Energy-preserving numerical schemes of high accuracy for one-dimensional Hamiltonian systems

Jan L Cieśliński and Bogusław Ratkiewicz

Uniwersytet w Białymstoku, Wydział Fizyki, ul. Lipowa 41, 15-424 Białystok, Poland
E-mail: janek@alpha.uwb.edu.pl and bograt@poczta.onet.pl

Received 27 December 2010, in final form 24 February 2011
Published 21 March 2011
Online at stacks.iop.org/JPhysA/44/155206

Abstract
We present a class of non-standard numerical schemes which are modifications of the discrete gradient method. They preserve the energy integral exactly (up to the round-off error). The considered class contains locally exact discrete gradient schemes and integrators of arbitrary high order. In numerical experiments we compare our integrators with some other numerical schemes, including the standard discrete gradient method, the leap-frog scheme and a symplectic scheme of fourth order. We study the error accumulation for a very long time and the conservation of the energy integral.

PACS numbers: 45.10.—b, 02.60.Cb, 02.70.—c, 02.70.Bf
Mathematics Subject Classification: 65P10, 65L12

1. Introduction

Geometric numerical integration consists in preserving geometric, structural and physical properties of the considered differential equations. Our aim is to improve the accuracy of geometric integrators, modifying them in an appropriate way, without losing their excellent qualitative properties (including the long-time behaviour, stability and the energy conservation). In this paper, we focus on the discrete gradient scheme for one-dimensional Hamiltonian systems, i.e. Hamiltonian system with one degree of freedom.

Discrete gradient numerical schemes have been introduced many years ago in order to integrate numerically $N$-body systems of classical mechanics with possible applications in molecular dynamics and celestial mechanics [19] (see also [14, 16, 18, 29]). Discrete gradient schemes preserve exactly (up to round-off errors) the total energy. Some of them also preserve other first integrals, e.g. angular momentum. More recently, discrete gradient methods have been extended and developed in the context of geometric numerical integration [21]. Quispel and his coworkers constructed numerical integrators preserving all integrals of motion of a given system of ordinary differential equations [22, 23, 27, 28].

1 Permanent address: I LO, ul. Śródmieście 31, 16-300 Augustów, Poland.
In general, geometric numerical integrators are very good in preserving qualitative features of simulated differential equations but it is not easy to enhance their accuracy. Symplectic algorithms can be improved using appropriate splitting methods, see, e.g., [2, 20, 30]. Energy-preserving integrators of high order are less common; one can mention here the average vector field method which has been improved by bootstrapping [24] and collocation [15]. Our research is concentrated on improving the efficiency of the standard discrete gradient method (which is not symplectic).

In this paper, we continue our earlier research [9–11], extending the theoretical framework on arbitrary one-dimensional Hamiltonian systems. Numerical experiments are carried out in the case of the simple pendulum equation for extremely long times (we test even 100 millions of periods). We study the accuracy of our new methods, namely the accumulation of the global error and conservation of the energy integral.

2. Non-standard discrete gradient schemes

In this paper, we confine ourselves to one-dimensional Hamiltonian systems

\[
\dot{x} = H_p, \quad \dot{p} = -H_x, \tag{2.1}
\]

where \( H = H(x, p) \) is a given function, subscripts denote partial differentiation and the dot denotes the total derivative with respect to \( t \). The Hamiltonian \( H(x, p) \) is an integral of motion (the energy integral).

We consider the following class of non-standard (compare [25]) discrete gradients schemes:

\[
\begin{align*}
\frac{x_{n+1} - x_n}{\delta_n} & = \frac{H(x_{n+1}, p_{n+1}) + H(x_n, p_{n+1}) - H(x_{n+1}, p_n) - H(x_n, p_n)}{2(p_{n+1} - p_n)}, \\
\frac{p_{n+1} - p_n}{\delta_n} & = \frac{H(x_n, p_{n+1}) + H(x_n, p_n) - H(x_{n+1}, p_{n+1}) - H(x_{n+1}, p_n)}{2(x_{n+1} - x_n)},
\end{align*} \tag{2.2}
\]

where \( \delta_n \) is an arbitrary positive function of \( h, x_n, p_n, x_{n+1}, p_{n+1}, \) etc (the time step is denoted by \( h \)). The subscript \( n \) indicates that \( \delta_n \) may depend on the step \( n \). On the right-hand side we use the symmetrized version of the coordinate increment discrete gradient [18].

In the separable case, i.e. \( H = T(p) + V(x) \), the scheme (2.2) becomes

\[
\begin{align*}
\frac{x_{n+1} - x_n}{\delta_n} & = \frac{T(p_{n+1}) - T(p_n)}{p_{n+1} - p_n}, \\
\frac{p_{n+1} - p_n}{\delta_n} & = \frac{V(x_{n+1}) - V(x_n)}{x_{n+1} - x_n}.
\end{align*} \tag{2.3}
\]

In numerical experiments we mostly test the case \( T(p) = \frac{1}{2}p^2 \), where further simplification occurs, see (4.1).

The system (2.2) is a consistent approximation of (2.1) if we add the condition

\[
\lim_{h \to 0} \frac{\delta_n}{h} = 1. \tag{2.4}
\]

The case \( \delta_n = h \) yields the standard discrete gradient method (GR) [14, 18, 19, 29].

