MATHEMATICS OF COMPUTATION
Volume 72, Number 244, Pages 1803–1816
S 0025-5718(03)01546-1
Article electronically published on May 29, 2003

A TECHNIQUE TO CONSTRUCT SYMMETRIC VARIABLE-STEPSIZE LINEAR MULTISTEP METHODS FOR SECOND-ORDER SYSTEMS

B. CANO AND A. DURÁN

Abstract. Some previous works show that symmetric fixed- and variable-stepsize linear multistep methods for second-order systems which do not have any parasitic root in their first characteristic polynomial give rise to a slow error growth with time when integrating reversible systems. In this paper, we give a technique to construct variable-stepsize symmetric methods from their fixed-stepsize counterparts, in such a way that the former have the same order as the latter. The order and symmetry of the integrators obtained is proved independently of the order of the underlying fixed-stepsize integrators. As this technique looks for efficiency, we concentrate on explicit linear multistep methods, which just make one function evaluation per step, and we offer some numerical comparisons with other one-step adaptive methods which also show a good long-term behaviour.

1. Introduction

In a recent paper [6], we studied the error growth with time when integrating periodic orbits which are solutions of initial value problems of the form

$$\begin{aligned} \ddot{Y}(t) &= F(Y(t)), \\ Y(t_0) &= \tilde{y}_0, \\ \dot{Y}(t_0) &= \tilde{v}_0. \end{aligned} \tag{1.1}$$

We assume F is a C^∞-function and that the integration is done with variable-stepsize linear multistep methods especially designed for these second-order systems. These integrators (denoted as VSLMM2s) are determined by the difference equation

$$\sum_{l=0}^{k} A_l(h_{n+k-1},\dots,h_n)Y_{n+l} = h_{n+k-1}^2 \sum_{l=0}^{k} B_l(h_{n+k-1},\dots,h_n)F(Y_{n+l}), \quad n>0, \tag{1.2}$$

and a starting procedure $(Y_0,\dots,Y_{k-1})^T$, where k is the stepnumber. In this formula Y_n approximates $Y(t_n)$, and h_n is the corresponding stepsize $t_{n+1}-t_n$. The

Received by the editor Deember 26, 2000 and, in revised form, April 30, 2002.
2000 *Mathematics Subject Classification.* Primary 65L06, 70F05, 70H33.
Key words and phrases. Explicit linear multistep methods, variable stepsizes, error growth, reversible second-order systems, symmetric integrators, efficiency, high-order methods.
This work was supported by DGICYT PB95–705 and JCL VA36/98.

interest of the generalization from fixed stepsize to variable stepsize was the integration of moderately eccentric orbits, like those which turn up in the solar system. In such a way, the coefficients $\{A_l\}$ cannot be constants any more but must depend on the stepsizes to keep the order, as stated in (1.2). In the study done in [6], it was assumed that, given a tolerance ϵ, the stepsize h_n is such that

$$h_n = \epsilon s(Y_n, \epsilon), \tag{1.3}$$

for a function s which satisfies

- $s_{\min} \leq s(Y_n, \epsilon) \leq s_{\max}$, with $s_{\min}, s_{\max} > 0$,
- s is C^∞ in both arguments and all the derivatives of s are bounded,

for ϵ small enough and Y_n in a bounded domain.

The main conclusion in [6] was that symmetric VSLMM2s whose underlying fixed-stepsize linear multistep method of second order (FSLMM2) has no double roots in their first characteristic polynomial except the root 1 lead to very advantageous error growth when integrating reversible systems, i.e., systems of the form (1.1) for which there exists a linear involution $\bar{\Lambda}$ such that

$$\bar{\Lambda} \circ F \circ \bar{\Lambda} = F.$$

Besides, the starting procedure for which we proved the best behaviour was that which considered $(Y_0, \dots, Y_{k-1})^T$ as exact. Therefore, we will concentrate in this paper on the exact starting values. On the other hand, the conditions for a VSLMM2 to be symmetric were proved there to be the following. In a first place, a sufficient condition on the coefficients of the method is that

$$\begin{aligned} A_l(h_{n+k-1}, \dots, h_n) &= A_{k-l}(h_n, \dots, h_{n+k-1}), \quad l = 0, \dots, k. \\ B_l(h_{n+k-1}, \dots, h_n) &= \frac{h_n^2}{h_{n+k-1}^2} B_{k-l}(h_n, \dots, h_{n+k-1}), \quad l = 0, \dots, k. \end{aligned} \tag{1.4}$$

Secondly, the stepsize function must also be symmetric in the sense that

$$s(Y_n, \epsilon) = s(Y_{n+1}, -\epsilon), \tag{1.5}$$

for s in (1.3). This condition is satisfied, for example, if

$$h_n = \frac{\epsilon}{2}(\tau(Y_n) + \tau(Y_{n+1})), \tag{1.6}$$

for some function τ [15]. Apart from symmetry, (1.6) gives reversibility with respect to $\bar{\Lambda}$ whenever

$$\tau(\bar{\Lambda} y) = \tau(y). \tag{1.7}$$

The aim of this paper is to construct explicit symmetric VSLMM2s in an effective and efficient way. Notice that explicit linear multistep methods just make one function evaluation per step in contrast with one-step methods which usually need many more, mainly when we require them to be high-order. Notice also that considering symmetric variable stepsizes means solving (1.6) implicitly. With explicit linear multistep methods, that means calculating new coefficients A_l, B_l each time a stepsize is rejected, but the function F is just evaluated once per step. That does not happen, for example, with explicit Runge-Kutta-Nyström methods, in which the function must be evaluated every time a stepsize is rejected. For one-step methods and first-order systems, some techniques have been developed in [10], [13], [17], [20] to avoid this problem of efficiency. The procedure is to reduce the problem to another reversible or Hamiltonian problem, in which fixed-stepsizes can be considered efficiently. The idea there is to consider a reparametrization of the problem which conserves reversibility or a modified Hamiltonian which has a common solution with the original. These techniques are very much suitable when integrating general first-order systems, but we cannot apply these ideas to our LMM2s because the reparametrized system would not have the form (1.1) any more (the first derivate would necessarily turn up). On the other hand, notice that symmetric LMMs for first-order systems lead to exponential error growth in general [8]. As there are only a few particular cases for which they do not [5], [9], there is a small range for finding optimal methods. Then, we concentrate on LMMs for second-order systems, for which the range is much wider. For FSLMM2s, there are already some optimal suggestions in [19]. Our purpose is to generalize them to variable stepsizes.

