Numerical methods for non conservative perturbations of conservative problems

M.P. Laburta[1], J.I. Montijano[1], L. Rández[1] and M. Calvo[1] *

September 4, 2014

[1] *IUMA – Departamento de Matemática Aplicada*
Universidad de Zaragoza. 50009-Zaragoza, Spain.
email: laburta,monti,randez,calvo@unizar.es

Abstract

In this paper the numerical integration of non conservative perturbations of differential systems that possess a first integral, as for example slowly dissipative Hamiltonian systems, is considered. Numerical methods that are able to reproduce appropriately the evolution of the first integral are proposed. These algorithms are based on a combination of standard numerical integration methods and certain projection techniques. Some conditions under which known conservative methods reproduce that desirable evolution in the invariant are analyzed. Finally, some numerical experiments in which we compare the behaviour of the new proposed methods, the averaged vector field method AVF proposed by Quispel and McLaren and standard RK methods of orders 3 and 5 are presented. The results confirm the theory and show a good qualitative and quantitative performance of the new projection methods.

AMS subject classification: 65L05, 65L06.

Keywords: initial value problems, numerical geometric integration, projection methods, dissipative systems, non conservative perturbed systems, explicit Runge–Kutta methods.

*This work was supported by D.G.I. project MTM2010-21630-C02.

1 Introduction

The dynamics of many mechanical systems as well as that of many other real life problems is often described by autonomous Hamiltonian systems. For these problems the Hamiltonian function, usually the energy of the system, is a constant of motion, and in the last decades there have been a lot of researches [2, 3, 4, 5, 6, 7, 10, 12, 13, 16, 18, 19, 20] to develop numerical integrators that are able to provide numerical approximations to the solution preserving the Hamiltonian, or any other invariant, along the integration.

In other cases, the energy of such system does not remain exactly constant due to the effect of external non-conservative forces, but its variation is small. Let us think for example about the effect of the friction in mechanical systems or else the air drag in the motion of artificial satellites. In the numerical simulation of these systems it is of great importance that the energy variation be properly reproduced by the numerical integrator [21, 22, 23], and this point is the main object of this paper.

We consider N dimensional autonomous systems of ordinary differential equations

$$z'(t) = f(z(t)), \tag{1}$$

with $f\colon D \subseteq \mathbb{R}^N \to \mathbb{R}^N$, having a scalar first integral $H(z)$, $H\colon \widehat{D} \subseteq \mathbb{R}^N \to \mathbb{R}$, $\widehat{D} \subseteq D$, that is, $\nabla H(z) \cdot f(z) = 0$ for all $z \in \widehat{D}$. We also consider a perturbed problem

$$y'(t) = f(y(t)) + \varepsilon g(y(t)), \tag{2}$$

where ε is a small positive parameter and $g(y)$ is a vector function of moderate size compared to the function $f(y)$ in a neighbourhood of the solution $y(t)$.

According to this, the variation of the scalar function $H(y)$ along the solution of (2) is governed by its derivative

$$\frac{\mathrm{d}}{\mathrm{d}t} H(y(t)) = \nabla H(y(t)) \cdot \big[f(y(t)) + \varepsilon g(y(t)) \big] = \varepsilon \nabla H(y(t)) \cdot g(y(t)). \tag{3}$$

We will assume that the function $\alpha(y) := \nabla H(y) \cdot g(y)$ is bounded by a positive function $u(t)$ in a neighbourhood of the solution $y(t)$, that is, for each t

$$|\alpha(y)| \leq u(t), \text{ for all } y \in \mathcal{B}(y(t), \delta), \tag{4}$$

where $\mathcal{B}(y(t), \delta) = \{y \in \mathbb{R}^N | \ \|y - y(t)\| \leq \delta\}$.

The above condition implies that

$$|H(y(t)) - H(y(t_0))| \leq \varepsilon \int_{t_0}^{t} u(s)\mathrm{d}s = \varepsilon v(t) \tag{5}$$

so $H(y(t))$ is not constant but its variation is small whenever ε is small. Note that since the solution $y(t)$ depends on ε, the function $\alpha(y(t))$ also depends

on ε. In the assumption (4) we are implicitly imposing that, for each t, $\alpha(y(t))$ is bounded when $\varepsilon \to 0$.

It is natural to ask under which conditions a numerical integrator provides approximations y_n satisfying a similar condition on the evolution of the function H, that is

$$|H(y_n) - H(y_0)| \leq \varepsilon v_h(t_n) \tag{6}$$

with $v_h(t)$ a positive function close to $v(t)$.

The above property is of great importance for example when we are interested in obtaining the time $t = t^*$ for which the first integral, let us say the energy, of the system attains a certain level $H = H^*$, that is, $H(y(t^*)) = H^*$. If the system is solved numerically, and we obtain an approximated solution $y_h(t)$, we will search for the time $\hat{t}$ such that $H(y_h(\hat{t})) = H^*$. The error $t^* - \hat{t}$ can be much smaller if the method satisfies condition (6). Note that by the mean value theorem

$$\begin{aligned} 0 = H(y(t^*)) - H(y_h(\hat{t})) =& H(y(t^*)) - H(y(\hat{t})) + H(y(\hat{t})) - H(y_h(\hat{t})) \\ =& \varepsilon\alpha(y(\xi))(t^* - \hat{t}) + H(y(\hat{t})) - H(y_h(\hat{t})) \end{aligned}$$

for some ξ between $\hat{t}$ and t^*. Therefore

$$t^* - \hat{t} = \frac{H(y_h(\hat{t})) - H(y(\hat{t}))}{\varepsilon\alpha(y(\xi))} \tag{7}$$

which means that the error $t^* - \hat{t}$ can be large, for small ε, unless $H(y_h(\hat{t})) - H(y(\hat{t}))$ is small. Since for a method with order p this quantity is of order $\mathcal{O}(h^p)$, this can be accomplished by taking the step size h small enough to compensate the ε factor in the denominator. However, if condition (6) is satisfied, $H(y_h(\hat{t})) - H(y(\hat{t}))$ is also of the order of $\mathcal{O}(\varepsilon)$ and the error $t^* - \hat{t}$ will be of the order of $\mathcal{O}(h^p)$ independently of the value of ε, assumed that $|\alpha(y(t))|$ is lower bounded by some non small value.

Observe that (6) is not satisfied for arbitrary small ϵ by all numerical methods. Let us suppose, for example, the scalar linear complex equation

$$y' = i\omega y - \varepsilon y,$$

whose solution is $y(t) = y_0 e^{(-\varepsilon + i\omega)t}$. For $\varepsilon = 0$, the problem has the first integral $H(y(t)) = H(y_0)$ where $H(y) = \bar{y}y = |y|^2$, whereas for any $\varepsilon > 0$, $H(y(t)) = e^{-2\varepsilon t}H(y_0)$ is a slowly decreasing function. Clearly $g(y) = -y$ and $\alpha(y) = -2|y|^2$, therefore, $|\alpha(y(t))| = 2|y_0|^2 e^{-2\varepsilon t}$ is not small unless t is larger than $1/\varepsilon$. Note that this equation satisfies (4) with $u(t) \leq 2(|y_0| + \delta)^2$ and therefore $v(t) \leq 2(|y_0| + \delta)^2 t$. This means that, for $t \leq 1/\varepsilon$, the first integral varies as εt.

If we integrate this problem with explicit Euler's method with step size h, we obtain $y_1 = (1 + hi\omega - h\varepsilon)y_0$. Consequently

$$H(y_1) - H(y_0) = (h^2\omega^2 - 2h\varepsilon + h^2\varepsilon^2)H(y_0),$$

which does not tend to zero when $\varepsilon \to 0$ and therefore this method does not satisfy (6). Moreover $H(y_1) - H(y(h)) = H(y_0)((h\omega)^2 - (h\varepsilon)^2 + \mathcal{O}(h^3\varepsilon^3))$ does not tend to zero when $\varepsilon \to 0$.

In general, for any method of order p applied to this problem we will get $H(y_n) - H(y_0) = \mathcal{O}(h^p) + \mathcal{O}(h\varepsilon)$, and only for methods with special properties the term $\mathcal{O}(h^p)$ will vanish. For example, if we integrate the problem with the implicit midpoint rule, we obtain $y_1 = y_0(1 + hi\omega/2 - h\varepsilon/2)/(1 - hi\omega/2 + h\varepsilon/2)$, giving

$$H(y_1) - H(y_0) = \frac{-2h\varepsilon}{(1 + h\varepsilon/2)^2 + (h\omega/2)^2} H(y_0),$$

which tends to zero when $\varepsilon \to 0$ independently of the step size h. It will be shown in section 3 that (6) is satisfied for this method and moreover $H(y_n) - H(y(t_n)) = \mathcal{O}(\varepsilon h^2)$ so that the error in (7) can be small for reasonable values of h independently of the value of ε.

