Example: chaos for double pendulum

We will apply the Dirac integrators constructed in the previous subsection to a scholar problem of a planar double pendulum in a gravity field: a system of 2 mass points attached



to rigid inextensible weightless rods (see Fig. 9). Although, it looks like a classical well-studied problem, it is a bit challenging for simulations. In the absence of the gravity field, this is a textbook example of an integrable system (energy and angular momentum are conserved, thus the Liouville–Arnold theorem can be applied). With the gravity, the “folkloric knowledge” says that it is chaotic, although as far as we know there is no rigorous proof of the fact. For the integrability, there is a semi-numerical proof of absence of an additional first integral in [26], using the computation of the monodromy group by a method presented in [25]. Anyway, the apparent chaoticity of the system results in its sensitivity to parameters and initial data in the numerical simulation.

From the point of view of the previous subsection, this is a typical example of a system with constraints: the distance ℓ_1 from the first mass point to the origin and the distance ℓ_2 between the two mass points are fixed. The system admits a parametrization in terms of angles, but we will pretend not to know it, to test the method.

We thus consider a mechanical system of two mass points given by the Lagrangian

$$L(\mathbf{q}_1, \mathbf{q}_2, \dot{\mathbf{q}}_1, \dot{\mathbf{q}}_2) = \frac{1}{2}m_1\|\dot{\mathbf{q}}_1\|^2 + \frac{1}{2}m_2\|\dot{\mathbf{q}}_2\|^2 - m_1gq_{1,y} - m_2gq_{2,y},$$

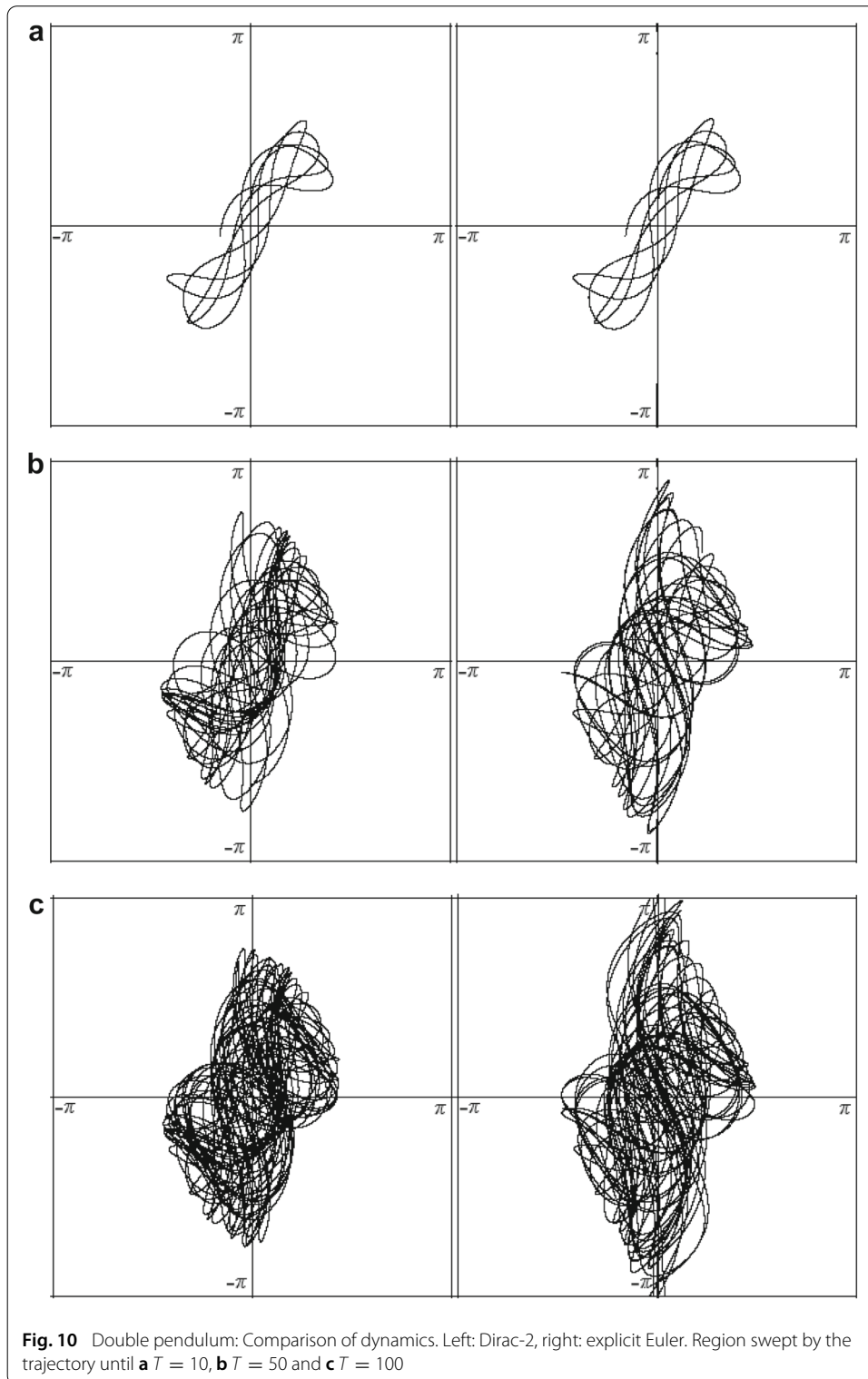
subject to the constraints

$$\varphi^1 \equiv \|\mathbf{q}_1\|^2 - \ell_1^2 = 0, \quad \varphi^2 \equiv \|\mathbf{q}_2 - \mathbf{q}_1\|^2 - \ell_2^2 = 0.$$

To recover the framework of the numerical method given by (20–22), we take $\alpha^a \equiv d\varphi^a$, $a = 1, 2$.

The typical result of simulations is shown in Fig. 10. Dirac-2 and explicit Euler methods are compared. For visualization (but not for computation), we use the angle representation of the double pendulum (see Fig. 9). Both algorithms start with the same initial data, and the same timestep $\Delta t = 0.0001$. They are in good agreement in the beginning as can be seen on the two top-graphics of Fig. 10. But already at time $T = 50$, the difference is visible (graphics in the middle). And towards $T = 100$ the difference becomes dramatic: for the Euler method the pendulum is making full turns instead of oscillation. And this is clearly a computation artifact, since decreasing the timestep one gets rid of the discrepancy and recovers the left picture for both methods. Note also that Dirac structure based method preserves the constraints much better than the Euler one: the error is $2.2 \cdot 10^{-6}$ compared to 0.06 respectively.

A similar effect is observed for other methods: trapezium, and even Runge–Kutta, which is of higher order. Moreover, there is another non-negligible convenience of the Dirac



structure based methods: the Lagrange multipliers are treated like other dynamical variables, there is no need to resolve “by hand” the equation related to constraints (22).

In many areas of mechanics, systems are often described by an underlying geometric structure. As observed in the two previous sections, making use of these structures leads

to more robust numerical schemes. In the last section, we propose an integrator which is suitable to general systems where no geometric structure is exploitable for numerical simulations.