The main difficulty comes from finding easy expressions for the coefficients $\{A_l\}$ and $\{B_l\}$ in (1.2), which is aggravated when the order of the method increases (we remind the reader that the higher the order the more the advantages of VSLMM2s over one-step Runge-Kutta-Nyström methods with respect to function evaluations). In [6], we constructed a 4th-step 4th-order VSLMM2s by imposing the conditions of order and symmetry and using a symbolic package to solve the equations. The expressions obtained there were already quite complicated and led to long rational expressions of third degree for B_l. Trying to get symbolic equivalent expressions for high-order methods is not an easy task, as well as the choice of appropiate parameters which make those expressions simple and coincide with their fixed-stepsize counterparts.

The paper is structured as follows. In section 2, we provide a technique to find the coefficients of an explicit symmetric VSLMM2 from its fixed-stepsize counterpart. We do not get a closed expression for the coefficients but they are calculated recursively. The procedure can be applied independently of the order of the underlying symmetric FSLMM2; we prove that it gives rise to a symmetric VSLMM2 with the same order. Here, we will assume the explicit FSLMM2 has as order its stepnumber, which is the best which can be achieved [11] among explicit LMM2s. As the method is symmetric, we always take into account that the stepnumber is even. In section 3, we give some details of implementation as well as a numerical comparison for different problems between the obtained VSLMM2s and symplectic as well as symmetric one-step methods which also lead to slow error growth [2], [7], [12]. Some concluding remarks are stated in section 4.

2. Effective construction of a symmetric VSLMM2

2.1. **Description of the technique for a particular case.** Let us assume in this section that $k = 4$. Then, we consider a 4th-order symmetric FSLMM2, from which to construct the corresponding VSLMM2. We will denote by A_i (B_i, respectively) the $A_i(h_{n+3}, \dots, h_n)$ ($B_i(h_{n+3}, \dots, h_n)$, respectively) once the stepsizes are known, and by $\tilde{A}_i$ ($\tilde{B}_i$, respectively) the coefficients of the fixed-stepsize method. Also to simplify notation, we will call $h_0, \dots, h_3$ to $h_n, \dots, h_{n+3}$, always taking into account that the stepsizes are changing with the stepnumber n. We will take as B_i

$$B_i = \frac{h_0}{h_3}\tilde{B}_i, \qquad i = 0, \dots, 4,$$

in such a way that the symmetry conditions (1.4) for the variable-stepsize method are satisfied for the coefficients of the second characteristic polynomial.

In [6] it is proved that it suffices to impose order 2 to a VSLMM2 associated to a 4th-order FSLMM to obtain in fact a 4th-order VSLMM2. To impose order 2, we need the formula to be exact for the polynomials of degree ≤ 3. Therefore, for the basis

$$\{1,(t-t_n),(t-t_n)(t-t_{n+4}),(t-t_n)(t-t_{n+2})(t-t_{n+4})\},$$

the following must be satisfied:

$$A_0+A_1+A_2+A_3+A_4=0, \tag{2.1}$$

$$A_1h_0+A_2(h_0+h_1)+A_3(h_0+h_1+h_2)+A_4(h_0+h_1+h_2+h_3)=0, \tag{2.2}$$

$$\begin{aligned}&-A_1h_0(h_1+h_2+h_3)-A_2(h_0+h_1)(h_2+h_3)-A_3(h_0+h_1+h_2)h_3\\&\qquad=2h_0h_3[\tilde{B}_1+\tilde{B}_2+\tilde{B}_3],\end{aligned} \tag{2.3}$$

$$\begin{aligned}&A_1h_0h_1(h_1+h_2+h_3)-A_3(h_0+h_1+h_2)h_2h_3\\&\qquad=2h_0h_3[\tilde{B}_1(h_0-2h_1-h_2-h_3)+\tilde{B}_2(h_0+h_1-h_2-h_3)\\&\qquad\qquad+\tilde{B}_3(h_0+h_1+2h_2-h_3)].\end{aligned} \tag{2.4}$$

At this moment, we have four equations and five unknowns, but the symmetry conditions (1.4) have not yet been imposed on the coefficients $\{A_l\}$. In order to get that, we suggest taking

$$A_1=\frac{C(h_3,h_2,h_1,h_0)}{h_0h_1(h_1+h_2+h_3)},\quad A_3=\frac{C(h_0,h_1,h_2,h_3)}{h_3h_2(h_0+h_1+h_2)},$$

for a function $C(\cdot,\cdot,\cdot,\cdot)$ which allows (2.4) to be satisfied. We will denote the right-hand side of (2.4) as $T(h_3,h_2,h_1,h_0)$. This function is odd in the sense that

$$T(h_3,h_2,h_1,h_0)=-T(h_0,h_1,h_2,h_3).$$

Because of this, we suggest considering

$$C(h_3,h_2,h_1,h_0)=\frac{1}{2}T(h_3,h_2,h_1,h_0)+\tilde{A}_1 3h_0\sqrt{h_1h_2}h_3,$$

in such a way that (2.4) is satisfied as well as the coincidence of A_1 and A_3 with $\tilde{A}_1$ and $\tilde{A}_3$ for fixed stepsizes. From here, the only thing to do is to solve for A_2 from (2.3), then for A_4 from (2.2), and finally for A_0 from (2.1). Notice that in this case, A_2 is symmetric in the sense of (1.4) because of the symmetry of A_1 and A_3, that of their coefficients in (2.3), and that of the right-hand side of (2.3). Then, A_4 and A_0 can also be seen to turn out symmetric.

