

UDC 519.872:519.217

PACS 07.05.Tp, 02.60.Pn, 02.70.Bf

DOI: 10.22363/2658-4670-2020-28-4-327-345

## On the realization of explicit Runge–Kutta schemes preserving quadratic invariants of dynamical systems

Yu Ying<sup>1</sup>, Mikhail D. Malykh<sup>2</sup>

<sup>1</sup> *Kaili University  
3, Kaiyuan Road, Kaili, 556011, China*

<sup>2</sup> *Peoples' Friendship University of Russia (RUDN University)  
6, Miklukho-Maklaya St., Moscow, 117198, Russian Federation*

(received: October 23, 2020; accepted: November 12, 2020)

We implement several explicit Runge–Kutta schemes that preserve quadratic invariants of autonomous dynamical systems in Sage. In this paper, we want to present our package `ex.sage` and the results of our numerical experiments.

In the package, the functions `rrk_solve`, `idt_solve` and `project_1` are constructed for the case when only one given quadratic invariant will be exactly preserved. The function `phi_solve_1` allows us to preserve two specified quadratic invariants simultaneously. To solve the equations with respect to parameters determined by the conservation law we use the elimination technique based on Gröbner basis implemented in Sage. An elliptic oscillator is used as a test example of the presented package. This dynamical system has two quadratic invariants. Numerical results of the comparing of standard explicit Runge–Kutta method RK(4,4) with `rrk_solve` are presented. In addition, for the functions `rrk_solve` and `idt_solve`, that preserve only one given invariant, we investigated the change of the second quadratic invariant of the elliptic oscillator. In conclusion, the drawbacks of using these schemes are discussed.

**Key words and phrases:** Explicit Runge–Kutta method, quadratic invariant, dynamical system, Sage

### 1. Quadratic invariant and conservative RK scheme

One of most widespread mathematical models is an autonomous system of ordinary differential equations, i.e., the system of the form

$$\begin{cases} \frac{dx}{dt} = f(x), & t \geq 0, \\ x(0) = x^0, \end{cases} \quad (1)$$

where:  $t$  is an independent variable, commonly interpreted as time;  $x$  is a vector  $(x_1, \dots, x_n)$ ;  $f$  is a vector function  $(f_1, f_2, \dots, f_n)$ , when in applications

© Ying Y., Malykh M. D., 2020



This work is licensed under a Creative Commons Attribution 4.0 International License

<http://creativecommons.org/licenses/by/4.0/>

its element  $f_i$  ( $i = 1, 2, \dots, n$ ) is often taken as a rational or an algebraic function of the coordinates  $x_1, \dots, x_n$  or one can be reduced to this form by some transformation of variables.

**Definition 1 (Goriely [1]).** If there exists a function  $I$  of  $x$ , such that, for any solution  $x(t)$  of system (1), the condition

$$\nabla I(x)f(x) = \sum_{k=1}^n \frac{dx_k}{dt} \frac{\partial I}{\partial x_k} = \sum_{k=1}^n f_k(x(t)) \frac{\partial I}{\partial x_k} = 0,$$

holds, then  $I$  is called the first integral or invariant of the system (1). If  $x(t)$  is any exact solution of system (1) then  $I(x(t))$  is independent of  $t$ . If  $I$  is a polynomial of degree 2 with respect to  $x$  then it is called a quadratic invariant.

Any quadratic invariant after a linear transformation can be rewritten in the form

$$I(x(t)) = x(t)^T S x(t) = \langle Sx(t), x(t) \rangle = \text{const}, \tag{2}$$

where  $\langle \cdot, \cdot \rangle$  denote the Euclidean inner product on  $\mathbb{R}^n$  and  $S \in \mathbb{R}^{n \times n}$  is a symmetric, constant matrix.

To determine an uniform grid (with a step  $\Delta t$ ) of the time interval  $[0, T]$  we take

$$t_n = n\Delta t \quad (n = 0, \dots, N).$$

We will interpret  $\{x^n\}$  as an approximation to the exact solution  $x(t)$  at time  $t_0 + n \Delta t$ , i.e.

$$x(t_0 + n\Delta t) \approx x^n.$$

For the system (1) Runge–Kutta scheme (RK scheme) with  $s$  stages can be written as

$$k_i = x^n + \Delta t \sum_{j=1}^s a_{ij} f(k_j), \quad i = 1, 2, \dots, s \tag{3}$$

and

$$x^{n+1} = x^n + \Delta t \sum_{i=1}^s b_i f(k_i). \tag{4}$$

Below the parameters  $a_{ij}$  and  $b_i$  ( $i = 1, 2, \dots, s, j = 1, 2, \dots, s$ ) will be arranged in an array

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

where

$$c_i = \sum_{j=1}^s a_{ij},$$

is known as the Butcher table [2], [3] and will be called the coefficients of the RK scheme. The RK method is explicit if  $a_{ij} = 0$  for  $i \leq j$ , otherwise is implicit.

We want to construct numerical solutions  $x^0, x^1, \dots, x^N$  such that the quadratic invariant  $I(x)$  is preserved numerically, i.e.

$$\langle Sx^n, x^n \rangle = \langle Sx^0, x^0 \rangle \quad i = 1, \dots, N. \tag{5}$$

In this case the RK method will be called S-conservative RK scheme.

Ref. [4]–[6] indicated that the RK method preserves the quadratic first integrals of system (1) iff the coefficients of such RK method satisfy

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \quad i, j = 1, \dots, s. \tag{6}$$

Such Runge–Kutta methods are called *symplectic*.

Obviously, no explicit RK schemes satisfies the symplectic condition [6], [7]. Unfortunately, during using the implicit schemes, we must solve a system of non-linear algebraic equations at each step. This is very complex problem, so implicit schemes require more resources than explicit RK schemes [8]. Furthermore numerical solutions of nonlinear system (for ex., by the Newton method) introduce new errors that sometimes we cannot estimate effectively [9]. Thus, the integrals could not be preserved exactly. For this reason, many authors try to construct numerical methods for solving the system of differential equations (1) with the preservation of algebraic integrals without the need to solve nonlinear algebraic equations.

