

RESEARCH ARTICLE

Numerical simulations for the Toda lattices Hamiltonian system: Higher order symplectic illustrative perspective

Asif Mushtaq^{1*}, Amna Noreen¹, Muhammad Asif Farooq²

1 Seksjon for matematikk, FLU, Nord Universitet, N-8049 Bodø, Norway, **2** Department of Mathematics, School of Natural Sciences (SNS), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan

* asif.mushtaq@nord.no



Abstract

In this paper we apply some higher order symplectic numerical methods to analyze the dynamics of 3-site Toda lattices (reduced to relative coordinates). We present benchmark numerical simulations that has been generated from the HOMsPY (Higher Order Methods in Python) library. These results provide detailed information of the underlying Hamiltonian system. These numerical simulations reinforce the claim that the symplectic numerical methods are highly accurate qualitatively and quantitatively when applied not only to Hamiltonian of the Toda lattices, but also to other physical models. Excepting exactly integrable models, these symplectic numerical schemes are superior, efficient, energy preserving and suitable for a long time integrations, unlike standard non-symplectic numerical methods which lacks preservation of energy (and other constants of motion, when such exist).

OPEN ACCESS

Citation: Mushtaq A, Noreen A, Farooq MA (2019) Numerical simulations for the Toda lattices Hamiltonian system: Higher order symplectic illustrative perspective. PLoS ONE 14(4): e0215054. <https://doi.org/10.1371/journal.pone.0215054>

Editor: Joseph Páez Chávez, Escuela Superior Politecnica del Litoral, ECUADOR

Received: September 29, 2018

Accepted: March 26, 2019

Published: April 18, 2019

Copyright: © 2019 Mushtaq et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: Data are available in figshare via the following DOI: [10.6084/m9.figshare.7762775](https://doi.org/10.6084/m9.figshare.7762775).

Funding: This work was supported by FLU, Nord University, Bodø Norway.

Competing interests: The authors have declared that no competing interests exist.

1 Introduction

Hamiltonian equations of motion belong to a class of ordinary differential equations (ODEs) which in general are difficult or mostly impossible to solve analytically. Consider a separable Hamiltonian written in the form

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \mathbf{p}^T M \mathbf{p} + V(\mathbf{q}) = T(\mathbf{p}) + V(\mathbf{q}), \tag{1}$$

where $T(\mathbf{p})$ is the non-relativistic kinetic energy, $V(\mathbf{q})$ is the potential energy and M is the inverse mass matrix. The autonomous Hamiltonian equations of motion constitute a system of first order ordinary differential equations,

$$\dot{q}^a = \frac{\partial H}{\partial p_a}, \quad \dot{p}_a = -\frac{\partial H}{\partial q^a}, \quad a = 1, \dots, \mathcal{N} \tag{2}$$

where q^a and p_a are generalized coordinates of positions and momenta, respectively. The $\dot{}$ denotes differentiation with respect to time t , and $\mathbf{H} = H(\mathbf{q}, \mathbf{p})$. The initial conditions at $t = 0$

can be written as,

$$q^a(0) = q_0^a, \quad p_a(0) = p_{a0}.$$

We define Eq (2) in abbreviated form as,

$$z = \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix}, \quad \nabla \mathbf{H} = \begin{pmatrix} \partial \mathbf{H} / \partial q^a \\ \partial \mathbf{H} / \partial p_a \end{pmatrix}, \tag{3}$$

$$\mathbf{J} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{pmatrix}, \tag{4}$$

where \mathbf{J} is a skew-symmetric matrix. Further \mathbf{I} and $\mathbf{0}$ represent the $(\mathcal{N} \times \mathcal{N})$ unit and zero matrices, respectively. The compact conservative Hamiltonian system in differential form is,

$$\dot{z} = \mathbf{J}^{-1} \nabla \mathbf{H}(z). \tag{5}$$

The above equations of motion are equivalent to Newton’s second law of mechanics with conservative forces. The dynamics generated by these equations define, for each evolution time, a mapping between regions of phase space. A general feature of these mappings is that they preserve the volume of the regions being mapped (and some related properties, collectively referred to as the symplectic structure). The standard non-symplectic numerical integrators, that have been used to solve general initial value problems numerically, do not preserve this qualitative behaviour, or the constants of motion for the system. Examples of such numerical integrators are the classical Runge-Kutta methods of different order, as found in standard integration packages.

By contrast the geometrical numerical integrators have gained popularity in the scientific community, due to their geometry preserving properties, in order to find qualitatively better solutions to Hamiltonian problems. In physical systems energy preservation, symmetries, time-reversal invariance, symplecticity, angular momentum, phase-space volume and dissipation are some key and crucial components to understand geometric properties. Detailed discussions of symplectic integrators with geometric properties have been given in [1–4].

Since the symplectic solvers have been widely accepted to be superior than the conventional numerical methods for solving the Hamiltonian systems, Mushtaq *et. al.* [5] constructed a well behaved class of higher order symplectic integrators schemes based on the extensions of the Störmer-Verlet scheme for Hamiltonians like Eq (1). An overview of these extensions are presented in Section 3. In this paper, we apply these schemes to the Toda lattice models.

The new proposed (*KiMoKi*) schemes involve extensive calculations of higher order derivatives of the Hamiltonian; hence it becomes a nightmare to do correct implementations manually. A collection of Python program *HOMsPy* (Higher Order Methods in Python) has been developed and presented by Mushtaq and Olaussen [8] to overcome these cumbersome and error-prone calculations for higher accuracy. More details, with implementation of many applications, can be found tutorial on *HOMsPY* by Mushtaq [9].

The structure of the rest of this paper is as follows: In Section 2 we review the Toda lattice models. These are integrable, nonlinear systems that have a number of extra constants of motions beyond standard ones like energy and momentum. The form of these can be described in a very concise manner by Eqs (7) and (8). In Section 3 we review the construction of the *KiMoKi* class of symplectic numerical solvers. In Section 4 we present and discuss the numerical simulations of the Toda lattice models by use of these methods. In Section 5 we

conclude the main body of paper with brief remarks. Some technical details are delegated to appendices.

2 Toda lattices

The periodic Toda lattice with \mathcal{N} sites (or particles) can, in suitable dimensionless coordinates, be specified by the Hamiltonian [10],

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \sum_{a=1}^{\mathcal{N}} p_a^2 + \sum_{a=1}^{\mathcal{N}} (\exp(q_a - q_{a+1}) - 1), \tag{6}$$

where q_a and p_a are phase-space coordinates of positions and momenta respectively, and the index a is interpreted modulo \mathcal{N} (i.e., $q_{\mathcal{N}+1} \equiv q_1$). This mode belongs to a more general class of lattice models where the nearest-neighbour potential, $\exp(q) - 1$, is replaced by an arbitrary function $V(q)$. Another famous member of this class is the Fermi-Pasta-Ulam-Tsingou problem, with $V(q) = q^2/2 - \alpha q^3/3 + \beta q^4/4$. The original study by Fermi *et. al.* [11] only treated linear chains with fixed endpoints, and parameter choices for which $\alpha\beta = 0$.

Integrability is one of the most important properties of the Toda lattices. The model describes a set of equal mass particles moving on a ring with exponentially decreasing nearest neighbour interactions. The $2\mathcal{N}$ phase-space coordinates can be used to define a symmetric, periodic tridiagonal $\mathcal{N} \times \mathcal{N}$ (time dependent) matrix,

$$L = \begin{pmatrix} p_1 & v_1 & 0 & \cdots & v_{\mathcal{N}} \\ v_1 & p_2 & v_2 & \cdots & 0 \\ 0 & v_2 & p_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ v_{\mathcal{N}} & 0 & \cdots & v_{\mathcal{N}-1} & p_{\mathcal{N}} \end{pmatrix}, \tag{7}$$

where $v_a = -e^{(q_a - q_{a+1})/2}$. It was shown by Flaschka [12], using theory developed by Lax [13], that the eigenvalues λ_a of L remain unchanged if the evolution $q_a(t), p_a(t)$ is generated by the Hamiltonian of Eq (6). This means that all quantities

$$\mathcal{C}_n \equiv \sum_{a=1}^{\mathcal{N}} \lambda_a^n = \text{Tr } L^n \tag{8}$$

are constants of motions. The first two are familiar, general expressions,

$$\mathcal{C}_1 = \sum_{a=1}^{\mathcal{N}} p_a \equiv P \quad (\text{total momentum}), \tag{9a}$$

$$\frac{1}{2} \mathcal{C}_2 = \sum_{a=1}^{\mathcal{N}} \left[\frac{1}{2} p_a^2 + v_a^2 \right] = H + \mathcal{N} \quad (\text{total energy}). \tag{9b}$$

The third one is a special consequence of integrability,

$$C_3 = \sum_{a=1}^{\mathcal{N}} [p_a^3 + 3p_a(v_{a-1}^2 + v_a^2)] + 6\delta_{\mathcal{N},3} v_1 v_2 v_3, \tag{9c}$$

where the last term is just an uninteresting constant, since $v_1 v_2 v_3 = 1$ when $\mathcal{N} = 3$.

It may not be easy to discover a general prescription like the one above. Alternative methods to find additional conservation laws (when one suspects that such exists) are the more *brute force* type of searches used by Göktaş *et al.* [14] and Hohler *et al.* [15]. They started by deducing the general form of a possible conservation law, with unknown coefficients, and next tried to explicitly solve for the coefficients with the help of computer algebra. This is a more pedestrian and cumbersome approach, but may be more likely to succeed when applied to models with unknown properties.