**Theorem 2.1.** The numerical scheme (2.2) preserves the energy integral exactly (up to round-off error), i.e. \( H(x_{n+1}, p_{n+1}) = H(x_n, p_n) \).

**Proof.** Multiplying left-hand sides of both equations (2.2) by their denominators we get identical expressions on the left-hand side. Therefore, numerators of right-hand sides are equal. \qed

2
Therefore, any $\delta_n$ satisfying non-restrictive condition (2.4) yields an energy-preserving numerical scheme. The main idea of this paper consists in finding $\delta_n$ such that the resulting numerical scheme is better than the standard gradient method. We consider and test two possibilities. First, we consider the so-called locally exact discretizations (section 4), and then we show that class (2.2) contains integrators of arbitrary high order. The corresponding $h$-series for $\delta_n$ is defined in a recurrent way (section 5).

3. Exact discretization

We consider an ordinary differential equation (ODE) with a general solution $x(t)$ (satisfying the initial condition $x(t_0) = x_0$), and a difference equation with the general solution $x_n$. The difference equation is the exact discretization of the ODE if $x_n = x(t_n)$.

It is well known that any linear ODE with constant coefficients admits the exact discretization in an explicit form [26], see also [1, 8, 25]. We summarize these results as follows.

**Theorem 3.1.** Any linear equation with constant coefficients, represented in the matrix form by

\[
\frac{dx}{dt} = Ax + b, \tag{3.1}
\]

(where $x = x(t) \in \mathbb{R}^n$, $b = \text{const} \in \mathbb{R}^n$ and $A$ is a constant invertible $n \times n$ matrix) admits the exact discretization given by

\[
x_{n+1} = e^{hA}x_n + (e^{hA} - I)A^{-1}b, \tag{3.2}
\]

where $h = t_{n+1} - t_n$ is the time step and $I$ is the identity matrix.

**Proof.** The general solution of (3.1) is given by

\[
x(t) = e^{(t-t_0)A}(x(t_0) + A^{-1}b) - A^{-1}b.
\]

Taking into account that $x_n = x(t_n)$ and, in particular, $x_0 = x(t_0)$, we get

\[
x_n = e^{(t_n-t_0)A}(x(t_0) + A^{-1}b) - A^{-1}b,
\]

\[
x_{n+1} = e^{(t_{n+1}-t_n)A}(x(t_0) + A^{-1}b) - A^{-1}b = e^{hA}(x_n + A^{-1}b) - A^{-1}b,
\]

which ends the proof. \(\square\)

**Example 3.2.** Exponential growth equation: $\dot{x} = ax$. Exact discretization: $x_{n+1} = e^{ah}x_n$ (a geometric series). Equivalent form:

\[
\frac{x_{n+1} - x_n}{\delta(h)} = ax_n, \quad \delta(h) = \frac{e^{ah} - 1}{a}. \tag{3.3}
\]

Note that $\lim_{h \to 0} \frac{\delta(h)}{h} = 1$.

**Example 3.3.** Harmonic oscillator: $\ddot{x} + \omega^2 x = 0$, $p = \dot{x}$. Exact discretization:

\[
x_{n+1} - 2\cos(\omega h)x_n + x_{n-1} = 0, \quad p_n = \frac{x_{n+1} - \cos(\omega h)x_n}{\omega^{-1}\sin(\omega h)}. \tag{3.4}
\]

Equivalent form:

\[
\frac{x_{n+1} - 2x_n + x_{n-1}}{\delta^2(h)} + \omega^2 x_n = 0, \quad \delta(h) = \frac{2}{\omega} \sin \frac{\omega h}{2}. \tag{3.5}
\]

Note that $\delta(h) \approx h$ for $h \approx 0$. 

3
Exact discretization seems to be of limited value because, in order to apply it, we need
to know the explicit solution of the considered system. However, there exist non-trivial
applications of exact discretizations. In the case of the classical Kepler problem we succeeded
in using the exact discretization of the harmonic oscillator in two different ways, obtaining
numerical integrators preserving all integrals of motion and trajectories [4–6]. Another fruitful
direction is associated with the so-called locally exact discretizations [5, 7, 10], see the next
section.

4. Locally exact discrete gradient schemes

First, we recall our earlier results concerning the case
\[ H = \frac{1}{2} p^2 + V(x), \]
see [9, 10]. We tested
the following class of numerical integrators
\[
\begin{align*}
\frac{x_{n+1} - x_n}{\delta_n} &= \frac{1}{2} (p_{n+1} + p_n), \\
\frac{p_{n+1} - p_n}{\delta_n} &= -\frac{V(x_{n+1}) - V(x_n)}{x_{n+1} - x_n},
\end{align*}
\]
where \( \delta_n \) is a function defined by
\[
\delta_n = \frac{2}{\omega_n} \tan \frac{h \omega_n}{2}, \quad \omega_n = \sqrt{V''(x)}, \tag{4.2}
\]
and, in general, \( x \) may depend on \( n \). For simplicity, we formally assume \( V''(x) > 0 \). However,
in the case of non-positive \( V''(x) \) one can use the same formula (either for \( V''(x) < 0 \), the
imaginary unit cancels, or for \( V''(x) = 0 \), we compute the limit \( \omega_n \to 0 \) obtaining
\( \delta_n = h \)); for details and final results see [10].