To overcome these difficulties Bueno and Mastroserio [10] suggested a method that uses explicit RK schemes for the construction of new finite-difference schemes which exactly preserve invariants. Below we will call it the Bueno method for shorthand. Of course, these new schemes are not standard RK schemes, but they are usually called an *explicit RK scheme preserving invariants* [8]. These schemes preserve only one specified invariant. We implemented several such schemes in Sage and investigated what happens to other invariants. Next, we investigated the method from th article [11] by Calvo et al. which is an extension of the Bueno method and can be used as a conservation one or more invariants. Below we will call it the Calvo method for shorthand.

## 2. Explicit RK scheme of preserving one quadratic invariant

### 2.1. The Bueno method

To make the explicit RK scheme conservative, we follow to Bueno and Mastroserio [10]. We scale the weights  $b_i$  by a parameter  $\gamma_n \in \mathbb{R}$  at the step  $n$ , i.e. use

$$x_\gamma^{n+1} = x^n + \Delta t \sum_{i=1}^s \gamma_n b_i f(k_i) \tag{7}$$

instead of  $x^{n+1}$  obtained by the formula (4) numerical solution after one time step.

Using the shorthand

$$\Delta_n = \sum_{i=1}^s b_i f(k_i),$$

the parameter  $\gamma_n$  could be estimated by the conservative condition, i.e.

$$\langle Sx_\gamma^{n+1}, x_\gamma^{n+1} \rangle - \langle Sx^n, x^n \rangle = \gamma_n \Delta t (2\langle Sx^n, \Delta_n \rangle + \gamma_n \Delta t \langle \Delta_n, \Delta_n \rangle). \quad (8)$$

Thus, we preserve the invariant  $\langle Sx, x \rangle$ , if we take

$$\gamma_n = -\frac{2\langle Sx^n, \Delta_n \rangle}{\Delta t \langle \Delta_n, \Delta_n \rangle}. \quad (9)$$

For Runge–Kutta schemes of order  $p$  this expression is close to 1, i.e.

$$\gamma_n = 1 + \mathcal{O}(\Delta t^{p-1})$$

as  $\Delta t \rightarrow 0$ , see Buono et al. [10, prop. 4] for  $p = 4$  and Zhang et al. [8, lemma 3.3]. Thus, the new numerical solution  $x_\gamma^{n+1}$  can be considered as an approximation either to  $x(t_n + \Delta t)$  with the RK weights scaled by  $\gamma_n$ , or to  $x(t_n + \gamma_n \Delta t)$  with the time  $\Delta t$  scaled by  $\gamma_n$ . Zhang et al. [8] denote the method defined by (3) and (7) with the interpretation

$$x_\gamma^{n+1} \approx x(t_n + \gamma_n \Delta t)$$

as the *relaxation Runge–Kutta method* (RRK), while the method using

$$x_\gamma^{n+1} \approx x(t_n + \Delta t)$$

this is called the *increment direction technique* (IDT). Note that the value of the scalar parameter  $\gamma_n$  at each step depends on the quadratic invariant that appears in *rrk* or *idt* method.

**Theorem 1 (Zhang et al. [8]).** *Let the original RK scheme be defined by (3) and (4) has order  $p$ , then the method defined by (3) and (7) has:*

- (RRK method) *If the solution  $x_\gamma^{n+1}$  is interpreted as an approximation to  $x(t_n + \gamma_n \Delta t)$ , the method has order  $p$ .*
- (IDT method) *If the solution  $x_\gamma^{n+1}$  is interpreted as an approximation to  $x(t_n + \Delta t)$ , the method has order  $p - 1$ .*

In our package `ex.sage` for Sage [12] the function `rrk_solve(P1,F,ics)` returns the numeric points  $(0, x^0), (\gamma_0 \Delta t, x^1), \dots$  with the parameters:

- `P1` is a quadratic invariant;
- `F` is the right sides of system (1)
- `ics` is the initial condition
- the default  $\Delta t = 0.1, T = 10$
- 4-stage explicit RK scheme with the Butcher table

$$\begin{array}{c|cccc}
 0 & 0 & 0 & 0 & 0 \\
 \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\
 \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\
 1 & 0 & 0 & 1 & 0 \\
 \hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
 \end{array}$$

Function `idt_solve()` returns the numerical points  $(0, x^0), (\Delta t, x^1), \dots$  with the parameters:

- `P1` is a quadratic invariant;
- `F` is the right sides of the system (1)
- `ics` is the initial condition
- the default  $\Delta t = 0.1, T = 10$
- 4-stage explicit RK scheme with coefficients the same as the previous table.

Users can redefine these variables in both functions, for ex., by adding `dt=0.01` or new explicit RK method.

## 2.2. Elliptic function test

To test this routine, we investigate a nonlinear oscillator. By the definition of Jacobi functions [13],  $p = \operatorname{sn} t$ ,  $q = \operatorname{cn} t$ ,  $r = \operatorname{dn} t$  is a particular solution to a nonlinear autonomous system of differential equations

$$\dot{p} = qr, \quad \dot{q} = -pr, \quad \dot{r} = -k^2 pq \quad (10)$$

with the initial conditions

$$p = 0, \quad q = r = 1 \text{ at } t = 0.$$

This autonomous system has two quadratic integrals of motion

$$p^2 + q^2 = 1 \quad \text{and} \quad k^2 p^2 + r^2 = 1 \quad (11)$$

We can solve the autonomous system (10) by `rrk` or `idt` methods that preserve only the first or second integral. For certainly, we take  $k = 1/2$  and indicated above initial condition.

```

sage: var('p, q, r')
sage: load('ex.sage')
sage: k=1/2
sage: s=4
sage: F=[r*q, -p*r, -k^2*p*q]
sage: list_of_integral=[p^2+q^2, k^2*p^2+r^2]
sage: ics=[p==0, q==1, r==1]
sage: idt_solve(P1=list_of_integral[0], ics=ics, F=F, dt=0.2, T=1)
sage: rrk_solve(P1=list_of_integral[0], ics=ics, F=F, dt=0.2, T=1)
sage:
  ↪ B1=rrk_solve(P1=list_of_integral[0], ics=ics, F=F, dt=0.1, T=40)
sage:
  ↪ B2=rrk_solve(P1=list_of_integral[1], ics=ics, F=F, dt=0.1, T=40)

