

## COMPOSITION METHODS FOR DIFFERENTIAL EQUATIONS WITH PROCESSING\*

S. BLANES<sup>†</sup>, F. CASAS<sup>‡</sup>, AND A. MURUA<sup>§</sup>

**Abstract.** We construct numerical integrators for differential equations up to order 12 obtained by composition of basic integrators. The following cases are considered: (i) composition for a system separable in two solvable parts, (ii) composition using as basic methods a first-order integrator and its adjoint, (iii) composition using second-order symmetric methods, and (iv) composition using fourth-order symmetric methods. Each scheme is implemented with a processor or corrector to improve their efficiency, and this can be done virtually cost-free.

**Key words.** composition methods, geometric integrators, processing technique, cheap postprocessor, initial value problems

**AMS subject classifications.** 65L05, 65L70, 65D30

**DOI.** 10.1137/030601223

**1. Introduction.** A widely used technique in the geometric integration of ordinary differential equations (ODEs) is to compose one or more low-order basic one-step methods (usually first or second order) with appropriately chosen weights to achieve a higher-order scheme. The resulting composition method then inherits the relevant geometric properties that the basic scheme shares with the exact solution (if these properties are preserved by composition). In close connection with composition is the splitting idea: if the differential equation can be split into two or more parts that are either solvable or simpler to integrate than the original system, it is also possible to build integrators by composition of the corresponding flows. These splitting methods are frequently used in celestial mechanics, quantum mechanics, molecular dynamics, and accelerator physics and, in general, for solving numerically Hamiltonian systems, as well as Poisson systems and reversible differential equations [8, 16, 21].

It has been widely recognized that the class of numerical integrators which preserve the geometric properties of the exact flow (the so-called *geometric integrators*) provide a better description of the system than standard methods, even with respect to the accumulation of numerical errors along the evolution, making them the best option for carrying out a long time integration. However, for short time integrations they are typically less efficient. For this reason it is important to build more efficient high-order geometric integrators which are also competitive in this setting.

---

\*Received by the editors October 16, 2003; accepted for publication (in revised form) February 18, 2005; published electronically February 3, 2006. The work of the first and second authors was partially supported by Fundació Caixa Castelló–Bancaixa through project P1.1B2002-22 and by Generalitat Valenciana through project GRUPOS03-002.

<http://www.siam.org/journals/sisc/27-6/60122.html>

<sup>†</sup>Instituto de Matemática Multidisciplinar, Universidad Politécnica de Valencia, 46022 Valencia, Spain (serblaza@imm.upv.es). This author's work was supported by Ministerio de Ciencia y Tecnología (Spain) through a contract in the Programme Ramón y Cajal 2001.

<sup>‡</sup>Departament de Matemàtiques, Universitat Jaume I, 12071 Castellón, Spain (Fernando.Casas@uji.es).

<sup>§</sup>Konputazio Zientziak eta A. A. saila, Informatika Fakultatea, EHU/UPV, Donostia/San Sebastián, Spain (ander@si.ehu.es). This author's work was supported by Ministerio de Educación y Ciencia under project MTM2004-00535, cofinanced by the European Regional Development Fund (ERDF).

Given the ODE

$$(1.1) \quad x' = f(x), \quad x_0 = x(0) \in \mathbb{R}^D,$$

with  $f : \mathbb{R}^D \rightarrow \mathbb{R}^D$  and associated vector field (or Lie operator associated with  $f$ )

$$(1.2) \quad F = \sum_{i=1}^D f_i(x) \frac{\partial}{\partial x_i},$$

a one-step numerical integrator for a time step  $h$ ,  $\psi_h : \mathbb{R}^D \rightarrow \mathbb{R}^D$ , can be regarded as a smooth family of maps with parameter  $h$  such that  $\psi_0$  is the identity map. The integrator  $\psi_h$  is said to be of order  $q$  if  $\psi_h = \varphi_h + \mathcal{O}(h^{q+1})$  as  $h \rightarrow 0$ , where  $\varphi_h$  is the  $h$ -flow of the ODE (1.1). In other words, an approximation to the exact solution  $x(h)$  is given by  $x_h = \psi_h(x_0) = \varphi_h(x_0) + \mathcal{O}(h^{q+1})$ .

In this work we consider methods obtained by enhancing an integrator  $\psi_h$  with processing, i.e., algorithms of the form

$$(1.3) \quad \hat{\psi}_h = \pi_h \circ \psi_h \circ \pi_h^{-1}.$$

Here  $\psi_h$  is referred to as the *kernel* and the map  $\pi_h : \mathbb{R}^D \rightarrow \mathbb{R}^D$  as the *postprocessor* or *corrector*. The method  $\psi_h$  is of *effective order*  $q$  if a postprocessor  $\pi_h$  exists for which  $\hat{\psi}_h$  is of (conventional) order  $q$  [6]. Application of  $\hat{\psi}_h$  over  $n$  steps with constant step size  $h$  leads to

$$\hat{\psi}_h^n = \pi_h \circ \psi_h^n \circ \pi_h^{-1}.$$

The computation of the preprocessor  $\pi_h^{-1}$  has to be done only at the beginning of the integration; then the kernel  $\psi_h$  acts once per step, and finally  $\pi_h$  is evaluated when output is required.

