ELSEVIER

Applied Numerical Mathematics 39 (2001) 245–259

APPLIED NUMERICAL MATHEMATICS

www.elsevier.com/locate/apnum

High-order Runge–Kutta–Nyström geometric methods with processing

S. Blanes [a,*], F. Casas [b], J. Ros [c]

[a] *Department of Applied Mathematics and Theoretical Physics, Cambridge, CB3 9EW, UK*
[b] *Departament de Matemàtiques, Universitat Jaume I, 12071-Castellón, Spain*
[c] *Departament de Física Teòrica and IFIC, Universitat de València, 46100-Burjassot, Valencia, Spain*

Abstract

We present new families of sixth- and eighth-order Runge–Kutta–Nyström geometric integrators with processing for ordinary differential equations. Both the processor and the kernel are composed of explicitly computable flows associated with non trivial elements belonging to the Lie algebra involved in the problem. Their efficiency is found to be superior to other previously known algorithms of equivalent order, in some case up to four orders of magnitude. © 2001 IMACS. Published by Elsevier Science B.V. All rights reserved.

Keywords: Geometric integration; Initial value problems; Processing technique

1. Introduction

One of the most used techniques in geometric integration of ordinary differential equations is to compose one or more low-order basic methods with appropriately chosen weights in order to achieve a higher-order scheme. The resulting composition algorithm preserves the favorable geometric property the basic method shares with the exact solution. Often one has a differential equation $\dot{x} = X(x)$ such that the function X can be written as a sum of two contributions $X = X_1 + X_2$ and the systems $\dot{x} = X_i(x)$, $i = 1, 2$, can both be solved analytically. Then the exact flows corresponding to X_i are taken as the basic methods.

When this approach is applied to second-order systems of ODE of the special form

$$\ddot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^l$ and $\boldsymbol{f} : \mathbb{R}^l \to \mathbb{R}^l$, the numerical algorithms are usually termed as Runge–Kutta–Nyström (RKN) methods. More specifically, introducing the new variables $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{v})^{\mathrm{T}}$, with $\boldsymbol{v} = \dot{\boldsymbol{x}}$, and the functions $\boldsymbol{f}_A = (\boldsymbol{v}, \boldsymbol{0})$ and $\boldsymbol{f}_B = (\boldsymbol{0}, \boldsymbol{f}(\boldsymbol{x}))$, Eq. (1) can be written as

$$\dot{\boldsymbol{z}} = \boldsymbol{f}_A + \boldsymbol{f}_B, \tag{2}$$

[*] Corresponding author.
E-mail address: s.blanes@damtp.cam.ac.uk (S. Blanes).

PII: S0168-9274(00)00035-0

and the systems $\dot{z} = \boldsymbol{f}_A$ and $\dot{z} = \boldsymbol{f}_B$ can be integrated in closed form with exact flow

$$\begin{aligned} z_A(t) &= \mathrm{e}^{tA} z_A(0) = (\boldsymbol{x}_0 + t\boldsymbol{v}_0, \boldsymbol{v}_0)^{\mathrm{T}}, \\ z_B(t) &= \mathrm{e}^{tB} z_B(0) = \bigl(\boldsymbol{x}_0, \boldsymbol{v}_0 + t\boldsymbol{f}(\boldsymbol{x}_0)\bigr)^{\mathrm{T}}. \end{aligned} \tag{3}$$

Here $\boldsymbol{x}_0$, $\boldsymbol{v}_0$ denote initial conditions (at $t = 0$) and $A \equiv \boldsymbol{f}_A \cdot \nabla$, $B \equiv \boldsymbol{f}_B \cdot \nabla$ are the Lie operators associated with $\boldsymbol{f}_A$ and $\boldsymbol{f}_B$, respectively ($\nabla = (\nabla_{\boldsymbol{x}}, \nabla_{\boldsymbol{v}})$). If we write the exact solution as $z(t) = \mathrm{e}^{t(A+B)} z_0$, the evolution operator $\mathrm{e}^{t(A+B)}$ for one time step $h = t/N$ may be approximated, for example, by

$$\mathrm{e}^{h(A+B)} \simeq \mathrm{e}^{hH_a} \equiv \prod_{i=1}^{s} \mathrm{e}^{ha_i A} \mathrm{e}^{hb_i B} \tag{4}$$

or

$$\mathrm{e}^{h(A+B)} \simeq \mathrm{e}^{hH_s} \equiv \prod_{i=1}^{r} S(w_i h), \tag{5}$$

where

$$S(h) = \exp\left(\frac{h}{2}A\right) \exp(B) \exp\left(\frac{h}{2}A\right)$$

is the leapfrog method. The corresponding approximate solution $z_\alpha(t) = \mathrm{e}^{tH_\alpha} z_0$, $\alpha = a, s$, evolves then according to the Lie group whose Lie algebra $L(A, B)$ is generated by A and B with the usual Lie bracket of vector fields [1]. If A and B are Hamiltonian vector fields, it lies in a subgroup of the group of symplectic maps and the method is called a symplectic integrator; if A and B are skew-Hermitian, then the approximation is unitary, etc.

The coefficients a_i, b_i or w_i of this kind of method are determined by imposing that

$$H_a = A + B + \mathrm{O}\bigl(h^n\bigr) \tag{6}$$

and similarly for H_s. This allows us to obtain an nth-order approximation to the exact solution. Observe that the efficiency of the compositions (4) or (5) is highly dependent of the number of flows involved and the coefficients appearing in the term $\mathrm{O}(h^n)$. In practice, methods like (4) are obtained by means of the Baker–Campbell–Hausdorff (BCH) formula, which makes it necessary to solve a system of polynomial equations in the coefficients. The solution of this system can be extraordinarily involved even for moderate values of n, so that various symmetries are imposed on (4) and (5) to reduce the number of determining equations, especially for high-order methods. For instance, if the composition is palindromic (also called left–right symmetric or self-adjoint) then H_a has not odd powers of h. The price to be paid is an increment in the number of flows to be composed in each step. Note that in (5) the individual stages $S(t)$ are themselves symmetric.

For the special decomposition (2) one can check that the Lie bracket $[B, [B, [B, A]]]$ is identically zero, and additional simplifications in the analysis occur. One of the most important examples is the class of Hamiltonian systems of the form $H = T(\boldsymbol{p}) + V(\boldsymbol{q})$ where $T(\boldsymbol{p})$ is quadratic in $\boldsymbol{p}$. Then the function $\boldsymbol{f}$ in (1) is the gradient of the potential $-V(\boldsymbol{q})$. The case of a Hamiltonian

$$H(\boldsymbol{q}, \boldsymbol{p}) = \mathcal{A}(\boldsymbol{q}, \boldsymbol{p}) + \mathcal{B}(\boldsymbol{q}), \tag{7}$$

with $\mathcal{A}$ quadratic in $\boldsymbol{p}$, is also included.