```

```

sage: G=line([[t,p] for [t,p,q,r] in
↪ B1],color='red',axes_labels=['t','p'])+point([[t,p] for
↪ [t,p,q,r] in B2],frame=true)
sage: max(abs(k^2*p^2+r^2-1) for [t,p,q,r] in B2 )
sage: max(abs(p^2+q^2-1) for [t,p,q,r] in B2)
sage: max(abs(k^2*p^2+r^2-1) for [t,p,q,r] in B1 )
sage: max(abs(p^2+q^2-1) for [t,p,q,r] in B1)
sage: G1=line([[t,k^2*p^2+r^2-1] for [t,p,q,r] in
↪ B1],axes_labels=['$t$', '$k^2p^2+r^2-1$'],tick_formatter=[None,RR(10e-
↪ 18).n(digits=1)],frame=true)
sage: G2=line([[t,p^2+q^2-1] for [t,p,q,r] in
↪ B2],axes_labels=['$t$', '$p^2+q^2-1$'],tick_formatter=[None,RR(10e-
↪ 18).n(digits=1)],frame=true)

```

In figure 1 we can see a graph of the solution found by rk method with exact conservation of  $p^2 + q^2 = 1$ . Rrk give a condensation of the greed points in those arches of the graph where the curvature has a maximum.

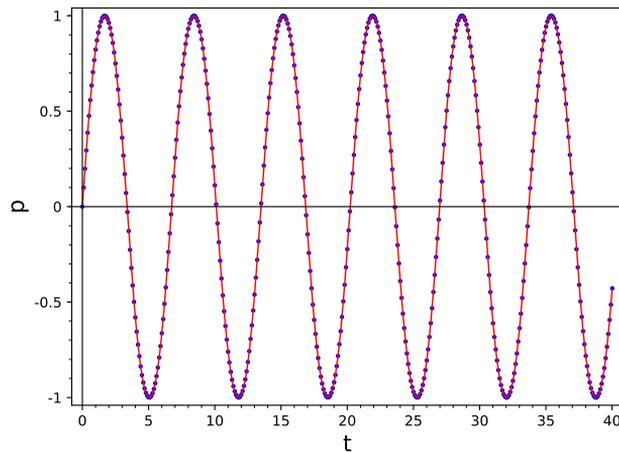


Figure 1. Graph of  $p(t)$ , rk,  $dt = 0.1$

The second integral  $k^2p^2 + r^2 - 1$  is not exactly preserved, but its value fluctuates with a small amplitude of  $10^{-7}$  (figure 2). We also use rk with exact conservation of  $k^2p^2 + r^2 = 1$ . In this case first integral grows by leaps bounds and quickly becomes larger than  $10^{-3}$  (figure 3). Thus, the preservation of one integral does not preserve others.

Let's compare the Buono method with the standard rk4 at the same step size  $dt$  (which denote  $\Delta t$  in the rk method).

```

sage: var('p,q,r,t')
sage: QRK=desolve_system_rk4(F,[p,q,r], ics=[0,0,1,1,], ivar=t,
↪ end_points=20)

```

```
sage: G3=line([[t,p^2+q^2] for [t,p,q,r] in
↳ B2],color='red')+point([[t,p^2+q^2] for [t,p,q,r] in
↳ QRK],axes_labels=['$t$', '$p^2+q^2$'], tick_formatter=[None,
↳ RR(10e-18).n(digits=1)],frame=true)
sage: G4=line([[t,k^2*p^2+r^2-1] for [t,p,q,r] in
B1],axes_labels=['$t$', '$k^2p^2+r^2-1$'],tick_formatter=[None,
↳ RR(10e-18).n(digits=1)], color='red') +
↳ point([[t,k^2*p^2+r^2-1] for [t,p,q,r] in QRK],frame=true)
```

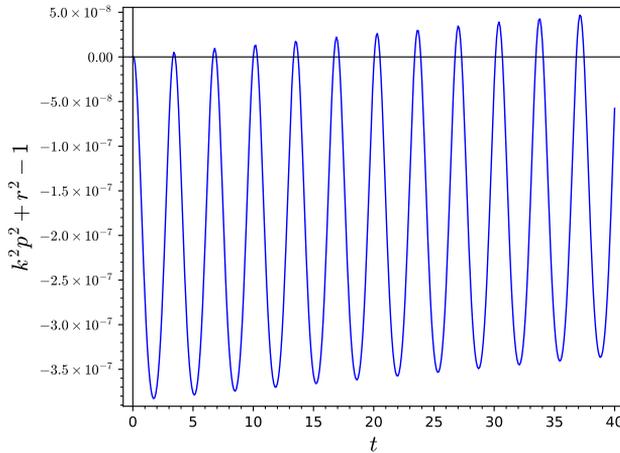


Figure 2. Second invariant for rk4 method with exact conservation of first invariant  $p^2 + q^2 = 1$

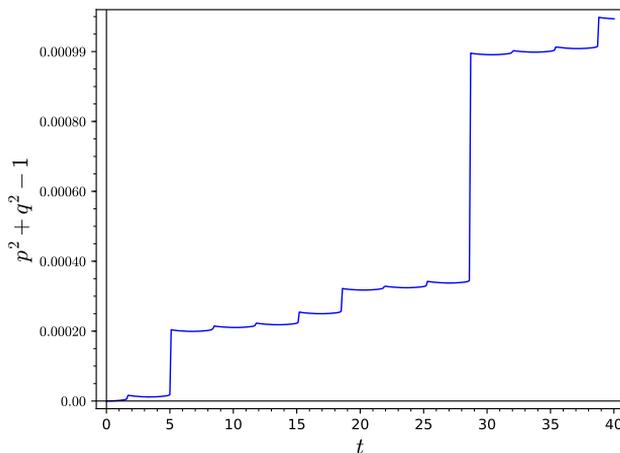


Figure 3. First integral for rk4 with exact conservation of  $k^2p^2 + r^2 = 1$

The rk4 method with the exact conservation of the first invariant  $p^2 + q^2 = 1$  preserves both integrals better than rk4 (figure 5) but the rk4 with the exact conservation of  $k^2p^2 + r^2 = 1$  preserves the first integral worse than rk4 (figure 4).