The analysis of the order conditions of the method  $\hat{\psi}_h$  has shown that many of them can be satisfied by  $\pi_h$ , so that  $\psi_h$  must fulfill a much reduced set of restrictions [1, 3, 7, 14]. This allows us to take kernels of effective order  $q$  involving far fewer function evaluations than a conventional integrator of order  $q$ .

In the course of constructing composition methods of a given order, the processing technique also presents additional advantages. In general, it is necessary to solve numerically a system of nonlinear polynomial equations in the coefficients (the order conditions), whose number and complexity grow very rapidly with the order. Numerically finding every solution and eventually determining the optimal solution turns into a very complicated task for nonprocessed methods, even for relatively small orders. However, this problem simplifies for processed composition schemes because the analysis has to be carried out only for the kernel. Since the number of order conditions for the kernel is smaller, one can analyze in more detail the set of solutions, even for moderate orders, and eventually find very efficient methods [25, 3, 1, 15].

Once a kernel is chosen, it remains for us to build a postprocessor such that the whole method has the required order. We can also impose additional conditions to reduce the leading order error terms. Then the coefficients of the postprocessor  $\pi_h$  usually have to satisfy a large number of conditions. Many different solutions can be found, and since usually their different contributions affect only higher-order terms, we can choose one with moderately small coefficients (to avoid large higher-order error terms), which considerably simplifies the numerical search.

In any case, very often the postprocessors thus constructed are more expensive than the kernel  $\psi_h$ , so that the use of processed schemes in this form has to be confined to problems in which intermediate results are not frequently required; otherwise the overall efficiency of the methods is decreased. To overcome this difficulty, a technique is developed in [2] which allows us to obtain approximations to the postprocessor virtually cost-free and without loss of accuracy. The key point is to replace  $\pi_h$  by a new integrator  $\hat{\pi}_h \simeq \pi_h$  obtained from the intermediate stages in the computation of  $\psi_h$ . It is recommended that one have a very accurate preprocessor  $\pi_h^{-1}$  but, on the other hand, the postprocessor can safely be replaced by a cheap approximation satisfying only the order conditions.

In the analysis of composition methods it is useful to apply the notion of adjoint of a given integrator  $\psi_h$  [21]. By definition, this is the method  $\psi_h^*$  such that  $\psi_h^* = \psi_{-h}^{-1}$ . A method that is its own adjoint is called self-adjoint or (time-)symmetric. In this case,  $\psi_{-h} \circ \psi_h = \text{id}$ . The order of a symmetric method is necessarily even. In addition, only even powers of  $h$  appear in the asymptotic expansion of the global error [21] so that the order equations are considerably simplified and its number is reduced. Time-symmetry is often a desirable property when one aims to discretize the differential equation in such a way that qualitative features of the system are preserved [8, 16]. In the context of processed methods, if the kernel  $\psi_h$  is symmetric and the postprocessor  $\pi_h$  is an even function of  $h$  ( $\pi_{-h} = \pi_h$ ), then the processed integrator  $\hat{\psi}_h$  has also the time-symmetry property.

In this paper we construct and analyze new high-order processed geometric integrators. We consider several families of kernels which are symmetric compositions of different basic methods of first, second, and fourth order and whose postprocessors can be implemented essentially at no additional computational cost. The procedure is largely based on the theoretical analysis of the processing technique done in [2], to which we will refer often in what follows.

## 2. Families of composition methods.

**2.1. Classes of methods.** Next we present the different classes of schemes analyzed, whose common feature is that they are defined as the composition of simpler maps.

1. Let us assume that the ODE (1.1) can be written as  $x' = f_a(x) + f_b(x)$  and that the vector field  $F$  is split accordingly as  $F = F_a + F_b$ . Suppose that the corresponding  $h$ -flows  $\varphi_h^{[a]}$  and  $\varphi_h^{[b]}$  can be computed exactly. Then one can form the following group  $\mathcal{G}_1$  of schemes:

$$(2.1) \quad \mathcal{G}_1 = \left\{ \varphi_{\beta_{2s}h}^{[b]} \circ \varphi_{\beta_{2s-1}h}^{[a]} \circ \cdots \circ \varphi_{\beta_2h}^{[b]} \circ \varphi_{\beta_1h}^{[a]} : s \geq 1, (\beta_1, \dots, \beta_{2s}) \in \mathbb{R}^{2s} \right\}.$$

2. Let  $\chi_h : \mathbb{R}^D \rightarrow \mathbb{R}^D$  be any first-order integration scheme. An important class  $\mathcal{G}_2$  of integrators is obtained by considering compositions of  $\chi_h$  and its adjoint  $\chi_h^*$ , namely,

$$(2.2) \quad \mathcal{G}_2 = \left\{ \chi_{\alpha_{2s}h} \circ \chi_{\alpha_{2s-1}h}^* \circ \cdots \circ \chi_{\alpha_2h} \circ \chi_{\alpha_1h}^* : s \geq 1, (\alpha_1, \dots, \alpha_{2s}) \in \mathbb{R}^{2s} \right\}.$$

