ELSEVIER

Applied Numerical Mathematics 28 (1998) 1–16

APPLIED NUMERICAL MATHEMATICS

Variable step implementation of geometric integrators

M.P. Calvo *, M.A. López-Marcos, J.M. Sanz-Serna

Departamento de Matemática Aplicada y Computación, Facultad de Ciencias, Universidad de Valladolid, Prado de la Magdalena s/n, 47005 Valladolid, Spain

Abstract

We compare experimentally several techniques for combining geometric integrators with variable time steps. In particular, we study modifications of the Verlet method due to Leimkuhler and a technique for symplectic integration based on Poincaré transformations suggested by Hairer and Reich independently. We conclude that it is feasible to develop symplectic variable step size codes that, for Hamiltonian problems, are competitive with standard software. We also analyze the error growth of the new algorithms when integrating periodic orbits.

Keywords: Variable step sizes; Symplectic integrators; Reversible methods; Kepler's problem

1. Introduction

In this paper we consider variable step size symplectic and reversible methods for the numerical integration of differential systems which are both Hamiltonian and reversible.

In the recent literature the question of numerical integration of differential systems by so-called "geometric" [15] integrators has been widely addressed. Geometric integrators are numerical methods which preserve some characteristic features of the exact flow of the system.

A first instance of such methods is given by *symplectic* integrators [16]. It is well known that for each t the flow φ_t of a Hamiltonian system of ordinary differential equations

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} &= -\nabla_q H(\boldsymbol{p},\boldsymbol{q}), \\ \frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} &= \nabla_p H(\boldsymbol{p},\boldsymbol{q}) \end{aligned} \tag{1}$$

is a canonical transformation in phase space. Symplectic integrators are numerical one-step methods for which the one-step mapping ψ_h also preserves the symplectic structure of phase space.

* Corresponding author. E-mail: maripaz@cpd.uva.es.

0168-9274/98/$19.00
PII: S0168-9274(98)00035-X

A second example of geometric integrators is given by *reversible* methods. Let us consider a differential system

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}) \tag{2}$$

and an involution ρ (a linear mapping with $\rho^2 = \mathrm{Id}$). The system (2) is said to be ρ-reversible if

$$\boldsymbol{f}(\rho(\boldsymbol{x})) = -\rho(\boldsymbol{f}(\boldsymbol{x})).$$

This implies that for each t, the flow φ_t of the system (2) is a ρ-reversible mapping, that is to say, $(\varphi_t)^{-1} = \rho\varphi_t\rho$. A numerical one-step method is called ρ-reversible if when applied to a ρ-reversible system (2) the numerical mapping ψ_h is a ρ-reversible mapping, i.e., for each h, $(\psi_h)^{-1} = \rho\psi_h\rho$.

These two families of geometric integrators are particularly relevant in the numerical integration of the Newton equations of mechanics

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = \boldsymbol{F}(\boldsymbol{q}), \qquad \frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = \boldsymbol{p}, \tag{3}$$

which are reversible with respect to the involution $\rho([\boldsymbol{p}, \boldsymbol{q}]^{\mathrm{T}}) = [-\boldsymbol{p}, \boldsymbol{q}]^{\mathrm{T}}$, and are also Hamiltonian if $\boldsymbol{F}$ is the gradient of a scalar potential.

Symplectic and reversible integrators with constant step sizes have revealed themselves very suitable for the integration of Hamiltonian and reversible systems of ordinary differential equations [2,3]. For instance, it has been proved in [3] that, when integrating periodic orbits in some problems, the errors grow only linearly with time.

On the other hand, the use of variable step sizes in the numerical integration of differential systems is advisable in order to advance with small step sizes when the solution changes rapidly and with large step sizes when the solution is only slowly varying. For general integrators, the use of variable step sizes significantly improves the efficiency of the method. Therefore the question arises of combining geometric integration with variable step sizes.

For symplectic methods, the use of a standard variable step size strategy results in a decrease in efficiency [2,17]. Backward error analysis which allows to get favorable results for symplectic methods implemented with fixed step size [5,16] is no longer valid when a symplectic integrator is used with standard variable step size strategies.

This difficulty in the use of symplectic integrators was the main reason for the study of reversible methods because, in the context of numerical integration of reversible systems, several step size strategies [7,9–11,18] have been found that retain the good properties observed in fixed step size implementations [3]. Most reversible variable step size methods in [7,9–11,18] are implicit. However, Leimkuhler has developed in [11] a Verlet-like method that is explicit. Here we focus our attention on the strategy developed in [9,11], where the main idea is to introduce a time transformation

$$\frac{\mathrm{d}t}{\mathrm{d}\tau} = g(\boldsymbol{p}, \boldsymbol{q}), \tag{4}$$

and to write the differential system in terms of the new independent variable τ:

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}\tau} &= -g(\boldsymbol{p}, \boldsymbol{q})\nabla_{\boldsymbol{q}} H(\boldsymbol{p}, \boldsymbol{q}), \\ \frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}\tau} &= g(\boldsymbol{p}, \boldsymbol{q})\nabla_{\boldsymbol{p}} H(\boldsymbol{p}, \boldsymbol{q}). \end{aligned} \tag{5}$$

Integrating the differential system (5) with fixed step size h provides variable step size numerical solutions to (1). If the function g is such that the reparameterized system (5) is reversible, then results for reversible integration with fixed step size [3] can be applied to (5) and they imply favorable properties for the numerical solution viewed as a variable step size approximation to (1).

The question of symplectic integration with variable step size without losing the good long-time behaviour of the fixed step symplectic integration has been an open problem for several years. In particular, the technique in (4), (5) does not work because in nontrivial cases the transformed system (5) fails to be Hamiltonian. Recently a solution to this problem by using Poincaré transformations has been suggested by Hairer [6] and Reich [14]. The idea is as follows: if $(\boldsymbol{p}_0, \boldsymbol{q}_0)$ is the initial condition for (1) and H_0 is the value of the Hamiltonian function H at $(\boldsymbol{p}_0, \boldsymbol{q}_0)$, one integrates with a symplectic method and fixed step size h the Hamiltonian system with Hamiltonian function $g(H - H_0)$, i.e.,

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}\tau} &= -g(\boldsymbol{p},\boldsymbol{q})\nabla_q H(\boldsymbol{p},\boldsymbol{q}) - \big[H(\boldsymbol{p},\boldsymbol{q}) - H_0\big]\nabla_q g(\boldsymbol{p},\boldsymbol{q}), \\ \frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}\tau} &= g(\boldsymbol{p},\boldsymbol{q})\nabla_p H(\boldsymbol{p},\boldsymbol{q}) + \big[H(\boldsymbol{p},\boldsymbol{q}) - H_0\big]\nabla_p g(\boldsymbol{p},\boldsymbol{q}), \end{aligned} \tag{6}$$

augmented by Eq. (4). Note that differentiability of g is required. The difference between (6) and the reparameterized system (5) is just a perturbation which vanishes along the solution of (1) with initial condition $(\boldsymbol{p}_0, \boldsymbol{q}_0)$, but that makes (6) Hamiltonian. When the symplectic method chosen to integrate (6) is the Lobatto IIIA-B pair [19] the resulting variable step size algorithm is very similar to one of the methods suggested by Leimkuhler.