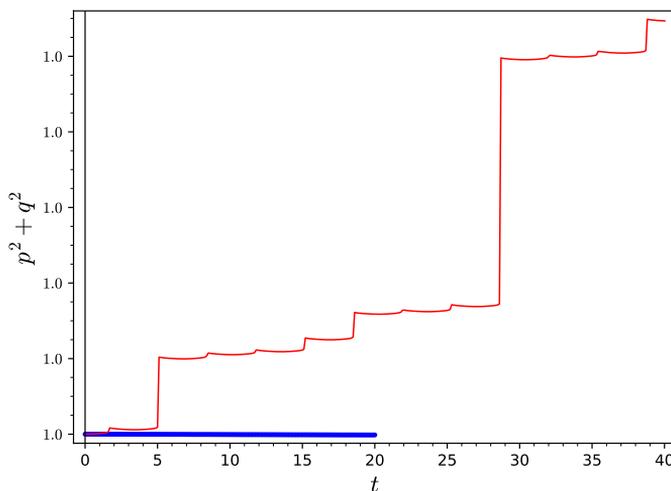


Figure 4. First invariant for rrk method with exact conservation of second invariant  $k^2 p^2 + r^2 = 1$  (red) and for standard rk4

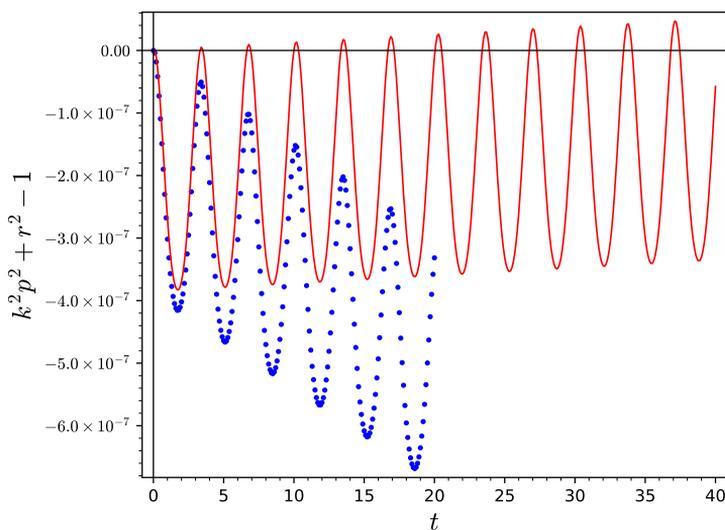


Figure 5. Second invariant for rrk method with exact conservation of first invariant  $p^2 + q^2 = 1$  (red) and for standard rk4 (blue)

From the numerical experiments, we came to the conclusion that preserving multiple integrals requires a different approach.

### 3. The Calvo method for one invariant

#### 3.1. The Calvo method

Let us take two popular explicit RK methods for examples: the RK4 and Euler methods. RK4 method has 4 stages and 4th approximation order. We calculate four axillary quantities at  $n$ th step

$$\begin{aligned} k_1 &= x^n, \\ k_2 &= x^n + \frac{1}{2}f(k_1)\Delta t, \\ k_3 &= x^n + \frac{1}{2}f(k_2)\Delta t, \\ k_4 &= x^n + f(k_3)\Delta t \end{aligned}$$

and then the quantity

$$\phi(x^n) = x^n + \Delta t \left( \frac{1}{6}f(k_1) + \frac{1}{3}f(k_2) + \frac{1}{3}f(k_3) + \frac{1}{6}f(k_4) \right) \tag{12}$$

which is used in standard way as  $x^{n+1}$ . Similarly, by the Euler method, we calculate in the step  $n$  one axillary quantity  $k_1 = x^n$  and the quantity

$$\phi_1(x^n) = x^n + f(k_1)\Delta t \tag{13}$$

which is used in standard way as  $x^{n+1}$ . We can describe this scheme by Butcher table with additional row:

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
$\phi$	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$
$\phi_1$	1	0	0	0

Calvo et al. [11] extend the Buono method by coupling these two schemes: in the step  $n$  we take

$$x_\lambda^{n+1} = \phi(x^n) - \lambda_n(\phi(x^n) - \phi_1(x^n)), \tag{14}$$

where  $\lambda_n \in \mathbb{R}$  is the scalar parameter that can be determined by the conservation law, i.e.

$$I(x_\lambda^{n+1}) = I(x^n). \tag{15}$$

Using (12) and (13) we have

$$\begin{aligned} x_\lambda^{n+1} = x^n + & \left( \frac{1}{6} - \frac{5}{6}\lambda_n \right) f(k_1)\Delta t + \left( \frac{1}{3} + \frac{1}{3}\lambda_n \right) f(k_2)\Delta t + \\ & + \left( \frac{1}{3} + \frac{1}{3}\lambda_n \right) f(k_3)\Delta t + \left( \frac{1}{6} + \frac{1}{6}\lambda_n \right) f(k_4)\Delta t. \end{aligned} \tag{16}$$

Thus, (15) gives an algebraic equation to determining the parameter  $\lambda_n$  at each step. Calvo et al [11] proved that approximation order of this new scheme is equal to 3. They investigated more general case when coupling any two explicit RK scheme.

If  $I$  is a quadratic invariant then we have a quadratic equation for  $\lambda_n$ , one of the roots of which goes to 0 at  $\Delta t \rightarrow 0$  and the other goes to  $\infty$ . In numerical experiments we choose the parameter  $\lambda_n$  as real number which is close to the value 0. In CAS sagemath, we use symbolic calculation to solve this equation (15) with respect to  $\lambda_n$ , and use the function `roots` to get this value, since our calculation is performed in the ring  $\mathbb{R}$  (Real Field with 53 bits of precision). So, there is a small error, since in  $\overline{\mathbb{Q}}$  will get the exact value, but it is very time-consuming.

### 3.2. Elliptic function test

We implement the described scheme in Sage as the function `project_1()`.

Function `project_1()` returns the numerical points  $(0, x^0), (\Delta t, x^1), \dots$  with the parameters:

- `list_of_integral` is an invariant required to be conserved;
- `F` is the right sides of the system (1)
- `ics` is the initial condition
- `dt` is the step size, `T` is the end point of time  $t$ .