This family  $\mathcal{G}_2$  of integrators clearly has (with the composition) a group structure. As shown in [13], if  $\sum_{i=1}^s \beta_{2i-1} = \sum_{i=1}^s \beta_{2i}$ ,  $\mathcal{G}_2$  is closely related to  $\mathcal{G}_1$

when  $\chi_h = \varphi_h^{[a]} \circ \varphi_h^{[b]}$ ,  $\beta_i = \alpha_i + \alpha_{i-1}$  ( $i = 1, \dots, 2s$ ), and  $\alpha_0 = \alpha_{2s} = 0$ . Consistency of the schemes requires that  $\sum_{i=1}^s \beta_{2i-1} = \sum_{i=1}^s \beta_{2i} = \sum_{i=1}^{2s} \alpha_i = 1$ , and the order conditions at higher orders are equivalent in both cases [13]. Methods in  $\mathcal{G}_2$  are more general and can also be used if the system is split into more than two parts, i.e.,  $x' = f_a + f_b + f_c + \dots$ , just taking  $\chi_h = \varphi_h^{[a]} \circ \varphi_h^{[b]} \circ \varphi_h^{[c]} \circ \dots$  ( $\chi_h^* = \dots \circ \varphi_h^{[c]} \circ \varphi_h^{[b]} \circ \varphi_h^{[a]}$ ).

3. Another well-known class  $\mathcal{G}_3$  of integrators [13] is introduced as follows. If  $\mathcal{S}_h^{[2]} : \mathbb{R}^D \rightarrow \mathbb{R}^D$  is any second-order self-adjoint integrator, then

$$(2.3) \quad \mathcal{G}_3 = \left\{ \mathcal{S}_{\alpha_s h}^{[2]} \circ \dots \circ \mathcal{S}_{\alpha_1 h}^{[2]} : s \geq 1, (\alpha_1, \dots, \alpha_s) \in \mathbb{R}^s \right\}.$$

This family of integrators also has group structure. In fact, if  $\mathcal{S}_h^{[2]}$  is chosen as  $\mathcal{S}_h^{[2]} = \chi_{h/2} \circ \chi_{h/2}^*$ , then  $\mathcal{G}_3$  is a subgroup of  $\mathcal{G}_2$  (in [18], it is shown that any self-adjoint integrator can be written in this form; the choice of  $\chi_h$  is not unique, however).

4. Finally, if  $\mathcal{S}_h^{[4]} : \mathbb{R}^D \rightarrow \mathbb{R}^D$  is any fourth-order self-adjoint integrator, then another class  $\mathcal{G}_4$  of integrators is

$$(2.4) \quad \mathcal{G}_4 = \left\{ \mathcal{S}_{\alpha_s h}^{[4]} \circ \dots \circ \mathcal{S}_{\alpha_1 h}^{[4]} : s \geq 1, (\alpha_1, \dots, \alpha_s) \in \mathbb{R}^s \right\}.$$

Methods in  $\mathcal{G}_4$  could be compositions of schemes in  $\mathcal{G}_l$ ,  $l = 1, 2, 3$ . They can be useful in some problems where, due to their particular structure, it is possible to build a very efficient integrator  $\mathcal{S}_h^{[4]}$ , as is the case for the second-order differential equation  $x'' = g(x)$  [11].

Composition integrators  $\psi_h \in \mathcal{G}_l$  ( $l = 1, 2, 3, 4$ ) are time-symmetric whenever they have left-right palindromic sequences of coefficients  $\alpha_i$ —more specifically, if the following apply:

1. In  $\mathcal{G}_1$ , if either  $\beta_{2s} = 0$ ,  $\beta_{2s-i} = \beta_i$  or  $\beta_1 = 0$ ,  $\beta_{2s+1-i} = \beta_{i+1}$ . Then one of the maps  $\varphi_{\beta_i h}^{[a]}$  and  $\varphi_{\beta_i h}^{[b]}$  is computed  $s$  times and the other  $s - 1$  times. Now the computations done for the last map can be reused for the first map in the following step. (This feature is known as the FSAL (first same as last) property.) Then both maps  $\varphi_{\beta_i h}^{[a]}$  and  $\varphi_{\beta_i h}^{[b]}$  are computed  $s - 1$  times.
2. In  $\mathcal{G}_l$ ,  $l = 2, 3, 4$ , if  $\alpha_{\bar{s}+1-i} = \alpha_i$  with  $\bar{s} = 2s$  or  $\bar{s} = s$ .