For simplicity, we have supposed that the perturbation term $g(y)$ is autonomous. However, with appropriate modifications, the results are also valid if a term $g(y, t)$ is considered instead, whenever the bound (4) is satisfied.

The paper is organized as follows: In Section 2 we present the new methods, based on certain projection techniques, and prove that property (6) is satisfied, seeing also that this condition has some benefits in the error of the function H. In Section 3 we show that (6) holds for other conservative methods under some non restrictive conditions on the method. Finally, in Section 4 we present some numerical experiments confirming the results in previous sections.

2 Projection methods for perturbed conservative systems

Projection methods have already been considered as a proper tool to obtain numerical approximations that preserve the invariants when considering conservative problems. These techniques can also be useful for non conservative problems that are perturbations of conservative ones. They have already been proved to be useful for example to preserve the monotonicity of Lyapunov functions [8, 15], or to preserve dissipation for dissipative systems [11], and we are proposing here the same type of algorithms based on projection to solve problems that possess a function $H(y)$ that has a slow variation in a neighbourhood of the solution $y(t)$. The main idea is to obtain first an approximation H_{n+1} to the value of the function at the next grid point $H(y(t_{n+1}))$ and then to project the numerical approximation onto the manifold $H(y) = H_{n+1}$. Thus, we propose the following Projection Runge–Kutta (PRK) algorithm:

Algorithm PRK

- First we compute a numerical approximation $\widetilde{y}_{n+1}$ to $y(t_{n+1})$ by means of a standard (non preserving) method, such as a Runge–Kutta scheme. We also compute a (piecewise) continuous extension $y_h(t)$ provided by it.

- We take a quadrature formula $(\bar{b}_i, \bar{c}_i)$, $i = 1, \ldots, k$ in $[0, 1]$ with coefficients $\bar{b}_i$ positive and nodes in $[0, 1]$. For example, a Gaussian formula.

- We compute an approximation H_{n+1} to the value $H(y(t_{n+1}))$ by integrating (3):

$$\begin{aligned} H_{n+1} = & \quad H(y_n) + h \sum_{i=1}^{k} \bar{b}_i \nabla H(y_h(t_n + \bar{c}_i h)) \cdot (f + \varepsilon g)(y_h(t_n + \bar{c}_i h)) \\ = & \quad H(y_n) + \varepsilon h \sum_{i=1}^{k} \bar{b}_i \alpha(y_h(t_n + \bar{c}_i h)) \end{aligned} \tag{8}$$

- We compute the new approximation y_{n+1} by projecting $\tilde{y}_{n+1}$ onto H_{n+1} along a direction of projection w_n:

$$y_{n+1} = \tilde{y}_{n+1} + \lambda_n w_n,$$

  where λ_n is a real parameter which, once determined the direction vector w_n, will be calculated so that y_{n+1} satisfies

$$H(y_{n+1}) = H(\widetilde{y}_{n+1} + \lambda_n w_n) = H_{n+1}. \tag{9}$$

  The direction vector must satisfy certain conditions as shown in [6]. In [9], it has been studied the way w_n can be chosen to maximize the dispersion order of the projected approximation.

Remark 2.1. *The PRK algorithm can be applied to any differential system $y'(t) = w(y(t))$ for which there exists a function $H(y)$ that varies slowly in a neighbourhood of the solution, that is, $|\nabla H(y) \cdot w(y)| \leq \varepsilon^*$ is small. Note that we can always decompose $w(y) = f(y) + \tilde{g}(y)$ with $f(y)$ orthogonal to $\nabla H(y)$, and consequently $|\nabla H(y) \cdot w(y)| = |\nabla H(y) \cdot \tilde{g}(y)| \leq \varepsilon^*$ will be small. Then, we can write $y' = f(y) + \varepsilon g(y)$ with $g(y) = \tilde{g}(y)/\varepsilon^*$. For $\varepsilon = \varepsilon^*$ we recover the original system.*

We will see next that, in particular, the proposed algorithm satisfies the condition (6).

Theorem 2.1. *i) The numerical solution provided by the algorithm PRK satisfies*

$$|H(y_{n+1}) - H(y_0)| \leq \varepsilon v_h(t_{n+1})$$

with

$$v_h(t_{n+1}) = h \sum_{j=0}^{n} \sum_{i=1}^{k} \bar{b}_i u(t_j + \bar{c}_i h), \tag{10}$$

$u(t)$ being the bound in (4).

ii) If the continuous method used in the algorithm PRK has order $\geq p-1$, the quadrature formula has order $\geq p$ and $\alpha(y) := \nabla H(y) \cdot g(y)$ is a Lipschitz continuous function, then the projected solution satisfies

$$H(y_{n+1}) - H(y(t_{n+1}; t_n, y_n)) = \mathcal{O}(\varepsilon h^{p+1}),$$

where $y(t; t_n, y_n)$ represents the local solution of (2) *that satisfies $y(t_n) = y_n$.*

Proof. From (8) and (9) it is clear that

$$H(y_{n+1}) - H(y_0) = h\varepsilon \sum_{j=0}^{n} \sum_{i=1}^{k} \bar{b}_i \alpha(y_h(t_j + \bar{c}_i h))$$

Assuming that the step size is small enough to ensure that the numerical approximation $y_h(t_j + \bar{c}_i h)$ belongs to the neighbourhood $\mathcal{B}(y(t_j + \bar{c}_i h), \delta)$, part i) has been proved.

On the other hand, it is clear from (3) that

$$H(y(t+h)) = H(y(t)) + \varepsilon \int_t^{t+h} \alpha(y(s)) \mathrm{d}s.$$

By construction

$$\begin{aligned} H(y_{n+1}) = & \quad H(y_n) + \varepsilon h \sum_{i=1}^{k} \bar{b}_i \alpha(y_h(t_n + \bar{c}_i h)) \\ = & \quad H(y_n) + \varepsilon \int_{t_n}^{t_n+h} \alpha(y(s; t_n, y_n)) \mathrm{d}s \\ + & \quad \varepsilon \left(- \int_{t_n}^{t_n+h} \alpha(y(s; t_n, y_n)) \mathrm{d}s + \int_{t_n}^{t_n+h} \alpha(y_h(s)) \mathrm{d}s \right) \\ + & \quad \varepsilon \left(- \int_{t_n}^{t_n+h} \alpha(y_h(s)) \mathrm{d}s + h \sum_{i=1}^{k} \bar{b}_i \alpha(y_h(t_n + \bar{c}_i h)) \right). \end{aligned}$$

Item ii) follows immediately from the order properties of the continuous method and the quadrature formula and from the Lipschitz condition for $\alpha(y)$. □

Computational cost of the algorithm

The PRK algorithm amounts some additional computational cost at each step to compute the integral and to project the solution onto the manifold $H(y) = H_{n+1}$.

- A continuous extension $y_h(t)$ (at least of order $p-1$) at k points $t_n + \bar{c}_i h$ must be evaluated. This requires at most one additional evaluation of the vector field $f + \varepsilon g$ for $p \leq 5$.

- The function $\alpha(y)$ must be evaluated at k points. Note that if we use a Gaussian quadrature formula, it is enough to take $k \simeq p/2$ to get the appropriate order.

- The equation (9) must be solved for λ_n. Since the equation is scalar, a simple method like the secant one can be used, and since its solution is close to zero, the convergence is usually extremely fast. In practice, the projection process amounts an average of two evaluations of the function $H(y)$ per step.

In conclusion, if the computational effort required to evaluate the functions $\alpha(y)$ and $H(y)$, which have vector argument but scalar result, is smaller than the one for the vector field, which has vector argument and result, the additional cost will be small. If the evaluations of those two functions have a cost greater than the one for the vector field, the additional cost can be important. Nevertheless, if we are interested for example in computing, with an error Tol, the time for which the first integral attains certain level H^*, with the standard method we should integrate the problem with an error tolerance ε Tol. If the method used has order p, this would mean that we must take about $(1/\varepsilon)^{1/p}$ times the number of steps, and therefore we are increasing the computational cost by this factor. For small ε this can be large and the PRK algorithm will be advantageous even if it requires certain additional cost.

In Theorem 2.1, which is the main result of this section, we have seen that the PRK algorithm satisfies the property (6). Next, we will see that, under some assumptions on the function $u(t)$, the function $v_h(t)$ is close to the continuous one $v(t)$.

