SIAM J. SCI. COMPUT.
Vol. 14, No. 5, pp. 1237–1252, September 1993

013

HIGH-ORDER SYMPLECTIC RUNGE–KUTTA–NYSTRÖM METHODS*

M. P. CALVO† AND J. M. SANZ-SERNA†

Abstract. A numerical method for ordinary differential equations is called symplectic if, when applied to Hamiltonian problems, it preserves the symplectic structure in phase space, thus reproducing the main qualitative property of solutions of Hamiltonian systems. The authors construct and test symplectic, explicit Runge–Kutta–Nyström (RKN) methods of order 8. The outcome of the investigation is that existing high-order, symplectic RKN formulae require so many evaluations per step that they are much less efficient than conventional eighth-order nonsymplectic, variable-step-size integrators even for low accuracy. However, symplectic integration is of use in the study of qualitative features of the systems being integrated.

Key words. Runge–Kutta–Nyström methods, symplectic integration, Hamiltonian problems, order conditions

AMS subject classifications. primary 65L05; secondary 70H05

1. Introduction. In this paper we are concerned with Runge–Kutta–Nyström (RKN) methods for the numerical integration of second-order systems of differential equations of the special form

$$d^2\mathbf{y}/dt = \mathbf{f}(\mathbf{y}), \qquad y = [y^1, y^2, \dots, y^N]^T, \tag{1.1}$$

or, equivalently, of first-order systems

$$\frac{d\mathbf{y}}{dt} = \dot{\mathbf{y}}, \qquad \frac{d\dot{\mathbf{y}}}{dt} = \mathbf{f}(\mathbf{y}).$$

For the RKN formula specified by the tableau

$$\begin{array}{c|ccc} \gamma_1 & \alpha_{11} & \cdots & \alpha_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_s & \alpha_{s1} & \cdots & \alpha_{ss} \\ \hline & b_1 & \cdots & b_s \\ \hline & \beta_1 & \cdots & \beta_s \end{array}, \tag{1.2}$$

the equations that describe the step $t_n \to t_{n+1} = t_n + h$ take the form

$$\mathbf{Y}_i = \mathbf{y}_n + h\gamma_i\dot{\mathbf{y}}_n + h^2\sum_{j=1}^{s}\alpha_{ij}\mathbf{f}(\mathbf{Y}_j),$$

$$\dot{\mathbf{y}}_{n+1} = \dot{\mathbf{y}}_n + h\sum_{i=1}^{s} b_i\mathbf{f}(\mathbf{Y}_i),$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\dot{\mathbf{y}}_n + h^2\sum_{i=1}^{s}\beta_i\mathbf{f}(\mathbf{Y}_i),$$

*Received by the editors April 20, 1992; accepted for publication (in revised form) September 9, 1992. This work was supported by Junta de Castilla y León project 1031-89 and by Dirección General de Investigación Científica y Técnica project PB89-0351.

†Departamento de Matemática Aplicada y Computacíon, Facultad de Ciencias, Universidad de Valladolid, Valladolid, Spain.

where $\mathbf{Y}_i$ denote the internal stages. Throughout the paper we suppose that in (1.2)

$$\beta_i = b_i(1-\gamma_i), \qquad 1 \le i \le s; \tag{1.3}$$

this is a standard assumption that significantly decreases the number of order conditions that must be imposed on the method coefficients to ensure a given order of consistency; see [10, Chap. 2, Lemma 13.13].

If the function $\mathbf{f}$ in (1.1) is the gradient of a scalar potential $-V = -V(\mathbf{y})$ and we set $\mathbf{p} = \dot{\mathbf{y}}$, $\mathbf{q} = \mathbf{y}$, then (1.1) may obviously be rewritten as

$$\frac{dp^I}{dt} = -\frac{\partial V}{\partial q^I}, \qquad \frac{dq^I}{dt} = p^I, \quad 1 \le I \le N. \tag{1.4}$$

This is the Hamiltonian system of ordinary differential equations

$$\frac{dp^I}{dt} = -\frac{\partial H}{\partial q^I}, \qquad \frac{dq^I}{dt} = \frac{\partial H}{\partial p^I}, \quad 1 \le I \le N$$

with Hamiltonian function

$$H = H(\mathbf{p},\mathbf{q}) = T(\mathbf{p}) + V(\mathbf{q}), \quad T(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T\mathbf{p}.$$

In mechanics the q variables represent Lagrangian coordinates, the p variables represent the corresponding momenta, T represents the potential energy, V represents the potential energy and H represents the total energy.

The recent literature has devoted much attention to the integration of Hamiltonian systems by means of canonical or symplectic methods; see [17] for a survey. A one-step numerical method is said to be canonical or symplectic if it preserves the so-called symplectic structure of the space of variables $(\mathbf{p}, \mathbf{q})$, thus reproducing the main qualitative property of solutions of Hamiltonian systems [2]. Suris [19] showed that the RKN method (1.2) is symplectic when applied to systems (1.4) if the coefficients satisfy the relations

$$b_i(\beta_j - \alpha_{ij}) = b_j(\beta_i - \alpha_{ji}), \qquad 1 \le i,j \le s; \tag{1.5}$$

see also [13]. On the other hand, if (1.2) does not possess redundant stages, (1.5) is also necessary for symplecticness; a rigorous proof of this necessity can be seen in [3] (cf. [1, §5]). In the remainder of the paper we use the expression "symplectic RKN method" to refer to RKN methods (1.2) that satisfy (1.5).

Okunbor and Skeel [14] studied the families of explicit, symplectic RKN methods with one, two, or three stages. The present authors [4], [5], [7] have constructed and tested an explicit, five-stage, fourth-order symplectic RKN method with optimized error constants. This method uses four function evaluations per step: the evaluation for the fifth stage of the current step coincides with the first evaluation in the next step (FSAL (first same as last) technique). In [15] Okunbor and Skeel construct explicit symplectic RKN formulae with five stages and seven stages and orders 5 and 6, respectively. For separable Hamiltonian systems [1], Yoshida [20] derives explicit, symplectic methods with order 8. When applied to problems of the form (1.1), Yoshida's methods reduce to RKN schemes of order 8 with 16 stages and 15 evaluations per step.

The experiments in [7] show that in the accurate long-time integration of problems of the form (1.4) the constant-step-size implementation of the fourth-order symplectic

formula constructed there is more efficient than a variable-step-size, fourth-order code based on an embedded RKN pair due to Dormand, El-Mikkawy, and Prince [8], [9]. The purpose of the present paper is to construct explicit, symplectic RKN methods of order 8 and to compare them with standard nonsymplectic RKN codes of the same order. The outcome of our investigation is that existing high-order, symplectic RKN formulae require so many evaluations per step that they are much less efficient than conventional eighth-order nonsymplectic, variable-step-size integrators, even for low accuracy. However, symplectic integration is of use in the study of qualitative features of the systems being integrated.