**2.2. Associated graded Lie algebra.** It is well known that for each infinitely differentiable map  $g : \mathbb{R}^D \rightarrow \mathbb{R}$ ,  $g(\varphi_h(x))$  admits an expansion of the form [19]

$$g(\varphi_h(x)) = e^{hF}[g](x) = g(x) + \sum_{k \geq 1} \frac{h^k}{k!} F^k[g](x), \quad x \in \mathbb{R}^D,$$

where  $F$  is the vector field (1.2). Similarly, for the maps  $\hat{\psi}_h$ ,  $\psi_h$ , and  $\pi_h$  one has

$$g(\hat{\psi}_h(x)) = e^{\hat{F}_h}[g](x), \quad g(\psi_h(x)) = e^{F_h}[g](x), \quad g(\pi_h(x)) = e^{P_h}[g](x),$$

with

$$\hat{F}_h = \sum_{k \geq 1} h^k \hat{F}_k, \quad F_h = \sum_{k \geq 1} h^k F_k, \quad P_h = \sum_{k \geq 1} h^k P_k;$$

i.e.,  $\hat{\psi}_h$ ,  $\psi_h$ , and  $\pi_h$  are formally the exact 1-flows of the vector fields  $\hat{F}_h$ ,  $F_h$ , and  $P_h$ , respectively.

If the kernel  $\psi_h$  belongs to  $\mathcal{G}_l$  ( $l = 1, 2, 3, 4$ ), then the corresponding ( $h$ -dependent) vector fields  $h^k F_k$  belong to a certain graded Lie algebra  $\mathcal{L} = \bigoplus_{k \geq 1} \mathcal{L}_k$  such that  $hF \in \mathcal{L}_1$ ,  $h^k F_k \in \mathcal{L}_k$  for each  $k \geq 1$  and  $[\mathcal{L}_k, \mathcal{L}_m] \subset \mathcal{L}_{k+m}$ . Each  $\mathcal{L}_k$  ( $k \geq 1$ ) is the subspace  $\mathcal{L}$  of vector fields affected by a  $k$ th power of  $h$ , and we denote  $n_k = \dim \mathcal{L}_k$ . We give  $\mathcal{L}$  explicitly in each case.

1. If  $\psi_h \in \mathcal{G}_1$ , then  $\mathcal{L} = \bigoplus_{k \geq 1} \mathcal{L}_k$  is the graded Lie algebra generated by the vector fields  $\{hF_a, hF_b\}$ . Here,  $hF_1 + h^2F_2 + \dots$  can be obtained, for instance, by repeated application of the Baker–Campbell–Hausdorff (BCH) formula [24] to the expression

$$\Psi_h = \exp(h\beta_1 F_a) \exp(h\beta_2 F_b) \cdots \exp(h\beta_{2s-1} F_a) \exp(h\beta_{2s} F_b)$$

since, formally,  $g \circ \psi_h = \Psi_h[g]$  and  $F_h = \log(\Psi_h)$  [2, 8].

2. For  $\mathcal{G}_2$ , we have for the basic method  $g(\chi_h(x)) = e^{Y_h}[g](x)$  with  $Y_h = \sum_{k \geq 1} h^k Y_k$  and for its adjoint  $g(\chi_h^*(x)) = e^{-Y_{-h}}[g](x)$ . Then

$$\Psi_h = \exp(-Y_{-h\alpha_1}) \exp(Y_{h\alpha_2}) \cdots \exp(-Y_{-h\alpha_{2s-1}}) \exp(Y_{h\alpha_{2s}}),$$

and  $F_h$  can be obtained by repeated application of the BCH formula, which shows that  $h^k F_k \in \mathcal{L}_k$  for each  $k \geq 1$  and  $\mathcal{L} = \bigoplus_{k \geq 1} \mathcal{L}_k$  is the graded Lie algebra generated by the vector fields  $\{hY_1, h^2Y_2, h^3Y_3, \dots\}$  where, by consistency,  $Y_1 = F$ .

3. For  $\mathcal{G}_3$ , the series of differential operators  $S_h$  associated with the integrator  $\mathcal{S}_h^{[2]}$ , i.e., such that  $g \circ \mathcal{S}_h^{[2]} = S_h[g]$ , can be written as  $S_h = \exp(Y_h)$ , where  $Y_h = hY_1 + h^3Y_3 + h^5Y_5 + \dots$ ,  $Y_1 = F$ , and

$$\Psi_h = \exp(Y_{h\alpha_1}) \cdots \exp(Y_{h\alpha_s}).$$

One similarly [13] arrives at an expansion  $hF_1 + h^3F_3 + h^4F_4 + \dots$ , where  $h^k F_k \in \mathcal{L}_k$  for the graded Lie algebra  $\mathcal{L} = \bigoplus_{k \geq 1} \mathcal{L}_k$  generated by the vector fields  $\{hY_1, h^3Y_3, h^5Y_5, \dots\}$ .

4. The case of  $\mathcal{G}_4$  can be considered as a particular case of  $\mathcal{G}_3$  with  $Y_3 = 0$ .

**2.3. Effective order conditions.** According to the theory developed in [2] the following theorem applies to the processed methods considered here.