The recent literature has devoted much attention to the integration of Eq. (1) by means of high-order structure-preserving composition schemes. Okunbor and Skeel [13] have constructed explicit symplectic RKN formulae of the type (4) with five stages and seven stages of orders 5 and 6, respectively. Yoshida [17] derived explicit, symplectic methods of orders 6 and 8 requiring seven and fifteen function evaluations per step as a composition of leapfrog steps (5). Calvo and Sanz-Serna [6] designed an optimized symmetric eighth-order scheme with 24 evaluations which they found to be superior to Yoshida's methods in tests. Finally, McLachlan [9], after a thorough analysis, did not find more efficient sixth-order schemes than the one obtained by Okunbor and Skeel, and built a symmetric composition of order 8 with seventeen function evaluations more efficient than the previous ones. Interestingly, all these eighth-order methods are not properly RKN algorithms: they work for all splittings $X = X_1 + X_2$, not just for those of the form (2), and the question of the existence of symmetric high-order Runge–Kutta–Nyström integrators more efficient than those composed of symmetric steps has been raised [11].

Some steps along this way have been taken recently: the use of the processing technique has allowed the present authors to develop highly efficient schemes of orders 4 and 6 [2], the improvement with respect to other algorithms being mainly due to the reduction in the number of evaluations. It has been suggested [16] that the use of processing is the most economical path to high order, because the number of determining equations diminishes.

The idea of processing was first introduced in the context of Runge–Kutta methods by Butcher [5] in 1969 and applied to the symplectic integration of Hamiltonian systems, among others, by Wisdom et al. [16], McLachlan [10] and López-Marcos et al. [8]. In order to reduce the number of evaluations per time step h the following composition is considered:

$$\mathrm{e}^{h\mathcal{H}(h)} = \mathrm{e}^{P}\mathrm{e}^{hK}\mathrm{e}^{-P}. \tag{8}$$

Then, after N steps, we have $\mathrm{e}^{t(A+B)} \approx \mathrm{e}^{t\mathcal{H}(h)} = \mathrm{e}^{P}(\mathrm{e}^{hK})^{N}\mathrm{e}^{-P}$. At first we apply e^{P} (the corrector or processor), then e^{hK} (the kernel) acts once per step, and e^{-P} is evaluated only when output is needed. Both the kernel and the processor are taken as compositions of the flows associated with A and B.

A general analysis of the processing technique in connection with symplectic integration has been done in [2]. There, the number of conditions to be satisfied by the kernel to attain a given order has been obtained. It has also been shown that the kernel completely determines the optimal method one can obtain by processing.

In this paper we apply the above analysis to the RKN case. By combining the processing technique with the use of several exactly computable flows generated by different elements belonging to the Lie algebra $L(A, B)$ we obtain a family of optimal sixth-order RKN methods more efficient than others previously published and some processed eighth-order schemes with less function evaluations per step. Although only autonomous systems have been mentioned so far, also time-dependent systems can be included by adding an extra variable.

These new methods are particularly effective when the function $\boldsymbol{f}$ in (1) can be written as

$$\boldsymbol{f}(\boldsymbol{x}, t) = \sum_{i=1}^{s} g_i(t)\boldsymbol{f}_i(\boldsymbol{x}), \tag{9}$$

where an explicit time dependence has been introduced through the functions g_i. This case embraces the physically important class of time-dependent non-linear oscillators [12].

2. General analysis

In the Lie algebra $L(A, B)$ generated by the vector fields A and B, we denote by $[L_1, L_2, \ldots, L_s]$ the nested Lie bracket product $[L_1, [L_2, \ldots, L_s]]$. Let $d(m)$ be the dimension of the space spanned by brackets of order m of A and B when $[B, B, B, A] = 0$, its first 8 values being $2, 1, 2, 2, 4, 5, 10$, and 15 [9] and denote by $\{E_{m,i}\}_{i=1}^{d(m)}$ a basis of this subspace. Our explicit choice of basis is given in Appendix A.

In addition to A and B there are other elements in $L(A, B)$ whose flow is explicitly and exactly computable. For instance, the operator $[B, A, B]$ can be written as

$$V_{3,1} \equiv [B, A, B] = 2 \sum_{i,j=1}^{l} f_i \frac{\partial f_j}{\partial x_i} \frac{\partial}{\partial v_j} \equiv \boldsymbol{g}^{(3)}(\boldsymbol{x}) \cdot \nabla_{\boldsymbol{v}} \tag{10}$$

and the corresponding flow is given by an expression similar to the second equation of (3) by replacing $\boldsymbol{f}$ with $\boldsymbol{g}^{(3)}$. In general, let C_i denote either A or B operators, and let us assign to C_i an index r_i with values -1 and 1, respectively. Then one can see that $[C_1, C_2, \ldots, C_s]$ has the form of Eq. (10) with an appropriate function $\boldsymbol{g}^{(s)}(\boldsymbol{x})$ as soon as, starting from the right, the sum of r_i adds up to 1. This only takes place when there is one more B than A operators, so that the total number s has to be odd.

When $s = 5$ there is only one independent element in $L(A, B)$ with the required structure, namely

$$V_{5,1} \equiv [B, B, A, A, B], \tag{11}$$

whereas for $s = 7$ we have

$$V_{7,1} \equiv [B, A, B, B, A, A, B], \qquad V_{7,2} \equiv [B, B, B, A, A, A, B]. \tag{12}$$

The expression of the corresponding functions $\boldsymbol{g}^{(s)}$ are collected in Appendix A. Observe that the operators given in Eqs. (10)–(12) correspond respectively to the basis elements $E_{3,2}$, $E_{5,4}$, $E_{7,8}$, and $E_{7,10}$. For easier reading we use a different notation in the text.

Then it is also possible to evaluate exactly $\exp(hC_{b,c,d,e,f})$, with

$$C_{b,c,d,e,f} = bB + h^2 c\, V_{3,1} + h^4 d\, V_{5,1} + h^6 (eV_{7,1} + fV_{7,2}), \tag{13}$$

b, c, d, e, and f being free parameters. Therefore, by replacing in the compositions (4), (5) or (8) all e^{hb_iB} factors by the more general ones $\mathrm{e}^{hC_{b_i,c_i,d_i,e_i,f_i}}$, we introduce several parameters with only one exponential and reduce the number of evaluations of the overall scheme. Its efficiency is then improved if the calculation of successive derivatives of $\boldsymbol{f}$ in (11) and (12) is not very expensive in terms of computational cost.

The operator $C_{b,c,d,e,f}$ can be considered a generalization of the so-called modified potential used frequently in the recent literature [8,14,16]. In the rest of the paper it will be referred to as modified function.