Borel–Laplace integrator

Consider an ordinary differential or a semi-discretized partial differential equation:

$$\frac{du}{dt} = F(u(t), t) \tag{23}$$

associated to an initial condition $u(t = 0) = u_0$. We look for a time series solution:

$$\check{u}(t) = \sum_{n=0}^{+\infty} u_n t^n. \tag{24}$$

Generally, the right-hand side of (23) can be expanded into

$$F(u(t), t) = \sum_{n=0}^{+\infty} F_n(u_0, \dots, u_n) t^n \tag{25}$$

where F^n is the n -th Taylor expansion of the function $F(u(t), t)$ at $t = 0$. Injecting (24) and (25) into Eq. (23) leads to the following recurrence relation

$$(n + 1)u_{n+1} = F_n(u_0, \dots, u_n) \quad \text{for all } n \in \mathbb{N}. \tag{26}$$

This relation enables to compute the terms of the series \check{u} . See also sections and for concrete examples. A Borel summation is then applied to series (24). This summation is essential if the convergence radius of the series is zero and the series is summable [2,20]. If the convergence radius is not zero, the Borel summation enlarges the domain of validity of the series.

The Borel sum of $\check{u}(t)$ is

$$S\check{u}(t) = [\mathcal{L} \circ \mathcal{P} \circ \mathcal{B}]\check{u}(t) \tag{27}$$

where \mathcal{B} is the Borel transform, \mathcal{P} is a prolongation along a semi-line in \mathbb{C} linking 0 to infinity (we will take the real positive semi-line), and \mathcal{L} is the Laplace transform along this semi-line. The theory of Borel summation can be found, for example, in [2,20–22]. Some other works on BPL, as a time integrator, can be found in [8,23]. In this section, we present very briefly the Borel-Padé-Laplace algorithm, integrated into a numerical scheme.

The Borel-Padé-Laplace summation integrator (BPL) consists in the following steps:

- Given an initial condition $u(t_0) = u_0$, compute a truncated series solution via recurrence (26): $\check{u}^N(t) = \sum_{n=0}^N u_n t^n$.
- Compute its Borel transform: $\mathcal{B}\check{u}^N(\xi) = \sum_{n=0}^{N-1} \frac{u_{n+1}}{n!} \xi^n$.
- Transform $\mathcal{B}\check{u}^N(\xi)$ into a rational fraction function via a Padé approximation: $P^N(\xi) = \frac{a_0 + a_1 t + \dots + a_{N_{num}} t^{N_{num}}}{b_0 + b_1 t + \dots + b_{N_{den}} t^{N_{den}}}$
The Padé approximation materializes the prolongation in the Borel summation procedure.
- Apply a Laplace transformation (at $1/t$) on $P(\xi)$ to obtain a numerical Borel sum $S\check{u}^N(t) = \int_0^{+\infty} P^N(\xi) e^{-\xi/t} d\xi$.
Numerically, the integral is replaced by a Gauss–Laguerre quadrature.

- Take $S\check{u}^N(t)$ as an approximate solution $u(t)$ of (23) within the integral $[t_0, t_1]$ where the residue of the equation is smaller than a parameter ϵ_{res} .
- Restart the algorithm with $u_0 = u(t_1)$ as initial condition to obtain an approximate solution for larger values of t .

At each iteration, $t_1 - t_0$ is considered as the (adaptative) time-step of the scheme. The average time-step will be used for comparisons in numerical experiments. Note that at each time, the approximate solution has an analytical representation as a Laplace integral. A continued fraction representation can also be used [9].

An advantage of BPL is that it is totally explicit, in contrast with symplectic integrators in general. Moreover, changing the order of the scheme is as easy as setting N to a different value. Note also that the resummation procedure can be done componentwise, enabling an easy parallelization on multi-core computers. However, no such optimization has been done in the present article.

In the following subsection, a partial analysis of the symplecticity property of BPL is presented.

High-order symplecticity

The numerical flow of BPL can be defined as

$$u_0 \mapsto S\check{u}^N(t).$$

Currently, no symplecticity result has yet been found on this scheme. Instead, it can be shown that a scheme based on the truncated series \check{u} without the resummation procedure is symplectic at order N , if the equation is symplectic.

The flow of the scheme based on the time series \check{u} is

$$\varphi_{t,\check{u}} : u_0 \mapsto \check{u}(t) = \sum_{n=0}^{+\infty} t^n u_n.$$

Lemma 1 *The flow of the scheme based on the time series, applied to the Hamiltonian Eq. (2) is*

$$\varphi_{t,\check{u}} = \sum_{n=0}^{+\infty} \frac{t^n}{n!} \mathbb{J} D^n \nabla H \quad \text{where} \quad D^n = \frac{d^{n-1}}{dt^{n-1}} \tag{28}$$

agreeing that $D^0 = 1$.

This can be straightforwardly deduced by injecting the time series \check{u} in (2) and identifying the coefficients of each t^n . Next, if the series is convergent then, inside the convergence disc, \check{u} is the exact solution. In this case, φ_t is symplectic. We reformulate this statement in the following theorem.

Theorem 2 *If the series is convergent then*

$$(\nabla \varphi_{t,\check{u}})^T \mathbb{J} \varphi_{t,\check{u}} = \mathbb{J}. \tag{29}$$

Corollary 3 *If the series is convergent then, for any $n \geq 1$,*

$$\sum_{k=0}^n \frac{1}{k!} \frac{1}{(n-k)!} (\mathbb{J} D^k \nabla H)^T \mathbb{J} (\mathbb{J} D^{n-k} \nabla H) = 0. \tag{30}$$

This corollary is obtained by injecting the series development (28) into (29) and identifying the coefficients of t^n for $n \geq 1$. For $n = 0$, we simply have

$$(\mathbb{J}\nabla H)^\top \mathbb{J} (\mathbb{J}\nabla H) = \mathbb{J}. \tag{31}$$

When the series is truncated at order N , the flow of \check{u}^N is

$$\varphi_{t,\check{u}^N} : u_0 \mapsto \check{u}^N(t) = \sum_{n=0}^N t^n u_n.$$

The following theorem shows that the scheme based on the truncated series is symplectic at order $N + 1$.

Theorem 4 *If the series is convergent then*

$$(\nabla\varphi_{t,\check{u}^N})^\top \mathbb{J} \nabla\varphi_{t,\check{u}^N} = \mathbb{J} + O(t^{N+1}) \tag{32}$$

for $t \in [0, \delta t]$ where δt is the convergence radius.

Indeed,

$$\begin{aligned} (\nabla\varphi_{t,\check{u}^N})^\top \mathbb{J} \nabla\varphi_{t,\check{u}^N} &= \sum_{n=0}^N \sum_{k=0}^n \frac{t^n}{k!(n-k)!} (\mathbb{J}D^k \nabla H)^\top \mathbb{J} (\mathbb{J}D^{n-k} \nabla H) \\ &\quad + \sum_{n=N+1}^{2N} \sum_{k=n-N}^N \frac{t^n}{k!(n-k)!} (\mathbb{J}D^k \nabla H)^\top \mathbb{J} (\mathbb{J}D^{n-k} \nabla H). \end{aligned}$$

Using (30) and (31), the theorem follows. Note that in Theorem 4, δt is generally small.

In the following subsections, BPL is implemented and tested on a Hamiltonian equation. Next, we present some experiments on non-Hamiltonian equations.