The simplest choice is \( \bar{x} = x_0 \), where \( V'(x_0) = 0 \) (small oscillations around the stable
equilibrium). In this case \( \delta_n \) does not depend on \( n \). The resulting scheme was first presented
in [9]; here we propose to name it MOD-GR. In [10] we considered the case \( \bar{x} = x_n \) (which
will be called GR-LEX) and its symmetric (time-reversible) modification \( \bar{x} = \frac{1}{2}(x_n + x_{n+1}) \n\) (GR-SLEX). In both cases \( \bar{x} \) is changed at every step.

Definition 4.1. A numerical scheme \( x_{n+1} = \Psi(x_n, h) \) for an autonomous equation
\( \dot{x} = F(x) \) is locally exact if its linearization around any fixed \( \bar{x} \) is identical with the exact discretization
of the differential equation linearized around \( \bar{x} \).

We use local exactness as a criterion to select numerical schemes of high accuracy from
a family of non-standard integrators, e.g. from (2.2). Our working algorithm to derive such
‘locally exact modifications’ of numerical integrators of the form (2.2) assumes that \( \delta_n \) depends
only on \( \bar{x}, \bar{p} \) (or, in the more general case, on \( \bar{x} \) and \( h \)). The following theorem extends results
of [5, 10] on the case of the general time-independent Hamiltonian \( H = H(x, p) \).

Theorem 4.2. The discrete gradient scheme (2.2) with
\[
\delta_n = \frac{2}{\omega_n} \tan \frac{h \omega_n}{2}, \quad \omega_n = \sqrt{H_{xx} H_{pp} - H_{xp}^2}, \tag{4.3}
\]
(where \( \omega_n \) is evaluated at \( \bar{x}, \bar{p} \)) is locally exact.

Proof. We have to linearize the continuous system (2.1), and then to find the exact discretization
of the obtained linearization. Therefore, we put \( x = \bar{x} + \xi, \quad p = \bar{p} + \eta \) into (2.1) and neglect
all terms of order greater than 2. Thus, we get
\[
\dot{\xi} = H_p + H_{p\xi} \xi + H_{p\eta} \eta, \quad \dot{\eta} = -H_t - H_{t\xi} \xi - H_{t\eta} \eta. \tag{4.4}
\]
The exact discretization of the system (4.4) is given by
\[
\begin{pmatrix}
\xi_{n+1} \\
\eta_{n+1}
\end{pmatrix} = e^{hA} \begin{pmatrix}
\xi_n \\
\eta_n
\end{pmatrix} + (e^{hA} - I)A^{-1}b
\]
(4.5)

(compare theorem 3.1), where
\[
A = \begin{pmatrix}
H_{xp} & H_{pp} \\
-H_{xx} & -H_{xp}
\end{pmatrix}, \quad b = \begin{pmatrix}
H_p \\
-H_x
\end{pmatrix}.
\]
(4.6)

We proceed to the linearization of the discrete system (2.2). We substitute
\[
x_n = \bar{x} + \delta_n, \quad \eta_n = \bar{p} + \eta_n,
\]
and assume that \(\delta_n\) depends only on \(\bar{x}, \bar{p}\) and \(h\) (which is equivalent to taking only the first, constant, term of the Taylor expansion of \(\delta_n\) with respect to \(\xi_n, \eta_n\)). Then, we linearize the system (2.2) around \(\bar{x}, \bar{p}\) (neglecting terms of at least the second order with respect to \(\delta_n\)), obtaining
\[
\begin{align*}
\xi_{n+1} - \xi_n &= H_p + \frac{1}{2} H_{xp} (\xi_n + \xi_{n+1}) + \frac{1}{2} H_{pp} (\eta_n + \eta_{n+1}) , \\
\eta_{n+1} - \eta_n &= -H_x - \frac{1}{2} H_{xx} (\xi_n + \xi_{n+1}) - \frac{1}{2} H_{xp} (\eta_n + \eta_{n+1}) ,
\end{align*}
\]
(4.8)

where partial derivatives \(H_x, H_p, H_{xx}, H_{xp}\) and \(H_{pp}\) are evaluated at \(\bar{x}, \bar{p}\).

After simple algebraic manipulations we rewrite this linear system in the form
\[
\begin{pmatrix}
\xi_{n+1} \\
\eta_{n+1}
\end{pmatrix} = M \begin{pmatrix}
\xi_n \\
\eta_n
\end{pmatrix} + w,
\]
(4.9)

where
\[
M = \frac{1}{1 + \frac{1}{4} \omega_n^2 \delta_n^2} \begin{pmatrix}
1 + \delta_n H_{xp} - \frac{1}{2} \omega_n^2 \delta_n^2 & \delta_n H_{pp} \\
-\delta_n H_{xx} & 1 - \delta_n H_{xp} - \frac{1}{2} \omega_n^2 \delta_n^2
\end{pmatrix},
\]
\[
w = \frac{\delta_n b}{1 + \frac{1}{4} \omega_n^2 \delta_n^2} + \frac{\frac{1}{2} \delta_n^2 A b}{1 + \frac{1}{4} \omega_n^2 \delta_n^2}.
\]
(4.10)

and \(\omega_n\) is defined by (4.3). Taking into account (4.6), we get
\[
M = e^{hA}, \quad w = (e^{hA} - I)A^{-1}b.
\]
(4.12)