This technique, when combined with processing, constitutes a new way to achieve high order methods deferring the explosion in the number of stages, which is typical of standard composition schemes. With this goal in perspective, let us return now to the composition (8).

By repeated application of the BCH formula the kernel and processor generators K and P can be written as a power series in h:

$$K = A + B + \sum_{i=2}^{\infty} \left\{ h^{i-1} \sum_{j=1}^{d(i)} k_{i,j} E_{i,j} \right\}, \qquad P = \sum_{i=1}^{\infty} \left\{ h^{i} \sum_{j=1}^{d(i)} p_{i,j} E_{i,j} \right\}, \tag{14}$$

and therefore

$$\mathcal{H}(h) = \mathrm{e}^{P} K \mathrm{e}^{-P} = A + B + \sum_{i=2}^{\infty} \left\{ h^{i-1} \sum_{j=1}^{d(i)} f_{i,j} E_{i,j} \right\}, \tag{15}$$

where the $f_{i,j}$ coefficients are given in terms of polynomial relations involving $k_{i,j}$ and $p_{i,j}$ [2]. Particular nth-order integration methods require that $f_{i,j} = 0$ up to $i = n$. These equations cannot be solved for an arbitrary kernel using only the processor coefficients. More specifically, the kernel of an nth-order ($n \geqslant 2$) method for Eq. (2) must satisfy exactly $k(n) = d(n) - 1$ independent conditions [2]. If, in addition, the kernel is symmetric, then the number of conditions is considerably reduced: $k(2n) = k(2n-1)$. In Table 1 we collect the explicit form, in the basis we are using, of these conditions $N_{i,j} = 0$ up to eighth-order.

In this case the processor $P(h)$ can be chosen as an even function of h and the corresponding coefficients take very simple forms in terms of $k_{i,j}$ [2]: the only non-zero values are, for methods up to sixth-order,

$$p_{2,1} = k_{3,1}, \qquad p_{4,1} = k_{5,1}, \qquad p_{4,2} = k_{5,4} - \tfrac{1}{2} k_{3,1}^2, \tag{16}$$

whereas for order 8 we have, in addition,

$$\begin{aligned}
p_{6,1} &= \tfrac{1}{2}(k_{71} + k_{72}) - \tfrac{1}{2} k_{31} k_{51}, \\
p_{6,2} &= \tfrac{1}{2}(k_{73} + k_{74}) + \tfrac{1}{6} k_{31}^3 + k_{31} k_{51} - \tfrac{1}{2} k_{31} k_{53}, \\
p_{6,3} &= \tfrac{1}{2}(k_{75} + k_{76}) - \tfrac{5}{12} k_{31}^3 - k_{31} k_{51} + k_{31} k_{53}, \\
p_{6,4} &= \tfrac{1}{2}(k_{77} + k_{78}) - \tfrac{1}{6} k_{31}^3, \\
p_{6,5} &= \tfrac{1}{2}(k_{79} + k_{710}) + \tfrac{2}{9} k_{31}^3 + \tfrac{1}{2} k_{31} k_{51} - \tfrac{1}{3} k_{31} k_{53}.
\end{aligned} \tag{17}$$

Table 1
Conditions to be satisfied by a symmetric kernel of a RKN processing method up to order 8

Order 3	$N_{3,1} \equiv k_{3,1} - k_{3,2} = 0$
Order 5	$N_{5,1} \equiv k_{5,2} - k_{5,1} - \tfrac{1}{2} k_{3,1}^2 = 0$
	$N_{5,2} \equiv k_{5,3} + k_{5,4} - k_{3,2} k_{3,1} = 0$
Order 7	$N_{7,1} \equiv k_{7,1} - k_{7,2} + k_{5,1} k_{3,1} = 0$
	$N_{7,2} \equiv k_{7,4} - k_{7,3} + \tfrac{1}{3} k_{3,1}^3 - k_{5,1} k_{3,1} - k_{5,3} k_{3,1} = 0$
	$N_{7,3} \equiv k_{7,5} - k_{7,6} + \tfrac{5}{6} k_{3,1}^3 + k_{5,1} k_{3,1} - 2 k_{5,3} k_{3,1} = 0$
	$N_{7,4} \equiv k_{7,7} - k_{7,8} + k_{3,1}^3 - 2 k_{5,3} k_{3,1} = 0$
	$N_{7,5} \equiv k_{7,9} - k_{7,10} + \tfrac{5}{18} k_{3,1}^3 + k_{5,1} k_{3,1} - \tfrac{1}{3} k_{5,3} k_{3,1} = 0$

It is traditional to compare different integration methods by means of an effective error constant E_f: some measure of the first term in the local truncation error E_r adjusted for the complexity of the scheme. In our case E_r can be defined as

$$E_\mathrm{r} = \sqrt{\sum_{j=1}^{d(n+1)} |f_{n+1,j}|^2} \tag{18}$$

and the effective error can be taken as $E_\mathrm{f} = mE_\mathrm{r}^{1/n}$, where m is the number of B (or C) evaluations per time step. Interestingly, it has been shown [2] that the values of $f_{n+1,j}$ which minimize the value of E_r can be written as linear combinations of the functions $N_{n+1,j}$ which would determine the conditions for the kernel at order $n+1$. When $n=6$ these optimal values read

$$|f_{7,2i-1}| = |f_{7,2i}| = \tfrac{1}{2}|N_{7,i}|, \quad i = 1, \ldots, 5. \tag{19}$$

We see, then, that the kernel itself determines the minimum error one can achieve by processing. In the following we design a complete family of sixth-order optimal processing methods just by considering different symmetric kernels. The eighth-order case is also considered.

3. The new RKN processing methods

The two basic ingredients of an integrator with processing are the processor e^P and the kernel e^{hK}. From a practical point of view it is generally not necessary to use an element of the Lie group associated with $L(A,B)$ as a processor because its effects are not propagated by the numerical integrator [8]. Nevertheless, to be fully consistent with the demand of geometric integration, we take as processor the explicitly computable non-symmetric composition

$$\mathrm{e}^P = \prod_{i=1}^{r} \mathrm{e}^{hz_iA}\mathrm{e}^{hy_iB}, \tag{20}$$

where the replacement $\exp(hy_iB) \mapsto \exp(hC_{y_i,v_i,w_i})$ can be done when necessary. In any case, it is characterized by the number r of B (or C) evaluations needed to guarantee that the $\sum_{i=1}^{n-1} d(i)$ equations

$$p_{i,j} = p_{i,j}(z_k, y_k)$$

have real solutions for the coefficients.

As far as the kernel is concerned, due to the qualitatively different character of the operators A and B, we have to consider the following two types of composition:

(i) ABA-type composition: $(\sum_{i=1}^{s+1} a_i = \sum_{i=1}^{s} b_i = 1)$:

$$\mathrm{e}^{hK} = \mathrm{e}^{ha_1A}\mathrm{e}^{hb_1B}\mathrm{e}^{ha_2A}\cdots\mathrm{e}^{ha_sA}\mathrm{e}^{hb_sB}\mathrm{e}^{ha_{s+1}A} \tag{21}$$

with $a_{s+2-i} = a_i$ and $b_{s+1-i} = b_i$.