2.1 The 3-particle case

For $\mathcal{N} = 3$ it is simple to introduce center-of-mass and relative coordinates. A common physicists choice is Jacobi coordinates with corresponding conjugate momenta (see Appendix A),

$$\begin{aligned} X &= \frac{1}{3}(q_1 + q_2 + q_3), & x_1 &= q_1 - q_2, & x_2 &= \frac{1}{2}(q_1 + q_2) - q_3, \\ P &= p_1 + p_2 + p_3, & \pi_1 &= \frac{1}{2}(p_1 - p_2), & \pi_2 &= \frac{1}{3}(p_1 + p_2 - 2p_3). \end{aligned}$$

This separates the Hamiltonian into center-of-mass and internal contributions, $H = \frac{1}{6}P^2 + H_{\perp}$, with

$$H_{\perp}(\mathbf{x}, \boldsymbol{\pi}) = \pi_1^2 + \frac{3}{4}\pi_2^2 + v_1^2 + v_2^2 + v_3^2 - 3. \tag{10a}$$

In a similar manner we may rewrite $C_3 = \frac{1}{9}P^3 + 2PH_{\perp} + 3C_{3\perp} + 6$, with

$$C_{3\perp} = \pi_1^2 \pi_2 - \frac{1}{4}\pi_2^3 - \pi_1(v_2^2 - v_3^2) + \pi_2[v_1^2 - \frac{1}{2}(v_2^2 + v_3^2)]. \tag{10b}$$

A direct evaluation of $dC_{3\perp}/dt$, using the Hamilton equations generated by H_{\perp} , confirms that it vanishes. I.e., that $C_{3\perp}$ indeed is a constant of motion.

The Hamiltonian of Eq (10a) can be rewritten by a canonical scale transformation,

$$(\pi_1, \pi_2, x_1, x_2) = \left(-\frac{1}{4}p_2, \frac{1}{2\sqrt{3}}p_1, -4q_2, 2\sqrt{3}q_1\right),$$

followed by a change of time and mass units (see Appendix B), $t \rightarrow \sqrt{3}t, m \rightarrow \frac{1}{8}m$. This transforms Eq (10a) to the expression used by Lunsford and Ford [16],

$$H_{\perp}(\mathbf{q}, \mathbf{p}) = \frac{1}{2}(p_1^2 + p_2^2) + \frac{1}{24}[\exp(2q_2 + 2\sqrt{3}q_1) + \exp(2q_2 - 2\sqrt{3}q_1) + \exp(-4q_2) - 3]. \tag{11}$$

With the same transformations the conserved quantity of Eq (10b) can be expressed as

$$\begin{aligned} \tilde{C}_{3\perp} &\equiv -96\sqrt{3}C_{3\perp} \\ &= 8(p_1^2 - 3p_2^2)p_1 + (v_3^2 + v_2^2 - 2v_1^2)p_1 + \sqrt{3}(v_3^2 - v_2^2)p_2. \end{aligned} \tag{12}$$

If the potential in Eq (11) is expanded to third order in the coordinates q_1, q_2 , one obtains the Henon-Heiles [17] Hamiltonian

$$H_{H-H} = \frac{1}{2}(p_1^2 + p_2^2) + \frac{1}{2}(q_1^2 + q_2^2) + q_1^2 q_2 - \frac{1}{3}q_2^3.$$

This is the motivation for the form by Lunsford and Ford [16]. The *KiMoKi* solvers have

already been implemented for the Henon-Heiles model by Mushtaq [9], and used to detect chaotic and non-chaotic regions of that model. The former occur for oscillations of larger amplitude, for which the higher order terms in the Toda lattice Hamiltonian become of importance. This explains why chaotic behaviour may occur in the Henon-Heiles model, but not in the Toda lattice models (since they are integrable).

3 An overview to construct the higher order symplectic scheme

One idea for construction of a symplectic integrator for the evolution generated by a Hamiltonian,

$$H = T(\mathbf{p}) + V(\mathbf{q}),$$

is to replace it with an iterated sequence of short-time evolutions generated by respectively $T(\mathbf{p})$ (moves, which changes \mathbf{q} without changing \mathbf{p}) and $V(\mathbf{q})$ (kicks, which changes \mathbf{p} without changing \mathbf{q}), since each of these are exactly integrable. This is the Störmer-Verlet scheme, which in its symmetrized form has a global error scaling like the timestep squared, τ^2 . One way to achieve higher accuracy is by replacing T and V by *effective* quantities, $T \rightarrow T_{\text{eff}}$ and $V \rightarrow V_{\text{eff}}$, in a systematic manner. The effective quantities will depend on the timestep τ , and the wanted order N of accuracy, τ^N . In the *kick-push-move-kick* scheme proposed by Mushtaq *et al.* [5], V_{eff} is still a function of \mathbf{q} only (in addition to τ); hence it can still be treated a potential, only slightly changed. Then this is no longer possible for T_{eff} ; it must depend on both \mathbf{p} and \mathbf{q} . However, what is really needed is not the infinitesimal generator $T_{\text{eff}}(\mathbf{p}, \mathbf{q}; \tau)$, but its corresponding, sufficiently accurate, finite (but short) time generator $G(\mathbf{q}, \mathbf{P}; \tau)$. The latter can be constructed in a systematic manner:

$$G(\mathbf{q}, \mathbf{P}) = \sum_{k=0}^N G_k(\mathbf{q}, \mathbf{P}) \tau^k, \tag{13}$$

such that the transformation $(\mathbf{q}, \mathbf{p}) \rightarrow (\mathbf{Q}, \mathbf{P})$ is defined by

$$p_a = \frac{\partial G}{\partial q^a}, \tag{14a}$$

$$Q^a = \frac{\partial G}{\partial P_a}, \tag{14b}$$

which preserves the symplectic structure exactly, reproduces the time evolution generated by T_{eff} to order τ^N . Here Q^a is shorten for $q^a(t + \tau)$, and P_a shorten for $p_a(t + \tau)$. The change in momentum \mathbf{p} (of order τ^3 —i.e. a gentle *push*) is then defined through an implicit equation (but one which has turned out to be unproblematic to solve by iteration for all cases tried), while the change in position \mathbf{q} continues to be explicit. Hence, the evolution step generated by G consists of a *move*, accompanied with a gentle *push*.

Define K such that $2K + 2$ is the order of the method, where $K = 1, 2, 3$ and $K = 0$ corresponds to the Störmer-Verlet scheme. One full time step with this modification for *kick-push-move-kick* scheme is,

1. “Kick”:

$$q^a \rightarrow q^a, \tag{15a}$$

$$p_a \rightarrow p_a - \frac{\tau}{2} \frac{\partial}{\partial q^a} \sum_{k=0}^K V_{2k}(\mathbf{q}), \tag{15b}$$

2. “Push”:

$$q^a \rightarrow q^a, \tag{15c}$$

$$P_a = \frac{\partial}{\partial q^a} \sum_{k=0}^{2K+2} \tau^k G_k(\mathbf{q}, \mathbf{P}). \tag{15d}$$

But P_a is unknown yet. We have to solve a nonlinear equation to find P_a . However our generating function takes the form,

$$G(\mathbf{q}, \mathbf{P}) = q^a P_a + \frac{1}{2} P_a^2 \tau + \sum_{k=3}^{2K+2} \tau^k G_k(\mathbf{q}, \mathbf{P}). \tag{15e}$$

Hence Eq (14a) can be written as,

$$p_a = P_a + \frac{\partial}{\partial q^a} \sum_{k=3}^{2K+2} \tau^k G_k(\mathbf{q}, \mathbf{P})$$

or in a form suitable for an iterative solution,

$$P_a = p_a - \frac{\partial}{\partial q^a} \sum_{k=3}^{2K+2} \tau^k G_k(\mathbf{q}, \mathbf{P})$$

3. “Move”:

$$q^a \rightarrow Q^a = \frac{\partial}{\partial P_a} \sum_{k=0}^{2K+2} \tau^k G_k(\mathbf{q}, \mathbf{P}), \tag{15f}$$

$$P_a \rightarrow P_a. \tag{15g}$$

4. “Kick”:

$$q^a \rightarrow q^a, \tag{15h}$$

$$p_a \rightarrow p_a - \frac{\tau}{2} \frac{\partial}{\partial q^a} \sum_{k=0}^K V_{2k}(\mathbf{q}). \tag{15i}$$

The explicit expressions for V_{2k} and G_k were published by Mushtaq *et. al.* in ref [5]. For convenience, on request from a reviewer, they are included in Appendix D.

To implement these higher order methods for our Toda lattice Hamiltonian, we define the model by the code in Listing 1 of Appendix C, and use programs in the HOMsPy package to

automatically generate all the KiMoKi solver code. The complete code package is available as “Supporting information” for this paper.

4 Numerical simulations with HOMsPy

As has been mentioned before, numerical simulations on several Hamiltonian systems with the algorithm outlined by Eq (15) has compared favourable to conventional non-symplectic methods. We here present the results of additional simulations, of the model defined by Eq (11), which strengthen this evidence further.

As mentioned earlier that we implemented these numerical schemes as Python routines in HOMsPy. Python is an open source programming language which has gained increasing popularity in general, including (successful) applications for scientific computing. It is fast and easy to code and use for small “prototyping” tasks, since there is no need for explicit declaration of variables or a separate compilation cycle. It also comes with a huge repository of packages covering a large area of applications. Python is equipped with other features which facilitates development and encourages documentation of large well-structured program systems. Obviously, as an interpreted language, native Python is not suitable for performing extended numerical computations. But very often the code for such computations reduces to calls to pre-compiled library routines.