The purpose of this paper is to compare the approaches in [9,11] with those in [6] when applied to the numerical integration of differential systems which are both Hamiltonian and reversible. While the schemes in [9,11] are of order two, the idea in [6] can be combined with integrators of arbitrarily high order. We first compare an implicit and an explicit method from [9,11] versus a symplectic second order algorithm. However second order is often very inefficient and order four or even higher are desirable. We have also considered symplectic schemes of order four based on the idea of [6] and conventional software of the same order. Section 2 is devoted to the detailed description of the implementation of the methods being compared. Due to the high number of details involved in each algorithm (choice of monitor function, choice of nonlinear solver, etc.), we have adopted a formal style of presentation, to ensure that the interested reader may reproduce our experiments. A more 'journalistic' style would have resulted in a paper more readable but less precise. In Section 3 numerical results are reported. Our conclusions are given in Section 4. In Appendix A we establish a theoretical result which ensures in some cases linear error growth for the symplectic variable step size strategy mentioned above.

2. Numerical integrators

Here we present some numerical methods with variable step size for the integration of the Hamiltonian system (1). We assume that the Hamiltonian function has the form

$$H(\boldsymbol{p},\boldsymbol{q}) = \frac{1}{2}\|\boldsymbol{p}\|^2 + V(\boldsymbol{q}). \tag{7}$$

Although in [9] several choices of stepsize function g are cited, in this paper we restrict our attention to the arclength parameterization

$$g(\boldsymbol{p},\boldsymbol{q}) = \left(\|\boldsymbol{p}\|^2 + \|\nabla V(\boldsymbol{q})\|^2\right)^{-1/2} \tag{8}$$

and to the minimum distance parameterization

$$g(\boldsymbol{q}) = d_{\min}^{2\alpha}, \tag{9}$$

where $d_{\min}$ denotes, for particle motion, the smallest separation between particles and α is a positive constant. The choices (8) and (9) appear to be the most popular considered in the literature [6,9,10]. In [6] the arclength parameterization (8) has been modified to become

$$g(\boldsymbol{q}) = \left(2(H_0 - V(\boldsymbol{q})) + \|\nabla V(\boldsymbol{q})\|^2\right)^{-1/2}, \tag{10}$$

which does not depend on $\boldsymbol{p}$. However, in our numerical experiments we found that in practice the quantity under the square root in (10) may become negative and for this reason we abandoned (10).

2.1. *Implicit adaptive Verlet method (IAV)*

Huang and Leimkuhler [9] introduced a generalization of the Verlet method by using the second-order Lobatto IIIA-B partitioned Runge–Kutta formula for the scaled system (4), (5). The equations of the method are

$$\boldsymbol{p}_{n+1/2} = \boldsymbol{p}_n - \frac{h}{2} g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n) \nabla V(\boldsymbol{q}_n), \tag{11}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \frac{h}{2}\left[g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n) + g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1})\right] \boldsymbol{p}_{n+1/2}, \tag{12}$$

$$\boldsymbol{p}_{n+1} = \boldsymbol{p}_{n+1/2} - \frac{h}{2} g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1}) \nabla V(\boldsymbol{q}_{n+1}), \tag{13}$$

$$t_{n+1} = t_n + \frac{h}{2}\left[g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n) + g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1})\right]. \tag{14}$$

This discretization is reversible and preserves the angular momentum. Next we discuss the solution of each one of the equations (11)–(13).

If the function g depends on $\boldsymbol{p}$, then the first equation is implicit in $\boldsymbol{p}_{n+1/2}$ and we define the following iteration to solve it:

$$\boldsymbol{p}_{n+1/2}^{[\nu+1]} = \boldsymbol{p}_n - \frac{h}{2} g(\boldsymbol{p}_{n+1/2}^{[\nu]}, \boldsymbol{q}_n) \nabla V(\boldsymbol{q}_n), \quad \nu = 0, 1, \ldots. \tag{15}$$

We choose as a first iterant $\boldsymbol{p}_{n+1/2}^{[0]} = \boldsymbol{p}_n$, and stop the iteration when the relative change is less than a prescribed tolerance, that is

$$\frac{\|\boldsymbol{p}_{n+1/2}^{[\nu+1]} - \boldsymbol{p}_{n+1/2}^{[\nu]}\|}{\|\boldsymbol{p}_{n+1/2}^{[\nu+1]}\|} < \text{Tol}. \tag{16}$$

If the function g depends on $\boldsymbol{p}$ only through its norm, then (15) can be written as a scalar iteration. For example, if g represents the arclength parameterization (8), we compute the quantities $\beta^{[\nu]} = \|\boldsymbol{p}_{n+1/2}^{[\nu]}\|^2$ from the iteration

$$\beta^{[\nu+1]} = \|\boldsymbol{p}_n\|^2 + \left(\frac{h}{2} g(\boldsymbol{p}_{n+1/2}^{[\nu]}, \boldsymbol{q}_n)\right)^2 \|\nabla V(\boldsymbol{q}_n)\|^2 - h g(\boldsymbol{p}_{n+1/2}^{[\nu]}, \boldsymbol{q}_n)\langle \boldsymbol{p}_n, \nabla V(\boldsymbol{q}_n)\rangle,$$
$$\nu = 0, 1, \ldots.$$

Here

$$g(\boldsymbol{p}_{n+1/2}^{[\nu]}, \boldsymbol{q}_n) = (\beta^{[\nu]} + \|\nabla V(\boldsymbol{q}_n)\|^2)^{-1/2},$$

so that the iterants $\boldsymbol{p}_{n+1/2}^{[\nu]}$ do not need to be constructed. Then (16) can be written as

$$\frac{(h/2)\|\nabla V(\boldsymbol{q}_n)\| |g(\boldsymbol{p}_{n+1/2}^{[\nu]}, \boldsymbol{q}_n) - g(\boldsymbol{p}_{n+1/2}^{[\nu-1]}, \boldsymbol{q}_n)|}{\sqrt{\beta^{[\nu+1]}}} < \text{Tol}, \tag{17}$$

with $g(\boldsymbol{p}_{n+1/2}^{[-1]}, \boldsymbol{q}_n) = 0$. When the condition (17) is satisfied, we set

$$g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n) = (\beta^{[\nu]} + \|\nabla V(\boldsymbol{q}_n)\|^2)^{-1/2}$$

and then obtain $\boldsymbol{p}_{n+1/2}$ from (11).