(ii) BAB-type composition $(\sum_{i=1}^{s} a_i = \sum_{i=1}^{s+1} b_i = 1)$:

$$\mathrm{e}^{hK} = \mathrm{e}^{hb_1B}\mathrm{e}^{ha_1A}\mathrm{e}^{hb_2B}\cdots\mathrm{e}^{hb_sB}\mathrm{e}^{ha_sA}\mathrm{e}^{hb_{s+1}B} \tag{22}$$

with $a_{s+1-i} = a_i$ and $b_{s+2-i} = b_i$.

We designate the whole method of order n by the label

$$(n : X\text{-}s, r; l)$$

where $X = ABA, BAB$ indicates the particular character of the kernel and, as before, we can replace e^{hb_iB} by the exact flow corresponding to $C_{b_i,c_i,\ldots}$, in which case we add a number l to indicate the highest order of the nested Lie brackets included. Observe, then, that the criterion for estimating the computational effort needs to be reformulated. Although the cost of the evaluation of $C_{b,c,d,e,f}$ depends both on the problem being integrated and the number of non-zero coefficients in C, it turns out that, at least when $l = 3$ (i.e., $d = e = f = 0$), one evaluation of $C_{b,c}$ can be done at the cost of at most two independent B evaluations and typically much less because of reuse of certain calculations in the computer [2,8]. This happens also for a number of problems even if d, e and f are nonvanishing, as is the case for polynomial $\boldsymbol{f}_i$ in Eq. (9) or when the potential depends only on $\|\boldsymbol{x}\|$.

3.1. Sixth-order methods

The kernel has to satisfy the three first conditions in Table 1. We look for different solutions of the equations $N_{3,1} = N_{5,1} = N_{5,2} = 0$ and take the coefficients which produce the smallest value for the functions $N_{7,j}$ (and thus the minimum error). The efficiency of these new methods should be compared with the value $E_{\mathrm{f}} = 7E_{\mathrm{r}}^{1/6} = 1.0345$ achieved by the best RKN scheme with seven B evaluations, designed by Okunbor and Skeel [9,13]. Table 2 collects the coefficients (with the last a_i and b_i omitted) and effective errors achieved by the new schemes.

(i) $l = 0$. The minimum number of B operators needed to solve the kernel conditions is four. We have not found real solutions with an *ABA*-type composition, whereas the optimal effective error achieved by a *BAB* composition is $E_{\mathrm{f}} = 1.0355$, which is similar to the best unprocessed algorithm. With five B evaluations there exist an *ABA* kernel with $E_{\mathrm{f}} = 0.7329$. If one more B evaluation is considered we have obtained the $(6: BAB\text{-}6, 9; 0)$ scheme given in Table 2 with effective error $E_{\mathrm{f}} = 0.64$. Obviously, with seven evaluations we could get a better method because then there are three free parameters. On the other hand, because a sixth order method satisfy all the conditions for a kernel, we can consider the most efficient sixth order method given by Okunbor and Skeel. With this kernel we can build a method with effective error $E_{\mathrm{f}} = 0.61685$.

(ii) $l = 3$. Three C_{b_i,c_i} evaluations in the kernel are required to have real solutions and there is still a free parameter (any of the c_i). Both types of composition produce exactly the same optimal error $E_{\mathrm{r}}^{1/6} = 0.15$. By fixing $c_1 = 0$ we have the $(6: ABA\text{-}3, 6; 3)$ method which only requires one C and two B evaluations, giving error $E_{\mathrm{r}}^{1/6} = 0.1551$ and efficiency $0.4653 < E_{\mathrm{f}} < 0.6204$, depending of the estimated cost of a C evaluation.

(iii) $l = 5$. The three free parameters needed to solve the equations can also be obtained with two $C_{b,c,d}$ operators. There are no real solutions with *BAB* schemes, whereas the resulting *ABA* composition admits two sets of real solutions, one of which is used to construct the $(6: ABA\text{-}2, 6; 5)$ scheme of Table 2, with error $E_{\mathrm{r}}^{1/6} = 0.2719$ and efficiency $0.5438 < E_{\mathrm{f}} < 1.0876$.

(iv) $l = 7$. We can use the complete expression C_{b_i,c_i,d_i,e_i,f_i} to reduce significantly the errors attained by the preceding kernels. More specifically, as the coefficients e_i and f_i appear linearly in the functions $N_{7,4}$ and $N_{7,5}$, they can be chosen to cancel these functions. This is especially useful when $N_{7,4}$ and $N_{7,5}$ provide the highest contribution to the error term. For instance, if we replace C_{b_1,c_1,d_1} by C_{b_1,c_1,d_1,e_1} in the kernel of the $(6: ABA\text{-}2, 6; 5)$ scheme, with $e_1 = (215 + 56\sqrt{15})/1612800$, we obtain a method with error $E_{\mathrm{r}}^{1/6} = 0.2281$. When the replacement $C_{b_2,c_2} \mapsto C_{b_2,c_2,0,e_2,f_2}$ is done in the $(6: ABA\text{-}3, 6; 3)$ scheme, i.e., when the kernel

$$e^{hK} = e^{ha_1A} e^{hb_1B} e^{ha_2A} e^{hC_{b_2,c_2,0,e_2,f_2}} e^{ha_2A} e^{hb_1B} e^{ha_1A} \tag{23}$$

Table 2
Coefficients and effective errors of the new sixth-order RKN geometric integrators with processing