2.2. **General case. Proof of order and symmetry.** For general k, we suggest

$$B_i=\frac{h_0}{h_{k-1}}\tilde{B}_i,\quad i=0,\ldots,k, \tag{2.5}$$

and then, the coefficients $\{A_l\}$ should satisfy the equations which say that the formula is exact for the basis $\{p_l(t)\}_{l=0}^{k-1}$, recursively defined by

$$\begin{array}{rcl}p_0(t)&=&1,\\p_{2j+1}(t)&=&(t-t_j)p_{2j}(t),\quad j=0,\ldots,\frac{k}{2}-2,\\p_{2j+2}(t)&=&(t-t_{k-j})p_{2j+1}(t),\quad j=0,\ldots,\frac{k}{2}-2,\\p_{k-1}(t)&=&(t-t_{\frac{k}{2}})p_{k-2}(t),\end{array} \tag{2.6}$$

(again to simplify notation we are denoting t_{n+i} by t_i). That is, the selected basis is obtained by incorporating the monomials $(t-t_j)$ in a symmetric way. This is the key point to finding a symmetric method in the end. In such a way, the resulting method has order $\geq k-2$. Again we have k equations and $k+1$ unknowns, of which we must take advantage to impose symmetry. As the equation corresponding to the last element of the basis in (2.6) just involves the coefficients $A_{\frac{k}{2}-1}$ and $A_{\frac{k}{2}+1}$ of the first characteristic polynomial, we suggest taking

(2.7)
$$\begin{aligned} A_{\frac{k}{2}-1} &= (-1)^{\frac{k}{2}} \frac{C(h_{k-1},\dots,h_0)}{\prod_{l=0}^{\frac{k}{2}-2}(h_{\frac{k}{2}-2}+\cdots+h_l)h_{\frac{k}{2}-1}\prod_{l=\frac{k}{2}+1}^{k-1}(h_{\frac{k}{2}-1}+\cdots+h_l)}, \\ A_{\frac{k}{2}+1} &= (-1)^{\frac{k}{2}} \frac{C(h_0,\dots,h_{k-1})}{\prod_{l=0}^{\frac{k}{2}-2}(h_l+\cdots+h_{\frac{k}{2}})h_{\frac{k}{2}}\prod_{l=\frac{k}{2}+1}^{k-1}(h_{\frac{k}{2}+1}+\cdots+h_l)}, \end{aligned}$$

where

$$\begin{aligned} C(h_{k-1},\dots,h_0) &= \frac{1}{2}T(h_{k-1},\dots,h_0) \\ &\quad + (-1)^{\frac{k}{2}}\tilde{A}_{\frac{k}{2}-1}h_0\dots h_{\frac{k}{2}-2}\sqrt{h_{\frac{k}{2}-1}h_{\frac{k}{2}}}\,h_{\frac{k}{2}+1}\dots h_{k-1}\frac{(\frac{k}{2}-1)!(\frac{k}{2}+1)!}{2}, \end{aligned} \tag{2.8}$$

with

$$T(h_{k-1},\dots,h_0) = h_0h_{k-1}\sum_{l=1}^{k-1}\tilde{B}_l\ddot{p}_{k-1}(t_l),$$

(T is the right-hand side of the kth equation). The rest of the coefficients will be found in the order

$$A_{\frac{k}{2}}, A_{\frac{k}{2}+2}, A_{\frac{k}{2}-2}, A_{\frac{k}{2}+3}, A_{\frac{k}{2}-3}, \dots, A_k, A_0, \tag{2.9}$$

by solving backwards the "nearly" triangular system of equations which gives the consistency. This structure of the system is due to the selected basis (2.6).

To continue, let us consider the following notation. For each expression in which we want to change h_{k-l-1} to h_l, we will do it through the superindex R. Notice that

$$(t_l - t_i) = h_i + \cdots + h_{l-1} = (h_{k-i-1} + \cdots + h_{k-l})^R = -(t_{k-l} - t_{k-i})^R. \tag{2.10}$$

The following lemmas as well as the symmetry proved below are keys to determining the order of the method.

Lemma 2.1. *T is odd in the sense that*

$$T(h_{k-1},\dots,h_0) = -T(h_0,\dots,h_{k-1}).$$

Proof. The key point is (2.10), that p_{k-1} has odd degree and the indexes of its factors are symmetric. □

Lemma 2.2. *The underlying fixed-stepsize method is the one selected.*

Proof. Because of Lemma 2.1, $T(h,\dots,h) = 0$ and therefore in that case

$$A_{\frac{k}{2}-1} = \tilde{A}_{\frac{k}{2}-1}, \qquad A_{\frac{k}{2}+1} = \tilde{A}_{\frac{k}{2}+1}.$$

The rest of the coefficients must coincide with their fixed-stepsize counterparts because they are completely determined by the consistency equations. □

The following theorem assures the symmetry of the constructed VSLMM2.

Theorem 2.3. *The coefficients $\{A_l\}$ and $\{B_l\}$ satisfy* (1.4).