4.1 Preservation of constants of motion

We have already stressed the advantage of using symplectic solvers for numerical analysis of Hamiltonian models. This is most important when simulating long time series, where conventional numerical algorithms (like the Runge-Kutta methods usually implemented in numerical packages) can lead to a continuous degradation of important qualitative properties of Hamiltonian systems, like symplecticity (preservation of phase space volume and related quantities) and constants of motion. These methods have no built-in mechanisms for preserving such properties, as is illustrated in this subsection.

As the name suggests, symplectic solvers preserve symplecticity *exactly*. There will, of course, always be errors caused by numerical roundoff, but such errors do not depend on the accuracy of the method, only on the numerical precision being used. Symplectic solvers do not preserve most other constants of motion exactly, but the error (deviation from the initial value) will oscillate in a narrow band around zero. The width of this band scales with the accuracy of the method (i.e., order and timestep) in the expected way. For the symplectic (*KiMoKi*) algorithms used in this paper, applicable to Hamiltonians of the form $H = T(\mathbf{p}) + V(\mathbf{q})$, a constant of motion is preserved exactly if it is conserved separately by T and V . In this paper, one such example is the total momentum P of Eq (9a), while H of Eq (9b) and C_3 Eq (9c) are not.

It has been proven that symplectic integrators that preserve the Hamiltonian must actually be exact solvers (modulo errors introduced by finite numerical precision). There exist special methods for integrable models, as f.i. discussed by Kuznetsov and Vanhaecke [6] and Zullo [7]. The *KiMoKi* integrators, aimed for a more general class of problems, are not able to provide exact solutions, at least not when the conventional coordinates are used.

In this subsection we compare the *KiMoKi* solvers of order 2 and 4 with the RK23 (order 2) and RK45 (order 4) Runge-Kutta methods available through the `solve_ivp` routine in the `scipy.integrate` package, for the same values of the timestep τ (for the Runge-Kutta solvers, τ is the *maximum* timestep).

There are additional methods available in `solve_ivp`, but they are—for this comparison—inferior to the Runge-Kutta ones.

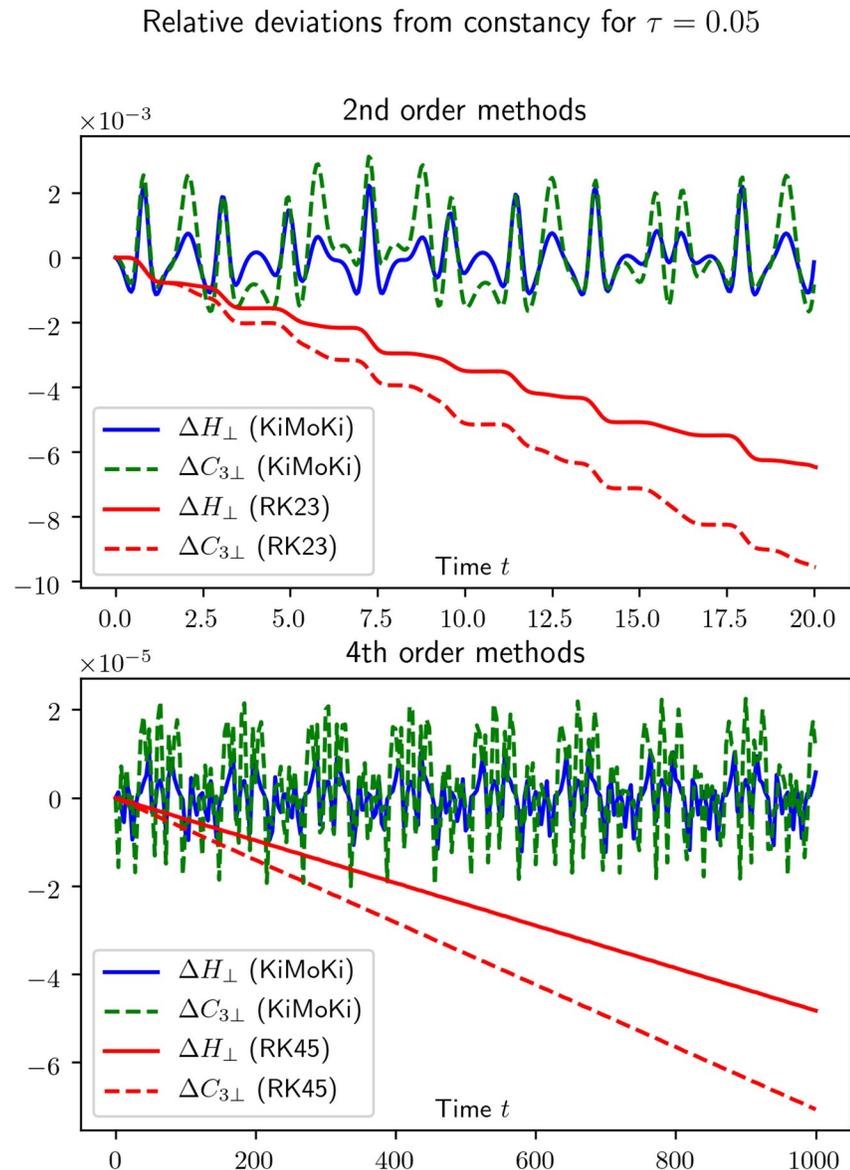


Fig 1. This figure illustrates how well exactly conserved quantities are preserved by our symplectic numerical solvers, compared to the standard Runge-Kutta methods implemented in `scipy.integrate.solve_ivp`. For the 3-site Toda lattice, reduced to relative coordinates, the constants of motion are the Hamiltonian H_{\perp} of Eq (11) and $C_{3\perp}$ of Eq (12). Here $\Delta H_{\perp} = [H_{\perp}(t) - H_{\perp}(0)]/H_{\perp}(0)$, calculated from the numerical solutions, and similar for $\Delta C_{3\perp}$. Here τ is the fixed timestep of the symplectic solvers.

<https://doi.org/10.1371/journal.pone.0215054.g001>

As can be seen from Figs 1 and 2, for short times the Runge-Kutta accuracy may very well be better than the *KiMoKi* ones, but as time increases it steadily becomes worse. By comparing Figs 1 and 2, we note that the time interval in which the Runge-Kutta methods are competitive becomes larger with decreasing τ .

4.2 Poincaré section technique (surface-of-section)

The reduced 3-site Toda lattice model we investigate has 4 degrees of freedom, $z \equiv (q_1, q_2, p_1, p_2)$. Even in this rather simple case it is a challenge to present and visualize how the solutions

Relative deviations from constancy for $\tau = 0.01$

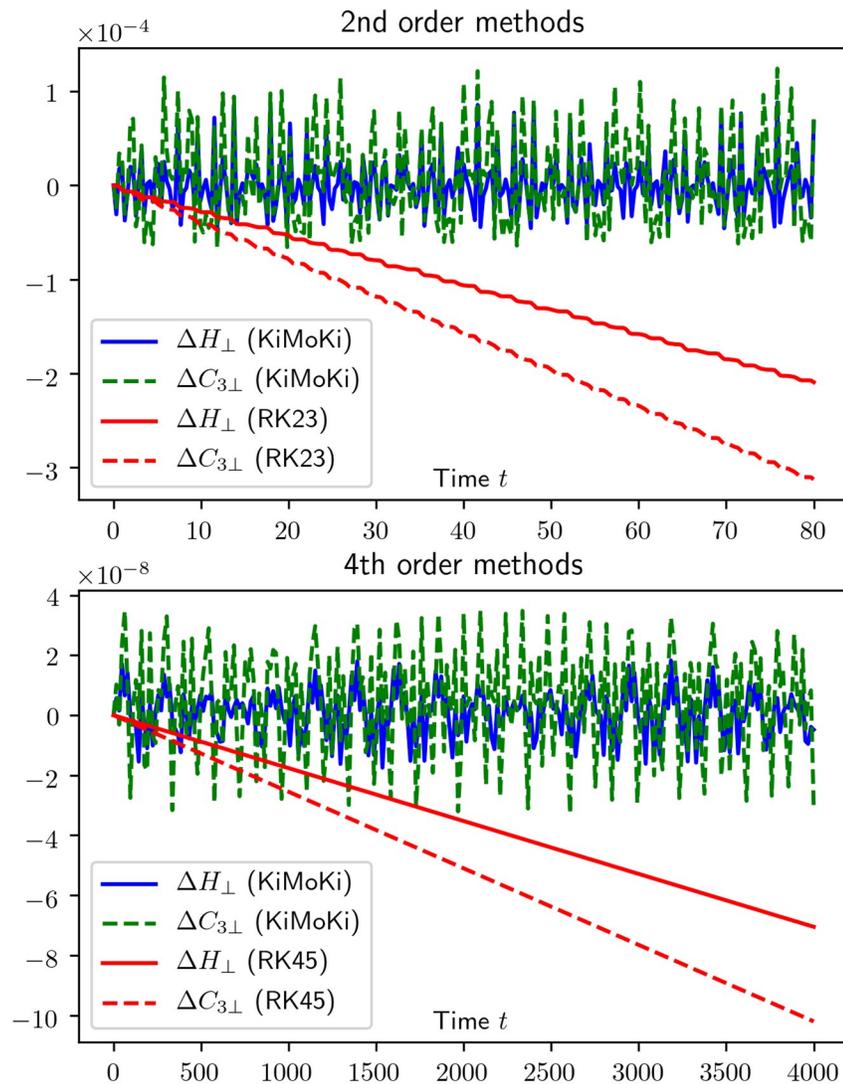


Fig 2. See the caption to Fig 1.