The structure of the paper is as follows. Section 2 reviews the theory of order conditions for symplectic RKN. Section 3 deals with the simplifying assumptions used later in the derivation of methods. In §4 we present a family of explicit, symplectic RKN methods. Specific order-7 methods within this family are constructed in §5. Section 6 is devoted to order-8 formulae, and §7 contains some numerical illustrations.

2. Order conditions for symplectic RKN methods. The conditions that must be imposed for an RKN method (1.2) for (1.1) to have order $\geq r$ are well known; the reader is referred to [10], whose terminology we follow. There is an order equation for each (rooted) SN-tree with r or fewer vertices (recall that (1.3) is assumed throughout). As an illustration, we have depicted in Fig. 1 the 10 SN-trees with six vertices. Furthermore, Table 1 displays the number m of SN-trees with r vertices, $1 \leq r \leq 10$. Clearly, for (1.2) to have order $\geq r$, the required number of order conditions is $\sum_{i=1}^{r} m_i$, a quantity that has also been tabulated in Table 1. Apparently, the generating function $M(z) = \sum_{i=1}^{\infty} m_i z^i$ for the sequence $\{m_r\}_{r=1}^{\infty}$ was first studied in [6].

In [6] we proved that the symplecticness conditions (1.5) act as simplifying assumptions, i.e., when (1.5) holds, not all order conditions are independent and some of them are implied by the remaining ones. For instance, for a symplectic method with order ≥ 5, the order condition associated with the tree $t_{6,2}$ is equivalent to the order condition associated with $t_{6,8}$. This comes about because $t_{6,2}$ and $t_{6,8}$ consist of the same vertices and edges and differ only in the location of the root. In other words, $t_{6,2}$ and $t_{6,8}$ are the same as unrooted SN-trees. For the same reason there is equivalence between $t_{6,3}$ and $t_{6,9}$, between $t_{6,4}$ and $t_{6,5}$, and among $t_{6,6}$, $t_{6,7}$ and $t_{6,10}$. Thus for a symplectic method with order ≥ 5 to have order 6 it is enough to impose five order conditions, one for each equivalence class $\{t_{6,1}\}$, $\{t_{6,2}, t_{6,8}\}$, $\{t_{6,3}, t_{6,9}\}$, $\{t_{6,4}, t_{6,5}\}$, $\{t_{6,6}, t_{6,7}, t_{6,10}\}$. The number m_r^* of corresponding equivalence classes for SN-trees with r vertices, $1 \leq r \leq 10$, is given in Table 1. The accumulated quantity $\sum_{i=1}^{r} m_i^*$ gives the total number of conditions for (1.2) subject to (1.5) to have order $\geq r$. A comparison of the third and fifth columns of Table 1 bears out the substantial reduction in order conditions implied by symplecticness.

3. Standard simplifying assumptions. Let us now leave aside the symplecticness conditions (1.5) and consider the well-known simplifying assumptions [10, Chap. 2, Lemma 13.14]

$$\sum_{j=1}^{s} \alpha_{ij} = \frac{\gamma_i^2}{2}, \qquad 1 \leq i \leq s, \tag{3.1}$$

that are often used in the construction of high-order RKN methods. When (1.2) satisfies (3.1), it is possible to disregard the order conditions associated with SN-trees with two or more vertices where at least one end vertex is fat. The order conditions for such trees are equivalent to order conditions for trees where all end vertices are meager. The basis

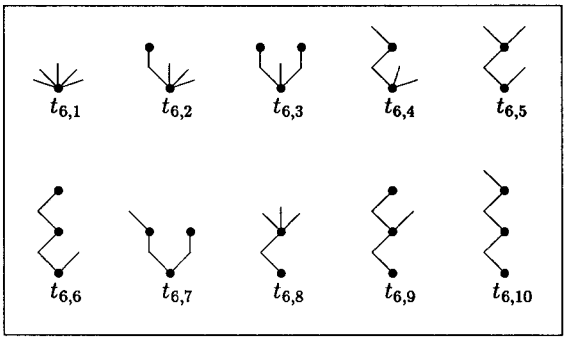

FIG. 1. SN-*trees with six vertices.*

TABLE 1

r	m_r	$\sum_{i=1}^{r} m_i$	m_r^*	$\sum_{i=1}^{r} m_i^*$	m_i'	$\sum_{i=1}^{r} m_i'$	$m_i'^*$	$\sum_{i=1}^{r} m_i'^*$
1	1	1	1	1	1	1	1	1
2	1	2	1	2	1	2	1	2
3	2	4	2	4	1	3	1	3
4	3	7	2	6	2	5	1	4
5	6	13	4	10	3	8	2	6
6	10	23	5	15	5	13	2	8
7	20	43	10	25	9	22	4	12
8	36	79	14	39	15	37	5	17
9	72	151	27	66	27	64	9	26
10	137	288	43	109	48	112	13	39

for this equivalence is illustrated in Fig. 2, where the order conditions for both trees are equivalent provided that the circle with the three branches at the bottom denotes in both cases the same arbitrary SN-tree. By iteration of the reduction in Fig. 2, the order condition for any tree with two or more vertices can be seen to be equivalent to order conditions for trees with only meager end vertices. For instance, in Fig. 1 the order condition for $t_{6,2}$ is the same as the order condition for $t_{6,1}$ and may be disregarded. For analogous reasons the order conditions for $t_{6,3}$, $t_{6,6}$, $t_{6,7}$, $t_{6,9}$ may be ignored, and this leaves five trees with six vertices to be considered. For general r we have the following result (caution: a prime does not mean differentiation!).