Let us take the elliptic function for example.

```
sage: load('ex.sage')
sage: var('p,q,r,k')
sage: k=1/2
sage: list_of_integral=[p^2+q^2,k^2*p^2+r^2]
sage: F=[r*q,-p*r,-k^2*p*q]
sage: ics=[p==0,q==1,r==1]
sage: B1=project_1(list_of_integral[0],F,ics,dt=0.1,T=20)
sage: max([abs(p^2+q^2-1) for [t,p,q,r] in B1])
sage: P1=line([[t,p] for [t,p,q,r] in
  ↪ B1],axes_labels=['t$', '$p$'],tick_formatter=[None,RR(10e-
  ↪ 15).n(digits=1)],frame=true, color='red')+point([[t,p] for
  ↪ [t,p,q,r] in B1])
sage: P1=line([[t,p^2+q^2-1] for [t,p,q,r] in
  ↪ B1],axes_labels=['t$', '$p^2+q^2-
  ↪ 1$'],tick_formatter=[None,RR(10e-
  ↪ 15).n(digits=1)],frame=true,
  ↪ color='red')
sage: P2=line([[t,k^2*p^2+r^2-1] for [t,p,q,r] in
  ↪ B1],axes_labels=['t$', '$k^2*p^2+r^2-
  ↪ 1$'],tick_formatter=[None,RR(10e-
  ↪ 15).n(digits=1)],frame=true)
sage: B2=project_1(list_of_integral[1],F,ics,dt=0.1,T=20)
sage: max([abs(p^2+q^2-1) for [t,p,q,r] in B2])
sage: P0=line([[t,p] for [t,p,q,r] in
  ↪ B2],axes_labels=['t$', '$p$'],tick_formatter=[None,RR(10e-
  ↪ 15).n(digits=1)],frame=true, color='red')+point([[t,p] for
  ↪ [t,p,q,r] in B2])
```

```
sage: P01=line([[t,p^2+q^2-1] for [t,p,q,r] in
↳ B2],axes_labels=['$t$', '$p^2+q^2-1$'],tick_formatter=[None,RR(10e-
↳ 15).n(digits=1)],frame=true)
sage: P02=line([[t,k^2*p^2+r^2-1] for [t,p,q,r] in
↳ B2],axes_labels=['$t$', '$k^2*p^2+r^2-1$'],tick_formatter=[None,RR(10e-
↳ 16).n(digits=1)],frame=true)
```

In figure 6 we can see that the solution founded by the Calvo method with exact conservation of the first invariant  $p^2 + q^2 = 1$  and the second invariant  $k^2 p^2 + r^2 = 1$  gives a condensation of greed points in those arches of the graph where the curvature has a maximum.

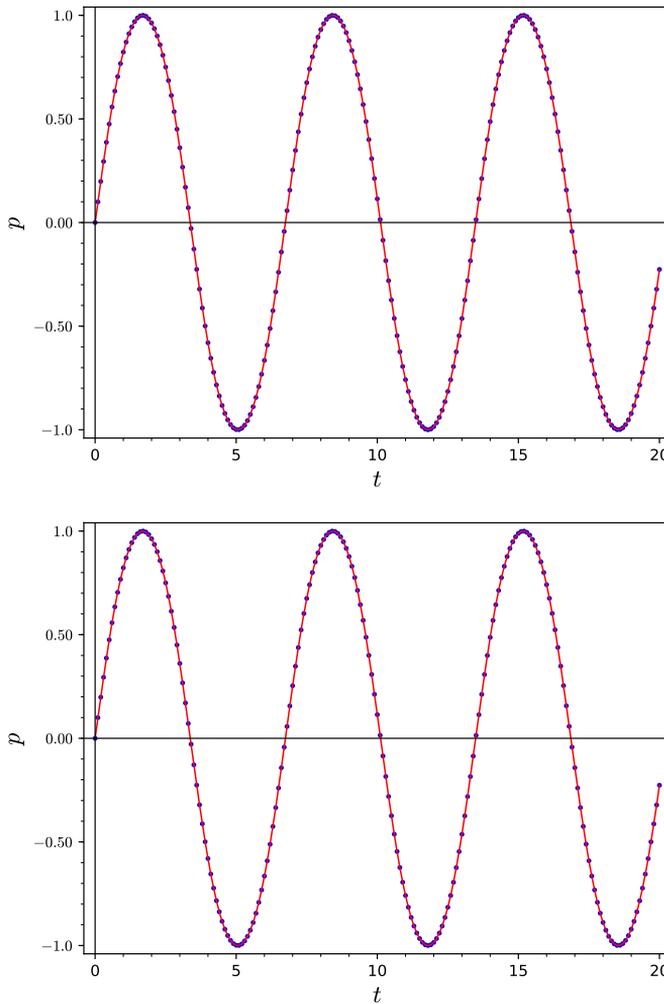


Figure 6. Graph of  $p(t)$ , for the Calvo method with exact conservation of  $p^2 + q^2 = 1$ ,  $dt = 0.1$ (up one) and that of  $k^2 p^2 + r^2 = 1$  (down one)

The second integral  $k^2 p^2 + r^2 = 1$  is not exactly preserved, but its value fluctuates with a small amplitude  $10^{-7}$  (figure 7) by the Calvo method with exact conservation of the first invariant  $p^2 + q^2 = 1$ . The Calvo method with exact conservation of the second invariant  $k^2 p^2 + r^2 = 1$  shows that the first invariant  $p^2 + q^2 = 1$  grows with an error of no more than  $10^{-5}$  (figure 8). Thus, preserving one integral does not preserve the other.