Another interesting situation arises when the function g depends on only $\boldsymbol{q}$ as in (9). In that case Eq. (11) is explicit.

Eq. (12) is implicit in $\boldsymbol{q}_{n+1}$, and we solve it by the iteration

$$\boldsymbol{q}_{n+1}^{[\nu+1]} = \boldsymbol{q}_n + \frac{h}{2}[g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n) + g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1}^{[\nu]})]\boldsymbol{p}_{n+1/2}, \quad \nu = 0, 1, \ldots, \tag{18}$$

choosing as a first iterant $\boldsymbol{q}_{n+1}^{[0]} = \boldsymbol{q}_n$. Again, the iteration is stopped under control of the relative change, that is

$$\frac{\|\boldsymbol{q}_{n+1}^{[\nu+1]} - \boldsymbol{q}_n^{[\nu]}\|}{\|\boldsymbol{q}_{n+1}^{[\nu+1]}\|} < \text{Tol}, \tag{19}$$

which can be written as

$$\frac{(h/2)\|\boldsymbol{p}_{n+1/2}\| |g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1}^{[\nu]}) - g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1}^{[\nu-1]})|}{\|\boldsymbol{q}_{n+1}^{[\nu+1]}\|} < \text{Tol}. \tag{20}$$

Here, $g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1}^{[-1]}) = -g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n)$. When (20) is satisfied, we set $g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1}) = g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1}^{[\nu]})$, and $\boldsymbol{q}_{n+1} = \boldsymbol{q}_{n+1}^{[\nu+1]}$.

Finally, Eq. (13) is explicit.

2.2. *Explicit adaptive Verlet method (EAV)*

Leimkuhler [11] proposed a fully explicit method by considering a new variable ρ that represents the reciprocal of the time-scaling factor. This variable is computed by a symmetric formula. The equations of the method are:

$$\boldsymbol{q}_{n+1/2} = \boldsymbol{q}_n + \frac{h}{2}\frac{1}{\rho_n}\boldsymbol{p}_n, \tag{21}$$

$$\boldsymbol{p}_{n+1/2} = \boldsymbol{p}_n - \frac{h}{2}\frac{1}{\rho_n}\nabla V(\boldsymbol{q}_{n+1/2}), \tag{22}$$

$$\rho_n + \rho_{n+1} = \frac{2}{g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1/2})}, \tag{23}$$

$$\boldsymbol{p}_{n+1} = \boldsymbol{p}_{n+1/2} - \frac{h}{2}\frac{1}{\rho_{n+1}}\nabla V(\boldsymbol{q}_{n+1/2}), \tag{24}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_{n+1/2} + \frac{h}{2}\frac{1}{\rho_{n+1}}\boldsymbol{p}_{n+1}, \tag{25}$$

$$t_{n+1} = t_n + \frac{h}{2}\left[\frac{1}{\rho_n} + \frac{1}{\rho_{n+1}}\right]. \tag{26}$$

This method is reversible and conserves the angular momentum. At the first step, we take $\rho_0 = 1/g(\boldsymbol{p}_0, \boldsymbol{q}_0)$.

If the monitoring function g only depends on $\boldsymbol{q}$, Eqs. (22) and (24) can be rewritten as a single equation

$$\boldsymbol{p}_{n+1} = \boldsymbol{p}_n - \frac{h}{2}\left[\frac{1}{\rho_n} + \frac{1}{\rho_{n+1}}\right]\nabla V(\boldsymbol{q}_{n+1/2}).$$

A method similar to (21)–(26) was suggested earlier by Leimkuhler in [9]. That method is explicit when g is independent of $\boldsymbol{p}$ and requires a scalar iteration if g depends on $\|\boldsymbol{p}\|$ and $\boldsymbol{q}$. Numerical experiments carried out by us, but not reported in this paper, revealed that the fully explicit method (21)–(26) provides an improvement on the earlier method of [9]. Therefore we will not consider further the earlier method.

2.3. *The variable step symplectic method (VS)*

The third method we have considered in the numerical experiments is the two-stage Lobatto IIIA-B pair applied to the Hamiltonian system (4), (6). This method is symplectic, reversible and preserves the angular momentum. The equations to advance a step of length h in the numerical integration of (4), (6) are as follows:

$$\begin{aligned}\boldsymbol{p}_{n+1/2} = \boldsymbol{p}_n - \frac{h}{2}&\big(g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n)\nabla V(\boldsymbol{q}_n)\\ &+ [H(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n) - H_0]\nabla_{\boldsymbol{q}} g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n)\big),\end{aligned} \tag{27}$$

$$\begin{aligned}\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \frac{h}{2}&\big([g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n) + g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1})]\boldsymbol{p}_{n+1/2}\\ &+ [H(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n) - H_0]\nabla_{\boldsymbol{p}} g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n)\\ &+ [H(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1}) - H_0]\nabla_{\boldsymbol{p}} g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1})\big),\end{aligned} \tag{28}$$

$$\begin{aligned}\boldsymbol{p}_{n+1} = \boldsymbol{p}_{n+1/2} - \frac{h}{2}&\big(g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1})\nabla V(\boldsymbol{q}_{n+1})\\ &+ [H(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1}) - H_0]\nabla_{\boldsymbol{q}} g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1})\big),\end{aligned} \tag{29}$$

$$t_{n+1} = t_n + \frac{h}{2}[g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_n) + g(\boldsymbol{p}_{n+1/2}, \boldsymbol{q}_{n+1})]. \tag{30}$$

As Eq. (27) is implicit in $\boldsymbol{p}_{n+1/2}$, we have solved it by fixed point iteration. More precisely, we have chosen the first iterant as $\boldsymbol{p}_{n+1/2}^{[0]} = \boldsymbol{p}_n$ and for $\nu = 0, 1, \ldots$ (cf. (15))

$$\boldsymbol{p}_{n+1/2}^{[\nu+1]} = \boldsymbol{p}_n - \frac{h}{2}\big(g(\boldsymbol{p}_{n+1/2}^{[\nu]}, \boldsymbol{q}_n)\nabla V(\boldsymbol{q}_n) + [H(\boldsymbol{p}_{n+1/2}^{[\nu]}, \boldsymbol{q}_n) - H_0]\nabla_{\boldsymbol{q}} g(\boldsymbol{p}_{n+1/2}^{[\nu]}, \boldsymbol{q}_n)\big). \tag{31}$$

The fixed point iteration has been stopped when (16) is satisfied. After stopping the fixed point iteration, the functions g and $\nabla_p g$ are evaluated at the last iterant $p_{n+1/2}^{[\nu+1]}$.