The proof reduces to showing that the system (4.12) is identically satisfied if \(\delta_n\) is given by (4.2). We easily verify that
\[
A^2 = -\omega_n^2 I, \quad \omega_n^2 = H_{xx} H_{pp} - H_{xp}^2.
\]
(4.13)

Hence
\[
e^{hA} = \cos h\omega_n + \omega_n^{-1}A \sin h\omega_n.
\]
(4.14)

The second equation of (4.12) is satisfied for \(\delta_n\) of any form. Indeed, using the first equation of (4.11) and then the first equation of (4.13), we get
\[
(M - I)A^{-1}b = \frac{\delta_n - \frac{1}{2} \omega_n^2 \delta_n^2 A^{-1}}{1 + \frac{1}{4} \omega_n^2 \delta_n^2} b = \delta_n + \frac{1}{2} \omega_n^2 A b = w.
\]
(4.15)
which is equivalent to the second equation of (4.12) provided that the first equation of (4.12) holds. Finally, the first equation of (4.12) is satisfied if and only if

$$1 - \frac{1}{2} \omega_n^2 \delta_n^2 = \cos h \omega_n, \quad \frac{\delta_n}{1 + \frac{1}{2} \omega_n^2 \delta_n^2} = \frac{\sin h \omega_n}{\omega_n},$$

(4.16)

where we took into account (4.11) and (4.14). From the first equation we compute

$$\frac{1}{1 + \frac{1}{2} \omega_n^2 \delta_n^2} = \frac{1 + \cos h \omega_n}{2} = \cos^2 \frac{h \omega_n}{2} \quad (4.17)$$

and substituting it into the second equation of (4.16), we get (4.3). □

**Remark 4.3.** Assuming $\bar{x} = x_n, \bar{p} = p_n$ we get a numerical scheme called GR-Lex, while the choice $\bar{x} = \frac{1}{2} (x_n + x_{n+1}), \bar{p} = \frac{1}{2} (p_n + p_{n+1})$ yields another scheme, named GR-SLEX. The system (2.1) is symmetric (time reversible). The numerical scheme GR-SLEX preserves this property, while GR-Lex does not preserve it.

The discrete gradient schemes GR and MOD-GR are of second order. Locally exact discrete gradient schemes have higher order: GR-Lex is of third order and GR-SLEX is of fourth order, see [10]. In the next section we show how to construct discrete gradient schemes of any order.

### 5. Discrete gradient schemes of Nth order

We consider the family (2.2) of non-standard discrete gradient schemes for the Hamiltonian system (2.1). The family is parameterized by a single function $\delta_n$ and this function can be expressed by $x_n, p_n, x_{n+1}, p_{n+1}$ as follows:

$$\delta_n = \frac{2(x_n - x_{n+1})(p_{n+1} - p_n)}{H(x_{n+1}, p_{n+1}) + H(x_n, p_{n+1}) - H(x_{n+1}, p_n) - H(x_n, p_n)}. \quad (5.1)$$

Replacing here $x_{n+1}, p_{n+1}$ by the exact solution $x(t_{n+1}), p(t_{n+1})$ we formally obtain $\delta_n$ corresponding to the exact integrator. In practice, we can replace $x_{n+1}, p_{n+1}$ by truncated Taylor expansions (and truncate the final result).

Therefore, we take Taylor expansions truncated by neglecting terms of order higher than $N$ (see appendix A, formulae (A.1)) and compute

$$\frac{2([x]_n - x)([p]_{n+1} - p_n)}{H([x]_{n+1}, [p]_{n+1}) + H(x, [p]_{n+1}) - H([x]_{n+1}, p_n) - H(x, p_n)} = \sum_{k=0}^{N} a_k h^k + O(h^{N+1}),$$

where coefficients $a_k$ are functions of $x_n$ and $p_n$. Then, truncating the obtained result, we define

$$\delta_n^{[N]} = \sum_{k=1}^{N} a_k (x_n, p_n) h^k. \quad (5.2)$$

The first few coefficients read

$$a_1 = 1, \quad a_2 = 0, \quad a_3 = \frac{1}{12} (H_{xx} H_{pp} - H_{xp}^2 - H_{x} H_{app} - H_p H_{xp}),$$

$$a_4 = \frac{1}{24} (H_{xx}^2 H_{ppp} - H_{x}^2 H_{xpp} + H_{x} H_{pp} H_{xxx} - H_p H_{xx} H_{pp} - H_{xp} H_{xxx} + 3 H_{x} H_{xp} H_{pp}), \quad (5.3)$$
Figure 1. Error in the energy as a function of time ($t = nh$, $h = 0.25$) for the simple pendulum (initial data: $x_0 = 0$, $p_0 = 1.8$, exact energy $E_{ex} = 0.62$).

where all partial derivatives are evaluated at $x = x_n$, $p = p_n$. In the separable case, $H = T(p) + V(x)$, the formulae simplify

\[ a_1 = 1, \quad a_2 = 0, \quad a_3 = \frac{1}{12} T_{pp} V_{xx}, \quad a_4 = \frac{1}{24} (T_p T_{pp} V_{3x} - V_x V_{xx} T_{3p} - V_x V_{xx} T_{pp}), \]

\[ a_5 = \frac{1}{720} \left( 9 V_x^2 V_{xx} T_{4p} + 9 T_p^2 T_{pp} V_{4x} - 12 V_x V_{3x} T_{pp} - 12 T_p V_x V_{2xx} + 6 T_{pp} V_{2xx}^2 - 4 V_x V_{xx} T_{pp} V_{3x} \right). \]  

(5.4)

where $V_{kx}$ denotes the $k$th derivative of $V$ with respect to $x$, etc. The case $H = \frac{1}{2} p^2 + V(x)$ is discussed in more detail in [11], where explicit formulae for $\delta_n^{(N)}$ for $N \leq 11$ can be found.