In simulations, the truncation order of the series is set to $N = 10$ unless otherwise stated. The degree of the numerator and the denominator of the Padé approximant are $N_{num} = 4$ and $N_{den} = 5$. A singular value decomposition is used to strengthen the robustness of the Padé calculation [11]. Twenty Gauss–Laguerre roots are used for the quadrature.

The aim of these simulations is not to make an extensive comparison of BPL with classical schemes (this will be done in a forthcoming paper) but only to show the potential of the scheme in predicting long time dynamics.

Periodic Toda lattice

We consider again the periodic Toda lattice from “Periodic Toda lattice” section. The quality parameter ϵ_{res} of BPL is chosen such that the mean time step δt is approximately 0.1, and compare the results with that of RK4 and RK4sym (see Figs. 3 and 4).

Figure 11a presents the local relative errors on the Hamiltonian. As can be seen, BPL is much more accurate than RK4sym for $t \in [0, 5000]$. The value of the global error, defined as

$$E_H^{mean} = \frac{1}{t_f} \int_0^{t_f} \frac{|H(\mathbf{q}, \mathbf{p}) - H(\mathbf{q}^0, \mathbf{p}^0)|}{|H(\mathbf{q}^0, \mathbf{p}^0)|}, \quad t_f = 5000,$$

is $3.010 \cdot 10^{-4}$ with BPL and $2.261 \cdot 10^{-3}$ with RK4sym. BPL also preserves the eigenvalues of the matrix L in equation (18) with a fair precision, as seen in Fig. 11b.

Table 1 compares the CPU time needed for 5000 seconds of simulation. It shows that RK4 is the fastest but, as already mentioned, it is not accurate enough for $\Delta t = 0.1$. It

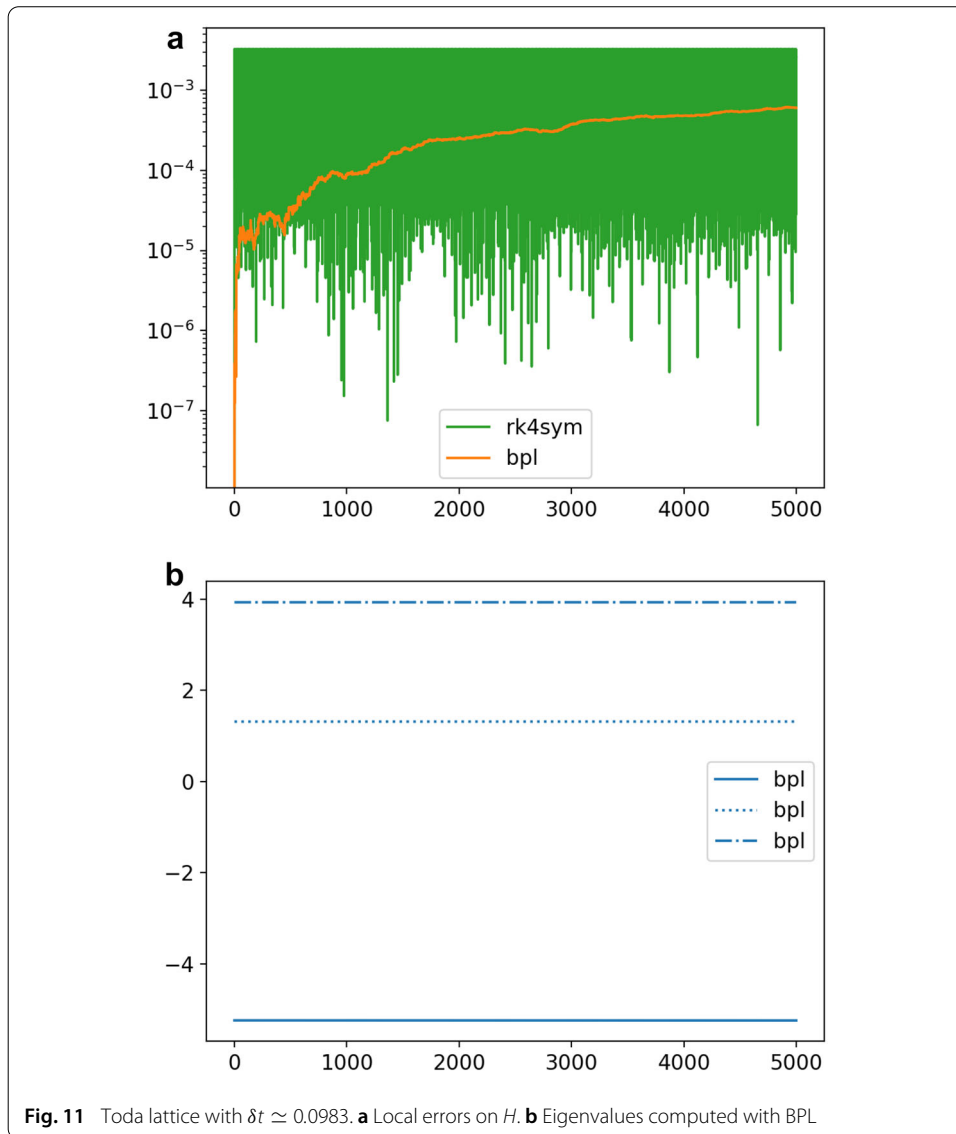


Table 1 Toda lattice

| | Mean time-step | CPU | Mean error |
|--------|----------------|--------|------------------------|
| RK4 | 0.1 | 107.44 | $3.7902 \cdot 10^{-1}$ |
| RK4sym | 0.1 | 128.74 | $2.261 \cdot 10^{-3}$ |
| BPL | 0.0983 | 259.31 | $3.010 \cdot 10^{-4}$ |

CPU and accuracy comparison, with almost the same (mean) time step

also shows that BPL, for approximately the same mean time step, is about twice as slow as RK4sym, but is 7.5 times more accurate.

In a second test, the different parameters (time step for RK4 and RK4sym and ϵ_{res} for BPL) are set such that the global accuracies are comparable. Table 2 shows the (mean) time steps and the CPU errors for E_H^{mean} around $2.43 \cdot 10^{-3}$. As can be seen, RK4sym needs 28 percent less time than BPL to achieve the same accuracy on H , due to its especially good property towards the preservation of the Hamiltonian. However, BPL is more than 8 times as fast as the classical RK4 scheme.