	$(6: BAB\text{-}6, 9; 0)$	$E_{\mathrm{f}} = 0.64$
$b_1 = 0.15$	$a_1 = 0.316$	
$b_2 = 0.3297455985640361$	$a_2 = 0.4312992634164797$	
$b_3 = -0.049363257050623707$		
$z_1 = -0.2079110832137436$	$z_2 = 0.4089657710426152$	$z_3 = 0.5630192496347863$
$z_4 = 0.009121373956442832$	$z_5 = -0.5602966606303723$	$z_6 = 0.7988679375711318$
$z_7 = -0.8711855319991359$	$z_8 = 0.8594189436382758$	$z_9 = -\sum_{i=1}^{8} z_i$
$y_1 = -0.015428952113728616$	$y_2 = 0.4245395527376832$	$y_3 = 0.1686944980146086$
$y_4 = -0.1611964864865696$	$y_5 = -0.4258477789489911$	$y_6 = -0.008262586834473168$
$y_7 = 0.008521397729269797$	$y_8 = 0.008980355902201032$	$y_9 = -\sum_{i=1}^{8} y_i$
	$(6: ABA\text{-}3, 6; 3)$	$E_{\mathrm{f}} \in (0.4653, 0.6204)$
$a_1 = -0.0682610383918630$	$b_1 = 0.2621129352517028$	
$c_1 = 0$	$c_2 = 0.0164011128160783$	
$z_1 = 0.07943288242455420$	$z_2 = 0.02974829169467665$	$z_3 = -0.7057074964815896$
$z_4 = 0.3190423451260838$	$z_5 = -0.2869147334299646$	$z_6 = -\sum_{i=1}^{5} z_i$
$y_1 = 1.3599424487455264$	$y_2 = -0.6505973747535132$	$y_3 = -0.033542814598338416$
$y_4 = -0.040129915275115030$	$y_5 = 0.044579729809902803$	$y_6 = -\sum_{i=1}^{5} y_i$
$v_1 = -0.034841228074994859$	$v_2 = 0.031675672097525204$	$v_3 = -0.005661054677711889$
$v_4 = 0.004262222269023640$	$v_5 = 0.005$	$v_6 = -0.005$
	$(6: ABA\text{-}2, 6; 5)$	$E_{\mathrm{f}} \in (0.5438, 1.0876)$
$a_1 = \frac{1}{4}(1 + \sqrt{1 + \frac{4}{\sqrt{15}}})$	$c_1 = \frac{5+\sqrt{15}}{240}$	$d_1 = \frac{4+\sqrt{15}}{2880}$
$z_1 = -0.029784067651958936$	$z_2 = 0.9445943038246405$	$z_3 = -1.908119437387469$
$z_4 = 1.651876569139561$	$z_5 = -0.2328222691635203$	$z_6 = -\sum_{i=1}^{5} z_i$
$y_1 = 0.1478939879876102$	$y_2 = -0.042209655271038353$	$y_3 = -0.000873366842778911$
$y_4 = 0.2180721303705606$	$y_5 = -0.3228830962443535$	$y_6 = -\sum_{i=1}^{5} y_i$
$v_1 = 0.007282272774510424$	$v_2 = -0.003668108110223575$	$v_3 = -0.000225427508528040$
$v_4 = 0.02$	$v_5 = -0.02$	$v_6 = 0$
$w_1 = 0.000384554838931473$	$w_2 = 0.000258018664435799$	$w_3 = 0$
$w_4 = 0$	$w_5 = 0$	$w_6 = 0$
	$(6: ABA\text{-}3, 6; 7)$	$E_{\mathrm{f}} \in (0.2238, 0.2984)$
$a_1 = -0.0682610383918630$	$b_1 = 0.2621129352517028$	
$c_1 = 0$	$c_2 = 0.0164011128160783$	
$d_2 = 0$	$e_2 = 0.0000186194612413481$	$f_2 = -0.0000063155794861591$
$z_1 = 0.1604630501234888$	$z_2 = -0.1222126706298830$	$z_3 = 0.1916801124727711$
$z_4 = 0.5630722377955035$	$z_5 = -0.7612758792358986$	$z_6 = -\sum_{i=1}^{5} z_i$
$y_1 = -0.012334538446142270$	$y_2 = -0.6610294848488182$	$y_3 = -0.023112349678219939$
$y_4 = 0.000181521815949959$	$y_5 = 2.3768244683666757$	$y_6 = -\sum_{i=1}^{5} y_i$
$v_1 = 0.013816178183636998$	$v_2 = -0.050288359617427786$	$v_3 = -0.013462400168471472$
$v_4 = 0.000603819193361427$	$v_5 = -0.01$	$v_6 = 0.01$

Table 3
Kernel coefficients of eighth-order RKN geometric integrators with processing. Coefficients of a processor are also included for one kernel

$(ABA\text{-}9; 0)$	
$a_1 = 0.00004683745923348969$	$b_1 = 0.3730012196073597$
$a_2 = -0.7458919296558489$	$b_2 = 0.039270822365231689$
$a_3 = -0.027911335134073806$	$b_3 = -0.032798861516437888$
$a_4 = 0.5888685556487076$	$b_4 = -0.1021725211468956$
$(ABA\text{-}4; 7)$	
$a_1 = 0.7129508732570782$	$b_1 = 0.5974070023507730$
$c_1 = -0.052876668399475798$	$a_2 = -0.4094021154865992$
$c_2 = 0.012122201874074444$	$d_2 = 0.003162537736573353$
$e_2 = 0.000516635479956932$	$f_2 = -0.000025513037513292$
$(8 : BAB\text{-}5, 14; 7)$	
$b_1 = 0.2585691647446146$	$c_1 = 0.007587869772563802$
$d_1 = 0.0001219127419188233$	$e_1 = 0.000005741889879702246$
$f_1 = -0.000002271708973531348$	$a_1 = 0.6954511641703808$
$b_2 = -0.1945897221635392$	$c_2 = 0.0005222572249380952$
$a_2 = -0.05$	
$z_1 = 0$	$y_1 = 0.3644761259072299$
$z_2 = -0.004624860718237988$	$y_2 = -0.2849544383272169$
$z_3 = 0.3423219445639433$	$y_3 = 0.2023898776842639$
$z_4 = 0.1760176996772205$	$y_4 = -0.2743578195701579$
$z_5 = 0.3625045293826689$	$y_5 = -0.00475975395524748$
$z_6 = -0.2729727321466362$	$y_6 = 0.1455974775779454$
$z_7 = -\sum_{i=1}^{6} z_i$	$y_7 = -\sum_{i=1}^{6} y_i$
$v_1 = 0.016298916362212911$	$v_2 = -0.019769812343547362$
$v_3 = 0.004608026684270971$	$v_4 = v_5 = v_6 = v_7 = 0$

is considered, the corresponding $(6 : ABA\text{-}3, 6; 7)$ method, given in Table 2, attains an error $E_{\mathrm{r}}^{1/6} = 0.0746$. Observe that, in this case, only one modified function C has to be evaluated per time step and then $0.2238 < E_{\mathrm{f}} < 0.2984$ for a number of problems.

3.2. Eighth-order methods

A symmetric kernel must satisfy the eight equations in Table 1. In order to have as many parameters as necessary to fulfill these conditions we may consider an *ABA* or *BAB* composition with at least nine B operators or include modified functions in the scheme: for instance, the minimum number of flows is achieved with only two $C_{b,c}$ and $C_{b,c,d,e,f}$ evaluations. In any case, the number of possibilities (and solutions) increases considerably with respect to $n = 6$.

In principle, a similar analysis can be carried out to obtain new families of optimal eighth-order methods with processing. This requires considering the space spanned by nested brackets of order 9 and the number of equations involved becomes prohibitively large ($d(9) = 26$). Some alternative approach for characterizing the accuracy of these methods should then be considered, such as some measure of the magnitude of the coefficients [9].