The gradient scheme (2.2) with $\delta_n = \delta_n^{(N)}$ is called GR-$N$. Its order is at least $N$, sometimes higher (e.g. GR-1 is of second order). Actually GR-1 and GR-2 are identical with GR.

6. Numerical experiments

The accuracy of high-order discrete gradient schemes was tested on the case of the simple pendulum, $H = \frac{1}{2} p^2 - \cos x$ (for simplicity always assuming $x_0 = 0$). We compared GR-3, GR-7 and GR-LEX with the discrete gradient method (GR), the leap-frog scheme (LF), fourth order explicit Runge–Kutta method (RK-4), high-order Taylor methods (TAY-$N$, see appendix A) and a fourth order symplectic scheme (SP-4, see appendix B). Computing global errors we use the exact solution of the simple pendulum equation, expressed in terms of elliptic integrals.

In previous papers [9–11], we focused on the stability and accuracy of the period (all motions of the pendulum are periodic). Here, we test the global error, accumulated after 120 periods (figures 4 and 5) and the accumulation of error after a very long time (up to $n = 10^8$ steps), figure 6. We also check the preservation of the energy integral by different numerical schemes, figures 1, 2 and 3. Details concerning the solution of implicit equations are the same as in [10], e.g., at every step we iterated until the accuracy $10^{-16}$ was obtained.
Figure 2. Energy as a function of time \((t = nh, \ h = 0.25)\) for the simple pendulum \((x_0 = 0, \ p_0 = 1.8, \ E_{\text{ex}} = 0.62)\). The line \(E = 0.62\) corresponds to GR, GR-LEX, GR-3 and GR-7. Other, scattered, points are produced by TAY-10.

Figure 3. Long time behaviour of the energy error for the simple pendulum. Initial data are the same as in figure 1, \(t = nh, \ h = 0.25\).

Symplectic methods are known to preserve almost exactly the energy integral [16], some positive results in the non-symplectic case are also known [12]. Figure 1 shows how accurate is the preservation of the energy by symplectic integrator SP-4 as compared with TAY-5 (permanent, fast growth of the energy) and RK-4 (energy is decreasing approaching the stable equilibrium value). A high order of a given scheme is not sufficient to assure the conservation of the energy. From the beginning TAY-10 produces small, but permanent, drift of the energy, while all discrete gradient schemes yield almost the exact value of the energy, see figure 2. According to theorem 2.1 all gradient schemes preserve the energy integral exactly (up to
round-off errors). Only after very long time one can note that also discrete gradient schemes have a slight drift of the energy. A curious phenomenon can be observed in figure 3. Gradient schemes and TAY-10 show a linear growth of the energy error (but the energy error of TAY-10 is always greater by six orders of magnitude), while the energy errors of symplectic schemes, LF and SP-4, vary in a large range but do not show any systematic time dependence. However, in the considered time interval, discrete gradient methods preserve the energy more accurately by several orders of magnitude than symplectic integrators like LF or SP-4.

Figures 4 and 5 show \( h \)-dependence of the global error calculated at \( t = 120T_{ex} \), where \( T_{ex} \) is the period of the exact solution (e.g., \( T_{ex} = 6.283 \, 342 \, 396 \) for \( p_0 = 0.02 \) and \( T_{ex} = 9.122 \, 196 \, 555 \) for \( p_0 = 1.8 \)). We observe that, usually, higher order integrators are
more accurate. An important exception is GR-LEX, of third order, which for \( p_0 = 0.02 \) and \( h > 0.3 \) is better than GR-7 (the behaviour of GR-SLEX is almost the same as GR-LEX). However, TAY-10 is clearly the best in this case. Only after very long time evolution TAY-10 becomes less accurate than GR and SP-4, although initially it was comparable with gradient schemes of high order, see figure 6. Several schemes in figure 6 show linear growth (at least for large \( t \)). It has been shown, see [3, 16], that symplectic integrators (under some mild conditions) have linear error growth. Results of our experiments suggest that after sufficiently large time some other schemes (e.g. RK-4, GR, GR-3) also accumulate error linearly. In the case of RK-4 this is a meaningless result, because after so long time the numerical ‘solution’ produced by RK-4 reduces to \( x_n \equiv 0 \). Finally, we point out that until \( n = 10^7 \) the global error of GR-LEX is smaller than the period (and the error of GR-7 is even smaller, by one order of magnitude). The global error of GR is smaller than that of SP-4, not saying about GR-3 or GR-7, see figure 6.

**Figure 6.** Global error as a function of time \( t = nh, h = 0.25 \) for the simple pendulum \((x_0 = 0, p_0 = 2.001)\).