Table 2 Toda lattice

| | Mean time-step | CPU | Mean error |
|--------|----------------|---------|-----------------------|
| RK4 | 0.0275 | 1475.49 | $2.130 \cdot 10^{-3}$ |
| RK4sym | 0.1 | 128.74 | $2.261 \cdot 10^{-3}$ |
| BPL | 0.125 | 179.08 | $2.897 \cdot 10^{-3}$ |

Time step and CPU comparison, with almost the same mean error on H . Final time: 5000

Duffing equation

In the next numerical experiment, consider the forced Duffing equation

$$\ddot{u} + r\dot{u} + au + bu^3 = c \cos(\omega t) \tag{33}$$

which describes nonlinear damped oscillators [16, 30]. To illustrate the series decomposition, let us write this equation as a first order one:

$$\begin{cases} \dot{u} = v, \\ \dot{v} = c \cos(\omega t) - rv - au - bu^3. \end{cases} \tag{34}$$

Thanks to a Taylor development of Eq. (34), the terms of the time series \check{u} and \check{v} at $t = t_0$ can be computed as follows:

$$\begin{cases} (n + 1)u_{n+1} = v_n \\ (n + 1)v_{n+1} = \frac{c \omega^n}{n!} \cos\left(\omega t_0 + \frac{n\pi}{2}\right) \\ -rv_n - au_n - b \sum_{n_1, n_2, n_3} u_{n_1} u_{n_2} u_{n_3}. \end{cases} \tag{35}$$

In the last term, the sum is over $n_1, n_2, n_3 \in \{0, \dots, n\}$ such that $n_1 + n_2 + n_3 = n$. Note that writing Eq. (33) as a first order equation is not mandatory (and increases lightly computation time and memory requirement). One can directly apply a Taylor development on Eq. (33) to compute the terms of \check{u} . Note also that for more complicated right-hand-side, automatic differentiation avoids to calculate the Taylor development by hand [3].

We first consider the force-free case with two sets of coefficients for which there is a first integral:

- *Case 1:* $a = 2/9, b = 1, r = -1,$
- *Case 2:* $a = 1, b = 1, r = 0.$

In *Case 1*, the first integral is

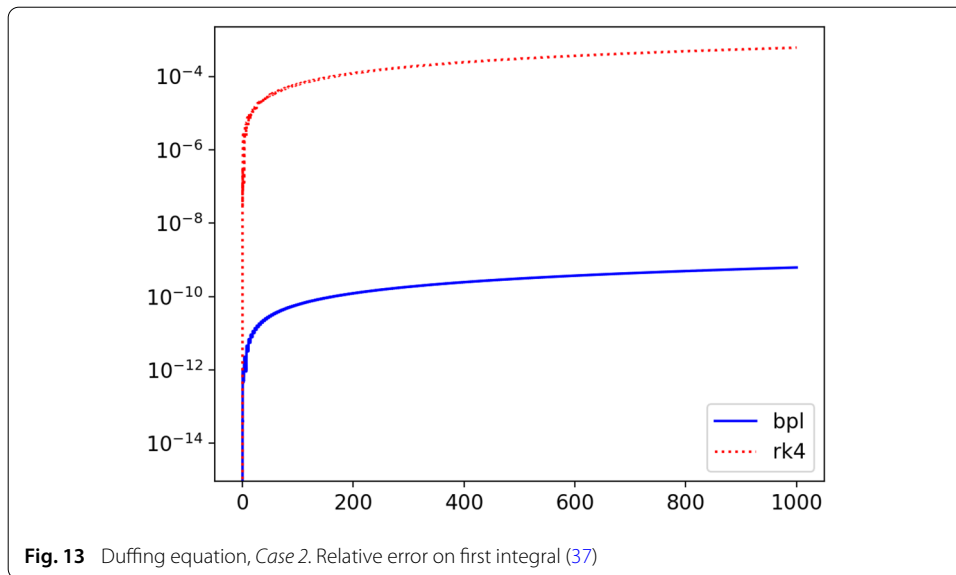
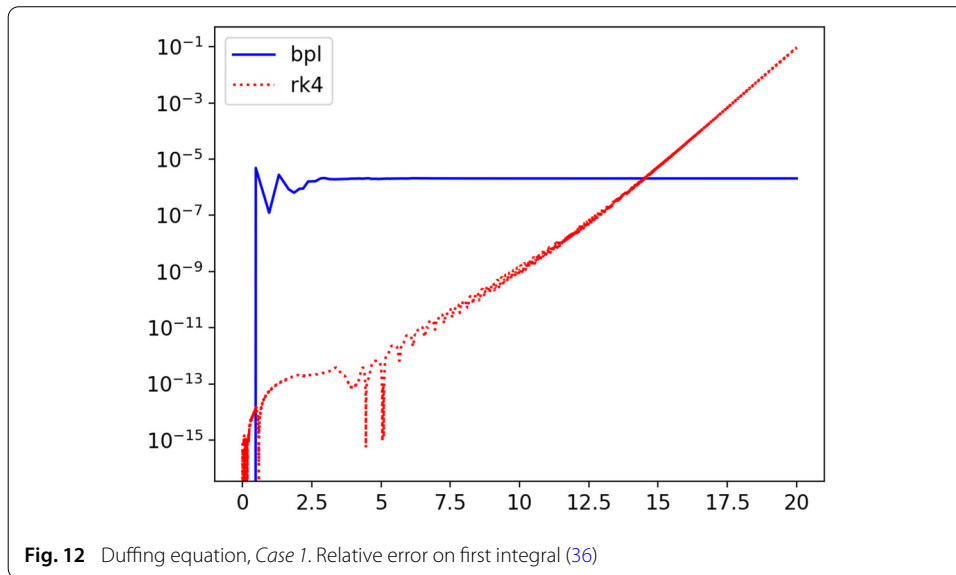
$$I = e^{-\frac{4t}{3}} \left(\dot{u}^2 - \frac{2}{3}u\dot{u} + \frac{1}{9}u^2 + \frac{1}{2}u^4 \right). \tag{36}$$

In *Case 2*, the equation can be written in a Hamiltonian form, with a Jacobi elliptic sine function as exact solution. The first integral (the Hamiltonian function) is