As in the IAV method, if g depends on p only through the norm $\|p\|$, then (31) can be replaced by a scalar iteration. For instance, when the step size function is the arclength parameterization (8), it is not difficult to see that $\nabla_q g(p,q) = -g(p,q)^3 V''(q)\nabla V(q)$, where V'' is the Hessian of V. So

$$p_{n+1/2}^{[\nu+1]} = p_n - \frac{h}{2}\left[C^{[\nu]}\nabla V(q_n) + D^{[\nu]}V''(q_n)\nabla V(q_n)\right], \quad \nu = 0, 1, \ldots,$$

where $C^{[\nu]}$ and $D^{[\nu]}$ are scalar quantities depending on $p_{n+1/2}$ only through its norm.

For monitoring functions g depending only on q, Eq. (27) is still implicit but only through the scalar quantity $\|p_{n+1/2}\|^2$

$$p_{n+1/2} = p_n - \frac{h}{2}\left(g(q_n)\nabla V(q_n) + \left[\frac{1}{2}\|p_{n+1/2}\|^2 + V(q_n) - H_0\right]\nabla_q g(q_n)\right). \tag{32}$$

This leads to a quadratic equation for $\|p_{n+1/2}\|^2$ as pointed out in [6]. When $\|p_{n+1/2}\|^2$ has been obtained, Eq. (32) determines $p_{n+1/2}$.

The relation (28) defining q_{n+1} is also implicit and we have used again fixed point iteration in order to solve it

$$\begin{aligned} q_{n+1}^{[\nu+1]} = q_n + \frac{h}{2}\big(&\left[g(p_{n+1/2}, q_n) + g(p_{n+1/2}, q_{n+1}^{[\nu]})\right]p_{n+1/2} \\ &+ \left[H(p_{n+1/2}, q_n) - H_0\right]\nabla_p g(p_{n+1/2}, q_n) \\ &+ \left[H(p_{n+1/2}, q_{n+1}^{[\nu]}) - H_0\right]\nabla_p g(p_{n+1/2}, q_{n+1}^{[\nu]})\big), \end{aligned} \tag{33}$$

starting with $q_{n+1}^{[0]} = q_n$ and with the stopping criterion (19). After stopping the fixed point iteration, the functions V, g and $\nabla_q g$ are evaluated at $(p_{n+1/2}, q_{n+1})$.

If the step size function g does not depend on p, as for (9), the fixed point iteration in order to solve Eq. (28) can be written as (cf. (18))

$$q_{n+1}^{[\nu+1]} = q_n + \frac{h}{2}\left[g(q_n) + g(q_{n+1}^{[\nu]})\right]p_{n+1/2}. \tag{34}$$

Eq. (29) is always explicit.

2.4. *Computational cost*

Here we discuss the computational cost required by the methods described above. We suppose that most of the work required in the integration of the Hamiltonian system (1) with Hamiltonian function (7) consists of the evaluation of the vector ∇V. For instance, in the simulation of the motion of N mutually interacting particles, this vector collects the N forces on each of the N particles, so that its evaluation has an $O(N^2)$ cost. Therefore we take as a cost measure of a numerical method the number of evaluations of the vector ∇V and of other quantities that require a similar work. This is the case for the computation of the smallest separation between particles because, in the motion of N particles, it requires an $O(N^2)$ cost, too.

All the methods previously described use one force evaluation per step. For instance, the IAV method requires $\nabla V(q_{n+1})$ in (13) which is reused in (11) at the next step. Besides that, terms involving g can require additional evaluations of ∇V.

Table 1
Cost of the methods. Here m is the number of iterations in the vector iteration

Method	Cost per step
IAV	m
EAV	1
VS	$m+1$

For the IAV method, (11) involves one evaluation of ∇V. The iteration (18) involves the evaluation of g on the preceding iterant. If the monitoring function is the arclength (8), this requires a new evaluation of ∇V per iteration and, for the particle separation parameterization (9), the computation of the smallest distance between particles. In the first iteration, these quantities do not need to be computed because we take $\boldsymbol{q}_{n+1}^{[0]} = \boldsymbol{q}_n$, so that, for the arclength parameterization, the value of $\nabla V(\boldsymbol{q}_{n+1}^{[0]})$ is known. In the case of particle separation, $g(\boldsymbol{q}_{n+1}^{[0]}) = g(\boldsymbol{q}_n)$ can be computed at negligible cost within the loops that compute $\nabla V(\boldsymbol{q}_n)$; note that for most problems $\nabla V(\boldsymbol{q})$ already requires the computation of all interparticle distances. So, if m is the number of iterations, solving the implicit equation (12) involves a cost of $m-1$ units.

The EAV method is explicit, so it requires only one evaluation of ∇V per step.

The VS method implies the following cost. Eq. (27) requires one evaluation of ∇V. Besides that, we assume that this effort can be used for the computation of g and H without additional work. On the other hand, if the arclength parameterization (8) is used, the evaluation of $\nabla_q g$ requires the Hessian-vector product $V''\nabla V$. In the case of an N-body problem with 2-body interactions the computation of ∇V and $V''\nabla V$ at the same point can be done at the cost of at most two independent ∇V evaluations [13]. So we consider that (27) involves two evaluations of ∇V.

The iteration (33) requires the evaluation of g, H and $\nabla_p g$ at the preceding iterant. If g is the arclength parameterization these values require only one evaluation of ∇V per iteration except the first one. Then, as in the IAV method, if m is the number of iterations required to get (19), solving (28) implies $m-1$ evaluations of ∇V.

Table 1 presents a summary of the cost of the methods.

The time stepping formulae of the three second-order methods IAV, EAV, VS are very closely related. In particular, the formulae for the IAV and VS methods only differ in a small term. The differences between the three methods are as follows. First of all, the adaptive Verlet methods are easier to implement than the variable step symplectic method, but adaptive Verlet methods are not symplectic (simplicity against symplecticity). Besides that, the IAV method is cheaper per step than the symplectic method, and the EAV even cheaper. On the other hand, the VS method is not applicable for all the choices of g (remember that $\nabla_q g$ and $\nabla_p g$ are needed).