**Table 1.** Computational cost of GR-11 and GR-LEX in comparison to GR. The relative cost of one step as a function of a number of iterations (simple pendulum, Newton’s method).

| Number of iterations | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     |
|----------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| GR-LEX/GR            | 1.29  | 1.19  | 1.15  | 1.12  | 1.10  | 1.08  | 1.07  | 1.07  | 1.06  |
| GR-3/GR              | 1.33  | 1.22  | 1.17  | 1.14  | 1.11  | 1.10  | 1.09  | 1.08  | 1.07  |
| GR-5/GR              | 1.53  | 1.36  | 1.27  | 1.22  | 1.18  | 1.16  | 1.14  | 1.12  | 1.11  |
| GR-7/GR              | 1.81  | 1.55  | 1.41  | 1.33  | 1.28  | 1.24  | 1.21  | 1.19  | 1.17  |
| GR-11/GR             | 2.88  | 2.27  | 1.96  | 1.77  | 1.64  | 1.55  | 1.48  | 1.43  | 1.39  |
| TAY-5/GR             | 1.15  | 0.78  | 0.59  | 0.47  | 0.39  | 0.34  | 0.30  | 0.26  | 0.24  |
| TAY-10/GR            | 3.14  | 2.13  | 1.61  | 1.29  | 1.08  | 0.93  | 0.81  | 0.72  | 0.65  |
| SP-4/GR              | 1.06  | 0.72  | 0.54  | 0.44  | 0.36  | 0.31  | 0.27  | 0.24  | 0.22  |
| LF/GR                | 0.22  | 0.15  | 0.11  | 0.09  | 0.07  | 0.06  | 0.06  | 0.05  | 0.04  |
| RK-4/GR              | 0.20  | 0.14  | 0.10  | 0.08  | 0.07  | 0.06  | 0.05  | 0.05  | 0.04  |
Table 2. Global error evaluated at \( t = 120T_\alpha \) for the simple pendulum (\( x_0 = 0 \)) as a function of \( p_0 \). Time steps are different in order to ensure similar computational cost.

| \( p_0 \) | GR | \( h = 0.25 \) | GR-LEX | \( h = 0.3 \) | GR-11 | \( h = 0.5 \) |
|---|---|---|---|---|---|---|
| 0.10 | 0.11 | 0.000 013 | 0.000 000 015 |
| 0.50 | 0.33 | 0.001 51 | 0.000 000 065 |
| 1.21 | 0.86 | 0.0188 | 0.000 000 050 |
| 1.80 | 0.29 | 0.0421 | 0.000 000 048 |
| 1.999 999 | 1.38 | 0.0382 | 0.000 001 84 |
| 1.999 999 99 | 1.14 | 0.0397 | 0.000 0926 |
| 2.000 000 01 | 3.18 | 0.0160 | 0.000 499 |
| 2.01 | 2.57 | 0.0159 | 0.000 000 169 |
| 2.50 | 0.53 | 0.0103 | 0.000 000 113 |

We point out that the computational cost of GR-LEX is only slightly higher than the cost of GR. Surprisingly enough, the computational cost of GR-\( N \) is also not so high as cumbersome formulae suggest. The crucial point is that \( \delta_0^{[N]} \) is computed outside the iteration loop. The relative cost depends mainly on the average number of iterations per one step, see table 1 (for higher number of iterations all gradient schemes have similar cost). Applying the Newton method we need a small number of iterations (usually 3, sometimes even less). Higher number of iterations (about 6) occurs in the neighbourhood of separatrix (\( p_0 \approx 2 \)). Table 2 compares the efficiency of GR, GR-LEX and GR-11. The advantage of GR-11 is obvious. We have to add, however, that the efficiency of GR-11 decreases if the function \( V(x) \) and, especially, its derivatives are more complicated.

7. Conclusions

Modifications presented in this paper essentially improve the discrete gradient method (in the one-dimensional case) keeping all its advantages. Modified gradient schemes GR-LEX, GR-SLEX, GR-\( N \) have important advantages:

- conservation of the energy integral (up to round-off errors),
- high stability and exact trajectories in the phase space,
- high accuracy (third, fourth and \( N \)th order, respectively),
- very good long-time behaviour of numerical solutions.

One can also use a variable time step, if needed [7]. This is an important advantage in comparison with symplectic methods. The computational cost, although higher after these modifications, is of the same order as the cost of the gradient scheme GR. Therefore, the efficiency of proposed methods is very high.

We point out, however, that numerical schemes (2.2), like all discrete gradient methods, are neither symplectic nor volume-preserving. Most of them, including GR-LEX and GR-\( N \) (\( N > 2 \)), are not symmetric (time reversible). GR and GR-SLEX are symmetric.

In the near future we plan to extend the range of applications of our approach on other numerical integrators and to generalize it on the multidimensional case (the crucial point is that \( h \) is replaced by a matrix instead of a scalar \( \delta_n \)) [7].
Acknowledgment

This research work has been partially supported by the grant no N N202 238637 from the Polish Ministry of Science and Higher Education. Main results of this paper were presented at the conference ‘BIT50 – Trends in Numerical Mathematics’, Lund, 17–20 June 2010.