Concerning the processor P, the coefficients z_k, y_k in Eq. (20) have to satisfy 26 equations. This unpractical number of conditions can be reduced by determining the coefficients of the composition $\mathrm{e}^{Q} = \prod_i \mathrm{e}^{hz_iA}\mathrm{e}^{hy_iB}$ such that

$$Q(h) = \tfrac{1}{2}P(h) + \mathrm{O}(h^7). \tag{24}$$

In this way only 16 equations are involved, but it follows that

$$\mathrm{e}^{Q(h)}\mathrm{e}^{Q(-h)} = \mathrm{e}^{P(h)} + \mathrm{O}(h^8) \tag{25}$$

because $P(h)$ is an even function of h. Observe that the same composition that defines $Q(h)$ but reversing the sign of h allows us to obtain $Q(-h)$.

Table 3 collects the coefficients of some compositions we have found for kernels of eighth-order methods. These include an *ABA* composition without modified functions, $(ABA\text{-}9; 0)$, a kernel involving the minimum number of exponentials, $(ABA\text{-}4; 7)$, and a *BAB* composition with two B, two $C_{b,c}$ and one $C_{b,c,d,e,f}$ evaluations. For this case we have also written the coefficients of a possible processor. The method thus obtained, $(8 : BAB\text{-}5, 2r; 7)$, with $r = 7$, should be considered only as a preliminary result of a more complete analysis which is being carried out at present [3].

4. Numerical examples

In order to test the efficiency of the new methods presented above, they are applied to some test-bench examples. Comparison is done with other schemes of similar asymptotic consistency. For order six, these are the most efficient seven-stage method designed by Okunbor and Skeel, OS6 [13], and the non-symplectic variable step embedded RKN method, DP6, presented in [4] such as is implemented in the subroutine D02LAF of the NAG library. Concerning the eighth-order, the methods we use are a symplectic integrator due to Yoshida [17] (Yos8), with 15 function evaluations, the 17 stages composition method obtained by McLachlan [9] (McL8) and the optimized symmetric scheme designed by Calvo and Sanz-Serna [6], with 24 evaluations per step (CSS8).

Example 1. The methods are applied first to the time-dependent Hamiltonian

$$H = \tfrac{1}{2}(p_1^2 + q_1^2) + \varepsilon\cos(q_1)g_1(t) + \varepsilon\sin(q_1)g_2(t) \tag{26}$$

with $g_1(t) = \sum_{k=1}^{m} \cos(w_k t)$, $g_2(t) = \sum_{k=1}^{m} \sin(w_k t)$. It describes the motion of a charged particle in a constant magnetic field perturbed by m electrostatic plane waves (propagating along the perpendicular direction of the motion), each with the same wavenumber and amplitude, but with differing temporal frequencies ω_k [7].

This Hamiltonian can be treated as an autonomous system by considering the additional coordinate $q_2 = t$ and the corresponding conjugate canonical momentum p_2 [1]. Then it has the form (7) with

$$\begin{aligned} \mathcal{A} &= p_2 + \tfrac{1}{2}(p_1^2 + q_1^2), \\ \mathcal{B} &= \varepsilon \cos(q_1) g_1(q_2) + \varepsilon \sin(q_1) g_2(q_2). \end{aligned} \tag{27}$$

Then, in terms of the variables $\boldsymbol{x} \equiv (q_1, t)$, $\boldsymbol{v} \equiv (p_1, p_2)$, the equations of motion can be written as Eq. (2) with

$$\boldsymbol{f}_A = (p_1, 1, -q_1, 0)^{\mathrm{T}}, \qquad \boldsymbol{f}_B = (\mathbf{0}, -\nabla_{\boldsymbol{x}} \mathcal{B})^{\mathrm{T}} \tag{28}$$

and the systems $\dot{z} = \boldsymbol{f}_A$, $\dot{z} = \boldsymbol{f}_B$ have the exact flows

$$\begin{aligned} \mathrm{e}^{hA} z_A(0) &= (q_{1_0} \cos h + p_{1_0} \sin h, h, -q_{1_0} \sin h + p_{1_0} \cos h, p_{2_0}), \\ \mathrm{e}^{hB} z_B(0) &= \big(\boldsymbol{x}_0, \boldsymbol{v}_0 - h \nabla_{\boldsymbol{x}} \mathcal{B}(\boldsymbol{x}_0)\big). \end{aligned} \tag{29}$$

Observe that, for this example, the additional computational cost of evaluating the modified functions $\boldsymbol{g}^{(s)}$ is almost negligible, in particular the most expensive $\partial_t \mathcal{B}$ is not needed for computing the trajectory.

In order to ensure resonance we choose $\omega_k = k\omega_0$, with ω_0 an integer. We take as initial conditions $q_{1_0} = 1$, $p_{1_0} = 0$ and parameters $\varepsilon = 0.1$, $w_0 = 7$ and $m = 10$, so that both parts of the Hamiltonian have a similar contribution. The numerical integration is carried out for 100 periods of the linear oscillator (up to a final time $t_{\mathrm{f}} = 100 \cdot 2\pi$) and the average error in distance with respect to the "exact" solution is evaluated during the last period. Here "exact" means obtained by integrating with a much shorter step size.

Fig. 1 shows, in a log-log scale, this error as a function of the number of e^B evaluations. Dash-dotted line corresponds to the non-symplectic method DP6, whereas dotted line (OS6) stands for the method of Okunbor and Skeel. The optimal processed schemes used are $(6 : BAB\text{-}6, 9; 0)$ (broken line, P6) and $(6 : ABA\text{-}3, 6; 7)$ (solid lines, PM6), with coefficients given in Table 2. Solid lines are obtained considering that the cost of the evaluation of $\mathrm{e}^{C_{b,c,d,e,f}}$ is one and two times the cost of e^B, respectively.

Observe the high superiority achieved by the new processed methods, in particular when modified functions are used regardless the increase of the cost of evaluating these functions, and how the theoretical efficiency obtained in the last section exhibits in practice.

The rest of the new sixth-order methods have, for this example, performances between the two processed methods given in the figure.