THEOREM 3.1. *Let m_r', $r \geq 2$ denote the number of* SN-*trees with r vertices without fat end vertices, and set $m_1' = 1$. Then*

$$m_r' = \sum_{\substack{k, j_1, j_3, \ldots, j_k \\ j_1 + 3j_3 + \cdots + kj_k = r-1}} \binom{m_2' + j_3 - 1}{j_3} \cdots \binom{m_{k-1}' + j_k - 1}{j_k}, \tag{3.2}$$

and the corresponding generating function $M'(z) = \sum_{r=1}^{\infty} m_r' z^r$ satisfies the equation

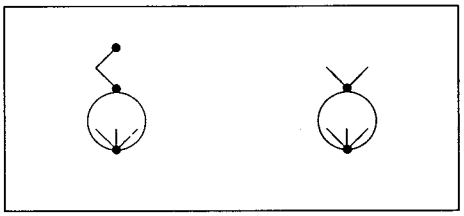

FIG. 2. *Equivalent* SN-*trees if* (3.1) *holds.*

$$M'(z) = \frac{z}{(1-z)(1-z^3)^{m'_2}\cdots(1-z^k)^{m'_{k-1}}\cdots}. \tag{3.3}$$

Proof. For $r \geq 2$ consider a special Nyström tree t with r vertices and remove its root. This gives rise to, say, j_1 graphs with one vertex, j_2 graphs with two vertices, etc. If t had no fat end vertex, then $j_2 = 0$ and for $k \geq 2$ each among the j_k graphs with k vertices consist of a meager vertex (that was a child of the root in the original t) followed by a special Nyström tree of order $k-1$ where all end vertices are meager. Hence for $k \geq 2$ the j_k graphs with k vertices can be chosen in

$$\binom{m'_{k-1} + j_k - 1}{jk}$$

different ways. This leads to (3.2). Equation (3.3) is a direct consequence of (3.2), along with the formula

$$\frac{1}{(1-z^r)^m} = \sum_{j=0}^{\infty} \binom{m+j-1}{j} z^{jr}. \qquad \square$$

The theorem makes it possible to recursively compute the m'_r. These have been tabulated in Table 1 for $1 \leq r \leq 10$. A comparison of the values of the quantities $\sum m_i^*$ and $\sum m'_i$ reveals that (1.5) and (3.1) leave roughly the same number of independent conditions to be considered. Therefore, in a sense (1.5) and (3.1) are as effective as simplifying assumptions. However, in (3.1) there are only s conditions to be imposed, whereas (1.5) comprises $s(s-1)/2$ relations (note that i and j play a symmetric role). Thus if one is not interested in Hamiltonian problems, (3.1) should clearly be preferred to (1.5). On the other hand, if to achieve symplecticness we impose (1.5), then we have reduced the number of order conditions by roughly the same amount we would have reduced that number by imposing the familiar simplifying assumptions (3.1).

The question arises of what happens when *both* (1.5) and (3.1) hold. For instance, for $r = 6, t_{6,1}, t_{6,2}, t_{6,3}$ are equivalent after (3.1) and $\{t_{6,2}, t_{6,8}\}$ and $\{t_{6,3}, t_{6,9}\}$ are equivalence classes for (1.5), so that $t_{6,1}, t_{6,2}, t_{6,3}, t_{6,8}, t_{6,9}$ all become equivalent. For the same reason, the order conditions for the remaining order-6 trees $t_{6,4}, t_{6,5}, t_{6,6}, t_{6,7}, t_{6,10}$ form a second equivalence class. Hence under (1.5) and (3.1) there are only two order conditions arising from order-6 trees. For general r let us say that two SN-trees t, t^* r vertices are S-equivalent if there exist a sequence of SN-trees $t_1, t_2, \ldots, t_k$ with $t_1 = t, t_k = t^*$, where t_i and t_{i+1}, $1 \leq i \leq k-1$, either are related as in Fig. 2 or differ only in the

location of the root. Thus if (1.5) and (3.1) hold and (1.2) has order $\geq r-1$, then order conditions for t and t^* are equivalent whenever t and t^* are S-equivalent.

THEOREM 3.2. *Let $m_r'^*$ denote the number of equivalence classes of* SN*-trees of order r under the relation* S. *Then the generating function $M'^*(z) = \sum_{r=1}^{\infty} m_r'^* z^r$ is given by*

$$M'^*(z) = M'(z) - \tfrac{1}{2}z(M'(z)^2 - M'(z^2)). \tag{3.4}$$

Proof. For a given $r \geq 2$ let us consider the m_r^* free or unrooted SN-trees. There is one such tree for each equivalence class based on (1.5). Those free trees that have one or more fat end vertex can be deleted in view of (3.1). Our task is to count the free trees that remain after such a deletion. As in [6], we resort to the notion of centroid of a free tree; see, e.g., [12], [18].

The following cases are possible for the free trees that remain.

(i) There is one centroid that is a meager vertex. By chopping off the centroid we obtain two (rooted) SN-trees. These must have the same order j, in view of the definition of centroid. Hence in this case r must be odd and $j = (r-1)/2$. For r odd $r > 3$ in view of the definition of m_j', $j \geq 2$, there are

$$\binom{m'_{r-1/2}+1}{2} = \frac{1}{2}m'_{(r-1)/2}(m'_{(r-1)/2}+1) \tag{3.5}$$

free trees in this case. For $r = 3$ obviously there is no free tree in this case.

(ii) There are two centroids. In this situation r must be even and the centroids are adjacent; one of them is fat and the other is meager. Chop off the meager centroid to get a rooted SN-tree of order $r/2$ without fat end vertices and to get a rooted SN-tree of order $r/2-1$ without fat end vertices. Therefore, for r even, $r > 4$, there are

$$m'_{r/2}m'_{r/2-1} \tag{3.6}$$

free trees in this category. For $r = 4$ this category is empty.

(iii) There is one centroid, and this is fat. The number of equivalence classes in this case is

$$m'_r - (m'_1 m'_{r-2} + \cdots + m'_{r-\max(3,r/2)}m'_{\max(3,r/2)}) \tag{3.7}$$

if r is even and

$$m'_r - (m'_1 m'_{r-2} + \cdots + m'_{r-\max(3,(r+1)/2)}m'_{\max(3,(r+1)/2)-1}) \tag{3.8}$$

if r is odd. These formulae are proved by an argument similar to that used to obtain [12, §2.3.4.4, formula (8)].

Formula (3.4) is a consequence of (3.5)–(3.8). □

The last column in Table 1 bears out the important reduction in the number of order conditions brought about by the combination of (1.5) and (3.1). In Fig. 3 we have depicted representatives of the 12 S-classes of equivalence to be considered for order ≥ 7, together with the corresponding order conditions.