<https://doi.org/10.1371/journal.pone.0215054.g002>

behave. One, quite popular and efficient method, is the Poincaré section technique (also known as surface-of-section), introduced by Henri Poincaré the early 20th century. Generally, for cases where energy is conserved, $H(q_1, q_2, p_1, p_2) = H_0$, each orbit is restricted to a 3-dimensional constant energy surface of 4-dimensional phase space. The points where one coordinate is kept fixed (for example $q_2 = 0$) define another, in general independent, 3-dimensional surface. The intersection of these two surfaces is therefore two-dimensional. It can be specified by two coordinates, for example (q_1, p_1) . In this example, the points $(q_1(t_n), p_1(t_n))$ where the constant energy orbit crosses the $q_2 = 0$ surface are therefore easy to visualize in two-dimensional plots. The times t_n of crossings, and the order of repeated crossings, will be lost (or visualized by other means).

Repeated crossings will generate a pattern which indicates the nature of the dynamics. In our case, where there is an additional constant of motion ($C_{3\perp}$), repeated crossings will appear on two smooth curves—one for each direction in which the ($q_2 = 0$)-plane is crossed, determined by the initial conditions. For ergodic motion, the crossings should spread smoothly over or more regions of the plane, according to density predicted by classical statistical mechanics. According to KAM theory, perturbations of integrable models are expected to lie in-between: For a finite fraction of initial condition, the crossings will appear on a smooth curve, while the rest will appear to be spread over a region of finite area.

Cheb-Terrab and de Oliveira [18] have written a MapleV R.3 routine for visualizing Hamiltonian dynamics by the Poincaré section technique. They employ the Toda lattice model for usage demonstration. We have implemented their algorithm in python, in combination with the *KiMoKi* solvers. The algorithm uses linear interpolation to determine crossings; hence it is of limited accuracy and is best used with short time-steps τ . (All our code could have been implemented in Maple, but this framework is not freely available to all.)

Fig 3 shows 4000 crossings of the orbit with the ($q_1 = 0$)-plane, 2000 in each direction, using *KiMoKi* solvers of order 2 (left panel) and 4 (right panel) with timestep $\tau = 0.005$. For the left panel the initial condition is $z_0 = (0, 1, 9.95, 10)$. The corresponding constants of motions are $H_0 = 100$, $C_{3\perp} = -15852.7$. For the right panel the initial condition is $z_0 = (0, 1, 19.98, 10)$. The corresponding constants of motion are $H_0 = 250$, $C_{3\perp} = 16117.7$.

Fig 4 shows 4000 crossings of the orbit with the ($q_2 = 0$)-plane, 2000 in each direction, using a *KiMoKi* solver of order 6 with timestep $\tau = 0.01$. For the left panel the initial condition is $z_0 = (1, 0, 22, 5.05)$. The corresponding constants of motion are $H_0 = 256$, $C_{3\perp} = 72099.45$. For the right panel $z_0 = (0, 0.1, 1.41, 0.1)$. The corresponding constants of motion are $H_0 = 1$, $C_{3\perp} = 23.51$.

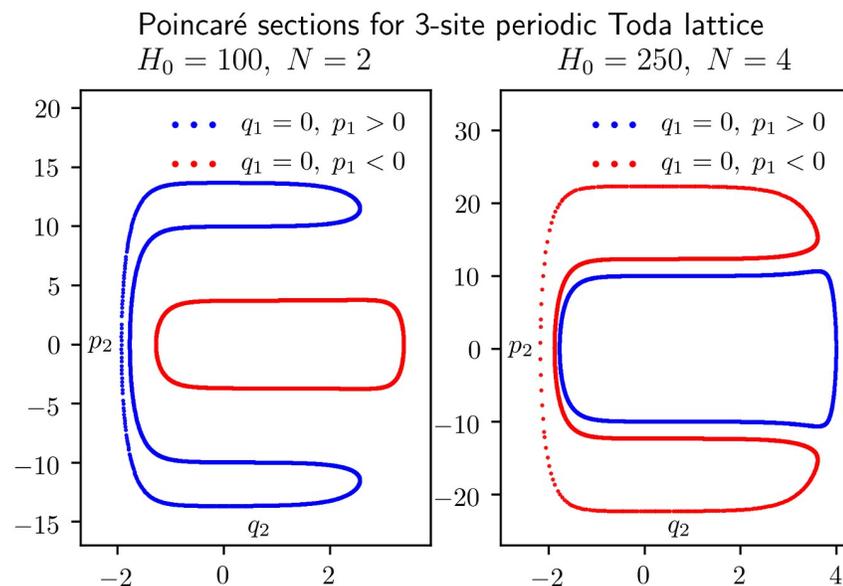


Fig 3. Poincaré sections for an orbit of the reduced 3-site Toda lattice model of Eq (11). Each panel shows 4000 crossings of the ($q_1 = 0$)-plane, 2000 in the positive direction ($p_1 > 0$, marked blue), and 2000 in the negative direction ($p_1 < 0$, marked red). The dynamics between each crossing is determined by *KiMoKi* solvers of order 2 (left panel) resp. 4 (right panel), with timestep $\tau = 0.005$. The initial condition is $z_0 = (0, 1, 9.95073, 10)$, with $H_0 = 100$, $C_{3\perp} = -15852.72982$, for the left panel, and $z_0 = (0, 1, 19.97541, 10)$, with $H_0 = 250$, $C_{3\perp} = 16117.70199$, for the right panel.

<https://doi.org/10.1371/journal.pone.0215054.g003>

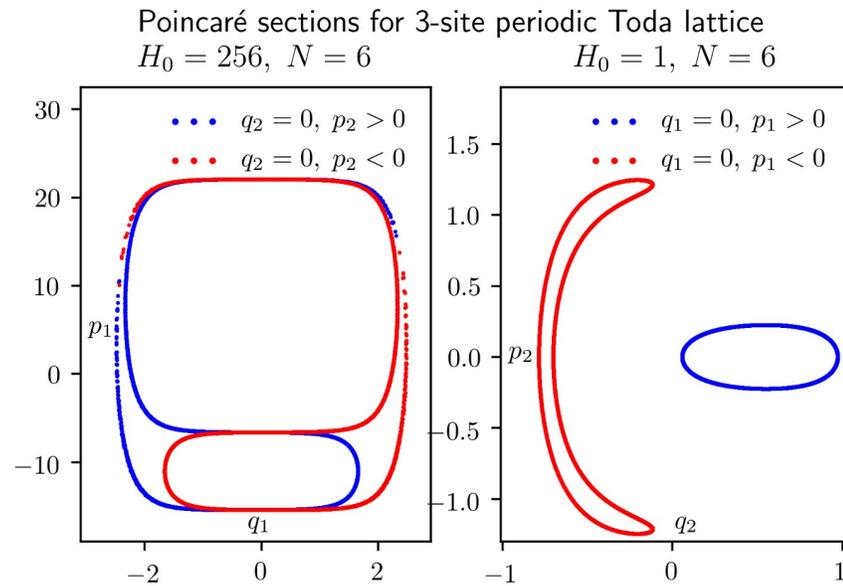


Fig 4. Poincaré sections for an orbit of the reduced 3-site Toda lattice model of Eq (11). Each panel shows 4000 crossings of the $(q_2 = 0)$ -plane, 2000 in the positive direction ($p_2 > 0$, marked blue), and 2000 in the negative direction ($p_2 < 0$, marked red). The dynamics between each crossing is determined by a *KiMoKi* solver of order 6, with timestep $\tau = 0.01$. The initial condition is $z_0 = (1, 0, 22, 5.04993)$, with $H_0 = 256$, $C_{3\perp} = 72099.45264$, for the left panel, and $z_0 = (0, 0.1, 1.40733)$, with $H_0 = 1$, $C_{3\perp} = 23.51188$ for the right panel.

<https://doi.org/10.1371/journal.pone.0215054.g004>

Fig 5 shows 4 000 crossings of the orbit with the $(q_2 = 0)$ -plane (left panel) or the $(q_1 = 0)$ -plane (right panel), 2 000 in each direction, using a *KiMoKi* solver of order 8 with timestep $\tau = 0.05$. For the left panel the initial condition is $z_0 = (0.1, 1, 0.1, 1.41)$. The corresponding constants of motion are $H_0 = 1$, $C_{3\perp} = -6.45$. For the right panel $z_0 = (0, 1, 6.93, 1)$. The corresponding constants of motion are $H_0 = 25$, $C_{3\perp} = 2597.68$.

4.3 3D camera plots of each orbit

An alternative method to visualize the solution behaviour is to make a projection to a 3-dimensional subspace, and display the orbit in a “3-dimensional” plot. This is best done in interactive sessions, since this allows one to vary the viewing direction over all possible spherical angles. Snapshots examples from such `matplotlib` sessions are shown in Figs 6 and 7, for phase space orbits $\{z(t) | 0 \leq t \leq 2\,000\}$. Each plot displays quasi-periodic motion on a two-dimensional surface determined by the initial value z_0 (or more precisely the corresponding constants of motion, H_0 and $C_{3\perp}$).