**THEOREM 1** (see [2]). *An integrator  $\psi_h$  has effective order  $p \geq q$  if and only if there exist vector fields  $P_1, \dots, P_{q-1}$  such that*

$$(2.5) \quad F_1 = F, \quad [P_{n-1}, F] = F_n + R_n(P_1, \dots, P_{n-2}, F_1, \dots, F_{n-1}), \quad 1 < n \leq q,$$

hold, where

$$R_n = - \sum_{j=1}^{n-2} [P_j, F_{n-j}] + \sum_{k \geq 2} \frac{(-1)^k}{k!} \sum_{j_1 + \dots + j_{k+1} = n} [P_{j_1}, [P_{j_2}, \dots [P_{j_k}, F_{j_{k+1}}] \dots]].$$

Under the premises of Theorem 1,  $[P_k, F] \in \mathcal{L}_{k+1}$ . As shown in [2], if such vector fields  $P_k$  ( $1 \leq k \leq q - 1$ ) exist, then they are not unique. We will hereafter consider vector fields  $P_k$  belonging to  $\mathcal{L}_k$ . Uniqueness is then achieved by requiring that  $P_k \in \mathcal{L}_k^*$ , where  $\mathcal{L}_k^*$  is a subspace of  $\mathcal{L}_k$  of minimal dimension,  $n_k^*$ , among all

subspaces  $V_k$  of  $\mathcal{L}_k$  such that  $[F, V_k] = [F, \mathcal{L}_k]$ . It must be stressed that such a requirement does not restrict the choice of the processed integrator [2]. Since the graded Lie algebras associated with  $\mathcal{G}_l$  ( $l = 1, 2, 3, 4$ ) are free, then  $\mathcal{L}_1 = \mathcal{L}_1^* \oplus F$ ,  $\mathcal{L}_k^* = \mathcal{L}_k$  for  $k > 1$ , and  $P_1 \in \mathcal{L}_1^*$ . Therefore,  $n_1^* = n_1 - 1$  and  $n_k^* = n_k$  for  $k > 1$ .

Let us denote by  $\{E_{k,i}\}_{i=1}^{n_k}$  a basis of  $\mathcal{L}_k$  with  $E_{1,1} = F$ . Then one can write

$$(2.6) \quad F_k = \sum_{i=1}^{n_k} f_{k,i} E_{k,i}, \quad R_k = \sum_{i=1}^{n_k} r_{k,i} E_{k,i}, \quad P_{k-1} = \sum_{i=1}^{n_{k-1}^*} p_{k-1,i} E_{k-1,i},$$

where  $P_0 = 0$ ,  $P_1 = \sum_{i=1}^{n_1} p_{1,i} E_{1,i}$  with  $p_{1,1} = 0$ , and the coefficients  $r_{k,i}$  can be written as polynomials in the indeterminates  $f_{l,j}, p_{l-1,j}$ ,  $l < k$ .

The order conditions (2.5) can be expressed as a system of (polynomial) equations in  $f_{k,i}, p_{k-1,i}$ . Such equations have a particularly simple form if the basis of each  $\mathcal{L}_{k+1}$  is constructed (from a given basis of  $\mathcal{L}_k$ ) in such a way that

$$(2.7) \quad E_{k+1, n_{k+1} - n_k^* + i} = [F, E_{k,i}], \quad 1 \leq i \leq n_k^*, \quad k \geq 1.$$

**THEOREM 2** (see [2]). *Assume that the chosen basis of each  $\mathcal{L}_k$  is such that (2.7) holds. The kernel  $\psi_h$  has effective order  $p \geq q$  if and only if it is consistent ( $F_1 = F$ ) and the vector fields (2.6) in Theorem 1 satisfy the equations*

$$(2.8) \quad f_{k,i} = -r_{k,i}, \quad 1 \leq i \leq l_k := n_k - n_{k-1}^*, \quad 2 \leq k \leq q,$$

$$(2.9) \quad p_{k-1,i} = -f_{k, l_k + i} - r_{k, l_k + i}, \quad 1 \leq i \leq n_{k-1}^*, \quad 2 \leq k \leq q.$$

If  $\psi_h$  is self-adjoint, then for even values of  $k$ , conditions (2.8) are automatically satisfied and equations (2.9) reduce to  $p_{k-1,i} = 0$ .

When the coefficients  $f_{k,i}$  are written in terms of the parameters  $\alpha \equiv (\alpha_1, \dots, \alpha_{\bar{s}})$  ( $\bar{s} = 2s$  or  $\bar{s} = s$ ) of the kernel  $\psi_h$ , repeated substitution of (2.9) into the recursive expressions (2.8)–(2.9) leads to an equivalent system of equations where (2.8) is replaced by a polynomial equation in the parameters  $\alpha_j$ ,  $N_{k,i}(\alpha) = 0$ , and (2.9) is transformed into an expression of the form  $p_{k-1,i} = Q_{k,i}(\alpha)$ , with  $Q_{k,i}$  a polynomial in  $\alpha$ .

The total number of equations  $N_{k,i}(\alpha) = 0$  required to achieve effective order  $q$  depends on the dimensions  $n_k$  in the following way.