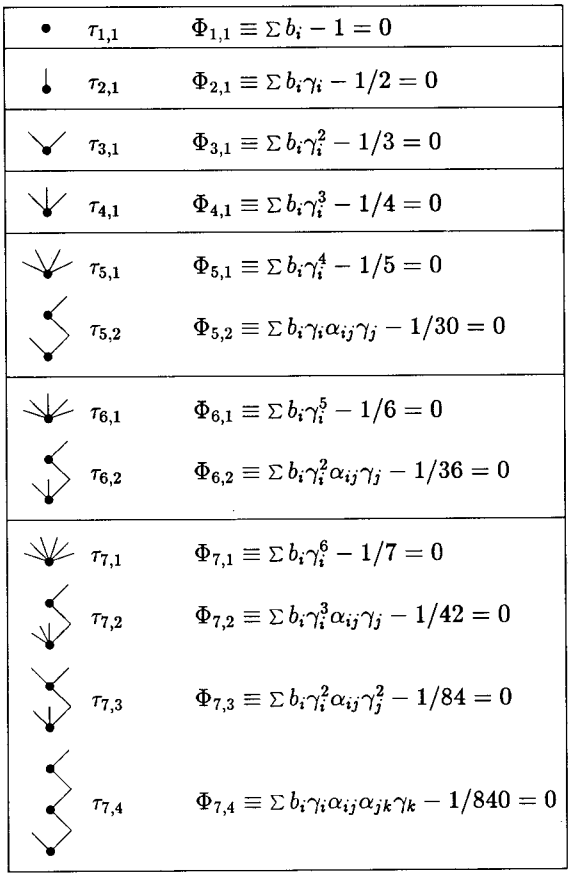

FIG. 3. *Order conditions for order* ≥ 7 *under* (1.5) *and* (3.1).

4. Family of explicit, symplectic RKN methods. Explicit symplectic RKN methods are of the form [14]

$$
\begin{array}{c|ccccc}
\gamma_1 & 0 & 0 & \cdots & 0 & 0 \\
\gamma_2 & b_1(\gamma_2-\gamma_1) & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\gamma_{s-1} & b_1(\gamma_{s-1}-\gamma_1) & b_2(\gamma_{s-1}-\gamma_2) & \cdots & 0 & 0 \\
\gamma_s & b_1(\gamma_s-\gamma_1) & b_2(\gamma_s-\gamma_2) & \cdots & b_{s-1}(\gamma_s-\gamma_{s-1}) & 0 \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s \\
\hline
 & \beta_1 & \beta_2 & \cdots & \beta_{s-1} & \beta_s
\end{array}
\tag{4.1}
$$

subject to (1.3). Thus with s stages there are $2s$ free parameters. For $s \geq 2$ we consider the subclass of methods given by

$$\gamma_1 = 0, \qquad \gamma_s = 1 \tag{4.2}$$

and

$$b_1 = \gamma_2/2; \quad b_i = (\gamma_{i+1} - \gamma_{i-1})/2, \quad 2 \leq i \leq s-1; \quad b_s = (1 - \gamma_{s-1})/2. \tag{4.3}$$

Note that (4.2) and (4.3) leave only $s-2$ free parameters $\gamma_2, \ldots, \gamma_{s-1}$ in (4.1). For arbitrarily fixed values of $\gamma_2, \ldots, \gamma_{s-1}$, the following properties are easily verified.

(i) Method (4.1)–(4.3) has the FSAL property: the sth stage of the current step coincides with the first stage of the next step. Thus (4.1)–(4.3) effectively require $s-1$ evaluations per step.

(ii) Method (4.1)–(4.3) satisfies the standard simplifying assumptions (3.1).

(iii) Method (4.1)–(4.3) has order ≥ 2, i.e., the conditions $\sum b_i = 1$, $\sum b_i \gamma_i = \frac{1}{2}$ are implied by the structure of tableau (4.1) and relations (4.2) and (4.3).

It is also useful to observe that a step of length h with (4.1)–(4.3) is a concatenation of $s-1$ steps of successive lengths $(\gamma_2 - \gamma_1)h, (\gamma_3 - \gamma_2)h, \ldots, (\gamma_s - \gamma_{s-1})h$ with the simple method

(4.4)

0	0	0
1	$\frac{1}{2}$	0
	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	0

that results after setting $s = 2$ in (4.1)–(4.3). Let us be more precise. Let ψ_h represent the transformation in $(\dot{\mathbf{y}}, \mathbf{y})$-space that effects a step of length h with (4.1)–(4.3), i.e., $(\dot{\mathbf{y}}_{n+1}, \mathbf{y}_{n+1}) = \psi_h(\dot{\mathbf{y}}_n, \mathbf{y}_n)$ if $(\dot{\mathbf{y}}_{n+1}, \mathbf{y}_{n+1})$ is the result of a step of length h from the preceding approximation $(\dot{\mathbf{y}}_n, \mathbf{y}_n)$. Let $\psi_h^{[2]}$ represent the corresponding transformation for (4.4). Then

$$\psi_h = \psi^{[2]}_{(\gamma_s - \gamma_{s-1})h} \cdots \psi^{[2]}_{(\gamma_2 - \gamma_1)h}. \tag{4.5}$$

This formula makes it easy to find the adjoint method ψ_h^* of ψ_h. Recall that by definition [10] ψ_h^* is the method such that ψ^*_{-h} inverts ψ_h, i.e., a step of length h with (4.1)–(4.3) followed by a step of length $-h$ with the adjoint method of (4.1)–(4.3) leaves the numerical solution unchanged. From (4.5)

$$\psi_h^* = [\psi^{[2]}_{(\gamma_s - \gamma_{s-1})h} \cdots \psi^{[2]}_{(\gamma_2 - \gamma_1)h}]^* = \psi^{[2]*}_{(\gamma_2 - \gamma_1)h} \cdots \psi^{[2]*}_{(\gamma_s - \gamma_{s-1})h},$$

but (4.4) is easily seen to be selfadjoint and hence

$$\psi_h^* = \psi^{[2]}_{(\gamma_2 - \gamma_1)h} \cdots \psi^{[2]}_{(\gamma_s - \gamma_{s-1})h}.$$

Comparison with (4.5) reveals that ψ_h^* is the method of the family (4.1)–(4.3) based on the abscissae $\gamma_2^*, \ldots, \gamma_{s-1}^*$ defined by $\gamma_i^* = 1 - \gamma_{s+1-i}$.

5. Constructing seventh-order methods. In this section we describe our experience in constructing order-7 methods of the form (4.1)–(4.3). Since both (1.5) and (3.1) hold, the last column in Table 1 shows that there are 12 order conditions to be imposed (see Fig. 3). However, the order conditions $\Phi_{1,1} = 0$, $\Phi_{2,1} = 0$ that guarantee order ≥ 2 are automatically satisfied, so that the choice of the $s - 2$ free parameters $\gamma_2, \ldots, \gamma_{s-1}$ should be directed toward enforcing the 10 remaining conditions

$$\Phi_{3,1} = 0, \quad \Phi_{4,1} = 0, \ldots, \quad \Phi_{7,4} = 0. \tag{5.1}$$

This suggests $s \geq 12$. The choice $s = 12$ leaves no freedom to tune the formula, and we set $s = 13$. We therefore undertook the task of numerically solving the nonlinear system (5.1) comprising 10 equations in the 11 unknowns $\gamma_2, \ldots, \gamma_{12}$. The solutions form curves in $\mathcal{R}^{11}$ that can be followed by continuation once a particular solution has been found.