4.4 Behavior of energy error

We have earlier in Section 4.1 and Figs 1 and 2 shown that the long time behaviour of the *KiMoKi* solvers are better than the standard Runge-Kutta solvers of the same order, with respect to preservation of constants of motion. In Fig 8 we show that this behaviour can be observed for all orders N of the *KiMoKi* solvers, with the accuracy increasing with N for a fixed timestep τ . As can be seen, the errors keep varying in an oscillating manner, with no noticeable increase in amplitude with time.

In Fig 9 we further show that the error scales with order N and timestep τ in the expected manner. I.e., proportional with τ^N , with a N -dependent constant of proportionality.

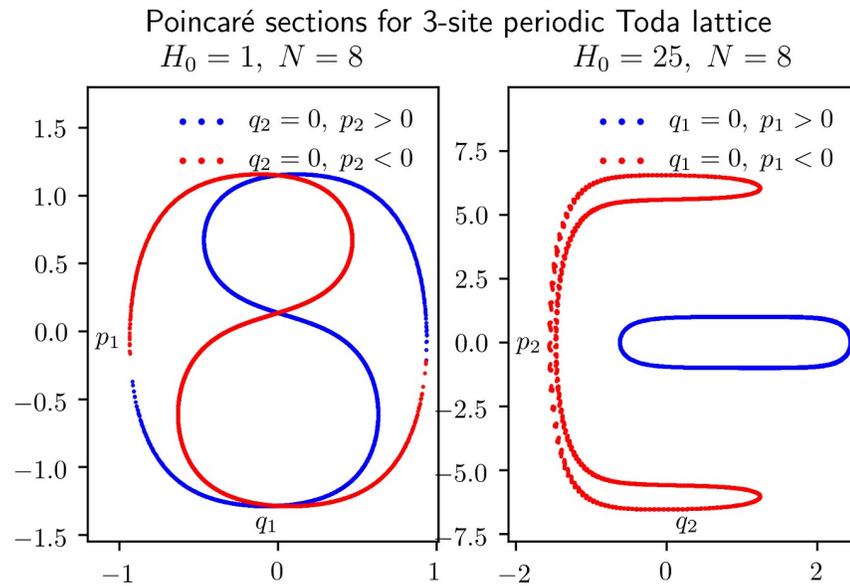


Fig 5. Poincaré sections for an orbit of the reduced 3-site Toda lattice model of Eq (11). The left panel shows 4 000 crossings of the $(q_2 = 0)$ -plane, 2 000 in the positive direction ($p_2 > 0$, marked blue) and 2 000 in the negative direction ($p_2 < 0$, marked red). The right panel similarly shows 4 000 crossings of the $(q_1 = 0)$ -plane. The dynamics between each crossing is determined by a *KiMoKi* solver of order 8, with timestep $\tau = 0.05$. The initial condition is $z_0 = (0.1, 0, 0.1, 1.40709)$, with $H_0 = 1, C_{3\perp} = -6.45412$, for the left panel, and $z_0 = (0, 1, 6.92943, 1)$, with $H_0 = 25, C_{3\perp} = 2597.68431$, for the right panel.

<https://doi.org/10.1371/journal.pone.0215054.g005>

3d-plots of solutions

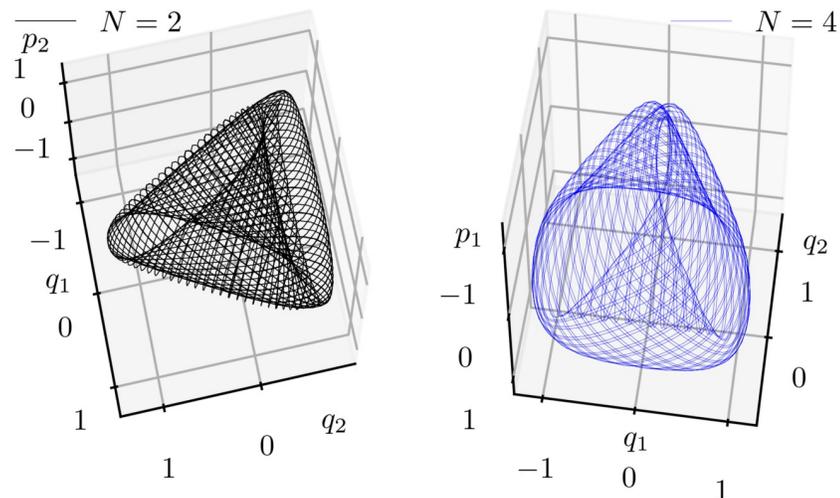


Fig 6. A quasi-periodic orbit $z(t)$ for times $0 \leq t \leq 2000$, found by the *KiMoKi* solvers of order $N = 2$ (left panel) and order $N = 4$ (right panel) with timestep $\tau = 0.1$, projected to respectively the (q_1, q_2, p_2) (left) and (q_1, q_2, p_1) (right) subspaces. The initial condition $z_0 = (0.1, 0, 0.1, 1.40709)$, with $H_0 = 1, C_{3\perp} = -6.45412$. The viewing angle is set to $(\vartheta, \varphi) = (68, 78)$ (left), respectively $(\vartheta, \varphi) = (-128, -8)$. Here ϑ is the elevation angle ($elev$) and φ the azimuth angle ($azim$).

<https://doi.org/10.1371/journal.pone.0215054.g006>

3d-plots of solutions

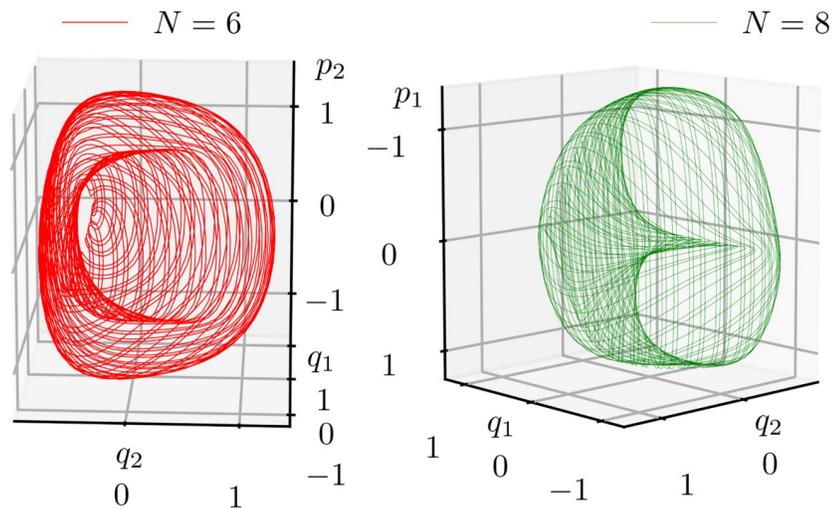


Fig 7. A quasi-periodic orbit $z(t)$ for times $0 \leq t \leq 2000$, found by the *KiMoKi* solvers of order $N = 6$ (left panel) and order $N = 8$ (right panel) with timestep $\tau = 0.1$, projected to respectively the (q_1, q_2, p_2) (left) and (q_1, q_2, p_1) (right) subspaces. The initial condition $z_0 = (0.1, 0, 0.1, 1.40709)$, with $H_0 = 1, C_{3\perp} = -6.45412$. The viewing angle is set to $(\vartheta, \varphi) = (15, -87)$ (left), respectively $(\vartheta, \varphi) = (-128, 133)$. Here ϑ is the elevation angle (elevation) and φ the azimuth angle (azimuth).

<https://doi.org/10.1371/journal.pone.0215054.g007>

5 Concluding remarks

In this paper, the *KiMoKi* algorithms for numerical solutions of the Hamilton equations for a Toda lattice model have been discussed and tested. These methods preserve the symplectic structure exactly (within the accuracy given by the employed numerical precision); For order $N = 2$ the method is equal to the Störmer-Verlet scheme, with long-time accuracy of order τ^2 ; it has been extended to methods of order τ^4, τ^6 and τ^8 . As demonstrated, the method works as expected (sometimes even better than expected) for the reduced 3-site Toda lattice model.

A brief summary of this work is as follows:

- The symplectic property is preserved provided we solve the non-linear Eq (14a) for *push* steps to sufficient accuracy.
- Without prior knowledge the quasi-periodic nature of the solutions can easily be detected from 3D plots of the orbits (projected to 3-dimensional subspaces). Further (but not independent) confirmation can be found by investigating the behaviour of the Poincaré section of each orbit.
- Although the *KiMoKi* solver do not preserve constants of motion exactly, the time oscillating error in these quantities do not systematically increase in “amplitude” with time. This amplitude can be reduced in a predictable manner by increasing the order N of the method, or decreasing the timestep τ , or both.