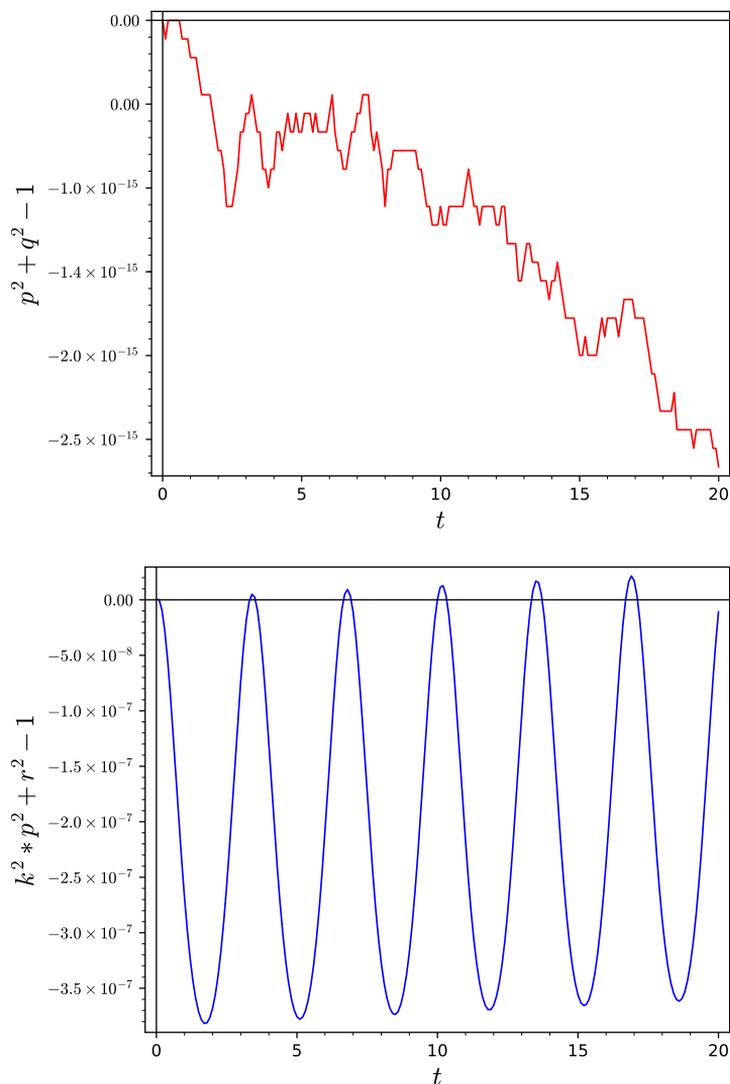


Figure 7. First and second integrals for the Calvo method with exact conservation of  $p^2 + q^2 = 1$

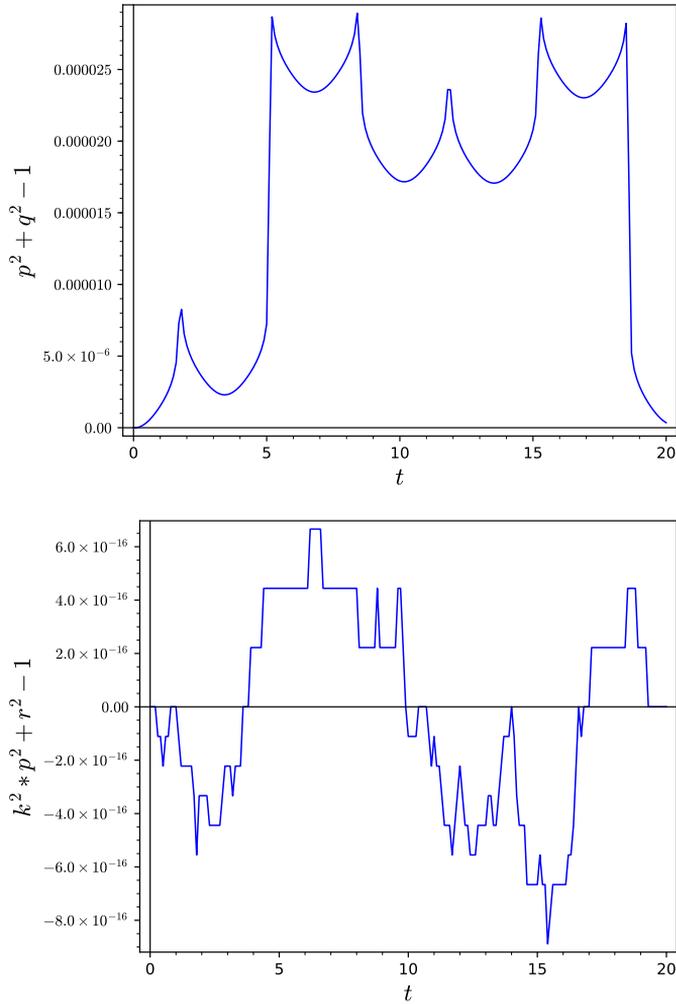


Figure 8. First and second integrals for the Calvo method with exact conservation of  $k^2 p^2 + r^2 = 1$

### 4. Scheme for preserving two invariants

Theoretically, the Calvo method allows to construct schemes that preserve several invariants. We take two pairs of RK methods defined by the following two extended Butcher tables:

0	0	0	0	0	and	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0		$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0		$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0		0	1	0	1	0
$\phi$	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$		$\phi$	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$
$\phi_1$	1	0	0	0		$\phi_2$	0	1	0	0

The first embedded RK method corresponding to the first table is constructed by combining the standard RK4 method, which has order 4, with the Euler method, which has order 1. The second embedded RK method corresponding to second table is constructed by combining the standard RK4 method, which has order 4, with a second order explicit RK scheme.

We are trying to find an explicit RK scheme of the type:

$$x_{\alpha\beta}^{n+1} = \phi(x^n) - \alpha_n(\phi(x^n) - \phi_1(x^n)) - \beta_n(\phi(x^n) - \phi_2(x^n)), \quad (17)$$

where  $\alpha_n, \beta_n \in \mathbb{R}$  are two scalar parameters, which can be determined by using the conservation laws, i.e.

$$I_i(x_{\alpha\beta}^{n+1}) = I_i(x^n), \quad i = 1, 2, \quad (18)$$

where  $I_1$  and  $I_2$  are two invariants of system (1).

By definition, we have

$$\phi(x^n) - \phi_1(x^n) = \left( -\frac{5}{6}f(k_1) + \frac{1}{3}f(k_2) + \frac{1}{3}f(k_3) + \frac{1}{6}f(k_4) \right) \Delta t$$

and

$$\phi(x^n) - \phi_2(x^n) = \left( \frac{1}{6}k_1 - \frac{2}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 \right) \Delta t.$$

Thus, (18) gives us a system of two equations with respect to two unknowns  $\alpha_n$  and  $\beta_n$ . In numerical experiment, we choose the parameters  $\alpha_n, \beta_n$  as real numbers close to the value 0.