Finding initial solutions of (5.1) to start the continuation procedure was not an easy task. After many unsuccessful attempts the following strategy was adopted. We began by considering the function

$$\lambda = \Phi_{3,1}^2 + \Phi_{4,1}^2 + \Phi_{5,1}^2 + \Phi_{5,2}^2 + \Phi_{6,1}^2$$

(see Fig. 3) of the variables $\gamma_2, \ldots, \gamma_{12}$. We minimized λ subject to bounds $-5 \leq \gamma_i \leq 5$, $2 \leq i \leq 12$, and to the equality constraints $\Phi_{6,2} = \Phi_{7,1} = \Phi_{7,2} = \Phi_{7,3} = \Phi_{7,4} = 0$. To this end the NAG routine E04UCF was used with the starting values for $\gamma_2, \ldots, \gamma_{12}$ generated randomly with the NAG routine G05DAF. Clearly, a solution of (5.1) is found whenever the objective function γ is successfully brought to its global minimum $\lambda = 0$ by the minimization routine.

For the continuation procedure we used one of the unknowns $\gamma_2, \ldots, \gamma_{12}$ as a continuation parameter. The particular unknown to be used at each step of the continuation procedure was determined as follows. Gaussian elimination with column pivoting was performed in the 10×11 Jacobian matrix of the system (5.1) evaluated at the current value of the solution. The parameter was chosen to be the unknown that was not used as a pivot, i.e., the unknown whose column would be in the 11th place if the columns were actually interchanged to carry out the pivoting. In a sense this identifies the unknown that is (locally in the solution curve) least constrained by (5.1) and that therefore is (locally) best suited for parametrizing the solutions of the system. Once the index $i_0, 2 \leq i_0 \leq 12$, of the unknown to be used as a parameter has been determined, we solved by Newton's method the 11×11 system given by (5.1), along with the equation $\gamma_{i_0} = \gamma_{i_0}^0 + \delta$, where $\gamma_{i_0}^0$ is the value of γ_{i_0} at the current solution and $\delta = 0.01$ denotes the increment in the parameter.

The coefficients of a specific method constructed by following this methodology are presented later in the paper.

6. Eighth-order methods. Once a method ψ_h of the class (4.1)–(4.3) with order $r = 7$ and $s = 13$ has been obtained, it is possible to use it so as to have eighth-order symplectic integration. In fact, it is enough to consider the method [16]

$$\bar{\psi}_h = \psi_{h/2}^* \psi_{h/2}. \tag{6.1}$$

A step of length h with the new method $\bar{\psi}_h$ consists of a step of length $h/2$ with the given order-7 formula followed by a step of length $h/2$ with the adjoint formula. The method $\bar{\psi}_h$ is symplectic as obtained by concatenating symplectic formulae and obviously has order ≥ 7. Furthermore, $\bar{\psi}_h$ is clearly selfadjoint, so that it has even order. Hence $\bar{\psi}_h$

is an eighth-order method. Note that $\bar{\psi}_h$ uses 24 evaluations per step as ψ_h and ψ_h^* are FSAL methods with 13 stages. The ratio number of stages per order is 26/8 versus the minimum 17/8 suggested by Table 1, but the situation is not bad at all. The approximation obtained after taking the first half-step $\psi_{h/2}$ is also globally accurate of the eighth order; the first half-step starts from an approximation with (global) error $O(h^8)$ and introduces a local error $O(h^8)$, so that the global error after the first half-step is $O(h^8)$. Hence output with global accuracy $O(h^8)$ is available after every 12 function evaluations. On the other hand, having 24 evaluations per step is certainly a drawback if variable step sizes are used: if a step is rejected, too many evaluations are wasted. Fortunately, symplectic integration should be used with constant step sizes [4], [5], [7] and so we feel that to find eighth-order formulae it is better to resort to the technique in (6.1) than to look directly at methods of the family (4.1)–(4.3).

The parameter in the continuation procedure used to find the seventh-order method ψ_h is chosen for the method $\bar{\psi}_h$ in (6.1) to have small error constants (see [8]). The $\dot{\mathbf{y}}$-truncation error and $\mathbf{y}$-truncation error of an eighth-order RKN method such as $\bar{\psi}_h$ have, respectively, the forms

$$h^9 \sum_j \Phi'_{9,j} \mathbf{F}_{9,j} + O(h^{10}) \tag{6.2}$$

and

$$h^9 \sum_k \Phi_{9,k} \mathbf{F}_{8,k} + O(h^{10}), \tag{6.3}$$

where $\mathbf{F}_{8,k}$ and $\mathbf{F}_{9,j}$ are elementary differentials that depend only on the system (1.1) being integrated and $\Phi'_{9,j}$, $\Phi_{9,k}$ are polynomials in the method coefficients α_{ij}, γ_i, β_i, b_i. In (6.2) the sum is extended to the 72 SN-trees of order 9, and in (6.3) the sum is extended to the 36 SN-trees of order 8 (see Table 1). (Note that in (6.2) and (6.3) the coefficients that are featured are those of the order-8 method, whereas in (5.1) we deal with the coefficients of the order-7 method. Also, in (6.2) and (6.3) all SN-trees are considered, whereas in (5.1) we took only one SN-tree per S-class of equivalence.) We try to minimize the Euclidean norm N of the vector with $72+36$ components $\Phi'_{9,j}$, $\Phi_{9,k}$. To this end, at each step of the continuation procedure described in §5 we evaluate N for the eighth-order method obtained $\bar{\psi}_h$, by means of (6.1), from the current seventh-order method ψ_h. The following coefficients identify the formula ψ_h that, among those we found, leads to the $\bar{\psi}_h$ with the lowest $N (N = 1.6 \times 10^{-5})$:

$$\gamma_2 = 0.60715821186110352503,$$

$$\gamma_3 = 0.96907291059136392378,$$

$$\gamma_4 = -0.10958316365513620399,$$

$$\gamma_5 = 0.05604981994113413605,$$

$$
\begin{aligned}
\gamma_6 &= 1.30886529918631234010,\\
\gamma_7 &= -0.11642101198009154794,\\
\gamma_8 &= -0.29931245499473964831,\\
\gamma_9 &= -0.16586962790248628655,\\
\gamma_{10} &= 1.22007054181677755238,\\
\gamma_{11} &= 0.20549254689579093228,\\
\gamma_{12} &= 0.86890893813102759275.
\end{aligned}
\tag{6.4}
$$

7. Numerical results. Although the numerical experiments presented in this section provide information on the advantages and disadvantages of symplectic integrators, they are limited in scope. More extensive testing is required before definite conclusions can be put forward.