Theorem 2.2. *If the positive function $u(t)$ in (4) is a polynomial of degree $\leq q$ and the PRK algorithm uses a quadrature formula with degree of precision at least q, the projected solution provided by the PRK algorithm satisfies*

$$|H(y_{n+1}) - H(y_0)| \leq \varepsilon v(t_{n+1}).$$

Proof. It is immediate by the definition of degree of precision. □

Lemma 2.1. *Let $u(t)$ be a positive $\mathcal{C}^1$ function such that $|u'(t)| \leq \lambda u(t)$ for all $t \geq t_0$ and some positive λ. Then, for all $h < 2/\lambda$*

$$h \max_{\sigma\in[t,t+h]} u(\sigma) \leq \frac{1}{1-h\lambda/2} \int_t^{t+h} u(s)\,ds, \quad \forall t \geq t_0.$$

Proof. Let $\hat{t}$ be such that $\max_{\sigma\in[t,t+h]} u(\sigma) = u(\hat{t})$. Since $u(t)$ is $\mathcal{C}^1$, for $s \in [t, t+h]$ we have

$$\max_{\sigma\in[t,t+h]} u(\sigma) - u(s) = u'(\xi)(\hat{t}-s) \leq \lambda \max_{\sigma\in[t,t+h]} u(\sigma)|\hat{t}-s|$$

where $\xi \in (t, t+h)$. Then

$$\begin{aligned} h \max_{\sigma\in[t,t+h]} u(\sigma) - \int_t^{t+h} u(s)\mathrm{d}s = & \int_t^{t+h} \big(\max_{\sigma\in[t,t+h]} u(\sigma) - u(s)\big)\mathrm{d}s \\ \leq & \ \lambda \max_{\sigma\in[t,t+h]} u(\sigma) \int_t^{t+h} |\hat{t}-s|\mathrm{d}s \\ \leq & \ \frac{h^2\lambda}{2} \max_{\sigma\in[t,t+h]} u(\sigma) \end{aligned}$$

and consequently,

$$(1-h\lambda/2)h \max_{\sigma\in[t,t+h]} u(\sigma) \leq \int_t^{t+h} u(s)\mathrm{d}s.$$

□

Theorem 2.3. *If the positive function $u(t)$ in (4) is $\mathcal{C}^1$ and satisfies $|u'(t)| \leq \lambda u(t)$ for all $t \geq t_0$ and some positive λ, then for all $h < 2/\lambda$ the projected solution provided by the PRK algorithm satisfies*

$$|H(y_{n+1}) - H(y_0)| \leq \varepsilon v_h(t_{n+1}),$$

where $v_h(t_{n+1})$ given in (10) satisfies

$$v_h(t_{n+1}) \leq \frac{1}{1-h\lambda/2} \int_{t_0}^{t_{n+1}} u(s)\,ds = \frac{1}{1-h\lambda/2} v(t_{n+1}).$$

Proof. From the value of $v_h(t_{n+1})$ in (10)

$$v_h(t_{n+1}) \leq h \sum_{j=0}^{n} \sum_{i=1}^{k} \bar{b}_i \max_{\sigma\in[t_j,t_j+h]} u(\sigma) = h \sum_{j=0}^{n} \max_{\sigma\in[t_j,t_j+h]} u(\sigma)$$

and using Lemma 2.1, for all $h < 2/\lambda$ we have

$$v_h(t_{n+1}) \leq \sum_{j=0}^{n} \frac{1}{1-h\lambda/2} \int_{t_j}^{t_j+h} u(s)\mathrm{d}s = \frac{1}{1-h\lambda/2} \int_{t_0}^{t_{n+1}} u(s)\mathrm{d}s.$$

□

Theorem 2.3 covers the case of functions $u(t)$ whose variation (increasing or decreasing) is at most exponential. The next theorem intends to cover the case of functions that decrease even faster than an exponential.

Theorem 2.4. *If the function $u(t)$ in (4) satisfies $u(t) \leq 1/(t+\beta)^\gamma$ for some $\beta > -t_0$ and $\gamma > 1$ then the function $v_h(t_n)$ given in (10) is bounded for all n.*

Proof. First, let us note that the function $r(t) = 1/(t+\beta)^\gamma$ has a bounded integral,

$$\int_{t_0}^{\infty} r(s)\mathrm{d}s < \infty.$$

On the other hand, $r(t)$ satisfies the conditions in Lemma 2.1 with $\lambda = \gamma/(\beta + t_0)$, which implies that

$$\begin{aligned} v_h(t_n) \leq & \; h \sum_{j=0}^{n-1} \max_{\sigma\in[t_j,t_j+h]} u(\sigma) \leq h \sum_{j=0}^{n-1} \max_{\sigma\in[t_j,t_j+h]} r(\sigma) \\ \leq & \; \frac{1}{1-h\lambda/2} \int_{t_0}^{t_n} r(s)\mathrm{d}s < \frac{1}{1-h\lambda/2} \int_{t_0}^{\infty} r(s)\mathrm{d}s < \infty. \end{aligned}$$

□

3 Conservative methods for perturbed conservative systems

Let us consider now conservative methods, that is, methods for which $H(z_n) = H(z_0)$ when they integrate conservative problems (1). We will see that, under some natural assumptions, they present also a good behaviour when they are applied to non-conservative perturbations of conservative systems. Let us consider again a differential system (2) and denote by $\phi_f(y,h)$ the flow map of a numerical method applied to a differential system with vector field f, starting from the point y, advancing a step h. If the method is conservative for the system (1), we will obtain a numerical approximation $z_1 = \phi_f(y_0,h)$ that satisfies $H(z_1) = H(y_0)$.

Here, we will assume that for each $t \geq 0$, $\nabla H(y)$ and $g(y)$ are bounded by positive functions in the neighbourhood $\mathcal{B}(y(t), \delta)$, i.e.

$$\|\nabla H(y)\| \leq \mu(t), \qquad \|g(y)\| \leq \eta(t), \text{ for all } y \in \mathcal{B}(y(t), \delta). \tag{11}$$

In addition to this, we define the functions

$$M_\delta(y) = \max_{\hat{y} \in \mathcal{B}(y,\delta)} \|\nabla H(\hat{y})\|, \qquad K_\delta(y) = \max_{\hat{y} \in \mathcal{B}(y,\delta)} \|g(\hat{y})\|.$$

Recall that by the nonlinear variation of constants formula, the exact solutions of (2) and (1) with the same initial condition satisfy

$$\|y(t) - z(t)\| \leq \varepsilon t K(y_0, t)$$

for all $t \in [0, t_{\max}]$ with some bounded function K.

In the next theorem we will show that under a similar assumption for the numerical method, bounds on the variations of the energy can be established.

Theorem 3.1. *Let $\phi_f(y, h)$ be the flow map of a conservative method on the vector field f. If for all y and small enough h*

$$\|\phi_{f+\varepsilon g}(y, h) - \phi_f(y, h)\| \leq L K_\delta(y) h \varepsilon \tag{12}$$

for some δ depending on h and some $L = L(y)$, then

$$|H(y_{n+1}) - H(y_n)| \leq \varepsilon h L M_\delta(y_n) K_\delta(y_n). \tag{13}$$

In addition, if the method has order p, then

$$H(y_{n+1}) - H(y(t_{n+1}; t_n, y_n)) = \mathcal{O}(\varepsilon h^{p+1}).$$

Proof. Since the method is conservative on f, $z_{n+1} = \phi_f(y_n, h)$ satisfies $H(z_{n+1}) = H(y_n)$. On the other hand, assuming that the step size is small enough so that y_{n+1} and z_{n+1} belong to the neighbourhood $\mathcal{B}(y_n, \delta)$

$$H(y_{n+1}) - H(z_{n+1}) = \int_0^1 \nabla H(s y_{n+1} + (1-s) z_{n+1}) \mathrm{d}s \cdot (y_{n+1} - z_{n+1}),$$

and by using (13)

$$\|y_{n+1} - z_{n+1}\| = \|\phi_{f+\varepsilon g}(y_n, h) - \phi_f(y_n, h)\| \leq L h \varepsilon K_\delta(y_n).$$

Further

$$|H(y_{n+1}) - H(y(t_n + h; t_n, y_n))| \leq |H(y_{n+1}) - H(y_n)| + |H(y_n) - H(y(t_n + h; t_n, y_n))|$$

but since

$$\begin{aligned}|H(y_n) - H(y(t_n + h; t_n, y_n))| =& \varepsilon \int_{t_n}^{t_{n+1}} |\nabla H(y(s; t_n, y_n)) \cdot g(y(s; t_n, y_n))| \, \mathrm{d}s \\ \leq& \varepsilon h M_\delta(y_n) K_\delta(y_n) = \mathcal{O}(h\varepsilon)\end{aligned}$$

then

$$|H(y_{n+1}) - H(y(t_n + h; t_n, y_n))| = \mathcal{O}(h\varepsilon).$$

On the other side, by the order p of the method, it is also $\mathcal{O}(h^{p+1})$. This implies that it is $\mathcal{O}(\varepsilon h^{p+1})$. □

Theorem 3.2. *Let $\phi_f(y, h)$ a conservative method on the vector field f satisfying (12) applied to a system* (2) *satisfying* (11)*. Then*

$$|H(y_{n+1}) - H(y_0)| \leq \varepsilon v_h(t_{n+1})$$

with

$$v_h(t_{n+1}) = hL \sum_{j=0}^{n} \mu(t_j)\eta(t_j). \tag{14}$$

Proof. First, using Theorem 3.1

$$\begin{aligned}|H(y_{n+1}) - H(y_0)| \leq& |H(y_{n+1}) - H(y_n)| + |H(y_n) - H(y_0)| \\ \leq& \varepsilon h L M_\delta(y_n) K_\delta(y_n) + |H(y_n) - H(y_0)| \\ \leq& \varepsilon h L \sum_{j=0}^{n} M_\delta(y_j) K_\delta(y_j).\end{aligned}$$

Taking stepsizes and δ in such a way that $\mathcal{B}(y_j, \delta) \subseteq \mathcal{B}(y(t_j), \delta)$,

$$|H(y_{n+1}) - H(y_0)| \leq \varepsilon h L \sum_{j=0}^{n} \mu(t_j)\eta(t_j).$$

□

By following analogous reasonings to the ones developed for Theorems 2.3 and 2.4, the next results can be proved.