$$\frac{1}{2}\dot{u}^2 + \frac{1}{2}u^2 + \frac{1}{4}u^4. \tag{37}$$

The initial conditions are $u(0) = 1, \dot{u}(0) = 0.$

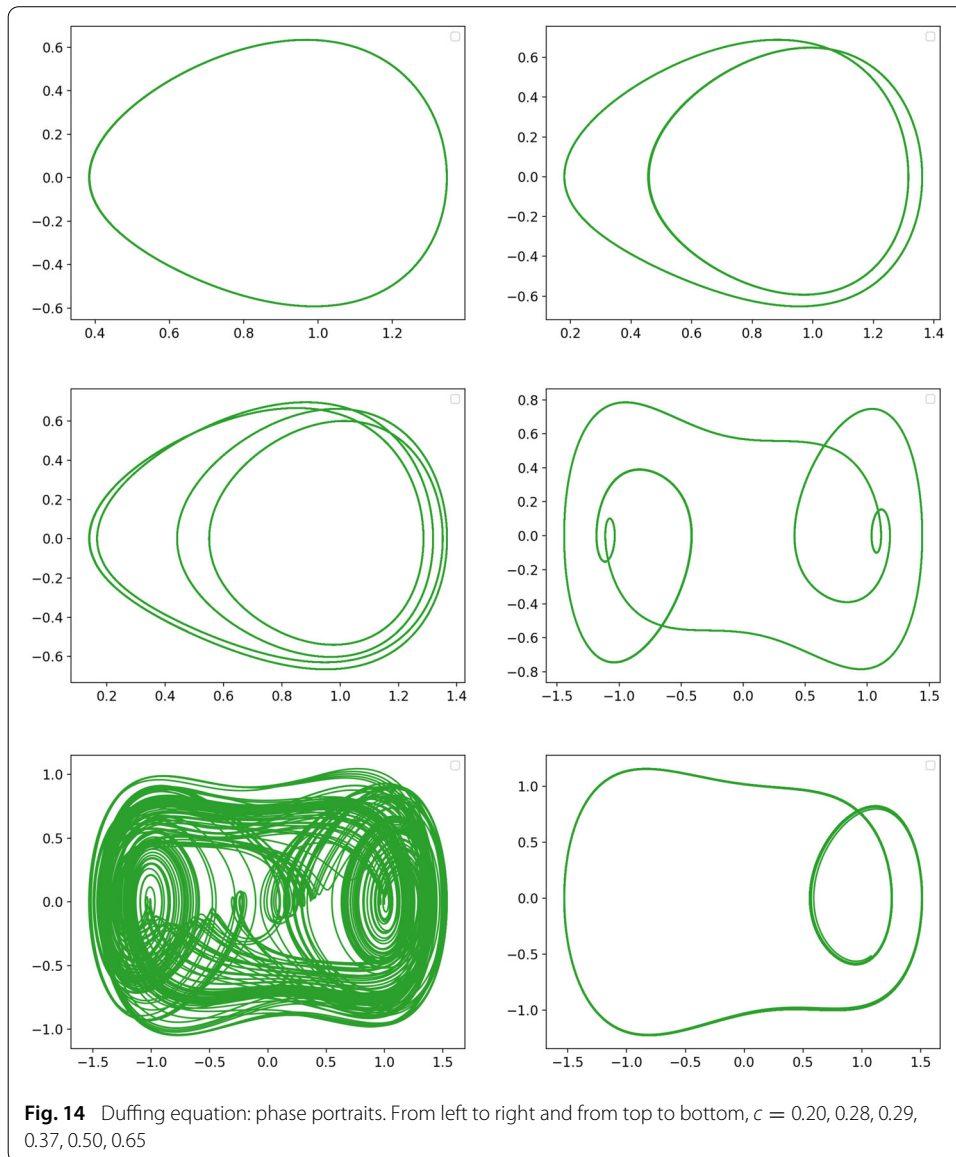
In *Case 1*, the quality criterium ϵ_{res} of BPL is set to 10^{-4} . The mean time step within the first 20 seconds is approximately $4.7 \cdot 10^{-4}$. Figure 12 plots the evolution of the relative error on the first integral, compared to the relative error of RK4 with a time step $4 \cdot 10^{-4}$.



As can be seen, the RK4 error is very small in the beginning of the simulation. It remains below 10^{-9} at $t = 10$. But when t is large, the RK4 error becomes big and reaches 50 percent at $t = 22.3$ s. With BPL, the error oscillates when t is small. The amplitude is of the same order as ϵ_{res} . Next, the error stabilizes rapidly around $1.27 \cdot 10^{-6}$.

In *Case 2*, the quality criterium ϵ_{res} of BPL is chosen such that the mean time step is around 0.106. For RK4, the time step is set to 0.1. The evolution of the relative error on the first integral (37) is plotted in Fig. 13 in logarithmic scale. As can be seen, the error of BPL is much smaller than that of RK4.

To end up with Duffing equation, some phase portraits obtained with BPL are computed. They corresponds to $a = -1$, $b = 1$, $r = .3$, $\omega = 1.2$ and c varying from 0.20 to 0.65. The initial conditions are $u(0) = 1$ and $\dot{u}(0) = 0$. Figure 14 presents the phase trajectories for $t \in [40, 1000]$, that is after the transient phase. These plots have been obtained with a rather loose value of ϵ_{res} for which the mean time step is around 0.5. But as can be seen,



when $c = 0.20, 0.28, 0.29, 0.37,$ and $0.65,$ the (multiple)-periodicity is very well captured even over a very long time interval. Indeed, the curves are closed. For $c = 0.50,$ the solution is chaotic but is bounded. These results are in agreement with those presented in [16].

In the last subsection, BPL is applied to a semi-discretized partial differential equation. It is compared to some other adaptative schemes. Since the system is big enough, it is worth to give an indication on the CPU simulation time.

Kortevog-de-Vries equation

Consider the Korteweg-de-Vries equation

$$\frac{\partial u}{\partial t} + c_0 \frac{\partial u}{\partial x} + \beta \frac{\partial^3 u}{\partial x^3} + \frac{\alpha}{2} \frac{\partial u^2}{\partial x} = 0 \tag{38}$$

which models waves on shallow water surfaces [17]. In this equation, the linear propagation velocity c_0 , the non-linear coefficient α and the dispersion coefficient β are positive constants, linked to the gravity acceleration g and the mean depth δ of the water by $c_0 = \sqrt{g\delta}$, $\alpha = \frac{3}{2}\sqrt{g/\delta}$ and $\beta = d^2c_0/6$.

The solution is assumed to be periodic with period X in space. Equation (38) is then discretized in space with a spectral method. The solution is approximated by its truncated Fourier series:

$$u(x, t) \simeq \sum_{|m| \leq M} \hat{u}^m(t) e^{im\omega x}, \tag{39}$$

where $M \in \mathbb{N}$ and $\omega = \frac{2\pi}{X}$. The injection of Eqs. (39) into (38) leads to a $(2M + 1)$ -dimensional ordinary differential equation:

$$\frac{d\hat{u}}{dt} = (-c_0i\omega m + i\beta\omega^3 m^3) \hat{u} + \frac{1}{2}i\alpha m\omega \hat{u} * \hat{u} \tag{40}$$

where the array \hat{u} contains the unknowns \hat{u}^m and the symbol $*$ denotes convolution product. Convolution operations are performed in physical space with the help of the fast Fourier transform and the standard dealiasing 3/2 rule is applied.

With BPL, the Fourier coefficient array $\hat{u}(t)$ is decomposed into a time series

$$\hat{u}(t) = \sum_{n=0}^N \hat{u}_n t^n. \tag{41}$$

Injected into Eq. (40), decomposition (41) leads to the following recurrence relation permitting an explicit computation of the series coefficients:

$$\hat{u}_{n+1} = \frac{1}{n+1} \left[(-c_0i\omega m + i\beta\omega^3 m^3) \hat{u}_n + \frac{1}{2}i\alpha m\omega \sum_{k=0}^n \hat{u}_k * \hat{u}_{n-k} \right], \tag{42}$$

for $n = 0, \dots, N - 1$. We take as initial condition the periodic prolongation of the function

$$u_0(x) = h \operatorname{sech}^2(\kappa x), \quad x \in \left[-\frac{X}{2}, \frac{X}{2} \right] \tag{43}$$

with $\kappa = \sqrt{3h/4\delta^3}$. The exact solution is

$$u(x, t) = u_0(x - ct) \tag{44}$$

where $c = c_0(1 + h/2\delta)$. We take $X = 24\pi$, $\delta = 2$, $g = 10$ and $h = \frac{1}{2}$. The period is $T \simeq 14.98s$. To begin with, the size of the system is set to $d = 128$ (that is the number of spectral discretization points is 129).

BPL is compared to two other schemes. The first one is the adaptative 4-th order Runge–Kutta scheme (still denoted RK4 in this subsection). This scheme is explicit. The second one is the exponential time differencing associated to RK4 (denoted ETDRK4), developed by Cox and Matthews in [7]. This scheme is based on an exact, exponential type, resolution of the linear part of the equation, followed by an explicit adaptative Runge–Kutta resolution of the non-linear part. The algorithm is not completely explicit since it requires the (pseudo-)inversion of a matrix. Moreover, it generally needs the evaluation of a matrix exponential, which is numerically expensive. This evaluation is done via a Padé approximants in simulations.