A Jacobi coordinates for few-body systems

Consider first a translation invariant Hamiltonian system with two non-relativistic particles of mass m_1 resp. m_2 , with position coordinates q_1 resp. q_2 . To exploit translation invariance it is

Long time accuracy of conservation laws

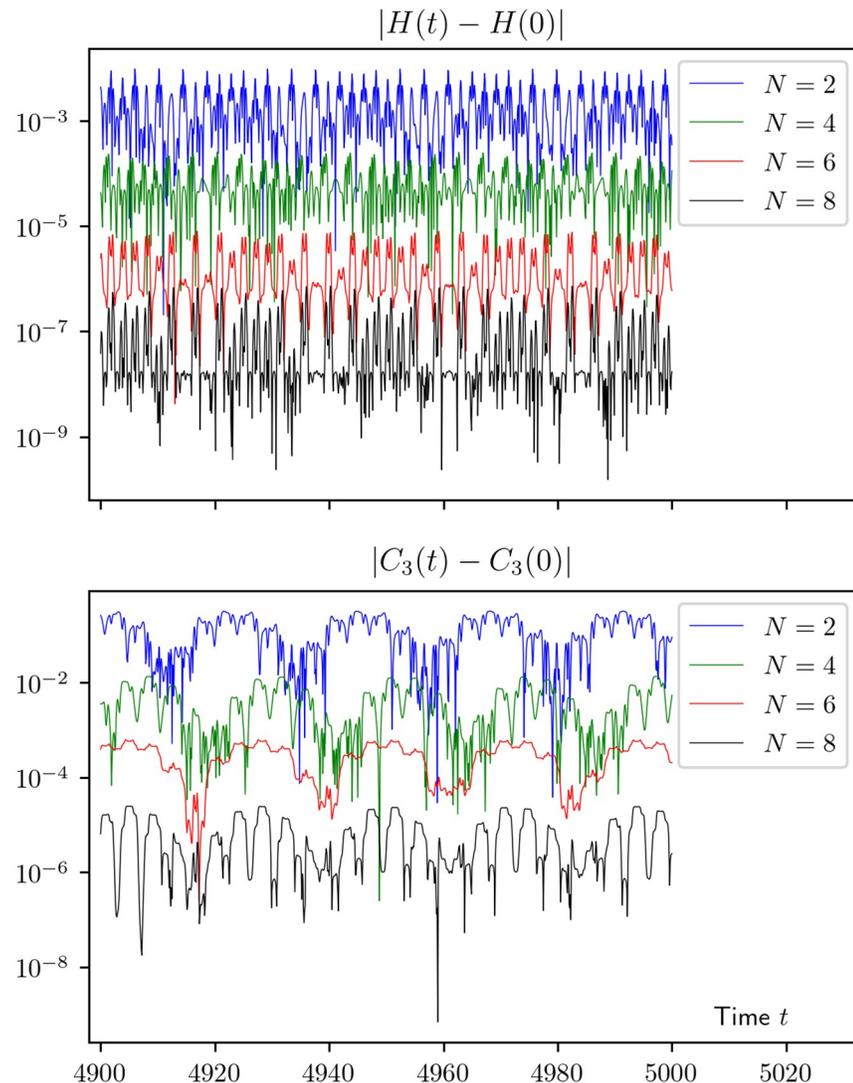


Fig 8. Long time energy error for solutions of a Toda lattice model computed by the *KiMoKi* solvers. An orbit $z(t)$ with initial value $z_0 = (0.1, 0, 0.1, 1.40709)$, corresponding to $H_0 = 1$ and $C_{3\perp} = -6.45412$. The solution is computed for times $0 \leq t \leq 5\,000$ with timestep $\tau = 0.1$; for better visibility only the last hundred time units are plotted.

<https://doi.org/10.1371/journal.pone.0215054.g008>

common to introduce center-of-mass and relative coordinates,

$$\begin{pmatrix} \mathbf{X} \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mu_1 & \mu_2 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{pmatrix}, \tag{16a}$$

where $\mu_j = m_j / (m_1 + m_2)$, for $j = 1, 2$. For common systems with conjugate momenta $\mathbf{p}_1 = m_1 \dot{\mathbf{q}}_1$ and $\mathbf{p}_2 = m_2 \dot{\mathbf{q}}_2$, the new momenta become

$$\begin{pmatrix} \mathbf{P} \\ \boldsymbol{\pi} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ \mu_2 & -\mu_1 \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix}. \tag{16b}$$

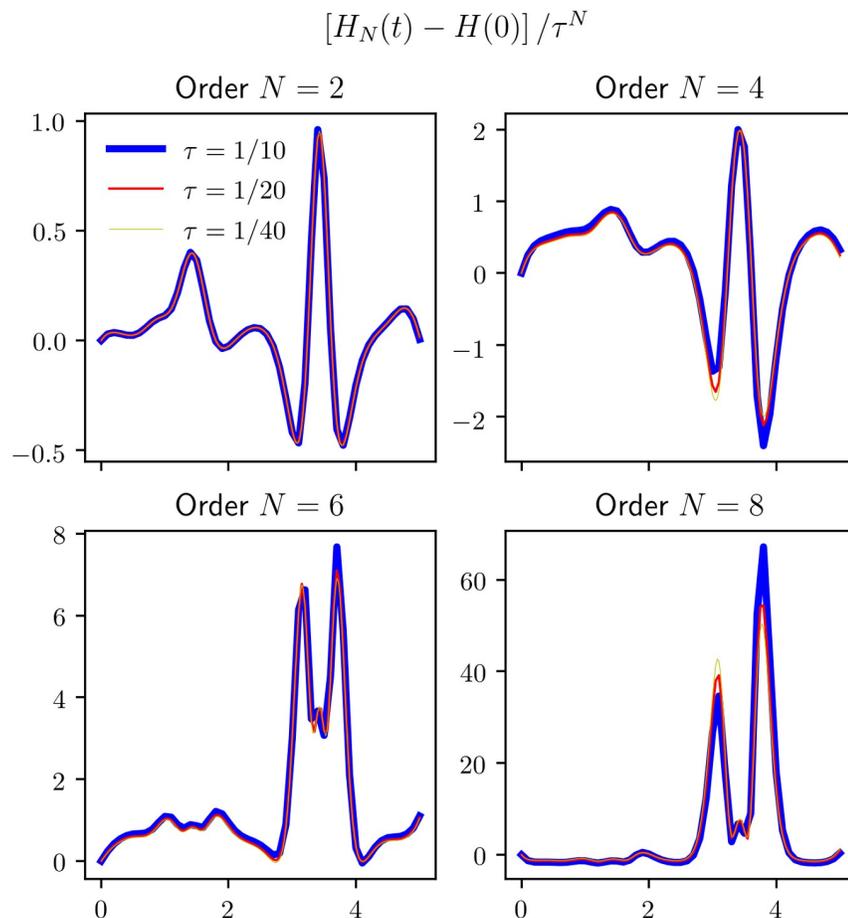


Fig 9. Scaled energy errors for some higher order symplectic integrators, when applied to the reduced 3-site Toda lattice Hamiltonian of Eq (11). The plots are for an orbit $z(t)$ with initial value $(0.1, 0, 0.1, 1.40709)$, corresponding to $H_0 = 1, C_{3\perp} = -6.45412$, computed with *KiMoKi* solvers of orders $N = 2, 4, 6, 8$, and timesteps $\tau = 0.025, 0.05$, and 0.1 .

<https://doi.org/10.1371/journal.pone.0215054.g009>

The linear transformation of Eq (16) is canonical (because the matrices M_q in Eq (16a) and M_p in Eq (16b) are related by $M_p^t = M_q^{-1}$) and maintains the diagonal form of the kinetic energy. In the equal-mass case, $\mu_1 = \mu_2 = \frac{1}{2}$ the matrices M_q and M_p do not become orthogonal. The latter, which can be obtained by scale transformations of X and x , may look simpler and more natural from a mathematical point of view. However, this would obscure physical interpretation of the coordinates.

The inverse of Eq (16) is

$$\begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} 1 & \mu_2 \\ 1 & -\mu_1 \end{pmatrix} \begin{pmatrix} X \\ x \end{pmatrix}, \tag{17a}$$

$$\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} \mu_1 & 1 \\ \mu_2 & -1 \end{pmatrix} \begin{pmatrix} P \\ \pi \end{pmatrix}. \tag{17b}$$

The extension to three particles is obvious for the center-of-mass coordinate, and one may further maintain the previous definition of one relative coordinate. As a second relative

coordinate, select the distance between the center-of mass of the first two particles, and the third one. Hence

$$\begin{pmatrix} X \\ \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mu_1 & \mu_2 & \mu_3 \\ 1 & -1 & 0 \\ \tilde{\mu}_1 & \tilde{\mu}_2 & -1 \end{pmatrix} \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{pmatrix}, \tag{18a}$$

where now $\mu_j = m_j/(m_1 + m_2 + m_3)$ for $j = 1, 2, 3$ and $\tilde{\mu}_j = m_j/(m_1 + m_2)$ for $j = 1, 2$. The new conjugate momenta becomes respectively

$$\begin{pmatrix} \mathbf{P} \\ \boldsymbol{\pi}_1 \\ \boldsymbol{\pi}_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ \tilde{\mu}_2 & -\tilde{\mu}_1 & 0 \\ \mu_3 & \mu_3 & -\mu_1 - \mu_2 \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} \tag{18b}$$

The inverse of Eq (18) is

$$\begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{pmatrix} = \begin{pmatrix} 1 & \tilde{\mu}_2 & \mu_3 \\ 1 & -\tilde{\mu}_1 & \mu_3 \\ 1 & 0 & -\mu_1 - \mu_2 \end{pmatrix} \begin{pmatrix} X \\ \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}, \tag{19a}$$

$$\begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \begin{pmatrix} \mu_1 & 1 & \tilde{\mu}_1 \\ \mu_2 & -1 & \tilde{\mu}_2 \\ \mu_3 & 0 & -1 \end{pmatrix} \begin{pmatrix} \mathbf{P} \\ \boldsymbol{\pi}_1 \\ \boldsymbol{\pi}_2 \end{pmatrix}. \tag{19b}$$

For the case of equal masses, $\mu_j = \frac{1}{3}$ and $\tilde{\mu}_j = \frac{1}{2}$.