In this way we have a system of algebraic equations for calculating parameters and we must solve this system at each step. Thus, we lose the main advantage of the exact methods. Sage has numerous tools for applying operations on the field of ideals. In our numerical experiments we use the elimination technique based on Gröbner basis [14] to solve (18). Namely, in each step  $n$ , after constructing multivariate polynomial ideal in variables  $\alpha_n, \beta_n$  generated by (18), we use Sage built-in function `elimination_ideal` to obtain an univariate equation in variable  $\beta_n$ . The function `roots` over ring  $\mathbb{R}$  is used to solve this equation. Substituting one of the value of  $\beta_n$  which is close to 0 to (18), the value of another parameter  $\alpha_n$  can be obtained by using the function `roots` again.

Consider elliptic function test.

We construct the function `phi_solve_1()` to implement the routine described above.

We implement the described scheme in Sage as the function `phi_solve_1()`. Function `phi_solve_1()` returns the numerical points  $(0, x^0), (\Delta t, x^1), \dots$  with the parameters:

- `list_of_integral` is two invariants that are required to be conserved;
- `F` is the right sides of the system (1);
- `ics` is the initial condition;
- `dt` is the step size, `T` is the end point of time  $t$ .

Let us take the elliptic function as example.

```

sage: var('p,q,r,t')
sage: k=1/2
sage: s=4
sage: F=[r*q,-p*r,-k^2*p*q]
sage: list_of_integral=[p^2+q^2,k^2*p^2+r^2]
sage: ics=[p==0,q==1,r==1]
sage: L=phi_solve_1(list_of_integral=list_of_integral, F=F,
  ↪ ics=ics, dt=0.1, T=20)
sage: max([abs(k^2*p^2+r^2-1) for [t,p,q,r] in L])
sage: G4=line([[t,k^2*p^2+r^2-1] for [t,p,q,r] in
  ↪ L],axes_labels=['$t$', '$k^2*p^2+r^2-1$'],tick_formatter=[None,RR(10e-
  ↪ 20).n(digits=1)],frame=true)
sage: max([abs(p^2+q^2-1) for [t,p,q,r] in L])
sage: G5=line([[t,p^2+q^2-1] for [t,p,q,r] in
  ↪ L],axes_labels=['$t$', '$p^2+q^2-1$'],tick_formatter=[None,RR(10e-
  ↪ 20).n(digits=1)],frame=true)

```

The Calvo method allows to preserve both invariants and thus significantly surpasses the other methods presented in the previous sections. Figure 9 shows that the error  $k^2 p^2 + r^2 - 1$  remains constant in size at  $10^{-16}$ , while figure 10 shows the error  $p^2 + q^2 - 1$  remains constant in size at  $10^{-13}$ . We can say that these errors are due to the implementation of the calculation in solving equation (18) over the ring  $\mathbb{R}$  instead of the algebraic closed field  $\overline{\mathbb{Q}}$  in CAS Sage [15]. From this point of view, we can conclude that this method can be considered as a method that exactly preserves exactly both quadratic invariants in the elliptic function test.

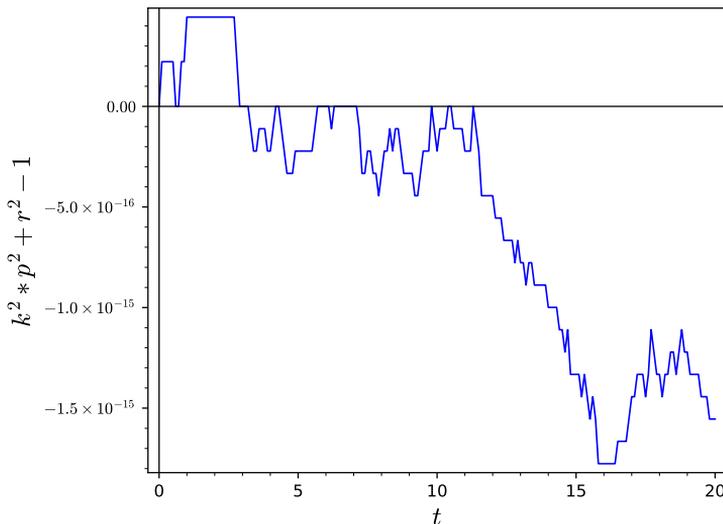


Figure 9. The first integral  $k^2 p^2 + r^2 - 1$  for  $T = 20$ ,  $dt = 0.1$

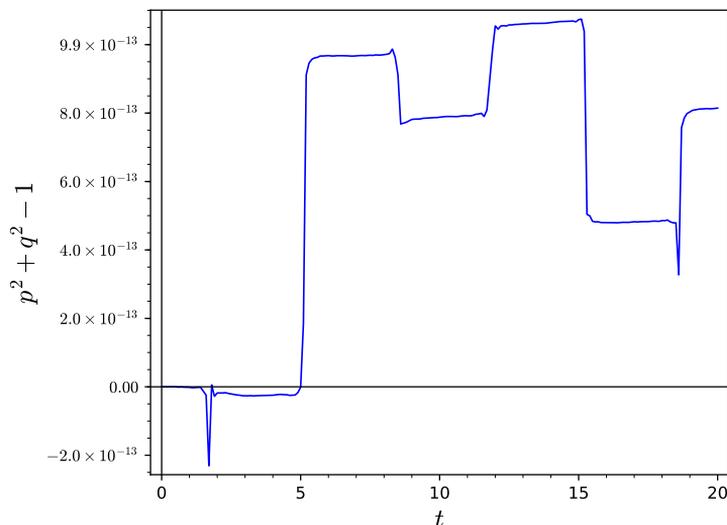


Figure 10. The second integral  $p^2 + q^2 - 1$  for  $T = 20$ ,  $dt = 0.1$

## 5. Conclusion

We have investigated several implementation of explicit RK schemes that preserve invariants. To preserve one invariant the Buono and Calvo methods require solving one algebraic equation with one unknown at each step. In our example, this equation is quadratic, so we can find its numerical solution without any difficulties. From the numerical experiments, we concluded that the exact conservation of one invariant is not an obstacle for changing of the other invariants. Thus, the conservation of multiple integrals requires a different approach.

The Calvo method which preserves several invariants has a drawback: it requires solving a system of algebraic equations with several unknown variables at each step, i.e. has the same drawback as standard implicit RK methods have. Fortunately, in our tests the system we obtained is much simpler than the system described by the midpoint scheme.