Proof. The symmetry of $\{B_l\}$ is obvious from (2.5), as is that of $A_{k/2-1}$ and $A_{k/2+1}$ from (2.7). Then, $A_{k/2}$ is solved from the equation corresponding to the $(k-1)$th-element of the basis (2.6). But the symmetry of the indexes of t_l associated to this element, that of its second derivative which now has even degree, as well as the already proved symmetry of $A_{k/2-1}$ and $A_{k/2+1}$ (which are the only other coefficients A_l which turn up in this equation), also make $A_{k/2}$ symmetric. From here, for $l = 2, \ldots, k/2$, we should solve consecutively for $A_{k/2+l}$ and $A_{k/2-l}$ in two adjacent equations going backwards:

$$\begin{aligned} \sum_{j=-l+1}^{l} A_{\frac{k}{2}+j} p_{k-2l+1}(t_{\frac{k}{2}+j}) &= h_0 h_{k-1} \sum_{j=1}^{k-1} \tilde{B}_j \ddot{p}_{k-2l+1}(t_j), \\ \sum_{j=-l}^{l} A_{\frac{k}{2}+j} p_{k-2l}(t_{\frac{k}{2}+j}) &= h_0 h_{k-1} \sum_{j=1}^{k-1} \tilde{B}_j \ddot{p}_{k-2l}(t_j). \end{aligned} \tag{2.11}$$

Notice also the following relations between p_{k-2l+1} and p_{k-2l}:

$$\begin{aligned} p_{k-2l+1}(t_j) &= (t_j - t_{\frac{k}{2}-l}) p_{k-2l}(t_j), \\ \ddot{p}_{k-2l+1}(t_j) &= 2\dot{p}_{k-2l}(t_j) + (t_j - t_{\frac{k}{2}-l}) \ddot{p}_{k-2l}(t_j). \end{aligned} \tag{2.12}$$

Now, because of the even and odd degrees of the corresponding polynomials and the symmetry of the indexes associated to it,

$$\begin{aligned} p_{k-2l}(t_j) &= p_{k-2l}(t_{k-j})^R, \\ \dot{p}_{k-2l}(t_j) &= -\dot{p}_{k-2l}(t_{k-j})^R, \\ \ddot{p}_{k-2l}(t_j) &= \ddot{p}_{k-2l}(t_{k-j})^R. \end{aligned} \tag{2.13}$$

From (2.11),

$$A_{\frac{k}{2}+l} p_{k-2l+1}(t_{\frac{k}{2}+l}) = h_0 h_{k-1} \sum_{j=1}^{k-1} \tilde{B}_j \ddot{p}_{k-2l+1}(t_j) - \sum_{j=-l+1}^{l-1} A_{\frac{k}{2}+j} p_{k-2l+1}(t_{\frac{k}{2}+j}), \tag{2.14}$$

$$A_{\frac{k}{2}-l} p_{k-2l}(t_{\frac{k}{2}-l}) = h_0 h_{k-1} \sum_{j=1}^{k-1} \tilde{B}_j \ddot{p}_{k-2l}(t_j) - \sum_{j=-l+1}^{l} A_{\frac{k}{2}+j} p_{k-2l}(t_{\frac{k}{2}+j}). \tag{2.15}$$

The matter to check is whether or not $A_{\frac{k}{2}-l} = A^R_{\frac{k}{2}+l}$, which is equivalent to proving whether or not

$$(t_{\frac{k}{2}+l} - t_{\frac{k}{2}-l}) A_{\frac{k}{2}-l} p_{k-2l}(t_{\frac{k}{2}-l}) = (t_{\frac{k}{2}+l} - t_{\frac{k}{2}-l}) A^R_{\frac{k}{2}+l} p_{k-2l}(t_{\frac{k}{2}+l})^R, \tag{2.16}$$

because of (2.13). Due to (2.15) and (2.12), the expression on the left in (2.16) can be written as

$$\begin{aligned} &(t_{\frac{k}{2}+l} - t_{\frac{k}{2}-l}) A_{\frac{k}{2}-l} p_{k-2l}(t_{\frac{k}{2}-l}) \\ &\quad = (t_{\frac{k}{2}+l} - t_{\frac{k}{2}-l}) \left(h_0 h_{k-1} \sum_{j=1}^{k-1} \tilde{B}_j \ddot{p}_{k-2l}(t_j) - \sum_{j=-l+1}^{l-1} A_{\frac{k}{2}+j} p_{k-2l}(t_{\frac{k}{2}+j}) \right) \\ &\qquad - A_{\frac{k}{2}+l} p_{k-2l+1}(t_{\frac{k}{2}+l}). \end{aligned}$$

Considering (2.14) and (2.12), this equals

$$h_0 h_{k-1} \sum_{j=1}^{k-1} \tilde{B}_j (t_{\frac{k}{2}+l} - t_j) \ddot{p}_{k-2l}(t_j) \\ - \sum_{j=-l+1}^{l-1} A_{\frac{k}{2}+j}(t_{\frac{k}{2}+l} - t_{\frac{k}{2}+j}) p_{k-2l}(t_{\frac{k}{2}+j}) - 2h_0 h_{k-1} \sum_{j=1}^{k-1} \tilde{B}_j \dot{p}_{k-2l}(t_j).$$

On the other hand, for the expression on the right in (2.16), due to (2.12) and (2.10),

$$(t_{\frac{k}{2}+l} - t_{\frac{k}{2}-l}) A^R_{\frac{k}{2}+l} p_{k-2l}(t_{\frac{k}{2}+l})^R = A^R_{\frac{k}{2}+l} p_{k-2l+1}(t_{\frac{k}{2}+l})^R.$$

Now, by using (2.14) in reverse order, (2.13) and (2.12), this is

$$h_0 h_{k-1} \sum_{j=1}^{k-1} \tilde{B}_j \Big(-2\dot{p}_{k-2l}(t_{k-j}) + (t_{\frac{k}{2}+l} - t_{k-j}) \ddot{p}_{k-2l}(t_{k-j}) \Big) \\ - \sum_{j=-l+1}^{l-1} A_{\frac{k}{2}-j}(t_{\frac{k}{2}+l} - t_{\frac{k}{2}-j}) p_{k-2l}(t_{\frac{k}{2}-j}).$$

So, after a change of indexes and the symmetry of $\{\tilde{B}_j\}$, (2.16) is proved. □

The following lemmas will be useful in the proof of the order of the method.