The precision criteria are calibrated such that the *a posteriori* errors of the three schemes have approximately the same magnitude, as can be seen in Fig. 15. This figure shows the difference between the predicted solutions with (44).

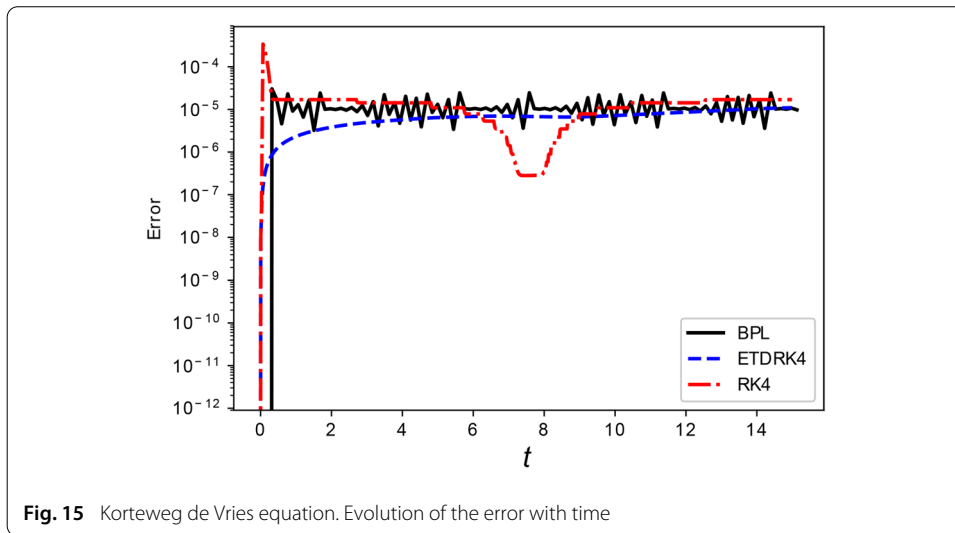


Fig. 15 Korteweg de Vries equation. Evolution of the error with time

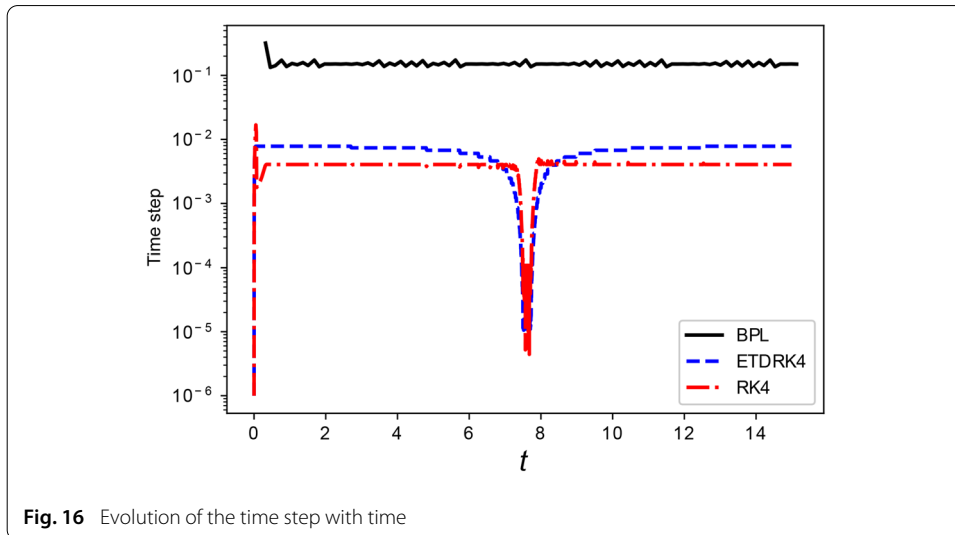


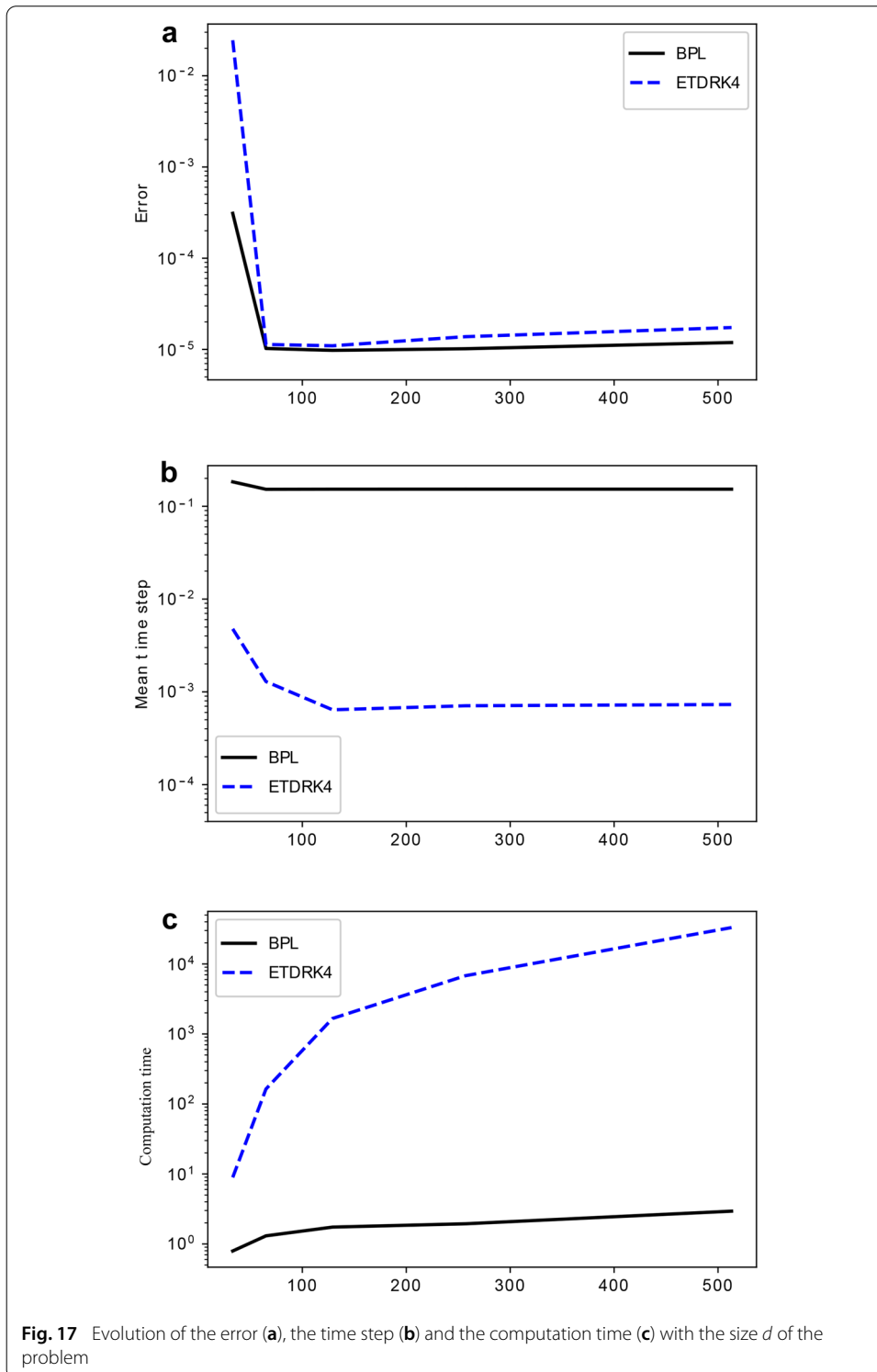
Fig. 16 Evolution of the time step with time