Theorem 3.3. *Let $\phi_f(y, h)$ a conservative method on the vector field f satisfying (12). If the function $\mu(t)\eta(t)$ is $\mathcal{C}^1$ and satisfies $|(\mu(t)\eta(t))'| \leq \lambda \mu(t)\eta(t)$ for all $t \geq t_0$, and some positive λ, then for all $h < 2/\lambda$ the numerical solution satisfies*

$$|H(y_{n+1}) - H(y_0)| \leq \varepsilon v_h(t_{n+1}),$$

where $v_h(t_{n+1})$ given in (14) *satisfies*

$$v_h(t_{n+1}) \leq \frac{L}{1 - h\lambda/2} \int_{t_0}^{t_{n+1}} \mu(s)\eta(s)\, ds.$$

Theorem 3.4. *Let $\phi_f(y,h)$ a conservative method on the vector field f satisfying (12). If the function $\mu(t)\eta(t) \leq 1/(t+\beta)^\gamma$ for some $\beta > -t_0$ and $\gamma > 1$ then the function $v_h(t_n)$ given in* (14) *is bounded for all n.*

Next, we will see that condition (12) is a natural condition that is satisfied by most relevant one step methods.

Proposition 3.1. *For small enough step size h, a Runge–Kutta method satisfies condition (12) for any differential system (1), (2) with f Lipschitz continuous.*

Proof. In compact form, a Runge–Kutta applied to both differential systems reads

$$Z = y_0 \otimes e + h(A \otimes I)F(Z)$$
$$z_1 = \phi_f(y_0,h) = y_0 + h(b \otimes I)^T F(Z)$$

and

$$Y = y_0 \otimes e + h(A \otimes I)(F(Y) + \varepsilon G(Y))$$
$$y_1 = \phi_{f+\varepsilon g}(y_0,h) = y_0 + h(b \otimes I)^T (F(Y) + \varepsilon G(Y)).$$

Therefore

$$\|Y-Z\| \leq h\|A\|\|F(Y)-F(Z)\|+h\varepsilon\|G(Y)\|\|A\| \leq hl\|A\|\|Y-Z\|+h\varepsilon\|G(Y)\|\|A\|$$

and

$$\|Y - Z\| \leq \frac{\|A\|}{1-hl\|A\|} h\varepsilon\|G(Y)\|$$

where l is the Lipschitz constant of f. Consequently, for h small enough, $Y_i \in \mathcal{B}(y_0,\delta)$ and

$$\|y_1 - z_1\| \leq \|b\| \frac{hl\|A\|}{1-hl\|A\|} h\varepsilon K_\delta(y_0) + h\varepsilon\|b\|K_\delta(y_0)$$

and condition (12) is satisfied with $L = \dfrac{\|b\|}{1-hl\|A\|}$. □

Proposition 3.2. *For small enough step size h, the averaged vector field method (AVF) [26],[10] satisfies condition (12) for any differential system (1), (2) with f Lipschitz continuous.*

Proof. The AVF method applied to both differential systems reads

$$z_1 = \phi_f(y_0,h) = y_0 + h\int_0^1 f(sz_1 + (1-s)y_0)\mathrm{d}s$$

and

$$y_1 = \phi_{f+\varepsilon g}(y_0,h) = y_0 + h\int_0^1 \big[f(sy_1 + (1-s)y_0) + \varepsilon g(sy_1 + (1-s)y_0)\big]\mathrm{d}s.$$

Therefore, for h such that $y_1 \in \mathcal{B}(y_0, \delta)$

$$\|y_1 - z_1\| \leq hl\|y_1 - z_1\| \int_0^1 s \, \mathrm{d}s + h\varepsilon K_\delta(y_0)$$

and

$$\|y_1 - z_1\| \leq h\varepsilon \frac{2}{2 - hl} K_\delta(y_0).$$

□

4 Numerical experiments

In this section we are going to present some numerical results to show the behaviour of the methods studied in the previous sections applied to some perturbed non-conservative problems. They refer to both qualitative and quantitative aspects of the numerical solution. All the figures, except the phase diagrams, have been represented in a log-log scale.

All the numerical methods have been implemented with variable step size. They are the following:

- BS32 is the 4-stage explicit embedded RK pair of order 3(2) derived by Bogacki and Shampine in [1] and implemented in MATLAB [27]. The control of the step size has been done by estimating the local error with the difference between the 3rd and 2nd order numerical solutions provided by that pair [17, II.4].

- pBS32 represents the projection method obtained in Section 2 according to the Algorithm PRK. The 3rd-order Bogacki–Shampine method described before has been used as standard method, and we have computed the dense output in each step with the corresponding Hermite interpolant of order 3, which does not need additional function evaluations over the pair BS32. This projection method pBS32 also uses the quadrature Gaussian formula with 2 nodes in $[0, 1]$. According to the theory developed in [9], we have chosen a simple way of getting the direction vector $w_n = \widehat{y}_{n+1} - \widetilde{y}_{n+1}$, where $\widetilde{y}_{n+1}$ is the numerical solution provided by the 3rd-order Bogacki–Shampine method, and $\widehat{y}_{n+1}$ represents the embedded RK with coefficients $\widehat{b}_2 = 0.33$, $\widehat{b}_3 = \frac{4}{9}\widehat{b}_2 + \frac{8}{27}$ and $\widehat{b}_1 = 1 - \widehat{b}_2 - \widehat{b}_3$.

- AVF denotes the "averaged vector field" method of order 2 derived by Quispel and McLaren in [26, Section 2]. When this method is applied to an autonomous differential system $y' = f(y)$, the equation that advances one step of size h from y_n is given by:

$$y_{n+1} = y_n + h \int_0^1 f((1 - \xi)y_n + \xi y_{n+1}) \, d\xi, \quad n = 0, 1, 2, \ldots. \tag{15}$$

It preserves energy for all canonical Hamiltonian vector fields. Estimations of the local error have been done by local extrapolation [17, II.4]. In the numerical experiments we have denoted by AVF(G4), AVF(G6) and AVF(G10), the method (15) that approximates its definite integral by using Gaussian quadrature formulas of orders 4, 6 and 10, respectively, in the interval $[0, 1]$. Functional iteration up to round off error has been used to find the numerical solutions of this implicit method.

- DP54 is the well known 7-stage explicit embedded RK pair of order 5(4) due to Dormand and Prince [14]. It is also implemented in MATLAB [27]. The local error has been estimated by subtracting the 5th and 4th order numerical solutions provided by this pair [17, II.4].

- pDP54 denotes the projection method obtained by following the Algorithm PRK, where the 5th order formula of DP54 has been taken as the basic method. The dense output is provided by the only fourth order interpolant that uses only the first six stages of that pair (see e.g. [25]), and we have considered the quadrature Gaussian formula with 3 nodes in $[0, 1]$. By following analogous ideas to those developed in [9] to obtain a direction vector w_n which gives rise to projected methods with low dispersion and dissipation errors, we have taken $w_n = \widehat{y}_{n+1} - \widetilde{y}_{n+1}$, where $\widetilde{y}_{n+1}$ is the numerical solution provided by the fifth-order method by Dormand and Prince, and $\widehat{y}_{n+1}$ represents the embedded RK with coefficients:

$$\begin{array}{lcllcl} \widehat{b}_1 &=& 0.1, & \widehat{b}_2 &=& 1, \\ \widehat{b}_3 &=& -0.768953928405587, & \widehat{b}_4 &=& 1.15647677385114, \\ \widehat{b}_5 &=& -0.767249955009483, & \widehat{b}_6 &=& 0.279727109563926. \end{array}$$

Our first test problem is Kepler's problem with atmospheric drag terms, given by:

$$\begin{array}{rcl} y_1'' &=& -\dfrac{y_1}{r^3} - \varepsilon \exp(-(r-0.5))\, y_1' \sqrt{(y_1')^2 + (y_2')^2} \\ y_2'' &=& -\dfrac{y_2}{r^3} - \varepsilon \exp(-(r-0.5))\, y_2' \sqrt{(y_1')^2 + (y_2')^2} \end{array} \tag{16}$$

where $r = \sqrt{y_1^2 + y_2^2}$. It is a simplified version of a model of the dynamics of an artificial satellite taking into account the effect of the air drag (see e.g. [24]).