We consider three explicit, symplectic methods, used with constant step sizes:

(i) S8: order 8, 26 stages, 24 evaluations per step, symplectic RKN formula associated with (6.4) by means as of (6.1). This has error constant $N = 1.6 \times 10^{-5}$.

(ii) Y8: order 8, 15 evaluations per step, symplectic method given in [20, Table 2, column D]. This is the order-8 method with the lowest value of N among those constructed in [20] and has $N = 4.4 \times 10^{-3}$.

(iii) S4: order 4, five stages, four evaluations per step, symplectic RKN formula constructed in [7].

As a reference standard (i.e., nonsymplectic) method we consider the following:

(iv) D8: order 8, nine stages, eight evaluations per step, RKN formula by Dormand, El-Mikkawy, and Prince [9, Table 1]. This has $N = 8.3 \times 10^{-7}$. An embedded order-6 method presented in [9] was used to estimate the error in a variable-step implementation. There are some printing errors in the method coefficients in the original [9], and the reader should see the corresponding corrigendum.

The values of N given above cannot be directly compared because the work per step is different for different methods. More informative error coefficient values can be obtained by assuming that a method with q function evaluations per step uses a step size of qh. If the method is of order p, this multiplies the values of N by q^p. For S8, Y8, and D8 the normalized values of the error coefficient turn out to be 1.8×10^6, 1.1×10^7, and 14, respectively. On raising these values to the power $-1/p = -1/8$, we obtain a crude measure of the efficiency of the various methods. The result is 0.16 for S8, 0.13 for Y8, and 0.71 for D8. The formulae S8 and Y8 are very demanding in function evaluations and are hence inefficient when compared with the nonsymplectic formula D8. This inefficiency is due to the $O(s^2)$ number of degrees of freedom in the RKN tableau that are used to enforce the symplecticness conditions (1.5).

Our first test problem was used in [7]. It corresponds to the Newton potential $V(q^1, q^2) = -1/\|\mathbf{q}\|$ (Kepler's problem) with initial condition

$$p^1 = 0, \quad p^2 - \sqrt{\frac{1+e}{1-e}}, \quad q^1 = 1 - e, \quad q^2 = 0.$$

Here e is the eccentricity of the orbit, $0 \leq e < 1$, that in the experiments to be reported is chosen to be $e = 0.5$. (The value of e does not significantly influence the outcome of the experiments.) The solution is 2π-periodic. Errors are measured in the Euclidean norm of $\mathcal{R}^4$.

Figure 4 corresponds to a final integration time $T = 810 \times 2\pi$ and depicts error at $t = T$ against number of function evaluations. The following runs are presented:

(i) S8, with step sizes $h = 2\pi/32, 2\pi/64, 2\pi/128$ (asterisks joined by a solid line).

(ii) Y8, with step sizes $h = 2\pi/64, 2\pi/128, 2\pi/256$ (plus signs joined by a dotted line).

(iii) S4, with step sizes $h = 2\pi/128, 2\pi/256, 2\pi/512, 2\pi/1024, 2\pi/2048$ ($\times$ joined by a dashed line).

(iv) D8, with absolute error tolerances $10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}, 10^{-11}, 10^{-12}$ (circles joined by dash-dot line).

Clearly, S8 is more efficient than Y8. Yoshida's method Y8 is more efficient than the lower-order method S4 when small errors are required. Otherwise, S4 is more efficient than Y8. However, the nonsymplectic method S8 is clearly more efficient than any of the symplectic formulae tested.

Figure 5 is similar to Fig. 4, but now the final time $T = 21870 \times 2\pi$ is longer. The runs depicted are as follows:

(i) S8, with step sizes $h = 2\pi/32, 2\pi/64, 2\pi/128$.

(ii) Y8, with step sizes $h = 2\pi/128, 2\pi/256$.

(iii) S4, with step sizes $h = 2\pi/256, 2\pi/512, 2\pi/1024, 2\pi/2048$.

(iv) D8, with absolute error tolerances $10^{-9}, 10^{-10}, 10^{-11}, 10^{-12}$.

The conclusions as to the relative efficiency of the methods are the same as above. However, the advantage of D8 is not so marked as before. This is due to the better error propagation properties of symplectic integrators [7]. In Fig. 6 we have depicted error against t (measured in periods) for S8 ($h = 2\pi/32$), Y8 ($h = 2\pi/128$), S4 ($h = 2\pi/256$), and D8 (TOL $= 10^{-9}$). Note that in this figure different methods are working differently; only the slopes in the different lines should be compared. In the symplectic methods the error grows linearly with t, whereas in the nonsymplectic method D8 the growth is as t^2, as is proved rigorously in [7]. Therefore, as the final integration time T increases, S8 and Y8 improve their efficiency relative to D8. However, the crossover point T for which S8 becomes more efficient than D8 is too large: perhaps T corresponds to millions of periods of the planet whose motion is being integrated.

From this experiment we conclude that if accurate solutions are needed, even for long integration times, a high-order standard code may easily be a better choice than a symplectic algorithm. For high-order RKN integrators too many ($O(s^2)$) degrees of freedom in the tableau are sacrificed to achieve symplecticness, and this sacrifice makes the formula very expensive relative to standard RKN methods. This should be compared with the conclusions in [7], where it is shown that for Kepler's problem fourth-order symplectic integrators are more efficient than fourth-order variable-step standard codes. In [7] the work per step of the fourth-order symplectic algorithm is $\frac{4}{3}$ of the work per step of the reference standard fourth-order algorithm and the advantages of symplecticness

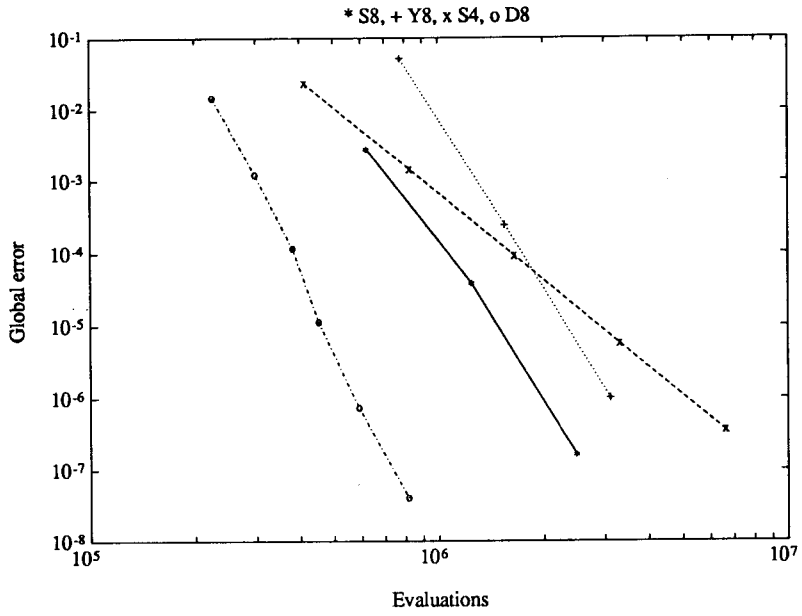

FIG. 4. *Efficiency plot at* $T = 810 \times 2\pi$.