Table 3 Time step, error and CPU time over one period, with $d = 128$

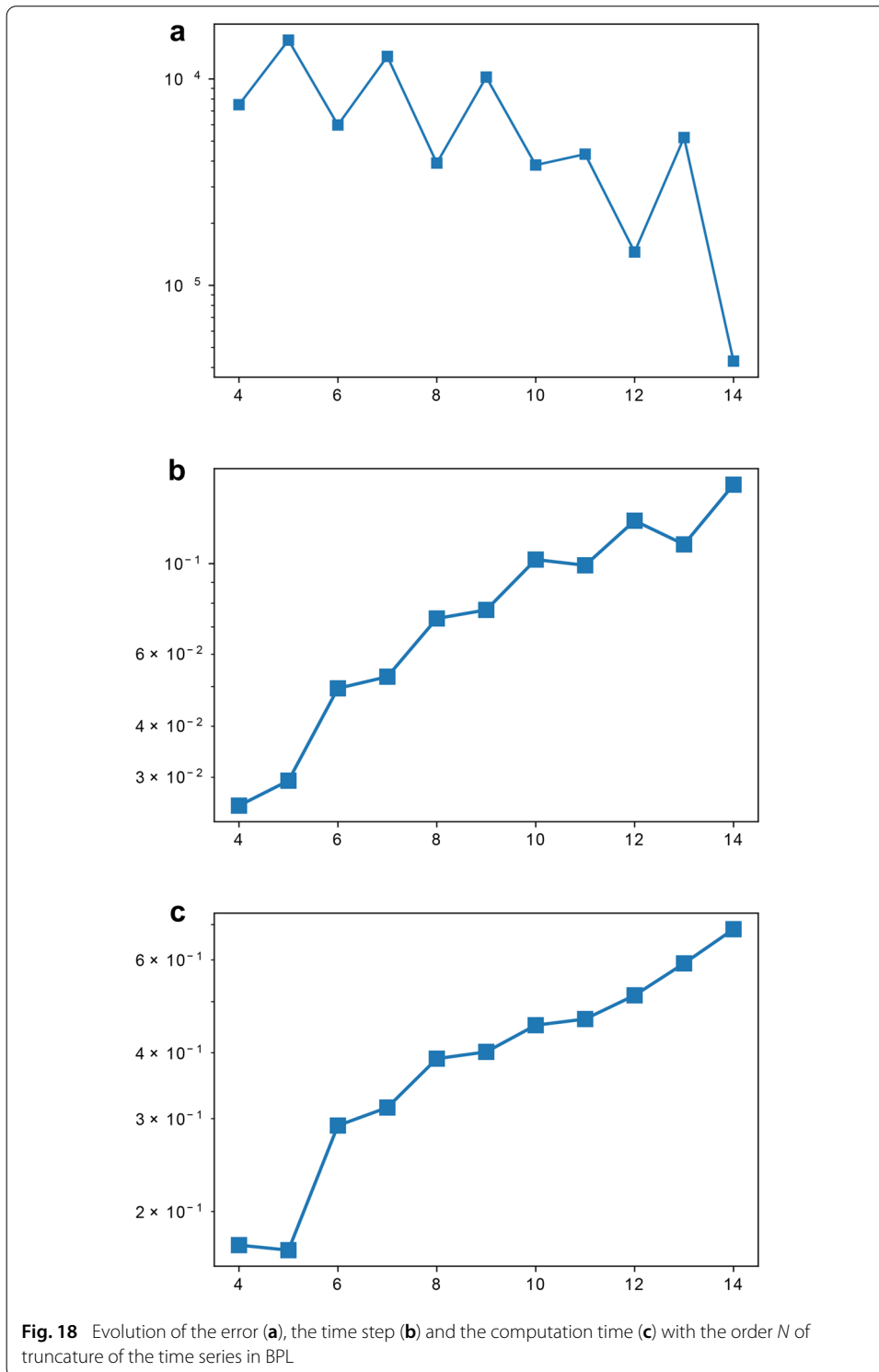
| | BPL | ETDRK4 | RK4 |
|------------------------|-----------------------|-----------------------|-----------------------|
| Mean time step | $1.53 \cdot 10^{-01}$ | $6.41 \cdot 10^{-04}$ | $1.88 \cdot 10^{-03}$ |
| L^2 error at $t = T$ | $9.76 \cdot 10^{-06}$ | $1.10 \cdot 10^{-05}$ | $1.69 \cdot 10^{-05}$ |
| Simulation time | 1.74 | $1.66 \cdot 10^{+03}$ | $5.98 \cdot 10^{+01}$ |

The time steps are presented in Fig. 16. In mean, the BPL time step is 238 times bigger than that of ETDRK4. This reflects on the CPU time. Indeed, as can be observed on Table 3, BPL is about 950 times faster than ETDRK4 for approximately the same precision. Note that, for this specific problem, the time step with RK4 has the same order as that of ETDRK4, but RK4 is more efficient than ETDRK4 in terms of computation time, since it requires neither numerical matrix (pseudo-)inversion nor exponential. Compared to BPL, RK4 takes 60 times more CPU time to reach one period.

In the next simulation, we analyse the behaviour of the schemes when the size d of the problem is increased. Figure 17a presents the L^2 error for $t = T$. It shows that the precision of BPL and ETDRK4 remains approximately the same, except when d is very



small. Figure 17b shows however that BPL requires much less iterations (100 iterations versus 20389 for ETDRK4 when $d = 512$ to reach one period). The time steps of BPL and ETDRK4 seem to have the same behaviour when d is large enough. Indeed, they tend to be independent of d , as suggested by Fig. 17b. However, the computation time increases



much more rapidly with ETDRK4. For BPL, the growth rate of the CPU time between $d = 128$ and $d = 512$ is 51 percent whereas, for ETDRK4, it is 381 percent.

In all of the previous simulations, the truncation order N of the time series in BPL was set to 10. In our last test, the effect of N on the quality of BPL is analysed. For this, the size of the problem is fixed to $d = 128$. Figure 18a shows that the time step increases with

N , passing from $\Delta t_{mean} = 0.0256$ for $N = 4$ to $\Delta t_{mean} = 0.156$ when $N = 14$. Despite the number of iterations is consequently reduced, the CPU time also increases with N , going from 0.686 to 0.173 s, as can be observed in Fig. 18b. This is caused by the fact that more coefficients of the series and more Padé coefficients have to be computed. As for it, the error fluctuates but globally decreases from $7.51 \cdot 10^{-5}$ to $4.30 \cdot 10^{-6}$. This fluctuation is not uncommon in series based approximations. It is interesting to note that whereas the error is divided by 17.5, the CPU time is multiplied only by 3.96 between $N = 4$ and $N = 14$. In other words, the precision increases faster than the CPU time when the order of the scheme is increased.