**COROLLARY 2.1.** *There exists a system of  $s(q) = n_q + 1$  polynomial equations on the parameters  $\alpha$  that guarantees that the corresponding kernel  $\psi_h \in \mathcal{G}_1$  (resp.,  $\psi_h \in \mathcal{G}_l$ ,  $l = 2, 3, 4$ ) has effective order  $q > 1$  for arbitrary basic vector fields  $\{F_a, F_b\}$  (resp.,  $\{Y_k\}_l$ ). If  $\psi_h$  is self-adjoint, this number reduces to  $\hat{s}(2) = n_1$ ,  $\hat{s}(2n) = n_1 + \sum_{k=2}^n (n_{2k-1} - n_{2k-2})$ ,  $n > 1$ .*

The numbers of effective order conditions given by Corollary 2.1 for each class of integrators are displayed in Table 1 for nonsymmetric,  $s(q)$ , and symmetric,  $\hat{s}(q)$ , kernels.

For a kernel of effective order  $q$  (i.e., satisfying (2.8) for  $k \leq q$  but not for  $k = q+1$ ) one can in principle determine a postprocessor such that equations (2.9) hold also for all  $k > q$ . We refer to that postprocessor as optimal, as it causes many terms of each  $\hat{F}_k = \sum_{i=1}^{n_k} \hat{f}_{k,i} E_{k,i}$  to cancel ( $\hat{f}_{k, n_k - n_{k-1}^* + i} = 0$ ,  $i = 1, \dots, n_{k-1}^*$ ), and denote by  $\mathbb{P}_k$  the set of maps  $\pi_h : \mathbb{R}^D \rightarrow \mathbb{R}^D$  whose Taylor expansion coincides with the optimal postprocessor up to order  $k$  (i.e., their difference is  $\mathcal{O}(h^{k+1})$ ). With this notation, the processed integrator  $\hat{\psi}_h$  in (1.3) is of order  $q$  if the kernel  $\psi_h$  is of effective order  $q$  and the postprocessor  $\pi_h \in \mathbb{P}_{q-1}$ . Note that optimality of a postprocessor depends on the chosen basis in each  $\mathcal{L}_k$  (see [2] for more details).

TABLE 1

Total number of effective order conditions for nonsymmetric ( $s(q)$ ) and symmetric ( $\hat{s}(q)$ ) kernels of effective order  $q$ .

Effective order $q$	$\mathcal{G}_1$		$\mathcal{G}_2$		$\mathcal{G}_3$		$\mathcal{G}_4$	
	$s(q)$	$\hat{s}(q)$	$s(q)$	$\hat{s}(q)$	$s(q)$	$\hat{s}(q)$	$s(q)$	$\hat{s}(q)$
1	2	–	1	–	1	–	1	–
2	2	2	2	1	1	1	1	1
3	3	–	3	–	2	–	1	–
4	4	3	4	2	2	2	1	1
5	7	–	7	–	3	–	2	–
6	10	6	10	5	3	3	2	2
7	19	–	19	–	5	–	3	–
8	31	15	31	14	6	5	3	3
9	57	–	57	–	9	–	4	–
10	100	40	100	39	12	8	4	4
11	187	–	187	–	19	–	6	–
12	335	127	335	126	26	15	7	6

*Remark.* For schemes in  $\mathcal{G}_1$  it is clear that  $\mathcal{L}_1 = \text{span}(E_{1,1} = F = F_a + F_b, E_{1,2} = F_a - F_b)$ ,  $\mathcal{L}_1^* = \text{span}(E_{1,1}^* = E_{1,2})$ , and thus  $n_1 = 2$ ,  $n_1^* = 1$ , whereas in  $\mathcal{G}_2$  one has  $\mathcal{L}_1 = \text{span}(E_{1,1} = F = Y_1)$ ,  $\mathcal{L}_1^* = \emptyset$ , so that  $n_1 = 1$ ,  $n_1^* = 0$ . Moreover,  $n_k$  for  $k \geq 2$  and  $s(k)$  for  $k \geq 3$  are identical in  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , but  $s(1) = s(2) = 2$  in  $\mathcal{G}_1$  and  $s(1) = 1$ ,  $s(2) = 2$  in  $\mathcal{G}_2$ . In consequence, *any first-order method in  $\mathcal{G}_1$  has always effective order 2, but this is not the case in  $\mathcal{G}_2$ .*

For example, let us consider the first-order scheme  $\chi_h = \varphi_h^{[a]} \circ \varphi_h^{[b]}$  belonging to  $\mathcal{G}_1$  and  $\mathcal{G}_2$ . If we take  $\pi_h = \varphi_{h/2}^{[b]} \in \mathcal{G}_1$ , then the processed method  $\pi_h \circ \chi_h \circ \pi_h^{-1}$  is the well-known second-order leapfrog scheme. On the other hand, because a general postprocessor in  $\mathcal{G}_1$ ,  $\exp(P_h) = \exp(h(p_{1,1}F_a + p_{1,2}F_b) + \mathcal{O}(h^2))$ , cannot be written as a composition in  $\mathcal{G}_2$  unless  $p_{1,1} = p_{1,2}$ , the effective order of  $\chi_h$  in  $\mathcal{G}_2$  is only one. However, if the kernel is a second-order method (as is the case of symmetric kernels), then  $p_{1,1} = p_{1,2} = 0$  and any composition postprocessor in  $\mathcal{G}_1$  can be expressed as a composition in  $\mathcal{G}_2$ ; the effective order conditions are equivalent in both cases.  $\square$