B Unit transformations

Most quantities in physical expressions, like the Hamiltonian

$$H = \frac{1}{2m} p^2 + V(q),$$

are dimensional. I.e., they carry units of time, length, and mass. When expressed in dimensionless form like in Eq (6) or Eq (10a), this means that the dimensionless time t , length ℓ , and mass m actually are expressed in terms of some reference quantities t_0, ℓ_0, m_0 . I.e., a dimensionless potential energy $V(q) = e^{(q_a - q_{a+1})}$ must be interpreted to mean $(m_0 \ell_0^2 / t_0^2) e^{(q_a - q_{a+1}) / \ell_0}$, and the factor $\frac{1}{2}$ in the dimensionless kinetic energy must be interpreted to mean $\frac{1}{2} m_0^{-1}$. In “units where $t_0 = \ell_0 = m_0 = 1$ ”. Consider now a change of reference units to

$$(\tilde{t}_0, \tilde{\ell}_0, \tilde{m}_0) = (t_0 / \lambda_t, \ell_0 / \lambda_\ell, m_0 / \lambda_m), \tag{20}$$

with all physical quantities fixed. It is rather obvious that dimensionless coordinates will

change as

$$t \rightarrow \tilde{t} = \lambda_t t, \tag{21a}$$

$$\mathbf{q} \rightarrow \tilde{\mathbf{q}} = \lambda_q \mathbf{q}, \tag{21b}$$

$$\mathbf{p} \rightarrow \tilde{\mathbf{p}} = (\lambda_m \lambda_t / \lambda_q) \mathbf{p}. \tag{21c}$$

In this context, the statement $t \rightarrow \lambda_t t$ is shorthand for i) a change of reference units $t_0 \rightarrow \tilde{t}_0 = t_0 / \lambda_t$, implying ii) the transformation of Eq (21a), often followed by iii) a symbol renaming back to the original one, $\tilde{t} \rightarrow t$.

The corresponding transformations of $V(q)$ and $\frac{1}{2}p^2$ are less obvious,

$$V(q) \rightarrow \tilde{V}(\tilde{q}) = (\lambda_m \lambda_t^2 / \lambda_q^2) V(\lambda_q \tilde{q}), \tag{21d}$$

$$\frac{1}{2}p^2 \rightarrow \frac{1}{2}\lambda_m^{-1} \tilde{p}^2, \tag{21e}$$

and cannot be deduced from the dimensionless expressions without knowledge of which physical quantity they represent (energy in this case).

C Code snippets

In this section we provide some information of how the routines in the HOMSPY package can be used to create the symplectic solvers for the Toda lattice Hamiltonian, and how these solvers can be used to solve an initial value problem from provided initial data.

The package itself can be downloaded from the CPC Program Library at <http://cpc.cs.qub.ac.uk/summaries/ADTV>, by providing the Catalogue Id AESD v1.0. The code, with accompanying information which need not be repeated here, is found by unpacking the downloaded tar-file. Since the package is written Python 2.7, we provide code snippets illustrating how the solvers can be accessed by Python 3 code.

One way to continue is to add the code in Listing 1 to the `examples/makeExamples.py` file. But for our project we made a new folder `TodaLattice`, and added it to a new file `makeTodaLattices.py`.

Listing 1. Constructing KiMoKi solvers for a Toda lattice model

```
def makeTodaLattices():
    # Choose names for coordinates and momenta
    q1, q2, p1, p2 = sympy.symbols(['q1', 'q2', 'p1', 'p2'])
    qvars = [q1, q2]; pvars = [p1, p2]
    # Define potential in terms of coordinates
    V = (exp(2*q2+2*sqrt(3)*q1) + exp(2*q2-2*sqrt(3)*q1) +
         exp(-4*q2) - 3)/24
    kimoki.makeModules('TodaLattices', V, qvars, pvars, DP = True,
                      MP = True, MAXORDER = 8, VERBOSE = True)
```

By running this code several files will be created. The most important one is `TodaLattices.py`, which contains the *KiMoKi* solvers up to 8th order. This file should *not* be modified manually; is not intended to be studied in detail by humans. But (in particular) the multiprecision code should be checked against unintended conversions to floating point numbers. F.i., if the final division `/24` in the above definition of `V` is replaced by a pre-multiplication `(1/24)*`, then this factor will be converted to a double precision number at an early stage, and thereby pollute all multiprecision accuracy.

The file `runTodaLattices.py` provide some usage examples, intended to be modified and extended. Since this file will be overwritten the next time `makeTodaLattices` is executed, it is recommended to work on a renamed copy.

The current version of the `HOMsPY` package, with all its output, is written in Python 2.7. But functions can be accessed from Python 3 through an interface like the one in Listing 2. Data exchange via pickle files may seem primitive, but this has the advantage of documenting (preserving) the arguments and data being used.

Listing 2. Python 3 code calling Python 2.7 function.

```
def get_ivpsoln(**kwargs):
    """Solve an initial value problem by use of KiMoKi.
    The KiMoKi solver routines are currently written in Python2.7. This
    is a simple Python 3 interface using 'subprocess.call()'. Arguments
    and results are communicated via pickle files, for which some care
    with protocol and encoding is required."""
    # Default arguments:
    args = {'argfile': 'ivpargs', 'soln': 'ivpsoln',
           'tau': 0.05, 'tmax': 50., 'order': 4,
           'z0': (0., 0., 5., 5.)}
    args = {**args, **kwargs} # Override defaults
    if not os.path.isfile(f"./{args['soln']}.pkl"):
        with open(f"{args['argfile']}.pkl", 'wb') as outfile:
            pickle.dump(args, outfile, protocol = 2)
            subprocess.call(["python2", "./run_todalattices.py",
                            "solve_ivp", args['argfile']])
    with open(f"{args['soln']}.pkl", 'rb') as infile:
        soln = pickle.load(infile, encoding='latin1')
    return soln
```

On the Python 2.7 side we copied `runTodaLattices.py` to `run_todalattices.py`. All existing functions except `computeSolution(...)` can be deleted, and the function in Listing 3 must be added.

Listing 3. Python 2.7 function called from Python 3.

```
def solve_ivp(argfile):
    """Interface to python3 code through a subprocess call.
    Calling arguments ('args'), and the returned solution ('zt') are
    communicated through pickle files."""
    # All arguments with default values
    args = {'tmax': 50., 'tau': 0.05, 'order': 4,
           'z0': (0., 0., 5., 5.), 'soln': "ivpsoln"}
    with open(argfile, 'rb') as infile:
        kwargs = pickle.load(infile)
    for key, value in kwargs.items():
        args[key] = value # Override defaults
    tmax, tau = args['tmax'], args['tau']
    nMax = 1+int(tmax/tau)
    z0 = numpy.array(args['z0'])
    zt = computeSolution(z0, tau, args['order'], nMax)
    with open("%s.pkl" % args['soln'], 'wb') as outfile:
        pickle.dump(zt, outfile)
```

Further, the last two blocks in `run_todalattices.py` were changed to the one of Listing 4.

Listing 4. Python 2.7 main entry point

```
funcs = {'solve_ivp': solve_ivp, 'find_sections': find_sections}
if __name__ == "__main__":
```

```

"""Execute the function named by sys.argv[1], with argument sys.argv
[2].
"""
argc = len(sys.argv)
func = sys.argv[1] if argc > 1 else "solve_ivp"
argfile = "%s.pkl" % sys.argv[2] if argc > 2 else "args.pkl"
funcs[func](argfile)

```

D Explicit expressions

On request from a reviewer we here for convenience include the explicit expressions used in the algorithms of Eq (15). The rest of this section is an essentially unedited copy of a section with the same name, previously published by Mushtaq and Olaussen [8]:

Explicit (but compact) expressions for the terms of order τ^2 , τ^4 , and τ^6 were given in [5, 19]. With the notation,

$$\partial_a \equiv \frac{\partial}{\partial q^a}, \quad \partial^a \equiv M^{ab} \partial_b, \quad p^a \equiv M^{ab} p_b, \quad D \equiv p_a \partial^a, \quad \bar{D} \equiv (\partial_a V) \partial^a,$$

where the *Einstein summation convention is employed* (an index which occur twice, once in lower position and once in upper position, are implicitly summed over all available values; i.e, $M^{ab} \partial_b \equiv \sum_b M^{ab} \partial_b$ —we generally use the matrix M to rise an index from lower to upper position), they are

$$T_2 = -\frac{1}{12} D^2 V \tau^2, \tag{22a}$$

$$T_4 = \frac{1}{720} (D^4 - 9 \bar{D} D^2 + 3 D \bar{D}) V \tau^4, \tag{22b}$$

$$T_6 = -\frac{1}{60480} (2 D^6 - 40 \bar{D} D^4 + 46 D \bar{D} D^3 - 15 D^2 \bar{D} D^2 + 54 \bar{D}^2 D^2 - 9 \bar{D} D \bar{D} D - 42 D \bar{D}^2 D + 12 D^2 \bar{D}^2) V \tau^6 \tag{22c}$$

$$V_2 = \frac{1}{24} \bar{D} V \tau^2, \tag{22d}$$

$$V_4 = \frac{1}{480} \bar{D}^2 V \tau^4, \tag{22e}$$

$$V_6 = \frac{1}{161280} (17 \bar{D}^3 - 10 \bar{D}_3) V \tau^6. \tag{22f}$$

In this last line we have introduced

$$\bar{D}_3 \equiv (\partial_a V) (\partial_b V) (\partial_c V) \partial^a \partial^b \partial^c. \tag{23}$$