Appendix A. Explicit Taylor schemes of Nth order

The Nth Taylor method for the system (2.1) is defined by

\[ x_{n+1}^{[N]} = \sum_{k=0}^{N} b_k h^k \frac{k!}{k!}, \quad p_{n+1}^{[N]} = \sum_{k=0}^{N} c_k h^k \frac{k!}{k!}, \]  

where the coefficients \( b_k, c_k \) are computed from Taylor’s expansion of the exact solution (see, for instance, [17], p 18). We assume \( x(t) = x_n, p(t) = p_n, t = t_n \) and expand \( x(t+h) \) and \( p(t+h) \) in the Taylor series:

\[ x(t+h) = \sum_{k=0}^{\infty} b_k h^k \frac{k!}{k!}, \quad b_k = \frac{d^k x(t)}{dt^k} \bigg|_{t=t_n}, \]  

\[ p(t+h) = \sum_{k=0}^{\infty} c_k h^k \frac{k!}{k!}, \quad c_k = \frac{d^k p(t)}{dt^k} \bigg|_{t=t_n}, \]  

where all derivatives are replaced by functions of \( x_n, p_n \) using (2.1) and its differential consequences, e.g.

\[ \dot{x} = \frac{d}{dt} \frac{\partial H}{\partial p} = H_p \dot{x} + H_{pp} \dot{p} = H_p \dot{x} + H_{pp} \dot{p} = H_p \dot{H}_x, \]  

where \( H_x, H_p, H_{xx}, \) etc are evaluated at \( x_n, p_n \). Thus, we get

\[ b_1 = H_p, \quad b_2 = H_p H_{xp} - H_x H_{pp}, \]  

\[ b_3 = H_x^2 H_{pp} + H_{xx} H_p^2 - 2 H_x H_p H_{pp} + H_p H_{xp} = H_x H_{pp} H_{xx} = H_x H_{pp} H_{xx}, \]  

\[ c_1 = -H_x, \quad c_2 = H_x H_{xp} - H_p H_{xx}, \]  

\[ c_3 = -H_x^2 H_{pp} - H_{xx} H_p^2 + 2 H_x H_p H_{pp} - H_x H_{pp} H_{xx} + H_x H_{pp} H_{xx}, \]  

and subsequent coefficients can be easily computed using the total derivative

\[ b_{k+1} = \frac{db_k}{dt} = H_p \frac{\partial b_k}{\partial x} - H_x \frac{\partial b_k}{\partial p}, \quad c_{k+1} = H_p \frac{\partial c_k}{\partial x} - H_x \frac{\partial c_k}{\partial p}. \]  

The formulae (A.5) simplify in the case \( H = T(p) + V(x) \) [11].

Appendix B. Explicit symplectic schemes of 2Mth order

Symplectic explicit integrators of arbitrary even order \( N = 2M \) can be derived by composition methods, see, e.g., [16]. In this section we present results of the pioneering paper [30], confining ourselves to the case \( H = \frac{1}{2} p^2 + V(x) \).

The numerical scheme SP-2M is defined by the following procedure. Having \( x_n, p_n \) we compute the next step, \( x_{n+1}, p_{n+1} \), as follows. We denote \( x_n = x^{(0)} \), \( p_n = p^{(0)} \) and perform \( K+1 \) iterations (where \( K = 3^{M-1} \))

\[ x^{[i]} = x^{[i-1]} + H_{xM}^{(i)} p^{[i-1]}, \quad p^{[i]} = p^{[i-1]} - H_{pM}^{(i)} V'(x^{[i]}), \]  

where \( H_{xM}^{(i)}, H_{pM}^{(i)} \) are evaluated at \( x^{[i]}, p^{[i]} \).

Other coefficients can be computed using the explicit formulas (A.4)–(A.6).
where \( c_M^{(1)}, d_M^{(1)} \) \((i = 1, 2, \ldots, K + 1)\) have to be carefully computed (see below) in order to ensure the required order. Then we identify \( x^{[K+1]} = x_{n+1}, p^{[K+1]} = p_{n+1} \).

The coefficients \( c_M^{(1)}, d_M^{(1)} \) are computed recursively. First, all coefficients for \( M = 1 \) are given by

\[
c_1^{(1)} = \frac{1}{2}, \quad d_1^{(1)} = 1, \quad c_1^{(2)} = \frac{1}{2}, \quad d_1^{(2)} = 0. \tag{B.2}
\]

Then, we express coefficients \( c_{m+1}^{(i)}, d_{m+1}^{(i)} \) by coefficients \( c_m^{(i)}, d_m^{(i)} \):

\[
\begin{align*}
d_{m+1}^{[2i]} &= d_{m+1}^{[2i+1]} = y_m d_m^{[i]} \quad (i = 1, \ldots, k), \\
d_{m+1}^{[k+1]} &= (1 - 2y_m) d_m^{[i]} \quad (i = 1, \ldots, k), \\
d_{m+1}^{[k+1]} &= 0, \\
c_{m+1}^{[i+1]} &= c_{m+1}^{[2i+1]} = y_m c_m^{[i+1]} \quad (i = 1, 2, \ldots, k), \\
c_{m+1}^{[k+1]} &= (1 - 2y_m)c_m^{[k+1]} \quad (i = 1, 2, \ldots, k - 1), \\
c_{m+1}^{[k+1]} &= (1 - y_m)(c_m^{[1]} + c_m^{[k+1]}), \\
c_{m+1}^{[2k+1]} &= (1 - y_m)\left(c_m^{[1]} + c_m^{[k+1]}\right),
\end{align*}
\]

where \( k = 3^{m-1} - 1 \) and

\[
y_m = \frac{1}{2 - 2^{1/(2m+1)}}. \tag{B.4}
\]