2.5. *Fourth-order methods*

In our numerical experiments we have also considered three methods of order four.

The first one is the variable step size symplectic integrator obtained when the two-stage Gauss method is used to integrate the Hamiltonian system (4), (6). We will refer to this method as VG4. As the method

is implicit we have solved the algebraic equations by fixed point iteration and the iteration is stopped under control of the relative change. The implementation follows the ideas described in [16]. According to the comments above, we consider that the computational cost of one evaluation of the right hand side of (4), (6) is twice the cost of evaluating the right hand side of the original Hamiltonian system (1).

The second scheme (SF4) is the symplectic Runge–Kutta–Nyström method proposed in [2], that has been implemented *with fixed step size*. The method is explicit, has optimized error constants and requires four function evaluations per step. The details for its efficient implementation can be found in [16].

The last fourth-order method (DPV4) is one of the well known embedded pairs of Runge–Kutta–Nyström methods developed by Dormand and his coworkers [4]. The algorithm is explicit and needs three function evaluations per step. It has been implemented with variable step sizes, as described in [16]. This method is included in the comparisons as a good example of a state-of-the-art, optimized, standard integrator.

3. Numerical experiments

3.1. Second-order methods

We performed numerical experiments in order to compare the IAV, EAV and VS methods described in Section 2. The implementation and cost measurements were performed in the way described in that section which is suited to 'general' problems. However it should be kept in mind that for specific problems our implementation may not be the best and our cost measurements may only be approximate.

The main test problem considered is the Kepler problem in two dimensions

$$H(\boldsymbol{p},\boldsymbol{q}) = \frac{1}{2}(p_1^2 + p_2^2) - \frac{1}{\sqrt{q_1^2 + q_2^2}},$$

with initial conditions $q_1(0) = 1 - e$, $q_2(0) = 0$, $p_1(0) = 0$, $p_2(0) = \sqrt{(1+e)/(1-e)}$, that correspond to an orbit of period 2π and eccentricity e.

In order to reduce the damage generated by roundoff, we used the compensated summation technique [8, Section 4.3] in the computation of the components of the solution (after the iteration in the implicit methods). Errors were measured in the Euclidean norm of $\mathbb{R}^4$. We also note that, while particle separation (9) cannot in general be used with the VS method, in Kepler's problem and other two particle problems $d_{\min}$ equals the distance between both particles and is therefore differentiable. In our experiments we took $\alpha = 1$.

We first took $e = 0.9$. Since geometric integration is of interest when the solution is required over a long time-interval we integrated for 1025 periods, i.e., for $0 \leqslant t \leqslant 2050\pi$. This time interval leads to expensive experiments: a typical run required 10^8 evaluations of ∇V and we needed to run many combinations h/Tol, different methods and different monitors (not all of them reported here). In the implicit methods IAV and VS we first determined, for each value of h, a suitable value of the tolerance Tol. For Kepler's problem and geometric integrators it can be shown (see [3] and Appendix A) that the error growth is linear in t. For each value of h we tried values of the tolerance $10^{-2}, 10^{-3}, \ldots, 10^{-15}$ and considered a tolerance to be too coarse if the errors at $t = 4 \times 2\pi, 16 \times 2\pi, 64 \times 2\pi, 256 \times 2\pi, 1024 \times 2\pi$ did not show a linear behaviour. For each method and each h we identified the largest acceptable tolerance, see

Table 2
Tolerances for the implicit methods in Kepler's problem $e = 0.9$

Algorithm		h			
Method	Monitor	4×10^{-4}	2×10^{-4}	1×10^{-4}	5×10^{-5}
IAV	Arclength		10^{-8}	10^{-9}	10^{-5}
IAV	Particle separation		10^{-4}	10^{-5}	10^{-5}
VS	Arclength	10^{-12}	10^{-13}	10^{-15}	10^{-15}
VS	Particle separation	10^{-11}	10^{-12}	10^{-13}	10^{-15}

Table 2. The comparisons are based on these tolerances. It is somewhat surprising that the VS method requires more stringent tolerances than the closely related IAV method.

We compared the different methods by measuring errors along the 1025th period at times $(1024 + j/20)2\pi$, $j = 1, 2, \ldots, 20$, and then averaging these 20 values. Cubic Hermite interpolation was used to obtain the solution at the required times.

Fig. 1 shows, for each method, the error versus cost measured as in Table 1. Circles correspond to the IAV method, crosses to the EAV method and stars to the VS method. Solid line represents the arclength parameterization and dashed line particle separation parameterization. The values of h used for the implicit methods are those presented in Table 2, and for the EAV method were $h = 2 \times 10^{-4}$, 1×10^{-4} and 5×10^{-5}. This figure shows that, for the EAV method and VS methods, arclength parameterization is more efficient than particle separation. For the IAV method both parameterizations are comparable. On the other hand, even though, as noted before, the IAV and VS methods seem to be method provides more accuracy with the same cost. The VS method with the arclength parameterization is the most efficient combination.

When the eccentricity is $e = 0.99$, we proceeded similarly. For this more expensive case we integrated for 65 periods (in this way the cost of the experiments is similar to that in the $e = 0.9$ case). In the implicit methods, we chose, for each h, the largest tolerance that made the errors at $t = 4 \times 2\pi, 8 \times 2\pi, 16 \times 2\pi, 32 \times 2\pi, 64 \times 2\pi$ have a linear behaviour. These values are presented in Table 3. We compared the methods by measuring the error at 20 equispaced times along the 65th period. Fig. 2 shows the results obtained. The values of h considered for the implicit methods are presented in Table 3, and for the EAV method were $h = 5 \times 10^{-5}$, 2.5×10^{-5} and 1.25×10^{-5}. The VS method again reveals itself to be more efficient than the adaptive Verlet methods, and for this method the arclength parameterization is better than particle separation.

From these experiments, we concluded that the VS method, in spite of the extra cost per step, is a better choice than the nonsymplectic methods IAV and EAV. It may be thought that the EAV method due to its explicit character would be of interest in large complicated problems. To examine this point we programmed the methods IAV, EAV, VS for the 256-particle molecular dynamics problem in [12,13]. The observed efficiency of the three methods was very similar, with EAV slightly better than the implicit methods. However, none of these three methods improved on the standard, constant step size Verlet method. Perhaps this could have been expected because in a problem with many particles close encounters between two particles are very frequent and there is not much scope for varying the step size.