**2.4. Cost saving using the processing technique.** For the families of composition methods considered in this work, the number of stages is frequently used as a measure of the computational cost. Here and in the following, the number of basic maps involved in a composition method or kernel will be referred to as the number of stages (i.e.,  $2s$  in (2.1) and (2.2) and  $s$  in (2.3) and (2.4)). In general, the minimum number of stages to obtain a method of order, or effective order,  $q$  is given by the total number of order conditions,  $N_q = \sum_{k=1}^q n_k$ , or effective order conditions,  $s(q) = n_q + 1$ . Then, the use of processed schemes is more advantageous as the value of

$$r_q = \frac{N_q}{s(q)} = \frac{\sum_{k=1}^q n_k}{n_q + 1}$$

increases. Observe that  $r_q$  depends on how fast  $n_k = \dim \mathcal{L}_k$  grows, and thus it is important to analyze its asymptotic behavior.

Let us suppose that  $\{X_1, X_2, \dots\}$  are a countable set of generators of a graded free Lie algebra with grades  $w_i = w(X_i)$  and suppose that the formal sum  $1 - \sum_{i=1}^{\infty} T^{w_i}$  converges to a rational function  $P(T) = q(T)/r(T)$ , as is the case for the Lie algebras previously considered. If the roots of  $q(T)$  and  $r(T)$  are denoted by  $\lambda_1, \dots, \lambda_m$  and

$\gamma_1, \dots, \gamma_{\bar{m}}$ , respectively, then  $n_k$  is given as [9, 16, 17]

$$(2.10) \quad n_k = \frac{1}{k} \sum_{d|k} \mu(d) \left( \sum_{j=1}^m \lambda_j^{-k/d} - \sum_{i=1}^{\bar{m}} \gamma_i^{-k/d} \right),$$

where the first sum ranges over all integers which divide  $k$ , and  $\mu(d)$  is the Möbius function:  $\mu(1) = 1$ ,  $\mu(d) = (-1)^q$  if  $d$  is the product of  $q$  distinct prime factors, and  $\mu(d) = 0$  otherwise. Let  $\rho = 1/\min\{\lambda_1, \dots, \lambda_m\} > 1$  denote the *entropy* of the system, so that  $n_k \sim \rho^k/k$  [16]. By instead assuming  $n_k \sim \rho^k$ , a rough estimation of the value of  $r_q$  is obtained as

$$(2.11) \quad N_q = \sum_{k=1}^q n_k \sim \sum_{k=1}^q \rho^k \sim \frac{\rho^{q+1}}{\rho - 1} \implies r_q \sim \frac{\rho}{\rho - 1},$$

and thus *the cost saving of the processing technique increases when  $\rho$  decreases*.

For  $\mathcal{G}_1$  and  $\mathcal{G}_2$  one has  $P(T) = 1 - 2T$  and  $P(T) = (1 - 2T)/(1 - T)$ , respectively, and  $\rho = 2$ . In  $\mathcal{G}_3$ ,  $P(T) = (1 - T - T^2)/(1 - T^2)$ , so  $\rho = 1.618$ , and in  $\mathcal{G}_4$ ,  $P(T) = (1 - T - T^2 + T^3 - T^5)/(1 - T^2)$ , so  $\rho = 1.443$ . It is worth noticing that the values of  $n_k$  (and thus the number of order conditions) collected in Table 1 can be obtained through the respective functions  $P(T)$  and (2.10).

**2.5. Order conditions for methods in  $\mathcal{G}_3$ .** For the sake of illustration we next consider kernels  $\psi_h \in \mathcal{G}_3$ . Recall that in that case  $\mathcal{L} = \bigoplus_{k \geq 1} \mathcal{L}_k$  is the graded free Lie algebra generated by the symbols  $\{Y_1, Y_3, Y_5, \dots\}$  ( $Y_1 = F$ ). Now  $\mathcal{L}_1^* = \emptyset$ ,  $\mathcal{L}_2 = \emptyset$ , so that  $n_k^* = n_k$  for all  $k$ . We derive the effective order conditions up to order 11 and the corresponding postprocessor coefficients when the basis of  $\mathcal{L}_k$ ,  $k \leq 11$ , is chosen according to (2.7) (see Table 2 for the particular basis taken).

TABLE 2  
Basis of  $\mathcal{L}_k$  ( $k \leq 11$ ) chosen for kernels in  $\mathcal{G}_3$ .