For the numerical experiments we have taken $\varepsilon = 10^{-4}$, eccentricity $e = 0.7$, and initial conditions

$$y_1(0) = 1 - e, \quad y_2(0) = 0, \quad y_1'(0) = 0, \quad y_2'(0) = \sqrt{(1+e)/(1-e)}.$$

The integrations have been carried out for $t \in [0, 245]$.

As it is well known, the Hamiltonian for the Kepler problem is given by [16, section I.2]:

$$H(y, y') = -1/r + ((y_1')^2 + (y_2')^2)/2.$$

Then, for this problem $\alpha(y) = \nabla H(y)g(y) = -\exp(-r+0.5)((y_1')^2+(y_2')^2)^{3/2}$. It can be numerically verified that, e.g. for $t < 1000$, the solution satisfies $0.4 < ((y_1')^2 + (y_2')^2)^{1/2} < 2.4$, $0.25 < r < 1.7$ and $0.02 \leq |\alpha(y(t))| \leq 19$ for all $\varepsilon \in [0, 10^{-4}]$. The maximum values correspond precisely to the maximum value of the parameter $\varepsilon = 10^{-4}$. In fact, $|\alpha(y(t))|$ is an almost periodic function that takes values from 0.02 to 19 along each 2π-period. Therefore, the energy decreases in average almost linearly on t with slope about 1.7ε, as it is shown in Figure 1. The factor 1.7 is approximately the integral $-(1/2\pi)\int_t^{t+2\pi} \alpha(y(s))\mathrm{d}s$. On the other hand, $|\mathrm{d}\alpha(y(t))/\mathrm{d}t|/|\alpha(y(t))| < 3$ for all $t \leq 1000$, which means that $u(t)$ satisfies the conditions of Theorem 2.3 with $\lambda \simeq 3$.

The left plot in Figure 1 shows the evolution of the energy H with time t for BS32, pBS32, AVF(G4) and AVF(G6) for error tolerance $atol = rtol = 10^{-3}$. The best results correspond to the projection method pBS32, followed by the conservative method AVF, whereas the standard one BS32 shows a much greater deviation from the real evolution of the energy. Let us mention that, for this tolerance, AVF gives rise to slightly better results when its definite integral is approximated by the 6th-order than by the 4h-order Gaussian quadrature formula, but they do not improve if the order of the quadrature formula is increased. The right plot in Figure 1 displays the error in H against time t for those methods.

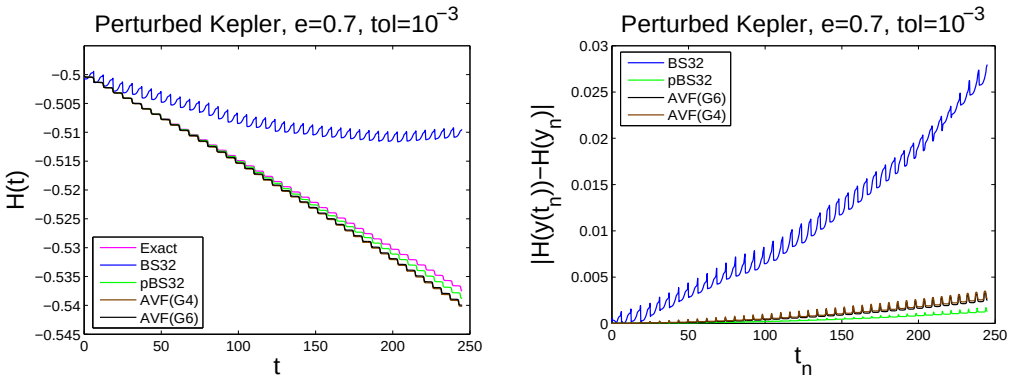

Figure 1: perturbed Kepler's problem (16): energy H (left) and error in H (right) against t, for methods BS32, pBS32, AVF(G4) and AVF(G6)

Observe that the Kepler energy is a monotone decreasing function with t. This monotonicity is preserved by pBS32 and AVF methods, whereas the numerical energy for standard method BS32 presents a wrong behaviour.

We are now interested in comparing the performance of the methods by computing the time t^* for which the energy of the system changes its value a 10%, that is, it attains the value $H(y(t^*)) = H^*$ with $H^* = 1.1H(y_0)$. This level of energy is attained for $t^* = 3.2202927214245 \times 10^2$. We have also computed the values $\hat{t}$ for which $H(y_h(\hat{t})) = H^*$, where $y_h(t)$ represents the numerical solution, for error tolerances from 10^{-3} to 10^{-8}. In Table 1 we present those values $\hat{t}$, the errors $t^* - \hat{t}$ and also the errors in H and the global errors at each $\hat{t}$, for BS32 (top) and pBS32 (bottom). Observe that, for each error tolerance, the errors $t^* - \hat{t}$ are smaller for pBS32 than for the standard method BS32, in agreement with (7). Errors in the energy and global errors are also smaller for the projected method than for the standard one. The method BS32 never attains the H^*-value for tolerance 10^{-3} since the numerical energy, after having a correct decreasing behaviour at the beginning, then it is an increasing function of time, as it is already made out in Figure 1.

It is remarkable in this table that the errors in the energy for BS23 are much smaller than the global errors. This is the reason why BS23 does not give errors in the energy $1/\varepsilon$ larger than pBS23. In any case this factor is at least 14, which makes pBS23 more efficient than BS23 in this aspect.

In Figure 2 the phase portraits for BS32, pBS32 and AVF(G4) are shown for error tolerance 10^{-3}. The numerical solutions are represented by points over the exact flow, in solid line, which has been computed by highly accurate numerical integration. It can be seen that the best behaviour corresponds to the new projection method pBS32. Even though the AVF method can provide a good approximation of the energy, it does not exhibit a good qualitative behaviour, and similar results are obtained if we approximate its integral in (15) with a quadrature formula of higher order. Naturally, as the error tolerance becomes smaller, the figures in the phase space are more precise.

Figure 3 shows the error in the energy H against the perturbation parameter ε at the final point of the integration interval. According to Theorem 2.1, that error must grow linearly with ε for pBS32, and this same kind of growth is expected for AVF taking into account Theorem 3.1 and Proposition 3.2. The method pBS32 behaves according to the theory, as it can be easily seen by comparing its graph with the dashed reference straight line with slope 1. Regarding the AVF method, to exhibit the correct behaviour it is crucial the way the integral is approximated. We have displayed the results for the numerical methods resulting from approximating the integral in (15) by the Gaussian quadrature formulas in [0,1] of orders 4, 6 and 10. The larger the order of the quadrature formula is, the more patent the expected behaviour is when $\varepsilon \to 0$. As expected, the error in H for the standard method BS32 does not decrease as ε does, since it is not an energy-preserving method.

Table 1: perturbed Kepler (16): time $\hat{t}$ for which the energy obtained using BS32 (top) and pBS32 (bottom) has been reduced in a 10%, together with error $t^* - \hat{t}$, energy-error at $\hat{t}$ and global error at $\hat{t}$

	BS32			
tol	$\hat{t}$	$t^* - \hat{t}$	$\lvert H(y(\hat{t})) - H^* \rvert$	$\lvert\lvert y(\hat{t}) - y_h(\hat{t}) \rvert\rvert$
10^{-3}	–	–	–	–
10^{-4}	3.5633999633776e+002	-3.4311e+001	5.5655e−003	3.3123e+000
10^{-5}	3.2728260303980e+002	-5.2533e+000	7.1248e−004	1.7931e+000
10^{-6}	3.2213356821941e+002	-1.0430e−001	4.7818e−005	5.9265e−002
10^{-7}	3.2203814782742e+002	-8.8757e−003	5.1014e−006	7.9495e−003
10^{-8}	3.2203014789989e+002	-8.7576e−004	5.1375e−007	8.1722e−004

	pBS32			
tol	$\hat{t}$	$t^* - \hat{t}$	$\lvert H(y(\hat{t})) - H^* \rvert$	$\lvert\lvert y(\hat{t}) - y_h(\hat{t}) \rvert\rvert$
10^{-3}	3.1023343442311e+002	1.1796e+001	2.5103e−003	2.8259e+000
10^{-4}	3.2168673878424e+002	3.4253e−001	4.3564e−004	9.5638e−001
10^{-5}	3.2197379433294e+002	5.5478e−002	3.7838e−005	5.4911e−002
10^{-6}	3.2202314859085e+002	6.1236e−003	3.6579e−006	4.6196e−003
10^{-7}	3.2202865146906e+002	6.2067e−004	3.6552e−007	4.5369e−004
10^{-8}	3.2202920993397e+002	6.2208e−005	3.6582e−008	4.5291e−005