Example 2. As a second example we consider the two-body gravitational problem, for which the equations may be written

$$\ddot{x}_i = -\frac{x_i}{(x_1^2 + x_2^2)^{3/2}}, \qquad i = 1, 2, \tag{30}$$

and take as initial conditions

$$x_1(0) = 1 - \mathrm{e}, \qquad x_2(0) = \dot{x}_1(0) = 0, \qquad \dot{x}_2(0) = \left(\frac{1 + \mathrm{e}}{1 - \mathrm{e}}\right)^{1/2}, \tag{31}$$

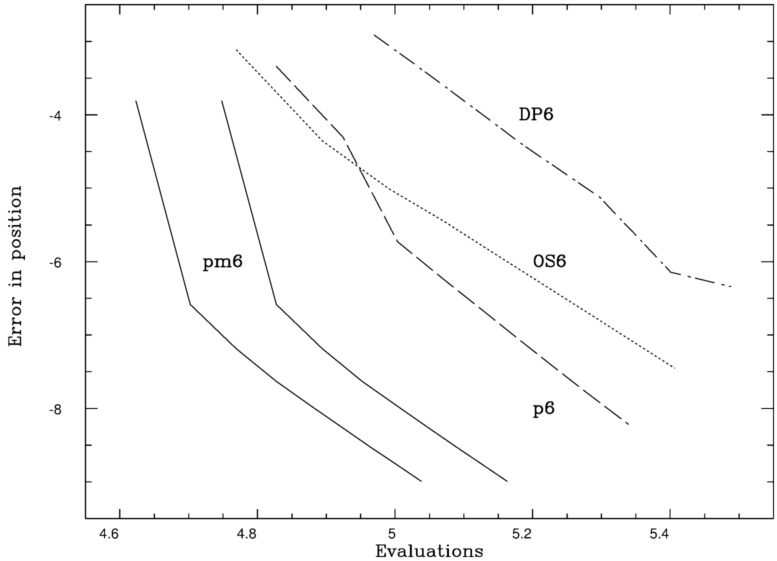

Fig. 1. Average distance between exact and numerical trajectories vs. number of function evaluations for the first example obtained with the new methods (6 : ABA-3, 6; 7) (solid lines) and (6 : BAB-6, 9; 0) (broken line). Results attained with standard symplectic (OS6) and variable-step (DP6) sixth-order integrators are also included.

which produce an orbit with eccentricity e. With the value $e = 0.5$, this orbit is determined numerically for 500 periods and the mean error in energy is computed during the last period. It should be remarked that the results achieved by the new schemes (as well as all symplectic integrators) are largely independent of the final time t_{f}, because the error in energy does not increase secularly, but this is not the case for the variable-step method DP6.

When modified functions $C_{b,c,d,e,f}$ are used into the algorithm, the following map has to be evaluated:

$$\mathrm{e}^{hC_{b,c,d,e,f}}\dot{x}_i = \dot{x}_i - x_i R, \tag{32}$$

where $R = G(b + F(4c + F(28d + F(280e + 360f))))$, and $G = h/(x_1^2 + x_2^2)^{3/2}$, $F = hG$. Notice that the increment in the computational cost with respect to the evaluation of e^{hbB} (which corresponds to $c = d = e = f = 0$) is only due to a few additional floating-point operations. In fact, by comparing CPU times of the same algorithm with and without using modified functions, we conclude that, for this example, the cost of $\mathrm{e}^{C_{b,c,d,e,f}}$ is approximately $\frac{4}{3}$ times the cost of e^{B}. We will consider this figure when counting the number of evaluations.

Fig. 2 shows the mean error as a function of the number of e^{B} evaluations for the same methods as that in the previous example. From the figure it is clear the higher performance of our new optimal processing methods with respect to the standard symplectic and non-symplectic integrators considered. This improvement is particularly noticeable when the modified function $C_{b,c,d,e,f}$ is incorporated into the schemes.

In Fig. 3 we compare the results achieved for this problem by the new eighth-order integrator whose coefficients are given in Table 3 (solid line denoted by pm8) with standard composition schemes. Observe that, whereas CSS8 and McL8 have been optimized in order to reduce the truncation error, this is not the case of pm8, and nevertheless it provides better results. This fact gives further momentum to the analysis

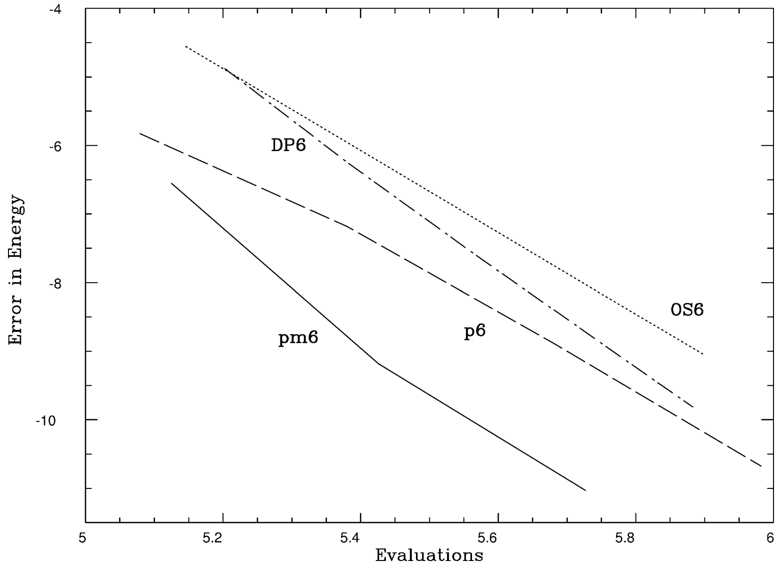

Fig. 2. Error in energy vs. number of function evaluations for the two-body problem with eccentricity $e = 0.5$ obtained with the methods of Fig. 1. In this case one evaluation of C is approximately equivalent to $\frac{4}{3}$ evaluations of B.

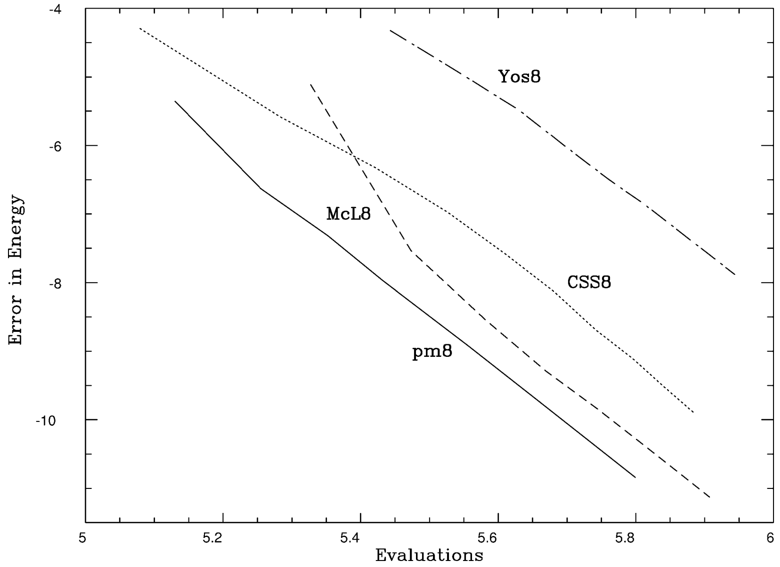

Fig. 3. Error in energy vs. number of function evaluations for the two-body problem obtained with the new method $(8 : BAB\text{-}5, 14; 7)$ (solid line, pm8) in comparison with other standard symplectic eighth-order algorithms.

and construction of more efficient eighth-order integrators by the combined use of modified functions and the processing technique.