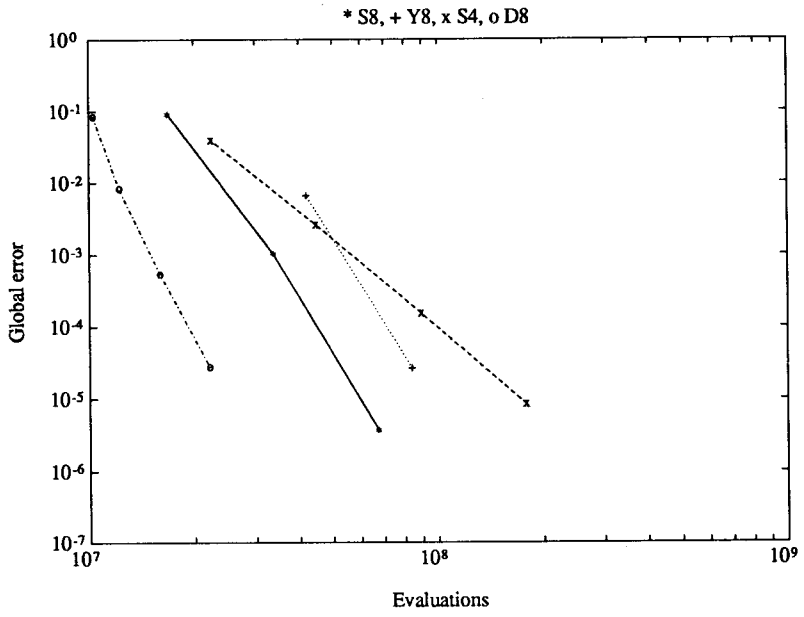

FIG. 5. *Efficiency plot at* $T = 21870 \times 2\pi$.

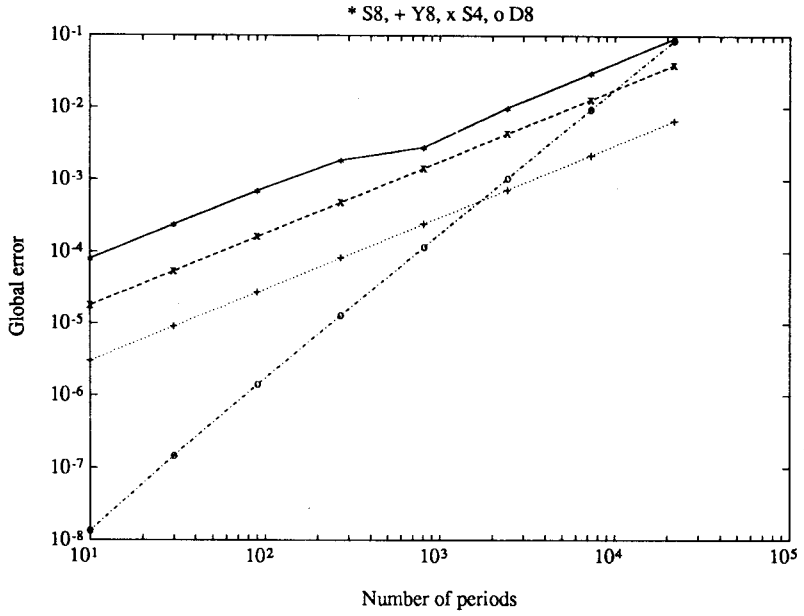

FIG. 6. *Error against t.*

make up for the increased cost per step. Here the work per step in S8 is three times the work per step in D8.

The second test problem is taken from Herbst and Ablowitz [11]. It originates from the sine-Gordon equation

$$u_{tt} - u_{xx} + \sin u = 0, \qquad 0 < x < L = 2\sqrt{2}\pi, \quad t > 0. \tag{7.1}$$

subject to periodic boundary conditions and to the initial conditions

$$u(x,0) = \pi + 0.1\cos(2\pi x/L), \qquad u_t(x,0) = 0. \tag{7.2}$$

Equation (7.1) may be thought of as describing the motion of a family of pendula. At each value of $x, 0 < x < L$, we have one pendulum. The term u_{xx} provides coupling between the motions of neighboring pendula. It represents a force that tries to keep a common value of the angle u for all the pendula. From the initial condition (7.2) we see that all pendula are initially left near the unstable equilibrium $u = \pi$. The pendula in $0 < x < L/4$ or $3L/4 < x < L$ start *above* the value $u = \pi$ and hence will increase u in order to approach the stable equilibrium at $u = 2\pi$. The pendula in $L/4 < x < 3L/4$ start *below* the value $u = \pi$ and will decrease u to approach the stable equilibrium at $u = 0$. This causes the term u_{xx} to become important. The effect of the restoring force is that the pendula are prevented from reaching the lowest $u = 2\pi$ or $u = 0$ positions and, rather, start going upward back to the initial positions, leading to a periodic motion. The solid curve in Fig. 7 represents u as a function of $t, 0 < t < 16L$, for the pendulum at $x = L/2$.

As in [11], (7.1) is discretized in space by the standard pseudospectral technique, with a mesh length $\Delta x = L/32$. This leads to a Hamiltonian system of the form (1.1),

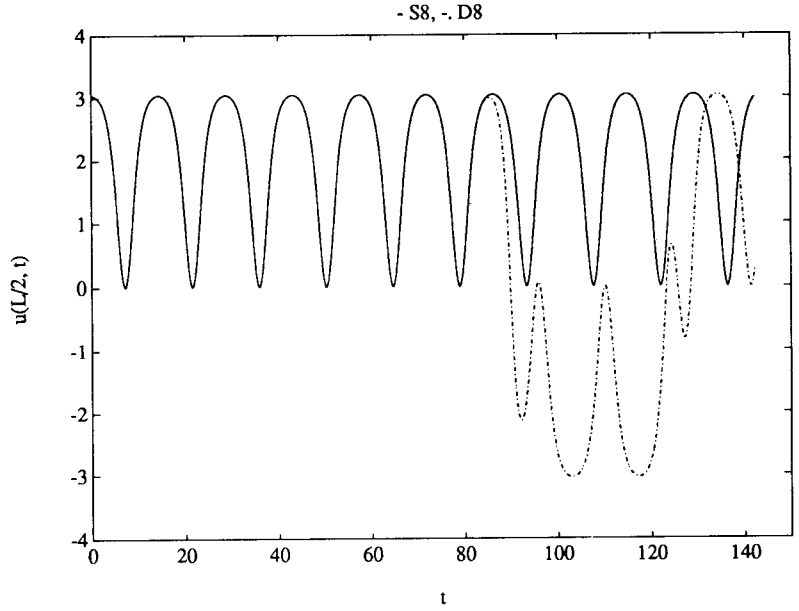

FIG. 7. $u(L/2, t)$ *against* t.

where the dependent variables $\mathbf{y}$ are the 32 discrete Fourier coefficients of the solution. This system of ordinary differential equations was integrated with the methods S8, Y8, S4, and D8 on $0 < t < 16L \approx 142.17$.