From the results above, we deduce the order of the method.

Theorem 2.4. *The constructed VSLMM2 has order k when the selection of the stepsize is symmetric* (1.5).

Proof. Remark 2.11 in [6] will be used, which says that whenever a symmetric VSLMM2 has order of consistency $\geq r-2$ and order r with fixed stepsize (r even), then the method also has order of consistency r with variable stepsizes.

Thanks to Lemma 2.1 and the symmetry of the right-hand side of (2.8), the last equation imposing order $\geq k-2$ will be satisfied. The rest of the equations are satisfied just because of the way in which the terms in (2.9) are determined. Since the VSLMM2 has order $\geq k-2$ and is symmetric and since the underlying fixed-stepsize method has order k, this implies that the VSLMM2 has in fact order k. □

3. Numerical study

3.1. **Implementation.** We describe some implementation details looking for the efficiency of the methods proposed.

At each change of stepsize, the coefficients of the method must be calculated solving the "nearly" triangular system which imposes that the formula of the method is exact for the basis (2.6). So, the first thing to do is to find the coefficients of that linear system as well as its independent term. Therefore, we need to calculate $\{p_l(t_j)\}$ and $\sum_{j=1}^{k-1} \tilde{B}_j \ddot{p}_l(t_j)$ for $l = 0, 1, \ldots, k-1$. For that, we use the recurrent relations (2.6), which imply

$$\begin{aligned} &p_l(t_j) = (t_j - t_Q) p_{l-1}(t_j), \\ (3.1) \qquad &\dot{p}_l(t_j) = p_{l-1}(t_j) + (t_j - t_Q)\dot{p}_{l-1}(t_j), \\ &\ddot{p}_l(t_j) = 2\dot{p}_{l-1}(t_j) + (t_j - t_Q)\ddot{p}_{l-1}(t_j), \quad j = 0, \ldots, k, \quad l = 1, \ldots, k-1, \end{aligned}$$

where t_Q is the corresponding point according to (2.6) (notice that (2.12) was a subset of these relations). Therefore, the computation of $\{p_l(t_j), \dot{p}_l(t_j), \ddot{p}_l(t_j)\}$ is obtained through the computation of the previous set $\{p_{l-1}(t_j), \dot{p}_{l-1}(t_j), \ddot{p}_{l-1}(t_j)\}$. This leads to a recurrent way to calculate the coefficients and independent vector of the system.

Some more implementation details can be given. First of all, we scale the stepsizes $h_j, j = 0, \ldots, k$, in order to avoid roundoff errors in the calculus of the coefficients of the system. On the other hand, $p_l(t)$ is just evaluated at the points t_j in which we know they do not vanish; this is the fact which makes the system "nearly" triangular.

A last remark concerns the computation of the independent terms $\sum_{j=1}^{k-1} \tilde{B}_j \ddot{p}_l(t_j)$ for $l = 2, \ldots, k-1$. After the calculation of $\ddot{p}_l(t_j)$, these terms require in principle $(k-2)(k-1)$ products. This can be reduced by using a slight modification of (3.1). We multiply $p_0(t_j)$ by $\tilde{B}_j$ and then apply the relations in (3.1). In such a way, no product is needed for the independent terms. As the coefficient matrix of the new system changes, we solve it and then scale each unknown appropiately. The latter just implies k products.

3.2. **Numerical experiments.** The main goal of the numerical experiments presented below is to compare the behaviour of the methods analysed in this article with other efficient one-step integrators of the same order, which are problem-adaptive and show a good long-term behaviour. We have considered VSLMM2s of orders 4 and 8 (denoted as VSLMM2-4 and VSLMM2-8, respectively) based on the respective FSLMM2s (R, S)

$$
\begin{aligned}
R(x) &= (x^2 + \frac{19}{10}x + 1)(x-1)^2, \qquad S(x) = \frac{53}{40}x^3 + \frac{5}{4}x^2 + \frac{53}{40}x, \\
R(x) &= x^8 - 2x^7 + 2x^6 - x^5 - x^3 + 2x^2 - 2x + 1, \\
S(x) &= \frac{1}{12096}[17671x^7 - 23622x^6 + 61449x^5 - 50516x^4 \\
&\qquad\qquad + 61449x^3 - 23622x^2 + 17671x].
\end{aligned}
$$

The first one was suggested in [6] and the second in [19]. They both make local error small. We study the efficiency of these methods with relation to Gauss integrators of 2 and 4 stages (denoted as VG4 and VG8, respectively) implemented in an adaptive way. We have chosen these methods for the comparison because VG4 was suggested as the most efficient for long-term integration for the Kepler problem among 4th-order integrators in [3] and among some 8th-order integrators in [21]. However, we do not state that they are the optimal ones among one-step methods. This comparison is just a first step in proving the efficiency of the methods suggested in the paper.

In order to reduce roundoff errors, both types of methods have been implemented by using the compensated summation technique [16] in the computation of the components of the solution. Also to this end, our methods have been performed with the stabilization process [11]. This consists of introducing the sequence $\{v_n\}$ such that if y_n is the numerical solution at level n, then

$$y_{n+1} = y_n + h_n v_n,$$

and then solving the resulting difference equation for the v_n.