Acknowledgements

The work of SB and JR has been supported by DGICyT under contract PB97-1139. FC is supported by the Collaboration Program UJI-Fundació Caixa Castelló 1996 under project no. P1B96-14.

Appendix A

As stated in the text we denote by $\{E_{n,i}\}_{i=1}^{d(n)}$ a basis of the space spanned by nested brackets of order n of A and B when $[B, B, B, A] = 0$. In this work we have taken:

$n = 1$	$E_{1,1} = A$, $E_{1,2} = B$
$n = 2$	$E_{2,1} = [A, B]$
$n = 3$	$E_{3,1} = [A, A, B]$, $E_{3,2} = [B, A, B]$
$n = 4$	$E_{4,1} = [A, A, A, B]$, $E_{4,2} = [B, A, A, B]$
$n = 5$	$E_{5,1} = [A, E_{4,1}]$, $E_{5,2} = [B, E_{4,1}]$
	$E_{5,3} = -[A, E_{4,2}]$, $E_{5,4} = [B, E_{4,2}]$
$n = 6$	$E_{6,1} = [A, E_{5,1}]$, $E_{6,2} = [B, E_{5,1}]$, $E_{6,3} = [A, E_{5,2}]$
	$E_{6,4} = [A, E_{5,4}]$, $E_{6,5} = [B, E_{5,2}]$,
$n = 7$	$E_{7,2j-1} = [A, E_{6,j}]$, $E_{7,2j} = [B, E_{6,j}]$, $j = 1, \ldots, 5$

Next we collect the functions $\boldsymbol{g}^{(s)}(\boldsymbol{x})$ appearing in the operators $V_{s,j}$, $s = 5, 7$, for autonomous systems, which are incorporated into the modified function $C_{b,c,d,e,f}$. The sum on repeated indices convention is adopted here:

$$(\boldsymbol{g}^{(5)})_j(\boldsymbol{x}) = 2f_k\left(f_l\frac{\partial^2 f_j}{\partial x_l\partial x_k} + 2\frac{\partial f_l}{\partial x_k}\frac{\partial f_j}{\partial x_l}\right), \tag{A.1}$$

$$\begin{aligned}&(\boldsymbol{g}_1^{(7)})_j(\boldsymbol{x})\\&\quad = 2f_m\left(4f_k\frac{\partial f_l}{\partial x_m}\frac{\partial^2 f_j}{\partial x_l\partial x_k} + f_kf_l\frac{\partial^3 f_j}{\partial x_m\partial x_l\partial x_k} + 4\frac{\partial f_k}{\partial x_m}\frac{\partial f_l}{\partial x_k}\frac{\partial f_j}{\partial x_l} + 3f_k\frac{\partial^2 f_l}{\partial x_m\partial x_k}\frac{\partial f_j}{\partial x_l}\right),\end{aligned} \tag{A.2}$$

$$(\boldsymbol{g}_2^{(7)})_j(\boldsymbol{x}) = 6f_kf_l\left(f_m\frac{\partial^3 f_j}{\partial x_m\partial x_l\partial x_k} + 3\frac{\partial^2 f_m}{\partial x_l\partial x_k}\frac{\partial f_j}{\partial x_m}\right). \tag{A.3}$$

It should be stressed that these expressions are also valid for the more general case

$$\ddot{\boldsymbol{x}} = A\dot{\boldsymbol{x}} + \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{f}(\boldsymbol{x}), \tag{A.4}$$

where A is a constant matrix and the system $\ddot{\boldsymbol{x}} = A\dot{\boldsymbol{x}} + \boldsymbol{h}(\boldsymbol{x})$ is exactly solvable.

References

[1] V.I. Arnold, Mathematical Methods of Classical Mechanics, 2nd Edition, Springer, New York, 1989.

[2] S. Blanes, F. Casas, J. Ros, Symplectic integrators with processing: A general study, SIAM J. Sci. Comput. 21 (1999) 711–727.

[3] S. Blanes, F. Casas, J. Ros, work in progress.

[4] R.W. Brankin, I. Gladwell, J.R. Dormand, P.J. Prince, W.L. Seward, Algorithm 670. A Runge–Kutta–Nyström code, ACM Trans. Math. Software 15 (1989) 31–40.

[5] J. Butcher, The effective order of Runge–Kutta methods, in: Proc. of Conference on the Numerical Solution of Differential Equations, Springer, Berlin, 1969, pp. 133–139.

[6] M.P. Calvo, J.M. Sanz-Serna, High-order symplectic Runge–Kutta–Nyström methods, SIAM J. Sci. Comput. 14 (1993) 1237–1252.

[7] J. Candy, W. Rozmus, A symplectic integration algorithm for separable Hamiltonian functions, J. Comput. Phys. 92 (1991) 230–256.

[8] M.A. López-Marcos, J.M. Sanz-Serna, R.D. Skeel, Explicit symplectic integrators using Hessian–vector products, SIAM J. Sci. Comput. 18 (1997) 223–238.

[9] R.I. McLachlan, On the numerical integration of ordinary differential equations by symmetric composition methods, SIAM J. Sci. Comput. 16 (1995) 151–168.

[10] R.I. McLachlan, More on symplectic correctors, in: J.E. Marsden, G.W. Patrick, W.F. Shadwick (Eds.), Integration Algorithms and Classical Mechanics, American Mathematical Society, Providence, RI, 1996, pp. 141–149.

[11] R.I. McLachlan, C. Scovel, A survey of open problems in symplectic integration, in: J.E. Marsden, G.W. Patrick, W.F. Shadwick (Eds.), Integration Algorithms and Classical Mechanics, American Mathematical Society, Providence, RI, 1996, pp. 151–180.

[12] A.H. Nayfeh, D.T. Mook, Nonlinear Oscillations, Wiley, New York, 1979.

[13] D.I. Okunbor, R.D. Skeel, Canonical Runge–Kutta–Nyström methods of orders five and six, J. Comput. Appl. Math. 51 (1994) 375–382.

[14] G. Rowlands, A numerical algorithm for Hamiltonian systems, J. Comput. Phys. 97 (1991) 235–239.

[15] J.M. Sanz-Serna, M.P. Calvo, Numerical Hamiltonian Problems, Chapman & Hall, London, 1994.

[16] J. Wisdom, M. Holman, J. Touma, Symplectic correctors, in: J.E. Marsden, G.W. Patrick, W.F. Shadwick (Eds.), Integration Algorithms and Classical Mechanics, American Mathematical Society, Providence, RI, 1996, pp. 217–244.

[17] H. Yoshida, Construction of higher order symplectic integrators, Phys. Lett. A 150 (1990) 262–269.