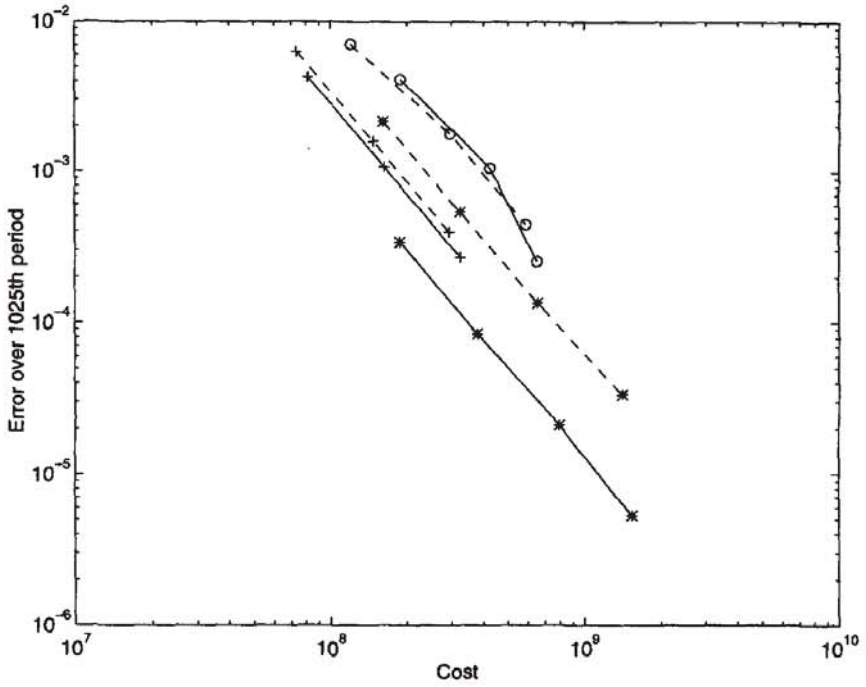

Fig. 1. Kepler's problem $e = 0.9$. Circles correspond to the IAV method, crosses to the EAV method and stars to the VS method. Solid line represents the arclength parameterization and dashed line particle separation parameterization.

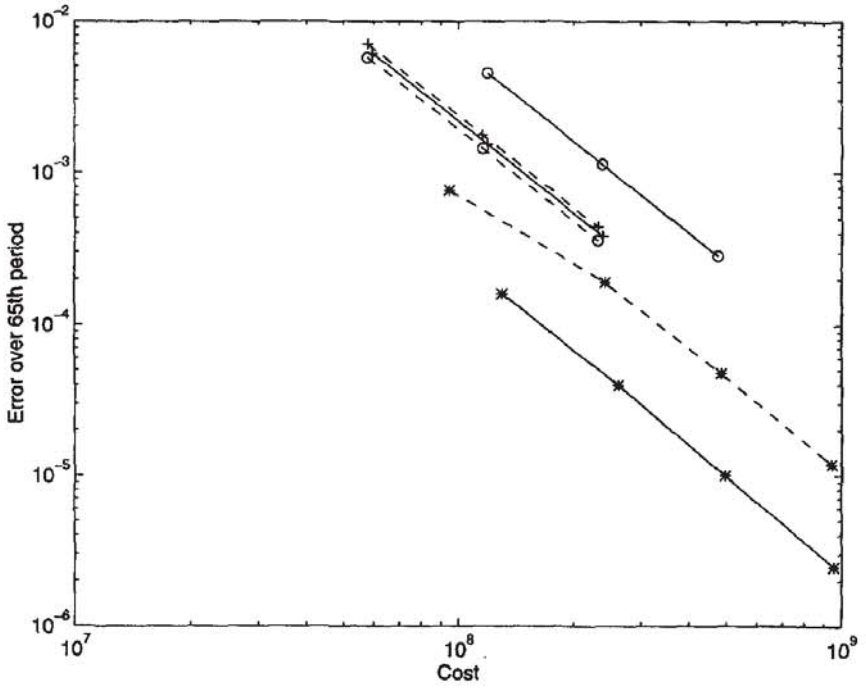

Fig. 2. Kepler's problem $e = 0.99$. Circles correspond to the IAV method, crosses to the EAV method and stars to the VS method. Solid line represents the arclength parameterization and dashed line particle separation parameterization.

Table 3
Tolerances for the implicit methods in Kepler's problem $e = 0.99$

Algorithm		h			
Method	Monitor	1×10^{-4}	5×10^{-5}	2.5×10^{-5}	1.25×10^{-5}
IAV	Arclength		10^{-6}	10^{-6}	10^{-6}
IAV	Particle separation		10^{-2}	10^{-2}	10^{-2}
VS	Arclength	10^{-14}	10^{-15}	10^{-15}	10^{-15}
VS	Particle separation	10^{-13}	10^{-14}	10^{-15}	10^{-15}

Table 4
Tolerances for the Gauss method in Kepler's problem

VG4		h				
Eccentricity	Monitor	0.1024	0.0512	0.0256	0.0128	0.0064
0.9	Arclength	10^{-13}	10^{-13}	10^{-13}	10^{-14}	10^{-15}
0.9	Particle separation	10^{-13}	10^{-13}	10^{-14}	10^{-15}	10^{-15}
0.99	Arclength	10^{-11}	10^{-11}	10^{-12}	10^{-14}	10^{-14}
0.99	Particle separation	10^{-13}	10^{-13}	10^{-14}	10^{-14}	10^{-14}

In fact, for many-particle problems the technique of multiple time steps [1] is probably a better choice than variable time steps.

3.2. *Fourth-order methods*

In this subsection we report the numerical results obtained with the fourth-order integrators introduced in Section 2. Again we considered Kepler's problem with the initial conditions already mentioned in the previous subsection. In the figures we have also included for comparison purposes the earlier results generated with the VS method.

With the variable-step Gauss method VG4 we have proceeded as for the second order implicit schemes. For each value of the step size h we have chosen the largest value of the tolerance for which linear error growth is observed (see Table 4). The numerical results shown in the figures correspond to these values of the tolerances.

In the figures we use circles for the fixed step size symplectic method SF4, $\times$ signs for the variable step size code of Dormand and his coworkers DPV4, crosses for the variable step size Gauss integrator VG4 and stars for the second order variable step size symplectic method VS. As in the previous figures solid line corresponds to the arclength parameterization and dashed line to particle separation.