As far as the Gauss methods are concerned, we have used fixed-point iteration to solve the corresponding algebraic equations [11]. We have tried different starting algorithms for the fixed-point iteration and elected the most efficient for the problems and the order of the methods considered. Specifically, for VG4 we have used the technique of Calvo [4] and, for VG8, the starting algorithm due to Laburta [18], with two additional evaluations per step. On the other hand, since the test problems considered are Hamiltonian systems, the variable-step size Gauss methods have been implemented by introducing a time transformation and integrating a reparameterized system with fixed stepsize. The use of Poincaré transformations avoids that the transformed system fails to be Hamiltonian, and requires the differentiability of the stepsize function [10].

Kepler's problem. The first test problem considered is Kepler's problem

$$\ddot{q}_i = -\frac{q_i}{(q_1^2+q_2^2)^{3/2}}, \qquad i=1,2,$$

with Hamiltonian function

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

and initial conditions $q_1(0)=1-e$, $q_2(0)=0$, $p_1(0)=0$, $p_2(0)=\sqrt{(1+e)/(1-e)}$, $0<e<1$. The solution describes the motion in a plane of a point attracted to the origin with a force inversely proportional to the distance squared. The orbit is 2π-periodic and consists of an ellipse with eccentricity e. In all the numerical experiments we have taken $e=0.9$, a severe test for which variable-stepsize methods are necessary due to the variability in the solution [1]. This problem is reversible with respect to the involution

$$\bar{\Lambda}(p_1,p_2,q_1,q_2)^T = (-p_1,p_2,q_1,-q_2)^T. \tag{3.2}$$

The stepsize function elected for both types of integrators is similar. In the case of the multistep methods, the equation (1.6) is solved iteratively with a given relative tolerance, while for the variable-stepsize Gauss methods, the stepsize function is incorporated in the system, as explained above. The function τ has been chosen as

$$\tau(q_1,q_2) = \frac{\pi}{2\sqrt{2}}(q_1^2+q_2^2)^{3/4},$$

a suitable selection already suggested in other papers [6], [7] among other possible selections. Notice that it satisfies (1.7) for $\bar{\Lambda}$ in (3.2).

Error growth with time. The first experiment we show concerns the behaviour in time of the error for the VSLMM2s considered in this paper. It is a way to check experimentally that the method obtained is symmetric. As was proved in [6], for this problem the growth with time of the first three terms of the asymptotic expansion of the global error provided by symmetric VSLMM2s with no parasitic root is linear, in contrast with the quadratic error growth shown by general methods. This is confirmed by Figure 3.1. In the left plot we have measured the error at final times $10T, 30T, \ldots, 21870T$ ($T=2\pi$ is the period of the problem) given by VSLMM2-4. The scale is logarithmic. Each line corresponds to a different tolerance ϵ; we have taken $\epsilon = 2\pi\times 10^{-3}, \pi\times 10^{-3}, \frac{\pi}{2}\times 10^{-3}$. The slopes of the lines show the linear growth with time of the error (see the line plotted in the lower right-hand corner of the picture). When the final time is multiplied by 3, the errors are multiplied by the

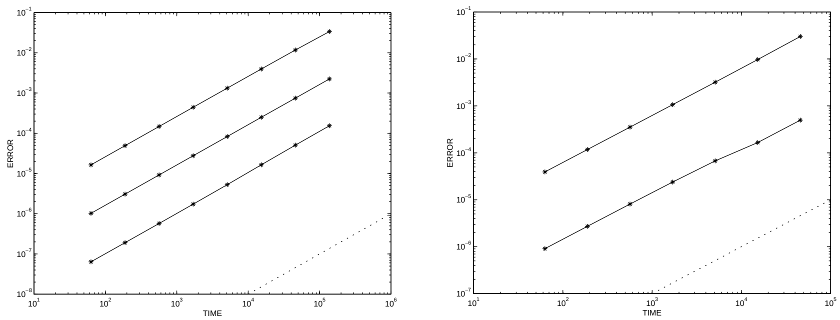

FIGURE 3.1. Error growth with VSLMM2-4 and VSLMM2-8

same number. On the other hand, by watching the distance between parallel lines, the order four of the method is noticed: errors corresponding to a same final time are divided by approximately 16 when the stepsize is halved. The same conclusion is reached from the right plot, which displays the growth with time of the error corresponding to VSLMM2-8. The final times are $10T, 30T, \ldots, 7290T$ and the tolerance $\epsilon = 2\pi/250, 2\pi/500$. Linear error growth is observed just till that time because from there on the fourth term of the asymptotic expansion, which does not grow linearly, becomes dominant. Notice that the values of ϵ considered above for the 4th-order method are smaller, which makes those quadratic terms become dominant in a longer time. However, if we had done our experiments in quadruple precision and considered smaller values of ϵ, we would have observed linear error growth over longer times also for the 8th-order methods.

Comparison of efficiency. Now we focus on the comparison in efficiency between VSLMM2-4, VSLMM2-8 and the corresponding variable-stepsize Gauss integrators. For the last integrators, we have taken a tolerance in the fixed-point iteration for which the error of the corresponding integrator grows linearly until the time in which we observe linear error growth with the corresponding VSLMM2. Thus, in order to compare the efficiency of the methods, it is sufficient to measure error and cost after one period of time.