Conclusion

In this article, we gave an overview of some time integrators for long-time simulations. Two geometric integrators and a general-purpose time integrator was presented.

Through numerical examples, the ability of symplectic integrator in preserving the Hamiltonian, the angular momentum or eigenvalues was observed. Moreover, it was shown that symplectic integrators are more robust compared to classical schemes when the time step is enlarged (in the example of Toda lattice) or when a perturbation is introduced (three-body problem).

Next, a way of constructing Dirac integrators for constrained system was given. Numerical experiments showed that respecting the Dirac structure at discrete level avoids numerical artifacts. As a consequence, Dirac integrators are able to reproduce the dynamics of the system over a long time.

Finally, we showed that BPL competes with symplectic integrators in predicting Hamiltonian dynamics (Toda lattice and *Case 2* of Duffing equation). For more general equations, BPL also preserves with high precision the first integral of the system, as well as the periodicity when the solution is periodic. Lastly, compared to two popular schemes, BPL appears to require less computation time.

Authors' contributions

All the authors contributed and participated to the elaboration of the article. All authors read and approved the final manuscript.

Author details

¹Laboratoire des Sciences de l'Ingénieur pour l'Environnement (LaSIE), UMR 7356 CNRS, Université de La Rochelle, Avenue Michel Crépeau, 17042 La Rochelle Cedex, France, ²Centre National de la Recherche Scientifique (CNRS), LaSIE, Université de La Rochelle, Avenue Michel Crépeau, 17042 La Rochelle Cedex, France.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Please contact author for data requests.

Funding

Not applicable.

Publisher's Note

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

References

1. Benettin G, Giorgilli A. On the hamiltonian interpolation of near-to-the identity symplectic mappings with application to symplectic integration algorithms. *J Statist Phys.* 1994;74(5):1117–43.
2. Borel E. Mémoire sur les séries divergentes. *Annales scientifiques de l'E.N.S.* 3eme. 1899;16:9–131.
3. Bücker M, Corliss G. Automatic differentiation: applications, theory, and implementations, vol. 50., Lecture notes in computational science and engineering Berlin: Springer; 2006.
4. Chenciner A, Montgomery R. A remarkable periodic solution of the three-body problem in the case of equal masses. *Ann Math.* 2000;152(3):881–901.
5. Cooper GJ. Stability of Runge–Kutta methods for trajectory problems. *IMA J Num Anal.* 1987;7(1):1–13.
6. Courant T. Dirac manifolds. *Trans Am Math Soc.* 1990;319(2):631–61.
7. Cox S, Matthews P. Exponential time differencing for stiff systems. *J Comput Phys.* 2002;176(2):430–55.
8. Deeb A, Hamdouni A, Liberge E, Razafindralandy D. Borel–Laplace summation method used as time integration scheme. *ESAIM Proc Surv.* 2014;45:318–27.
9. Deeb A, Hamdouni A, Razafindralandy D. Comparison between Borel–Padé summation and factorial series, as time integration methods. *Disc Contin Dynam Syst Serie S.* 2016;9(2):393–408.
10. Feng K, Qin M. Symplectic geometric algorithms for Hamiltonian systems. Berlin: Springer; 2010.
11. Gonnet P, Güttel S, Trefethen L. Robust Padé approximation via SVD. *SIAM Rev.* 2013;51(1):101–17.
12. Hairer E, Lubich C. The life-span of backward error analysis for numerical integrators. *Num Math.* 1997;76(4):441–62.
13. Hairer E, Lubich C, Wanner G. Geometric numerical integration illustrated by the Störmer–Verlet method. *Acta Num.* 2003;12:399–450.
14. Hairer E, Nørsett S, Wanner G. Solving ordinary differential equations I: nonstiff problems. 2nd ed., Springer series in computational mathematics Berlin: Springer; 1993.
15. Hairer W, Wanner G, Lubich C. Geometric numerical integration. Structure-preserving algorithms for ordinary differential equations. 2nd ed., Springer series in computational mathematics Berlin: Springer; 2006.
16. Jordan D, Smith P. Nonlinear ordinary differential equations: an introduction for scientists and engineers. 4th ed., Oxford texts in applied and engineering mathematics Oxford: Oxford University Press; 2007.
17. Korteweg D, de Vries G. On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves. *Philosop Magaz.* 1895;39(240):422–43.
18. Lasagni FM. Canonical Runge–Kutta methods. *Zeitschrift für Angewandte Mathematik Physik.* 1988;39(6):952–3.
19. Leok M, Ohsawa T. Discrete Dirac structures and implicit discrete Lagrangian and Hamiltonian systems. In: XVIII international fall workshop on geometry and physics, volume 1260 of AIP conference proceedings, pages 91–102. American Institut of Physics, Melville, 2010.
20. Ramis J-P. Séries divergentes et théories asymptotiques. In *Journées X-UPS 1991*, p. 7–67. 1991.
21. Ramis J-P. Les développements asymptotiques après Poincaré: continuité et... divergences. *Gazett Math.* 2012;134:17–36.
22. Ramis J-P. Poincaré et les développements asymptotiques (première partie). *Gazett Math.* 2012;133:34–72.
23. Razafindralandy D, Hamdouni A. Time integration algorithm based on divergent series resummation, for ordinary and partial differential equations. *J Comput Phys.* 2013;236:56–73.
24. Razafindralandy D, Hamdouni A, Chhay M. A review of some geometric integrators. *Adv Model Simul Eng Sci.* 2018;5(1):16.
25. Salnikov V. Effective algorithm of analysis of integrability via the Ziglin's method. *J Dynam Control Syst.* 2014;20(4):465–74.
26. Salnikov V. Integrability of the double pendulum—the Ramis' question. [arXiv:1303.4904](https://arxiv.org/abs/1303.4904), 2016
27. Salnikov V, Hamdouni A. From modelling of systems with constraints to generalized geometry and back to numerics. *ZAMM J Appl Math Mech.* 2019;1:1. <https://doi.org/10.1002/zamm.201800218>.
28. Sanz-Serna JM. Runge–kutta schemes for Hamiltonian systems. *BIT Num Math.* 1988;28(4):877–83.
29. Sanz-Serna JM. Symplectic integrators for Hamiltonian problems: an overview. *Acta Num.* 1992;1:243–86.
30. Thompson JMT, Stewart HB. Nonlinear dynamics and chaos. 2nd ed. New York: Wiley; 2002.
31. Tulczyjew WM. The legendre transformation. *Annal l'Inst Henri Poincaré.* 1977;27(1):101–14.
32. van der Schaft A. Port-Hamiltonian systems: an introductory survey. In: *International congress of mathematicians, Vol. 3.* European Mathematical Society, Zürich; 2006, p. 1339–65.
33. Yoshimura H, Marsden J. Dirac structures in Lagrangian mechanics. I. Implicit Lagrangian systems. *J Geom Phys.* 2006;57(1):133–56.
34. Yoshimura H, Marsden J. Dirac structures in Lagrangian mechanics. II. Variational structures. *J Geom Phys.* 2006;57(1):209–50.