The standard method D8 was run with absolute error tolerances in the range 10^{-8} to 10^{-13}. Smaller tolerances were not tried because we felt that they would be too close to the size of the round-off error associated with the evaluation of the force $\mathbf{f}$ (this requires a couple of discrete Fourier transforms). None of the values of TOL we tried led to a successful integration, and D8 was not able to come up with the right qualitative behavior of the solution. The dash-dot line in Fig. 7 corresponds to TOL $= 10^{-13}$, $x = L/2$; the computed solution is completely wrong for $t > 80$. For this value of the tolerance the D8 code uses 32,810 function evaluations.

On the other hand, the symplectic algorithms S8, Y8, S4 were all able to identify the right qualitative behavior when run with suitable values of the step length h. With $h = \frac{1}{4}$ method S8 cannot faithfully describe the behavior of the solution up to the final time $t = 16L$. Halving the value of h to $h = \frac{1}{8}$ (27,312 evaluations) leads to a successful integration (see the curve in Fig. 7). For Y8, h has to be reduced down to $h = \frac{1}{16}$ (34,125 evaluations) and S4 requires $h = \frac{1}{32}$ (18,200 evaluations). Hence S4 was the most efficient, followed by S8 and Y8. Additional experiments show that if the final integration time is increased, further reduction of h is necessary to attain the correct qualitative behavior.

Following a referee's suggestion, we also integrated (7.1) and (7.2) in time by non-symplectic methods implemented with constant step sizes. The classical fourth-order Runge–Kutta method required $h = \frac{1}{64}$. This implies 36,400 function evaluations, which is more than any of the symplectic methods we tried. On the other hand, the order-8 formula of Dormand, El-Mikkawy, and Prince, when implemented with constant step sizes,

was found to be able to identify the correct qualitative behavior. However, this needed $h = \frac{1}{64}$ and 72,800 function evaluations, i.e., more than twice as much computational effort, as the least efficient symplectic formula Y8.

Acknowledgments. The authors are grateful to J. de Frutos (Valladolid) for his help in programming the pseudospectral method and to R. Fletcher (Dundee) for his suggestions in connection with the continuation procedure. We also thank J. Dormand (Teeside), who kindly provided us with several of his RKN codes.

REFERENCES

[1] L. Abia and J. M. Sanz-Serna, *Partitioned Runge–Kutta methods for separable Hamiltonian problems*, Math. Comp., to appear.
[2] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, 2nd ed., Springer-Verlag, New York, 1989.
[3] J. P. Calvo, *Métodos Runge–Kutta–Nyström Simplécticos*, Ph.D. thesis, Department of Applied Mathematics and Computation, University of Valladolid, Spain, 1992.
[4] M. P. Calvo and J. M. Sanz-Serna, *Variable steps for symplectic integrators*, in Numerical Analysis, 1991, D. F. Griffiths and G. W. Watson, eds., Longmans Press, London, 1992, pp. 32–48.
[5] ———, *Reasons for a failure. The integration of the two-body problem with a symplectic Runge–Kutta–Nyström code with stepchanging facilities*, in Applied Mathematics and Computation Reports, Report 1991/7, Universidad de Valladolid, Spain, 1991.
[6] ———, *Order conditions for canonical Runge–Kutta–Nyström methods*, BIT, 32 (1992), pp. 131–142.
[7] ———, *The development of variable-step symplectic integrators, with applications to the two-body problem*, SIAM J. Sci. Comput., 14 (1993), pp. 936–952.
[8] J. R. Dormand, M. E. El-Mikkawy, and J. P. Prince, *High-order embedded Runge–Kutta–Nyström formulae*, IMA J. Numer. Anal., 7 (1987), pp. 235–250.
[9] ———, *High-order embedded Runge–Kutta–Nyström formulae*, IMA. J. Numer. Anal., 7 (1987), pp. 423–430, corrigendum in 11 (1991), p. 297.
[10] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations* I, *Nonstiff Problems*, Springer-Verlag, Berlin (1987).
[11] B. M. Herbst and M. J. Ablowitz, *Numerical homoclinic instabilities in the Sine-Gordon Equation*, Quaestiones Mathematicae, 15 (1992), pp. 345–363.
[12] D. E. Knuth, *Fundamental Algorithms*, 2nd ed., Addison-Wesley, Reading, MA, 1973.
[13] D. Okunbor and R. D. Skeel, *An explicit Runge–Kutta–Nyström method is canonical if and only if its adjoint is explicit*, SIAM J. Numer. Anal., 29 (1992), pp. 521–527.
[14] ———, *Explicit canonical methods for Hamiltonian systems*, Math. Comp., 59 (1992), pp. 439–455.
[15] ———, *Canonical Runge–Kutta–Nyström Methods of Orders 5 and 6*, J. Comput. Appl. Math., to appear.
[16] J. M. Sanz-Serna, *The numerical integration of Hamiltonian systems*, in Computational Ordinary Differential Equations, J. R. Cash and I. Gladwell, eds., Clarendon Press, Oxford, 1992, pp. 437–449.
[17] ———, *Symplectic integrators for Hamiltonian problems: An overview*, in Acta Numerica 1992, Cambridge University Press, Cambridge, NY, 1992, pp. 243–286.
[18] J. M. Sanz-Serna and L. Abia, *Order conditions for canonical Runge–Kutta schemes*, SIAM J. Numer. Anal., 28 (1991), pp. 1081–1096.
[19] Y. B. Suris, *The canonicity of mappings generated by Runge–Kutta type methods when integrating the systems* $\ddot{x} = -\partial U/\partial x$, Zh. Vychisl. Mat. i Mat. Fiz., 29 (1989), pp. 202–211. (In Russian.); same as U.S.S.R. Comput. Math. and Math. Phys., 29 (1989) pp. 138–144.
[20] H. Yoshida, *Construction of higher order symplectic integrators*, Phys. Lett. A, 150 (1990), pp. 262–268.