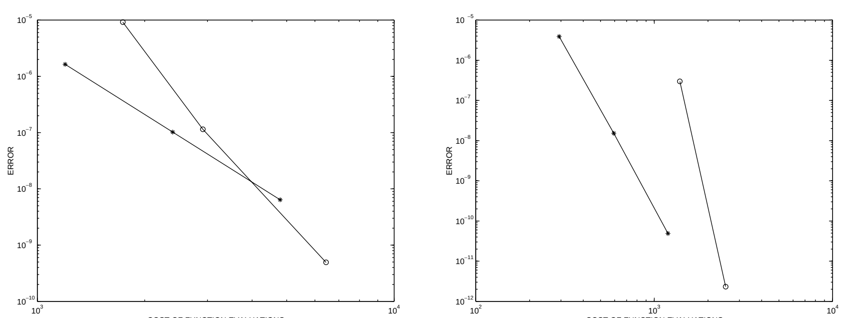

FIGURE 3.2. Error against function cost (* for VSLMM2-4(8); $\circ$ for VG4(8))

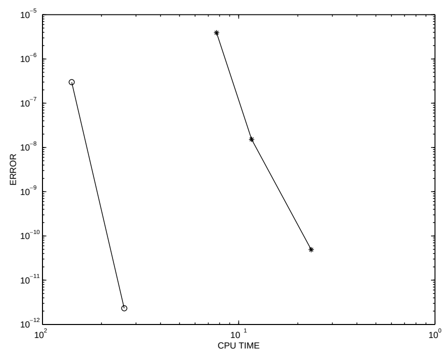

FIGURE 3.3. Error against CPU time (* for VSLMM2-8; ∘ for VG8)

Figure 3.2 compares the error (after one period) against the cost of function evaluations for methods of order 4 and 8, respectively. For Gaussian schemes, the system is written in first-order form and the formulas for the derivatives are more complicated that in the original case because of the introduction of a reparameterization and the evaluation of the Hamiltonian. In such a way, the cost of each function evaluation is at least double that in the original case. In our figures, we have considered it exactly double. Asterisks correspond to VSLMM2-4(8) (with tolerances $\epsilon = 2\pi/250, 2\pi/500, 2\pi/1000$) and circles correspond to VG4(8) (with $\epsilon = 0.2, 0.1, 0.05$ for VG4 and $\epsilon = 0.8, 0.4$ for VG8). Observe that, for errors greater than approximately 10^{-8}, VSLMM2-4 is less expensive, changing the situation for smaller errors. For a fixed cost greater than 4000, VG4 is more accurate. A different situation is presented when we compare the efficiency of the methods of order eight. The cheaper cost in function evaluations of VSLMM2-8 is manifest for the reasonable range of errors in double precision. By increasing the order of the methods, we render evident the improvement in efficiency in terms of function evaluation cost when we use multistep methods. We remind the reader again that the number of function evaluations per step does not grow when increasing the order of the multistep method while it necessarily increases for the one-step methods.

However, the advantage of VSLMM2-8 as far as evaluation cost is concerned is lost when we study the efficiency in terms of computational time for Kepler's problem. Figure 3.3 gives the error against CPU time for each method of order eight and shows the more efficient behaviour of VG8. Something similar happens with the 4th-order integrators. This is due to the fact that the calculus of the coefficients in the VSLMM2 is quite expensive compared with the operations needed by Gauss methods which do not come from the function evaluations.

Five-body problem. The better behaviour of VG8 in computational time observed in Figure 3.3 is associated to the simplicity of the equations of Kepler's problem. It is necessary to state the comparison when we integrate more complicated problems, increasing the number of variables or the complexity to evaluate the right-hand side of the equations involved.

To this end, we consider the five-body problem that describes the motion of five outer planets about the sun. If we denote by y_{1j}, y_{2j}, y_{3j}, $j = 1, \ldots, 5$, the coordinates of the jth body, then each of the 15 coordinates satisfies a second-order

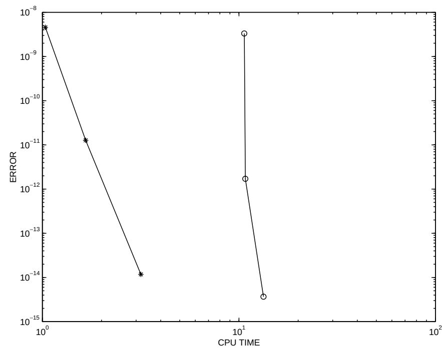

FIGURE 3.4. Error in energy against CPU time (* for VSLMM2-8; $\circ$ for VG8)

differential equation [14]

$$\ddot{y}_{ij} = g\left(-(m_0+m_j)\frac{y_{ij}}{r_{j^3}} + \sum_{k=1,k\neq j}^{5} m_k\left(\frac{y_{ik}-y_{ij}}{d_{jk}^3} - \frac{y_{ik}}{r_k^3}\right)\right), \tag{3.3}$$

where

$$r_j^2 = \sum_{i=1}^{3} y_{ij}^2, \quad d_{jk}^2 = \sum_{i=1}^{3}(y_{ij}-y_{ik})^2, \quad j,k = 1,\dots,5,$$

g is the gravitational constant, m_0 the mass of the sun and m_j the mass of the jth planet. We consider equations (3.3) to implement the multistep methods but they have to be rewritten in Hamiltonian form in order to be integrated by the Gauss schemes, leading to a system of 30 components.

We have measured the error in the Hamiltonian of the problem at time $t_f = 1000$ and represented this error against CPU time in Figure 3.4. Asterisks correspond to VSLMM2-8 and circles to VG8. The stepsize function considered for both methods is

$$\tau = \sum_{j=1}^{5} r_j^{3/4} + \sum_{i<j} d_{ij}^{3/4},$$

and we have chosen $\epsilon = 10^{-3}, 5\times 10^{-4}, 2.5\times 10^{-4}$ for VSLMM2-8 and $\epsilon = 4\times 10^{-3}, 2\times 10^{-3}, 10^{-3}$ for VG8. Note that more complexity of the problem translates to a better behaviour of the multistep method in computational time. This reveals the interest in the use of the methods proposed for moderately large problems.