In Fig. 3 we show error against cost for eccentricity 0.9. We observe that the best second order method (the VS code with the arclength parameterization) is not competitive with the fourth order integrators (not even with a fixed step method). On the other hand, a comparison among the fourth-order schemes shows a better performance of the codes which use variable step size. The most efficient methods are

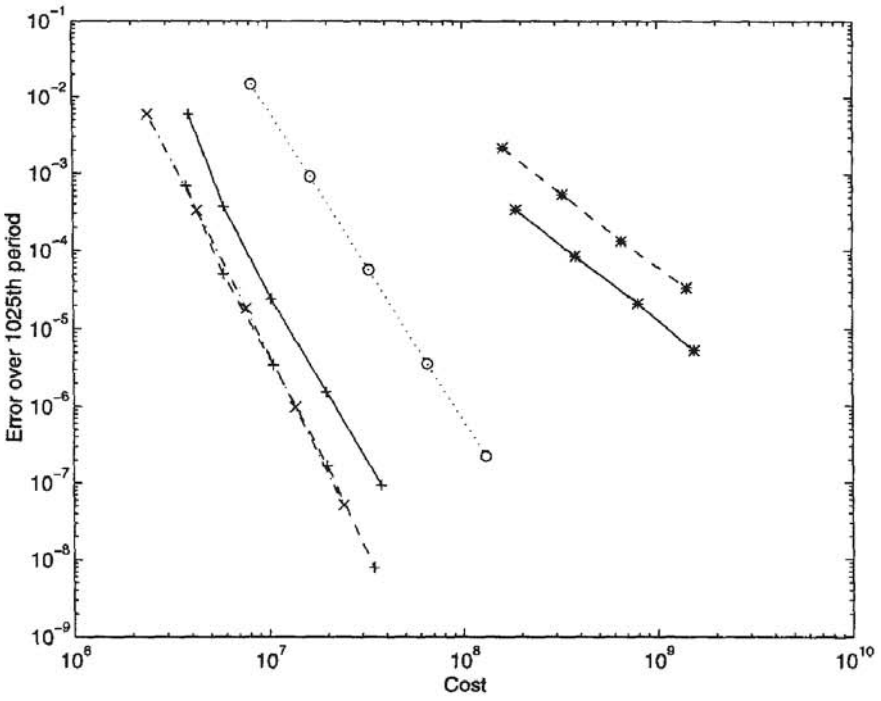

Fig. 3. Kepler's problem $e = 0.9$. Circles correspond to the SF4 method, $\times$ signs to the DPV4 method, crosses to the VG4 method and stars to the VS method. Solid line represents the arclength parameterization and dashed line particle separation parameterization.

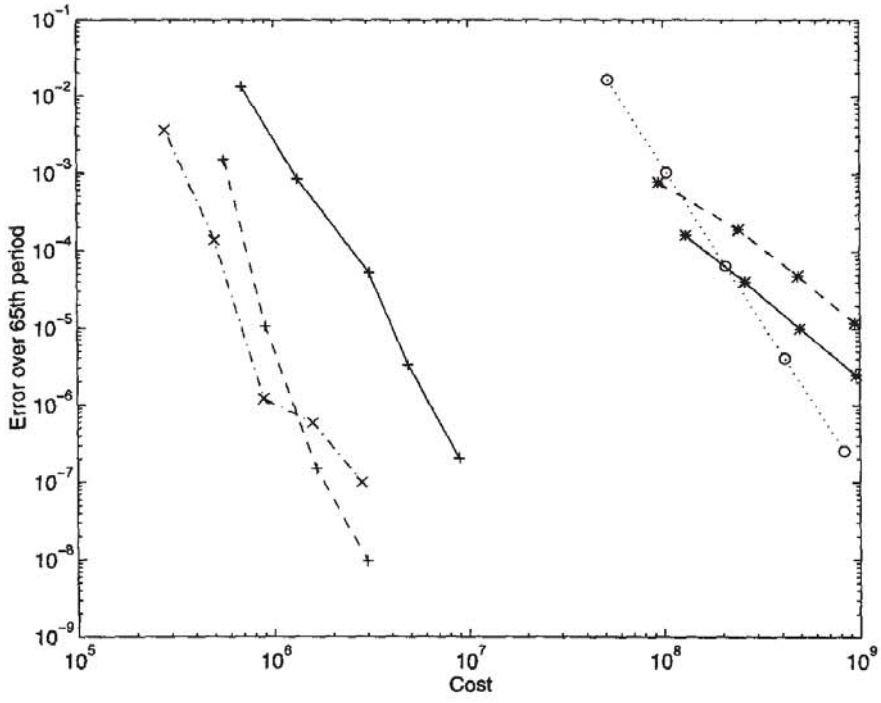

Fig. 4. Kepler's problem $e = 0.99$. Circles correspond to the SF4 method, $\times$ signs to the DPV4 method, crosses to the VG4 method and stars to the VS method. Solid line represents the arclength parameterization and dashed line particle separation parameterization.

the optimized explicit variable step Runge–Kutta–Nyström code DPV4 and the variable-step Gauss integrator VG4 with particle separation. The performance of these two methods is virtually the same.

In Fig. 4 we have represented the results corresponding to eccentricity 0.99. The first aspect to emphasize is that, for this higher eccentricity, the variable step second order symplectic integrator VS (which is implicit) can be more efficient than a fixed step size symplectic explicit method of order four. The necessity of using variable step increases with the eccentricity of the orbit, as expected. On the other hand, a comparison between the variable step fourth-order methods shows that for errors larger than 10^{-6} the explicit Runge–Kutta–Nyström scheme DPV4 is the most efficient of the methods being compared. When smaller errors are required, the variable-step Gauss method VG4 with particle separation is preferable. Notice that in Fig. 4 the errors correspond to the 65th period. For longer times the advantage of the symplectic variable step size method against the explicit Runge–Kutta–Nyström integrator becomes larger due to the different error growth, which is linear for the symplectic scheme (see Appendix A) and quadratic for the Runge–Kutta–Nyström method [2].

4. Conclusions

Our experiments and those in [6] have convinced us that there is much potential for combining geometric integration and variable time steps. It seems likely that for Hamiltonian problems a code based on high-order Gauss formulae with Poincaré transformations may outperform standard software. In the development of such Gauss code much attention should be given to the choice of the monitoring function, which significantly affects the efficiency of the overall algorithm.

The explicit technique developed by Leimkuhler [11] has the advantage of its simplicity and exhibits the good features of geometric integrators. However we think that, in very large problems, multiple time step techniques are a better alternative and, in smaller problems, one should use higher-order formulae.

The implicit technique of Leimkuhler is very similar to the method based on the Lobatto IIIA-B symplectic pair and Poincaré transformations. However the small correction at each step that renders the method symplectic leads to an improved performance. Besides, the Poincaré transformation can be used with arbitrary, possibly high-order, symplectic methods, while the approach in [9] seems limited to variants of the Verlet algorithm.