The *kick*-steps can still be integrated directly, since the V_{2k} 's only depend on \mathbf{q} . However, the T_{2k} 's (for $k \geq 1$) in general depend on both \mathbf{q} and \mathbf{p} ; hence the *move*-steps cannot be integrated directly. To overcome this problem we introduce a generating function

$$\mathcal{G}(\mathbf{q}, \mathbf{P}; \tau) = \sum_{0 \leq k \leq N} \mathcal{G}_k(\mathbf{q}, \mathbf{P}) \tau^k \tag{24}$$

such that the transformation $(q, p) \rightarrow (Q, P)$ defined implicitly by

$$Q^a = \frac{\partial \mathcal{G}}{\partial P_a}, \quad p_a = \frac{\partial \mathcal{G}}{\partial q^a}, \tag{25}$$

preserves the symplectic structure exactly, and reproduce the time evolution generated by T_{eff} to order τ^N . Here Q^a is shorthand for $q^a(t + \tau)$, and P_a shorthand for $p_a(t + \tau)$. The explicit expressions for the coefficients \mathcal{G}_k are

$$\mathcal{G}_0 = q^a P_a, \tag{26a}$$

$$\mathcal{G}_1 = \frac{1}{2} P^a P_a, \tag{26b}$$

$$\mathcal{G}_2 = 0, \tag{26c}$$

$$\mathcal{G}_3 = -\frac{1}{12} \mathcal{D}^2 V, \tag{26d}$$

$$\mathcal{G}_4 = -\frac{1}{24} \mathcal{D}^3 V, \tag{26e}$$

$$\mathcal{G}_5 = -\frac{1}{240} (3 \mathcal{D}^4 + 3 \bar{\mathcal{D}} \mathcal{D}^2 - \mathcal{D} \bar{\mathcal{D}} \mathcal{D}) V, \tag{26f}$$

$$\mathcal{G}_6 = -\frac{1}{720} (2 \mathcal{D}^5 + 8 \bar{\mathcal{D}} \mathcal{D}^3 - 5 \mathcal{D} \bar{\mathcal{D}} \mathcal{D}^2) V, \tag{26g}$$

$$\begin{aligned} \mathcal{G}_7 = & -\frac{1}{20160} (10 \mathcal{D}^6 + 10 \bar{\mathcal{D}} \mathcal{D}^4 + 90 \mathcal{D} \bar{\mathcal{D}} \mathcal{D}^3 - 75 \mathcal{D}^2 \bar{\mathcal{D}} \mathcal{D}^2 \\ & + 18 \bar{\mathcal{D}}^2 \mathcal{D}^2 - 3 \bar{\mathcal{D}} \mathcal{D} \bar{\mathcal{D}} \mathcal{D} - 14 \mathcal{D} \bar{\mathcal{D}}^2 \mathcal{D} + 4 \mathcal{D}^2 \bar{\mathcal{D}}^2) V, \end{aligned} \tag{26h}$$

$$\begin{aligned} \mathcal{G}_8 = & -\frac{1}{40320} (3 \mathcal{D}^7 - 87 \bar{\mathcal{D}} \mathcal{D}^5 + 231 \mathcal{D} \bar{\mathcal{D}} \mathcal{D}^4 - 133 \mathcal{D}^2 \bar{\mathcal{D}} \mathcal{D}^3 + 63 \bar{\mathcal{D}}^2 \mathcal{D}^3 \\ & - 3 \mathcal{D} \bar{\mathcal{D}}^2 \mathcal{D}^2 - 21 \mathcal{D}^2 \bar{\mathcal{D}}^2 \mathcal{D} + 4 \mathcal{D}^3 \bar{\mathcal{D}}^2 - 63 \bar{\mathcal{D}} \mathcal{D} \bar{\mathcal{D}} \mathcal{D}^2 \\ & + 25 \mathcal{D} \bar{\mathcal{D}} \mathcal{D} \bar{\mathcal{D}} \mathcal{D}) V. \end{aligned} \tag{26i}$$

The Eqs (22,26) define the *kick-move-kick* scheme for a general potential V . If one uses all the listed terms the local error becomes of order τ^9 , and the scheme will respect long-time conservation of energy to order τ^8 .

Acknowledgments

A. Mushtaq and A. Noreen would like to thank Prof. Kåre Olaussen for his valuable contributions in the collaborated research that forms the basis of this paper, and for informative discussions about Toda lattice models. They further thank the Department of Mathematics, FLU, Nord University for research support enabling parts of this work to be done.

Author Contributions

Conceptualization: Asif Mushtaq, Muhammad Asif Farooq.

Data curation: Asif Mushtaq.

Formal analysis: Asif Mushtaq.

Investigation: Asif Mushtaq, Amna Noreen.

Methodology: Asif Mushtaq.

Project administration: Asif Mushtaq.

Resources: Asif Mushtaq.

Software: Asif Mushtaq.

Supervision: Asif Mushtaq, Amna Noreen.

Validation: Asif Mushtaq.

Visualization: Asif Mushtaq.

Writing – original draft: Asif Mushtaq.

Writing – review & editing: Asif Mushtaq.

References

1. Hairer E, Lubich C, Wanner G. Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations. 2nd ed. Berlin: Springer; 2006. Available from: <https://doi.org/10.1007/3-540-30666-8>.
2. McLachlan RI, Quispel GRW. Geometric integrators for ODEs. *Journal of Physics A: Mathematical and General*. 2006; 39(19):5251–5285. <https://doi.org/10.1088/0305-4470/39/19/S01>
3. McLachlan RI, Quispel GR. In: Devore R, Iserles A, Süli E, editors. Six lectures on the geometric integration of ODEs. London Mathematical Society Lecture Note Series. Cambridge University Press; 2001. p. 155–210. Available from: <https://doi.org/10.1017/CBO9781107360198.008>.
4. Hairer E. In: Long-time energy conservation of numerical integrators. Foundations of computational mathematics, Santander 2005. Cambridge: Cambridge University Press; 2006. p. 162–180. Available from: <https://doi.org/10.1017/CBO9780511721571.005>.
5. Mushtaq A, Kværnø, Olaussen K. Higher-Order Geometric Integrators for a Class of Hamiltonian Systems. *International Journal of Geometric Methods in Modern Physics*. 2014; 11(01):1450009–1–20. <https://doi.org/10.1142/S0219887814500091>
6. Kuznetsov V, Vanhaecke P. Bäcklund transformations for finite-dimensional integrable systems: a geometric approach. *Journal of Geometry and Physics*. 2002; 44(1):1–40. [https://doi.org/10.1016/S0393-0440\(02\)00029-3](https://doi.org/10.1016/S0393-0440(02)00029-3)
7. Zullo F. Bäcklund transformations and Hamiltonian flows. *Journal of Physics A: Mathematical and Theoretical*. 2013; 46(14):145203. <https://doi.org/10.1088/1751-8113/46/14/145203>
8. Mushtaq A, Olaussen K. Automatic code generator for higher order integrators. *Computer Physics Communications*. 2014; 185(5):1461–1472. <https://doi.org/10.1016/j.cpc.2014.01.012>
9. Mushtaq A. Numerical Geometric Integration with HOMsPy. *IAENG Int J of App Math*. 2015; 45(4):383–391.
10. Toda M. Vibration of a Chain with Nonlinear Interaction. *Journal of the Physical Society of Japan*. 1967; 22(2):431–436. <https://doi.org/10.1143/JPSJ.22.431>
11. Fermi E, Pasta J, Tsingou M, Ulam SM. Studies of nonlinear problems. I; 1955. LANL report LA-1940. Available from: <https://www.osti.gov/servlets/purl/4376203>.
12. Flaschka H. The Toda lattice. II. Existence of integrals. *Phys Rev B*. 1974; 9:1924–1925. <https://doi.org/10.1103/PhysRevB.9.1924>
13. Lax PD. Integrals of nonlinear equations of evolution and solitary waves. *Communications on Pure and Applied Mathematics*; 21(5):467–490. <https://doi.org/10.1002/cpa.3160210503>
14. Ünal Göktaş, Hereman W, Erdmann G. Computation of conserved densities for systems of nonlinear differential-difference equations. *Physics Letters A*. 1997; 236(1):30–38.
15. Hohler EGB, Olaussen K. Conservation laws for the classical Toda field theories. *Modern Physics Letters A*. 1993; 08(35):3377–3385. <https://doi.org/10.1142/S0217732393003809>
16. Lunsford GH, Ford J. On the Stability of Periodic Orbits for Nonlinear Oscillator Systems in Regions Exhibiting Stochastic Behavior. *Journal of Mathematical Physics*. 1972; 13(5):700–705. <https://doi.org/10.1063/1.1666037>
17. Henon M, Heiles C. The applicability of the third integral of motion: Some numerical experiments. *Astronomical Journal*. 1964; 69:73–79. <https://doi.org/10.1086/109234>

18. Cheb-Terrab ES, de Oliveira HP. Poincaré sections of Hamiltonian systems. *Computer Physics Communications*. 1996; 95(2):171–189. [https://doi.org/10.1016/0010-4655\(96\)00032-X](https://doi.org/10.1016/0010-4655(96)00032-X)
19. Asif Mushtaq, Anne Kværnø and Kåre Olaussen. Systematic Improvement of Splitting Methods for the Hamilton Equations. *Proceedings of the World Congress on Engineering, London July 4–6. 2012; (I):247–251.*