4. Concluding remarks

We remark that, in principle, symplectic and symmetric one-step methods lead to slow error growth for longer times than symmetric VSLMM2s with exact starting values. However, for the latter, the range of times for which the same slope for the errors is observed is very large, and therefore they are still very useful. Notice also that making the tolerance smaller means getting a wider range in time for slow error growth whenever the precision of the machine is high enough. At the same time, for the numerical comparisons made in section 3, we took a tolerance for the fixed-point iteration in Gauss methods which just led to the same range of slow error growth as the compared VSLMM2s. If the tolerance for that fixed-point iteration had been smaller, the linear error growth would have been observed further, but also at a higher computational cost.

In the paper, we have considered 4th- and 8th-order methods for the comparisons, but higher order methods could also have been constructed with the same technique. We emphasize again that the higher the order the more the advantages of VSLMM2s over one-step integrators. In such a way, these methods are very much recommended to solve some problems in astronomy for which high accuracy and long-term integration is required.

References

[1] Arnold, V. I., *Mathematical Methods of Classical Mechanics*, 2nd. Ed., Springer, New York, 1989. MR **96c:**70001

[2] Calvo, M. P. and Sanz-Serna, J. M., *The development of variable-step symplectic integrators, with application to the two-body problem*, SIAM J. Sci. Comput., 14 (1993), pp. 936–952. MR **94g:**65068

[3] Calvo, M. P., López-Marcos, M. A. and Sanz-Serna, J. M., *Variable step implementation of geometric integrators.* Appl. Num. Math., 28 (1998), pp. 1–16. MR **99d:**65236

[4] Calvo, M. P., *High order initial interants for implicit Runge-Kutta methods: an improvement for variable-step symplectic integrators*, IMA J. Num. Anal. 22 (2002), pp. 153–166.

[5] Cano, B., *Integración numérica de órbitas periódicas con métodos multipaso*, PhD Thesis, Universidad de Valladolid, 1996.

[6] Cano, B. and Durán, A., *Analysis of variable-stepsize linear multistep methods with special emphasis on symmetric ones*, Math. Comp., posted on May 29, 2003, PII S 0025-5718(03)01538-2 (to appear in print).

[7] Cano, B. and Sanz-Serna, J.M., *Error growth in the numerical integration of periodic orbits, with application to Hamiltonian and reversible systems,* SIAM J. Num. Anal., 34 (1997), pp. 1391–1417. MR **98i:**65052

[8] Cano, B. and Sanz-Serna, J.M., *Error growth in the numerical integration of periodic orbits by multistep methods, with application to reversible systems*, IMA J. Num. Anal., 18 (1998), pp. 57–75. MR **99d:**65237

[9] Evans, N. W. and Tremaine, S., *Linear multistep methods for integrating reversible differential equations*, Astron. J **118** 1888 (1999).

[10] Hairer, E., *Variable time step integration with symplectic methods*, Appl. Numer. Math., 25 (1997), pp. 219–227. MR **99a:**65083

[11] Hairer, E., Nörsett, S. P. and Wanner, W. G., *Solving Ordinary Differential Equations I. Nonstiff Problems*, 2nd ed., Berlin, Springer, 1993. MR **94c:**65005

[12] Hairer, E. and Stoffer, D., *Reversible long-term integration with variable stepsizes*, SIAM J. Sci. Comput., 18 (1997), pp. 257–269. MR **97m:**65118

[13] Huang, W. and Leimkuhler, B. *The Adaptive Verlet method*, SIAM J. Sci. Comput. **18** (1997), pp. 239–256. MR **98g:**65063

[14] Hull, T. E., Enright, W. H. Fellen, B. M. and Sedgwick, A. E., *Comparing numerical methods for ordinary differential equations*, SIAM J. Numer. Anal., 9 (1972), pp. 603–637. MR **50:**3577

[15] Hut, P., Makino, J. and McMillan, S., *Building a better leapfrog*, Astrophys. J., 443 (1995), pp. L93–L96.

[16] Kahan, W., *Unconventional numerical methods for trajectory calculations*, Mathematics Dept., and Elect. Eng. & Computer Science Dept., University of California, 1993.

[17] Leimkuhler, B., *Reversible adaptive regularization I: Perturbed Kepler motion and classical atomic trajectories.* R. Soc. Lond. Philos, Trans. Ser. A Math. Phys. Eng. Sci., 357 (1999), pp. 1101–1133. MR **2000b:**70002

[18] Laburta, M. P., *Construction of starting algorithms for the RK-Gauss methods*, J. Comp. Appl. Math., 90 (1998), pp. 239–261. MR **99d:**65215

[19] Quinlan, G. D. and Tremaine, S., *Symmetric multistep methods for the numerical integration of planetary orbits*, Astron. J., 100 (1990), pp. 1694–1700.

[20] Reich, S., *Backward error analysis of numerical integrators*, SIAM J. Numer. Anal. **36** (1999), pp. 1549–1570. MR **2000f:**65060

[21] Sanz-Serna, J. M and Calvo, M. P., *Numerical Hamiltonian problems*, Chapman & Hall, London, 1994. MR **95f:**65006

Departamento de Matemática Aplicada y Computación, Facultad de Ciencias, Universidad de Valladolid, Valladolid, Spain
E-mail address: bego@mac.uva.es

Departamento de Matemática Aplicada y Computación, Facultad de Ciencias. Universidad de Valladolid, Valladolid, Spain
E-mail address: angel@mac.uva.es