Our second test problem comes from a semi-discretization of the perturbed one-dimensional wave equation where the velocity is taken equal to 1, given by:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} - \varepsilon \, \frac{\partial u}{\partial t}, \quad 0 < x < L, \ t > 0,$$

with the boundary conditions $u(0,t) = 0$, $u(L,t) = 0$. After using 4th-order central finite differences for the space derivative, we get the linear ODE system

$$y'' = -\frac{1}{12\Delta x^2} A \, y - \varepsilon \, y', \tag{17}$$

where $y(t) = (y_1(t), \ldots, y_M(t))^T$, with $y_i(t) \approx u(x_i, t)$, $x_i = i\Delta x$ the grid-points in the space interval $[0, L]$, $\Delta x = L/(M+1)$ the space-step, and A

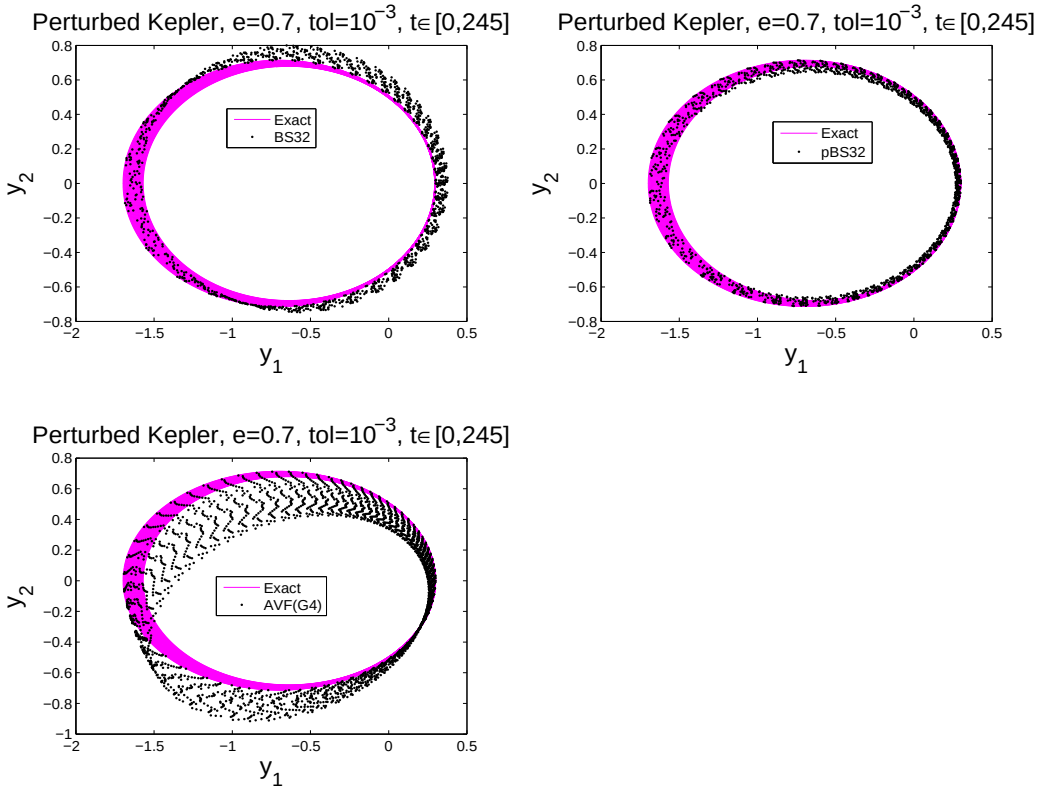

Figure 2: perturbed Kepler's problem (16): phase portraits, $t \in [0, 245]$, for methods BS32 (left top), pBS32 (right top) and AVF(G4) (bottom)

the symmetric pentadiagonal $M \times M$ matrix given by

$$A = \begin{pmatrix} 30 & -16 & 1 & & & \\ -16 & 30 & -16 & 1 & & \\ 1 & -16 & 30 & -16 & 1 & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & 1 & -16 & 30 \end{pmatrix}.$$

We have taken for this problem $\varepsilon = 10^{-3}$ as perturbation parameter, $L = 320$, $\Delta x = 1/4$, and $t \in [0, 300]$ as integration time interval. Let us note that the dimension of the first-order system equivalent to (17) is $2M = 2558$. The following initial conditions have been considered

$$y_i(0) = e^{-(x_i-10)^2}, \quad y_i'(0) = 2(x_i - 10)e^{-(x_i-10)^2}, \quad i = 1, \ldots, M.$$

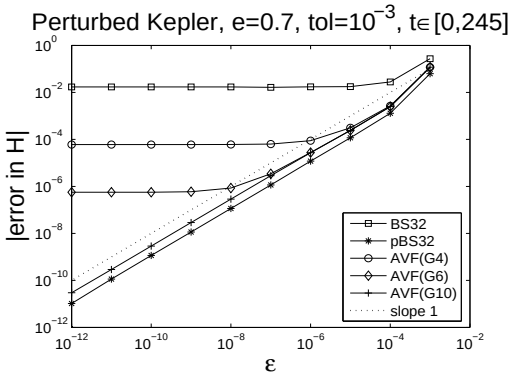

Figure 3: perturbed Kepler's problem (16): error in H against ε, for methods BS32, pBS32, AVF(G4), AVF(G6) and AVF(G10), and reference straight line with slope 1

A first integral for the un-damped semi-discretized wave equation is

$$H(y, y') = \frac{1}{24\Delta x^2} y^T A\, y + \frac{1}{2}\, y'^T\, y',$$

which is in fact its Hamiltonian function. The function $\alpha(y, y') = \nabla H(y, y') \cdot g(y, y')$ is given by

$$\alpha(y, y') = -y'^T y'.$$

It can be numerically verified that $\alpha(y)$ is a decreasing function along the solution $y(t)$ ranging from 5.013 at $t = 0$ to 3.712 at $t = 300$. Moreover, $|\mathrm{d}\alpha(y(t))/\mathrm{d}t|/|\alpha(y(t))| < 0.0031$ for all $t \leq 300$, that means that $u(t)$ satisfies the conditions of Theorem 2.3 with $\lambda \simeq 5 \times 10^{-3}$. The energy in an increasing function of t and is upper bounded by $H(y_0) + 5.013\,\varepsilon\, t$ and lower bounded by $H(y_0) + 3.7\,\varepsilon\, t$, as it is shown in Figure 4.

Figure 4 shows the evolution with time of the energy for error tolerance 10^{-3}. The picture on the left corresponds to the methods BS32, pBS32 and AVF(G4), whereas the one on the right corresponds to DP54 and its projection pDP54. The qualitative behaviour of the numerical energy is correct for all those methods unless for DP54, in the sense that it is a monotone decreasing function of t. The methods that produce closer values of H to those corresponding to the exact solution are the projection methods pBS32 and pDP54 and the conservative method AVF(G4). In fact in both pictures there is not distinction between the graphs for the exact energy and the numerical energy corresponding to those three methods. Clearly, standard methods give much worse results. Pictures in Figure 5 show the error in H as a function of time for the methods with order 2 and 3 at the top, and for the 5th-order methods at the bottom. These pictures make clear the superior performance of the projected and conservative methods over the standard

ones. The one on the right at the top allows us to compare the results between pBS32 and AVF(G4), which were indistinguishable in previous graphs, and it shows that the projection method pBS32 behaves slightly better than the conservative method AVF(G4).