## Acknowledgments

The authors are thanks Prof. L. A. Sevastianov for his discuss and suggestions with this paper. This paper was supported by the foundation of KaiLi university (Z1602) and by the Russian Foundation for Basic Research (RFBR) according to the research project No 18-07-00567.

## References

- [1] A. Goriely, *Integrability and nonintegrability of dynamical systems*. Singapore; River Edge, NJ: World Scientific, 2001. DOI: 10.1142/3846.

- [2] J. C. Butcher, “On Runge–Kutta processes of high order,” *Journal of the Australian Mathematical Society*, vol. 4, no. 2, pp. 179–194, 1964. DOI: 10.1017/S1446788700023387.
- [3] E. Hairer, G. Wanner, and S. P. Nørsett, *Solving ordinary differential equations I: Nonstiff Problems*, 3rd ed. New York: Springer, 1993, vol. 1. DOI: 10.1007/978-3-540-78862-1.
- [4] K. Burrage and J. C. Butcher, “Stability criteria for implicit Runge–Kutta methods,” *SIAM Journal on Numerical Analysis*, vol. 16, no. 1, pp. 46–57, 1979. DOI: 10.1137/0716004.
- [5] G. Cooper, “Stability of Runge–Kutta methods for trajectory problems,” *IMA journal of numerical analysis*, vol. 7, no. 1, pp. 1–13, 1987. DOI: 10.1093/imanum/7.1.1.
- [6] E. Hairer, G. Wanner, and C. Lubich, *Geometric numerical integration. Structure-preserving algorithms for ordinary differential equations*. Berlin Heidelberg New York: Springer, 2000.
- [7] J.-M. Sanz-Serna and M.-P. Calvo, *Numerical hamiltonian problems*. Courier Dover Publications, 2018.
- [8] H. Zhang, X. Qian, J. Yan, and S. Song, “Highly efficient invariant-conserving explicit Runge–Kutta schemes for nonlinear Hamiltonian differential equations,” *Journal of Computational Physics*, p. 109598, 2020.
- [9] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *et al.*, *Numerical recipes*. Cambridge University Press Cambridge, 1989.
- [10] N. Del Buono and C. Mastroserio, “Explicit methods based on a class of four stage fourth order Runge–Kutta methods for preserving quadratic laws,” *Journal of Computational and Applied Mathematics*, vol. 140, pp. 231–243, 2002. DOI: 10.1016/S0377-0427(01)00398-3.
- [11] M. Calvo, D. Hernández-Abreu, J. I. Montijano, and L. Rández, “On the preservation of invariants by explicit Runge–Kutta methods,” *SIAM Journal on Scientific Computing*, vol. 28, no. 3, pp. 868–885, 2006. DOI: 10.1137/04061979X.
- [12] Y. Ying. (2020). “Package ex.sage,” [Online]. Available: <https://malykhmd.neocities.org>.
- [13] Y. S. Sikorsky, *Elements of the theory of elliptic functions with applications to mechanics [Elementy teoriiy ellipticheskikh funktsiy s prilozheniyami v mekhanike]*. Moscow–Leningrad: ONTI, 1936, In Russian.
- [14] D. Cox, J. Little, D. O’Shea, and M. Sweedler, “Ideals, varieties, and algorithms,” *American Mathematical Monthly*, vol. 101, no. 6, pp. 582–586, 1994.
- [15] The Sage Developers, *Sagemath, the Sage Mathematics Software System (Version 7.4)*, <https://www.sagemath.org>, 2016.

**For citation:**

Y. Ying, M. D. Malykh, On the realization of explicit Runge–Kutta schemes preserving quadratic invariants of dynamical systems, *Discrete and Continuous*

Models and Applied Computational Science 28 (4) (2020) 327–345. DOI: 10.22363/2658-4670-2020-28-4-327-345.

**Information about the authors:**

**Ying, Yu** — Assistant Professor of Department of Algebra and Geometry, Kaili University (e-mail: [yingy6165@gmail.com](mailto:yingy6165@gmail.com), phone: +7(495)9550927)

**Malykh, Mikhail D.** — doctor of Physical and Mathematical Sciences, Assistant Professor of Department of Applied Probability and Informatics of Peoples' Friendship University of Russia (RUDN University); (e-mail: [malykh\\_md@pfur.ru](mailto:malykh_md@pfur.ru), phone: +7(495)9550927, ORCID: <https://orcid.org/0000-0001-6541-6603>, ResearcherID: P-8123-2016, Scopus Author ID: 6602318510)

УДК 519.872:519.217

PACS 07.05.Tr, 02.60.Pn, 02.70.Bf

DOI: 10.22363/2658-4670-2020-28-4-327-345

## О реализации явных схем Рунге–Кутты с сохранением квадратичных инвариантов динамических систем

Юй Ин<sup>1</sup>, М. Д. Малых<sup>2</sup>

<sup>1</sup> *Университет Кайли  
Кайли, 556011, Китай*

<sup>2</sup> *Российский университет дружбы народов  
ул. Миклухо-Маклая, д. 6, Москва, 117198, Россия*

Авторами реализовано несколько явных схем Рунге–Кутты, которые сохраняют квадратичные инварианты автономных динамических систем в Sage. В статье представлен пакет `ex.sage` и результаты численных экспериментов.

В пакете функции `gk_solve`, `idt_solve` и `project_1` построены для случая, когда только один заданный квадратичный инвариант будет сохранён точно. Функция `phi_solve_1` позволяет сохранить одновременно два указанных квадратичных инварианта. Для решения уравнений относительно параметров, определяемых законом сохранения, использована методика исключения на основе базисов Грёбнера, реализованная в Sage. В качестве тестового примера представленного пакета используется эллиптический осциллятор. Эта динамическая система имеет два квадратичных инварианта. Представлены численные результаты сравнения стандартного явного метода Рунге–Кутты RK(4,4) с `gk_solve`. Кроме того, для функций `gk_solve` и `idt_solve`, сохраняющих только один инвариант, исследовано изменение второго квадратичного инварианта эллиптического осциллятора. В заключение рассматриваются недостатки использования этих схем.

**Ключевые слова:** явный метод Рунге–Кутты, квадратичный инвариант, динамическая система, Sage