Acknowledgements

This research has been supported by project DGICYT PB95-0705.

Appendix A

In the numerical experiments we used the fact that the methods IAV, EAV, VS and VG4 lead to linear error growth when integrating Kepler's problem. For IAV and EAV this linear error growth was proved in [3]. In this appendix we prove linear error growth for VS, VG4 and any other symplectic method combined with the Poincaré transformation. Even though the result is presented for Kepler's problem it holds in more general situations listed in [3].

Theorem 1. *If the differential system* (4), (6) *associated with Kepler's problem is integrated with a fixed step size symplectic method of order r, the error grows linearly with time up to* $O(h^{2r})$ *terms. More precisely, for* $r \leqslant m \leqslant 2r-1$, $\|e_m^{(N)}\| = O(N)$ *as* $N \to \infty$, *where* $e_m^{(N)}$ *is the coefficient of* h^m *in the asymptotic expansion of the global error after N periods.*

Proof. Let us denote by (p_n, q_n, t_n) the numerical approximation to the solution of (4), (6) at time level $\tau_n = nh$. As in the proof of Lemma 4.4 in [3], let us assume that $t_n = T$, where T is the period of the periodic solution of (1) with initial condition (p_0, q_0). Backward error analysis results for symplectic integrators with fixed step size [5,16] ensure the existence of a perturbed Hamiltonian $\widetilde{K} = K + O(h^r)$, such that

$$(p_n, q_n) - (\widetilde{p}(\tau_n), \widetilde{q}(\tau_n)) = O(h^{2r}),$$

where $(\widetilde{p}(\tau), \widetilde{q}(\tau))$ is the exact solution of the Hamiltonian system with Hamiltonian function $\widetilde{K}$ and initial condition (p_0, q_0). As $\widetilde{K}$ is a conserved quantity for the perturbed solution, then

$$\widetilde{K}(p_n, q_n) = \widetilde{K}(p_0, q_0) + O(h^{2r}). \tag{A.1}$$

Furthermore, without loss of generality we can assume that $\widetilde{K}(p_0, q_0) = K(p_0, q_0) = 0$. Let us define

$$\widetilde{H}(p, q) = \frac{\widetilde{K}(p, q)}{g(p, q)} + H_0. \tag{A.2}$$

By (A.1) and (A.2),

$$\widetilde{H}(p_n, q_n) - \widetilde{H}(p_0, q_0) = O(h^{2r})$$

and then, as $(p_0, q_0) = (p(t_n), q(t_n))$, $\widetilde{G}_0^{\mathrm{T}} e^{(1)} = O(h^{2r})$, where $\widetilde{G}_0$ is the gradient of the perturbed Hamiltonian $\widetilde{H}$ evaluated at the initial condition and $e^{(1)}$ is the error in the solution of (1) after the first period. It follows that for the exact Hamiltonian

$$H(p_n, q_n) - H_0 = (G_0 - \widetilde{G}_0)^{\mathrm{T}} e^{(1)} + \widetilde{G}_0^{\mathrm{T}} e^{(1)} + O(h^{2r}) = O(h^{2r}).$$

Then the variable step size scheme used for the numerical integration of (1) preserves, after the first period, the energy H up to $O(h^{2r})$ terms. This preservation is enough in order to apply Lemma 4.4 and Theorem 4.1 in [3] which ensure linear error growth up to $O(h^{2r})$ terms.

References

[1] J.J. Biesiadecki and R.D. Skeel, Dangers of multiple time step methods, *J. Comput. Phys.* 109 (1993) 318–328.

[2] M.P. Calvo and J.M. Sanz-Serna, The development of variable-step symplectic integrators with application to the two-body problem, *SIAM J. Sci. Comput.* 14 (1993) 936–952.

[3] B. Cano and J.M. Sanz-Serna, Error growth in the numerical integration of periodic orbits, with application to Hamiltonian and reversible systems, *SIAM J. Numer. Anal.* 34 (1997) 1391–1417.

[4] J.R. Dormand, M.E.A. El-Mikkawy and P.J. Prince, Families of Runge–Kutta–Nyström formulae, *IMA J. Numer. Anal.* 7 (1987) 235–250.

[5] E. Hairer, Backward analysis of numerical integrators and symplectic methods, *Ann. Numer. Math.* 1 (1994) 107–132.

[6] E. Hairer, Variable time step integration with symplectic methods, *Appl. Numer. Math.* 25 (1997) 219–227.

[7] E. Hairer and D. Stoffer, Reversible long-term integration with variable step sizes, *SIAM J. Sci. Comput.* 18 (1997) 257–269.
[8] N.J. Higham, *Accuracy and Stability of Numerical Algorithms* (SIAM, Philadelphia, PA, 1996).
[9] W. Huang and B. Leimkuhler, The adaptive Verlet method, *SIAM J. Sci. Comput.* 18 (1997) 239–256.
[10] P. Hut, J. Makino and S. McMillan, Building a better leapfrog, *Astrophys. J.* 443 (1995) L93–L96.
[11] B. Leimkuhler, Reversible adaptive regularization I: Perturbed Kepler motion and classical atomic trajectories, DAMTP Technical Report 1997/NA08, University of Cambridge (1997).
[12] M.A. López-Marcos, J.M. Sanz-Serna and J.C. Díaz, Are Gauss–Legendre methods useful in molecular dynamics?, *J. Comput. Appl. Math.* 67 (1996) 173–179.
[13] M.A. López-Marcos, J.M. Sanz-Serna and R.D. Skeel, Explicit symplectic integrators using Hessian–vector products, *SIAM J. Sci. Comput.* 18 (1997) 223–238.
[14] S. Reich, Backward error analysis for numerical integrators, Preprint SC 96-21, Konrad-Zuse-Zentrum, Berlin (1996).
[15] J.M. Sanz-Serna, Geometric integration, in: I.S. Duff and G.A. Watson, eds., *The State of the Art in Numerical Analysis* (Clarendon Press, Oxford, 1997) pp. 121–143.
[16] J.M. Sanz-Serna and M.P. Calvo, *Numerical Hamiltonian Problems* (Chapman and Hall, London, 1994).
[17] R.D. Skeel and C.W. Gear, Does variable step size ruin a symplectic integrator?, *Physica D* 60 (1992) 311–313.
[18] D. Stoffer, Variable steps for reversible integration methods, *Computing* 55 (1995) 1–22.
[19] G. Sun, Symplectic partitioned Runge–Kutta methods, *J. Comput. Math.* 11 (1993) 365–372.