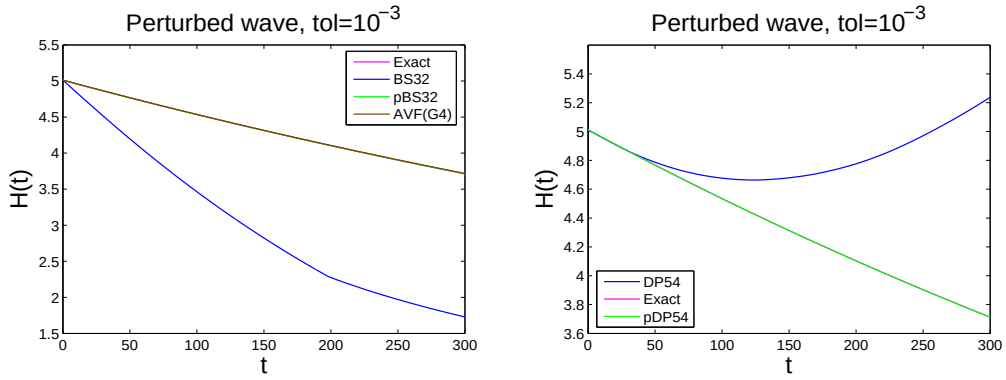

Figure 4: perturbed wave (17): energy H against t, for methods BS32, pBS32 and AVF(G4) (left), and methods DP54 and pDP54 (right)

Next, we include Table 2 to show the performance of the standard method BS32 and its projection pBS32, when we look for the time $\hat{t}$ for which the numerical energy attains a prefixed level H^*, as well as the error in H and the global error at $\hat{t}$. More precisely, for this problem we have taken H^* as the 75% of the initial energy of the system. The time t^* such that $H(y(t^*)) = H^*$ has turned out to be $t^* = 2.8768232264606 \times 10^2$. As it can be seen, the errors $t^* - \hat{t}$ are much smaller for the projected method than for the standard one and, additionally, they agree with the error tolerance imposed in the numerical integration. For each error tolerance, the error in the energy at the corresponding $\hat{t}$ is also quite smaller for pBS32 than for BS32. The global errors are of the same order of magnitude for both methods, slightly smaller for the projected method.

Table 3 shows the same type of information for DP54 and its projection pDP54, and analogous comments can be done. In this table there are no results for the largest tolerance because there is no such a time $\hat{t}$ since, as it can be appreciated on the right plot in Figure 4, the numerical energy corresponding to DP54, in spite of having a correct performance at the beginning, later on it presents a wrong increasing behaviour.

For this second problem the error in the first integral is of the same order of magnitude as the global error in both methods, BS23 and DP54, whereas it is much more smaller for pBS23 and pDP45, by a factor smaller than ε, as predicted by the theory.

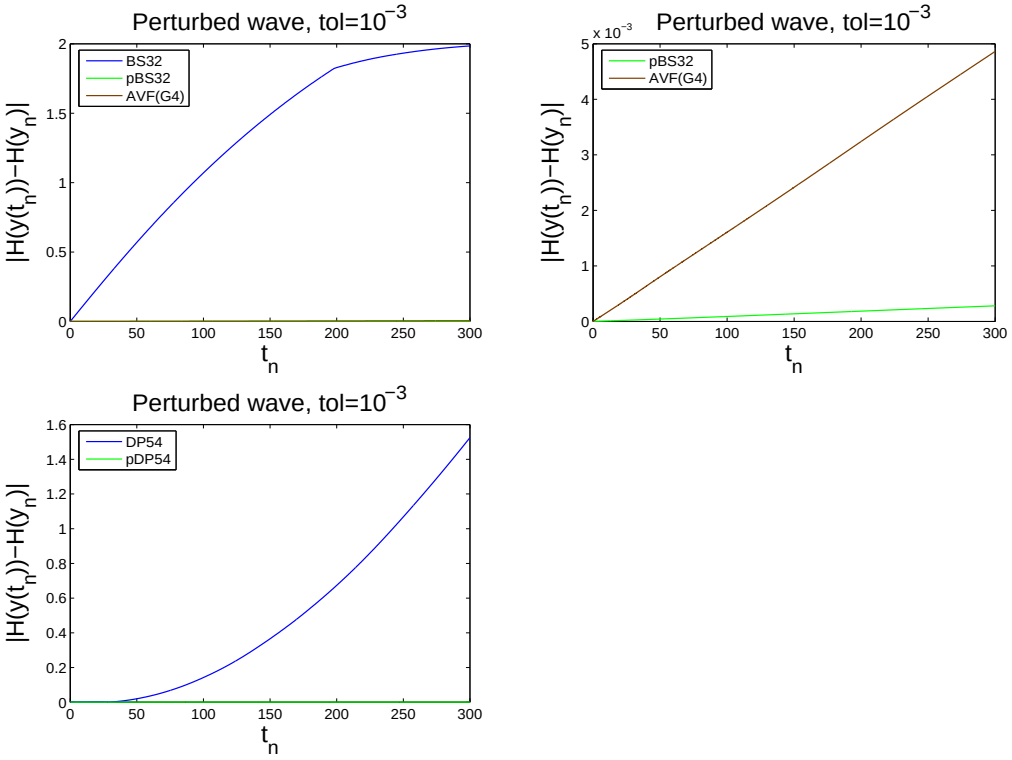

Figure 5: perturbed wave (17): error in H against t, for methods BS32, pBS32 and AVF(G4) (left top), methods pBS32 and AVF(G4) (right top), and methods DP54 and pDP54 (bottom)

In Figure 6 the error in the Hamiltonian at the end of the integration interval is represented as a function of ε for error tolerance 10^{-3}. That error grows linearly for the projection methods pBS32 and pDP54, and also for AVF(G4), as expected according Theorem 2.1 and Theorem 3.1, respectively. In relation with the averaged vector field method, it must be taken into account that, in order to be a conservative method, its definite integral should be exactly calculated. In fact, it is what happens for this problem because the 4th-order Gaussian quadrature formula is exact for it. The error in energy for the standard methods remain practically constant when $\varepsilon \to 0$.

Finally, in order to see the additional computational cost required by the PRK method with respect to the standard method, we study the efficiency of the proposed methods computing the global error at the end of the integration interval and representing it against the CPU time, in seconds, used in the integration. The results obtained using error tolerances from 10^{-4} to 10^{-11} are collected in Figure 7. The new method pBS32 employs about 2.5 times more CPU time than the standard one, and the time required by pDP54

Table 2: perturbed wave (17): time $\hat{t}$ for which the energy obtained using BS32 (top) and pBS32 (bottom) has been reduced in a 25%, together with error $t^* - \hat{t}$, energy-error at $\hat{t}$ and global error at $\hat{t}$

	BS32			
tol	$\hat{t}$	$t^* - \hat{t}$	$\lvert H(y(\hat{t})) - H^* \rvert$	$\lvert\lvert y(\hat{t}) - y_h(\hat{t}) \rvert\rvert$
10^{-3}	7.9176408362816e+001	2.0851e+002	8.7142e−001	4.4513e−001
10^{-4}	2.1631309599391e+002	7.1369e+001	2.7806e−001	1.4874e−001
10^{-5}	2.7770087979013e+002	9.9814e+000	3.7706e−002	2.0788e−002
10^{-6}	2.8663495502923e+002	1.0474e+000	3.9389e−003	2.1809e−003
10^{-7}	2.8757700109253e+002	1.0532e−001	3.9590e−004	2.1926e−004
10^{-8}	2.8767178340487e+002	1.0539e−002	3.9615e−005	2.1940e−005

	pBS32			
tol	$\hat{t}$	$t^* - \hat{t}$	$\lvert H(y(\hat{t})) - H^* \rvert$	$\lvert\lvert y(\hat{t}) - y_h(\hat{t}) \rvert\rvert$
10^{-3}	2.8771391324466e+002	-3.1591e−002	1.1874e−004	6.6320e−001
10^{-4}	2.8768451278965e+002	-2.1901e−003	8.2322e−006	1.0309e−001
10^{-5}	2.8768246709044e+002	-1.4444e−004	5.4293e−007	1.1454e−002
10^{-6}	2.8768232811620e+002	-5.4701e−006	2.0561e−008	1.1624e−003
10^{-7}	2.8768232283167e+002	-1.8561e−007	6.9766e−010	1.1646e−004
10^{-8}	2.8768232266350e+002	-1.7440e−008	6.5568e−011	1.1649e−005

is double the time for DP54. This is the price we must pay to get a more accurate value of the Hamiltonian, independent of the small parameter ε. On the contrary, for the same error tolerance, the projected methods give a slightly smaller global error than the corresponding standard ones, divided approximately by 1.9 for pBS32 and 1.25 for pDP54. Recall that the high CPU time required by AVF(G4) is due, not only to the iterations of the own method, but also to those required by the extrapolation technique. In any case, to get small global errors is not the aim of this work.

5 Conclusions

We have considered first-order autonomous differential systems which are systems possessing a scalar first integral weakly perturbed. For these sys-

Table 3: perturbed wave (17): time $\hat{t}$ for which the energy obtained using DP54 (top) and pDP54 (bottom) has been reduced in a 25%, together with error $t^* - \hat{t}$, energy-error at $\hat{t}$ and global error at $\hat{t}$.