In particular, the symplectic integrator SP-4 has the following coefficients:

\[
\begin{align*}
c_2^{[1]} &= c_2^{[4]} = \frac{1}{2(2 - 2^{1/3})}, \quad c_2^{[2]} = c_2^{[3]} = \frac{1 - 2^{1/3}}{2(2 - 2^{1/3})}, \\
d_2^{[1]} &= d_2^{[3]} = \frac{1}{2 - 2^{1/3}}, \quad d_2^{[2]} = -\frac{2^{1/3}}{2 - 2^{1/3}}, \quad d_2^{[4]} = 0. \tag{B.5}
\end{align*}
\]

The scheme SP-4 was independently presented in [13] and [30].

References

[1] Agarwal R P 2000 *Difference Equations and Inequalities* (New York: Dekker) chapter 3
[2] Blanes S 2001 High order numerical integrators for differential equations using composition and processing of low order methods *Appl. Numer. Math.* 37 289–306
[3] Calvo M P and Hairer E 1995 Accurate long-term integration of dynamical systems *Appl. Numer. Math.* 18 95–105
[4] Cieśliński J L 2007 An orbit-preserving discretization of the classical Kepler problem *Phys. Lett. A* 370 8–12
[5] Cieśliński J L 2009 On the exact discretization of the classical harmonic oscillator equation (in preparation) arXiv:0911.3672 [math-ph]
[6] Cieśliński J L 2011 *J. Differ. Eqns Appl.* (doi:10.1080/10236191003730563)
[7] Cieśliński J L 2010 Comment on ‘Conservative discretizations of the Kepler motion’ *J. Phys. A: Math. Theor.* 43 322001
[8] Cieśliński J L 2011 Locally exact modifications of numerical integrators arXiv:1101.0578 [math.NA]
[9] Cieśliński J L and Ratkiewicz B 2006 On simulations of the classical harmonic oscillator equation by difference equations *Adv. Differ. Eqns* 2006 40171
[10] Cieśliński J L and Ratkiewicz B 2009 Long-time behaviour of discretizations of the simple pendulum equation *J. Phys. A: Math. Theor.* 42 105204
[11] Cieśliński J L and Ratkiewicz B 2010 Improving the accuracy of the discrete gradient method in the one-dimensional case *Phys. Rev. E* 81 016704
[12] Faou E, Hairer E and Pham T L 2004 Energy conservation with non-symplectic methods: examples and counterexamples *BIT Numer. Math.* 44 699–709
[13] Forest E and Ruth R D 1990 Fourth-order symplectic integration Physica D 43 105–17

[14] Gonzales O 1996 Time integration and discrete Hamiltonian systems J. Nonlinear Sci. 6 449–67

[15] Hairer E 2010 Energy-preserving variant of collocation methods J. Numer. Anal. Indust. Appl. Math. 5 73–84

[16] Hairer E, Lubich C and Wanner G 2006 Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations 2nd edn (Berlin: Springer)

[17] Iserles A 2009 A First Course in the Numerical Analysis of Differential Equations 2nd edn (Cambridge: Cambridge University Press)

[18] Itoh T and Abe K 1988 Hamiltonian conserving discrete canonical equations based on variational difference quotients J. Comput. Phys. 77 85–102

[19] LaBudde R A and Greenspan D 1974 Discrete mechanics – a general treatment J. Comput. Phys. 15 134–67

[20] McLachlan R I and Quispel G R W 2002 Splitting methods Acta Numer. 11 341–434

[21] McLachlan R I and Quispel G R W 2006 Geometric integrators for ODEs J. Phys. A: Math. Gen. 39 5251–85

[22] McLachlan R I, Quispel G R W and Robidoux N 1998 Unified approach to Hamiltonian systems, Poisson systems, gradient systems and systems with Lyapunov functions or first integrals Phys. Rev. Lett. 81 2399–403

[23] McLachlan R I, Quispel G R W and Robidoux N 1999 Geometric integration using discrete gradients Phil. Trans. R. Soc. A 357 1021–45

[24] McLaren D I and Quispel G R W 2004 Integral preserving integrators J. Phys. A: Math. Gen. 37 L489–95

[25] Mickens R E 1994 Nonstandard Finite Difference Models of Differential Equations (Singapore: World Scientific)

[26] Potts R B 1982 Differential and difference equations Am. Math. Mon. 89 402–7

[27] Quispel G R W and Capel H W 1996 Solving ODE’s numerically while preserving a first integral Phys. Lett. A 218 223–8

[28] Quispel G R W and Turner G S 1996 Discrete gradient methods for solving ODE’s numerically while preserving a first integral J. Phys. A: Math. Gen. 29 L341–9

[29] Simo J C, Tarnow N and Wong K K 1992 Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics Comput. Methods Appl. Mech. Eng. 100 63–116

[30] Yoshida H 1990 Construction of higher order symplectic integrators Phys. Lett. A 150 262–8