	DP54			
tol	$\hat{t}$	$t^* - \hat{t}$	$\lvert H(y(\hat{t})) - H^* \rvert$	$\lvert\lvert y(\hat{t}) - y_h(\hat{t}) \rvert\rvert$
10^{-3}	–	–	–	–
10^{-4}	2.8624180729865e+002	1.4405e+000	5.4185e−003	8.8184e−002
10^{-5}	2.8693750393298e+002	7.4482e−001	2.8006e−003	3.5398e−003
10^{-6}	2.8759566169643e+002	8.6661e−002	3.2575e−004	3.3912e−004
10^{-7}	2.8767316752039e+002	9.1551e−003	3.4412e−005	3.3236e−005
10^{-8}	2.8768138898087e+002	9.3367e−004	3.5094e−006	3.2954e−006

	pDP54			
tol	$\hat{t}$	$t^* - \hat{t}$	$\lvert H(y(\hat{t})) - H^* \rvert$	$\lvert\lvert y(\hat{t}) - y_h(\hat{t}) \rvert\rvert$
10^{-3}	2.8767107854237e+002	1.1244e−002	4.2264e−005	4.0743e−001
10^{-4}	2.8768177850769e+002	5.4414e−004	2.0453e−006	9.1319e−002
10^{-5}	2.8768223805316e+002	8.4593e−005	3.1796e−007	3.2037e−003
10^{-6}	2.8768233521131e+002	-1.2565e−005	4.7230e−008	2.8382e−004
10^{-7}	2.8768232211774e+002	5.2832e−007	1.9858e−009	2.7038e−005
10^{-8}	2.8768232269738e+002	-5.1321e−008	1.9292e−010	2.6510e−006

tems, we have proposed numerical methods that are able to reproduce accurately the time evolution of the first integral in the approximate solution. Thus, we propose a projection technique that requires a standard Runge–Kutta (RK) method provided with dense output, also needing a quadrature formula with positive coefficients. We prove that these projection methods produce numerical approximations which give rise to the desirable evolution in the invariant. Moreover, we prove that, under some natural assumptions, conservative methods also have a good behaviour when they are applied to those perturbed systems. Numerical experiments have been carried out by using projection methods based on the 3rd-order Bogacki–Shampine RK method, as well as the 5th-order Dormand and Prince method, comparing them with the standard ones and also with the averaged vector field (conservative) method. These numerical results verify the theory and show the superior qualitative behaviour of the new projection method. They also show that the projected methods do not require much additional computational effort, and are therefore very efficient when accurate numerical first integral

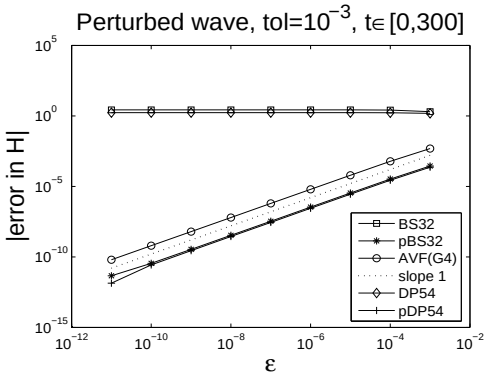

Figure 6: perturbed wave (17): error in H against ε, for methods BS32, pBS32, AVF(G4), DP54 and pDP54, and reference straight line with slope 1

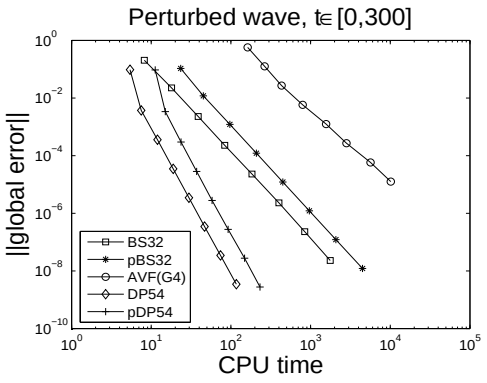

Figure 7: perturbed wave (17): global error against CPU time, for methods BS32, pBS32, AVF(G4), DP54 and pDP54

is needed.

Acknowledgements

The authors thank the anonymous referees for their very useful comments and suggestions that helped to improve greatly the paper.

References

[1] P. Bogacki, L.F. Shampine, A 3(2) pair of Runge–Kutta formulas, Appl. Math. Lett. 2 (1989), no. 4, 321–325.

[2] L. Brugnano, F. Iavernaro, Line Integral Methods which preserve all invariants of conservative problems, J. Comput. Appl. Math. 236 (2012), no. 16, 3905–3919.

[3] L. Brugnano, F. Iavernaro, D. Trigiante, Hamiltonian BVMs (HBVMs): a family of "drift-free" methods for integrating polynomial Hamiltonian systems, AIP Conf. Proc. 1168 (2009) 715–718.

[4] L. Brugnano, F. Iavernaro, D. Trigiante, A simple framework for the derivation and analysis of effective classes of one-step methods for ODEs, Appl. Math. Comput. 218 (2012) 8475–8485.

[5] L. Brugnano, M. Calvo, J.I. Montijano, L. Rández, Energy-preserving methods for Poisson systems, J. Comput. Appl. Math. 236 (2012), no. 16, 3890–3904.

[6] M. Calvo, D. Hernández–Abreu, J.I. Montijano, L. Rández, On the preservation of invariants by explicit Runge–Kutta methods, SIAM J. Sci. Comput. 28 (2006), no. 3, 868–885.

[7] M. Calvo, M.P. Laburta, J.I. Montijano, L. Rández, Approximate preservation of quadratic first integrals by explicit Runge–Kutta methods, Adv. Comput. Math. 32 (2010), no. 3, 255–274.

[8] M. Calvo, M.P. Laburta, J.I. Montijano, L. Rández, Projection methods preserving Lyapunov functions, BIT Numer. Math. 50 (2010) 223–241.

[9] M. Calvo, M.P. Laburta, J.I. Montijano, L. Rández, Runge–Kutta projection methods with low dispersion and dissipation errors, Adv. Comput. Math., in press.

[10] E. Celledoni, R.I. McLachlan, D.I. McLaren, B. Owren, G.R.W. Quispel, W.M. Wright, Energy-preserving Runge–Kutta methods, M2AN Math. Model. Numer. Anal. 43 (2009) 645–649.

[11] E. Celledoni, V. Grimm, R.I. McLachlan, D.I. McLaren, D. O'Neale, B. Owren, G.R.W. Quispel, Preserving energy resp. dissipation in numerical PDEs using the "Average Vector Field" method, J. Comput. Phys. 231 (2012), no. 20, 6770–6789.

[12] D. Cohen, E. Hairer, Linear energy-preserving integrators for Poisson systems, BIT Numer. Math. 51 (2011), no. 1, 91–101.

[13] M. Dahlby, B. Owren, T. Yaguchi, Preserving multiple first integrals by discrete gradients, Technical report, Norwegian University of Science and Technology, Trondheim, Numerics no. 11/2010, (`arXiv:1011.0478v4`).

[14] J.R. Dormand, P.J. Prince, A family of embedded Runge-Kutta formulae, J. Comp. Appl. Math. 6 (1980) 19–26.

[15] V. Grimm, G.R.W. Quispel, Geometric integration methods that preserve Lyapunov functions, BIT 45 (2005) 709–723.

[16] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration: Structure Preserving algorithms for Ordinary Differential Equations, Springer-Verlag, Berlin, 2002.

[17] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, Second Revised Edition, Springer-Verlag, Berlin, 1993.

[18] E. Hairer, Energy-preserving variant of collocation methods, J. Numer. Anal. Ind. Appl. Math. 5 (2010), no. 1-2, 73–84.

[19] F. Iavernaro, B. Pace, s-Stage Trapezoidal Methods for the Conservation of Hamiltonian Functions of Polynomial Type, AIP Conf. Proc. 936 (2007) 603–606.

[20] F. Iavernaro, D. Trigiante, High-order Symmetric Schemes for the Energy Conservation of Polynomial Hamiltonian Problems, J. Numer. Anal. Ind. Appl. Math. 4 (2009), no. 1-2, 87–101.

[21] R.I. McLachlan, G.R.W. Quispel, What kinds of dynamics are there? Lie pseudogroups, dynamical systems and geometric integration, Nonlinearity 14 (2001) 1689–1705.

[22] R.I. McLachlan, G.R.W. Quispel, Geometric integrators for ODEs, J. Phys. A 39 (2006), no. 19, 5251–5285.

[23] K. Modin, G. Söderlind, Geometric integration of Hamiltonian systems perturbed by Rayleigh damping, BIT 51 (2011) 977–1077.

[24] O. Montenbruck, E. Gil, Satellite Orbits: Models, Methods, and Applications, Springer, Berlin, 2001.

[25] B. Owren, M. Zennaro, Continuous explicit Runge-Kutta methods, Computational ordinary differential equations (London, 1989) Inst. Math. Appl. Conf. Ser. New Ser. 39, Oxford Univ. Press, New York, 1992, pp. 97–105.

[26] G.R.W. Quispel, D.I. McLaren, A new class of energy-preserving numerical integration method J. Phys. A: Math. Theor. 41 (2008) 045206 (7 pp.).

[27] L.F. Shampine, M.W. Reichelt, The MATLAB ODE suite, SIAM J. Sci. Comput. 8 (1997), no. 1, 1–22 .