$\mathcal{L}_i$	Basis of $\mathcal{L}_i$	
$\mathcal{L}_1$	$E_{1,1} = Y_1 = F$	
$\mathcal{L}_3$	$E_{3,1} = Y_3$	
$\mathcal{L}_4$	$E_{4,1} = [F, E_{3,1}]$	
$\mathcal{L}_5$	$E_{5,1} = Y_5$	$E_{5,2} = [F, E_{4,1}]$
$\mathcal{L}_6$	$E_{6,i} = [F, E_{5,i}]$	$i = 1, 2$
$\mathcal{L}_7$	$E_{7,1} = Y_7$	$E_{7,2} = [E_{3,1}, E_{4,1}]$
	$E_{7,2+i} = [F, E_{6,i}]$	$i = 1, 2$
$\mathcal{L}_8$	$E_{8,1} = [E_{3,1}, E_{5,1}]$	$E_{8,i+1} = [F, E_{7,i}]$
		$i = 1, \dots, 4$
$\mathcal{L}_9$	$E_{9,1} = Y_9$	$E_{9,i+1} = [E_{3,1}, E_{6,i}]$
		$i = 1, 2$
	$E_{9,3+j} = [F, E_{8,j}]$	$j = 1, \dots, 5$
$\mathcal{L}_{10}$	$E_{10,i} = [E_{3,1}, E_{7,i}]$	$i = 1, \dots, 3$
	$E_{10,3+j} = [F, E_{9,j}]$	$j = 1, \dots, 8$
$\mathcal{L}_{11}$	$E_{11,1} = Y_{11}$	$E_{11,i+1} = [E_{3,1}, E_{8,i}]$
		$i = 1, \dots, 5$
	$E_{11,7} = [E_{5,1}, E_{6,1}]$	$E_{11,7+j} = [F, E_{10,j}]$
		$j = 1, \dots, 11$

As previously stated, if  $\psi_h \in \mathcal{G}_3$  the corresponding series of differential operators can be written as  $\Psi_h = \exp(Y_{h\alpha_1}) \cdots \exp(Y_{h\alpha_s}) \equiv \exp(F_h)$ , with  $F_h = hF_1 + h^3F_3 + h^4F_4 + \dots$ , and each  $f_{k,i}$  in (2.6) is a polynomial in the coefficients  $\alpha_1, \dots, \alpha_s$ . Consistency ( $F_1 = F$ ) is achieved by requiring that  $\sum \alpha_i = 1$ . Since  $n_1 = 1$ ,  $n_1^* = n_2 = 0$ , then  $F_2 = 0$ ,  $P_1 = P_2 = 0$  [2]. If we consider a self-adjoint



kernel, a straightforward computation shows that

$$\begin{aligned}
 R_2 = R_3 = R_4 = R_5 = R_6 = 0, \quad R_7 = f_{3,1}p_{4,1}E_{7,2}, \quad R_8 = 0, \\
 R_9 = f_{5,1}p_{4,1}(E_{9,2} - E_{9,4}) \\
 + \left( f_{5,2}p_{4,1} + \frac{1}{2}p_{4,1}^2 + f_{3,1}p_{6,2} \right) E_{9,3} - p_{4,1} \left( f_{5,2} + \frac{1}{2}p_{4,1} \right) E_{9,6},
 \end{aligned}$$

with similar (but larger) expressions for  $R_{10}, R_{11}$ . Equations (2.8) and (2.9) finally lead, after rearranging and simplifying terms, to the order conditions collected in Table 3.

TABLE 3

Order conditions for the kernel and postprocessor up to order  $q$  in  $\mathcal{G}_3$  when the kernel is self-adjoint ( $f_{2k,i} = 0, p_{2k-1,j} = 0$ ).

$q$	Eff. order conditions	Postprocessor conditions
1	$f_{1,1} = 1$	
3	$f_{3,1} = 0$	
5	$f_{5,1} = 0$	$p_{4,1} = -f_{5,2}$
7	$f_{7,1} = f_{7,2} = 0$	$p_{6,i} = -f_{7,i+2}, \quad i = 1, 2$
9	$f_{9,1} = f_{9,2} = 0$	$p_{8,i} = -f_{9,i+3}, \quad i = 1, 2, 4, 5$
	$f_{9,3} = \frac{1}{2}f_{5,2}^2$	$p_{8,3} = -f_{9,6} - \frac{1}{2}f_{5,2}^2$
11	$f_{11,i} = 0, \quad i = 1, 2, 3, 4, 7$	$p_{10,i} = -f_{11,i+7}, \quad i = 1, 2, 4, 7, 8, 10, 11$
	$f_{11,5} = f_{5,2}f_{7,3}$	$p_{10,3} = -f_{11,10} - \frac{3}{2}f_{5,2}f_{7,3}; \quad p_{10,5} = \frac{1}{2}f_{5,2}f_{7,3} - f_{11,12}$
	$f_{11,6} = f_{5,2}f_{7,4}$	$p_{10,6} = -f_{11,13} - \frac{5}{2}f_{5,2}f_{7,4}; \quad p_{10,9} = \frac{3}{2}f_{5,2}f_{7,4} - f_{11,16}$

**3. Construction of the processor.** Once a kernel satisfying the effective order conditions is proposed, there is still much freedom in the choice of the preprocessor and postprocessor  $\pi_h$  to ensure that  $\hat{\psi}_h$  has the required order. The kernel fixes the accuracy that is possible to reach. This can be attained with the optimal postprocessor, defined as the exact 1-flow of an infinite series of vector fields  $P_h$ . In practice, one must look for a preprocessor and a postprocessor close to this optimal one (an approximate map in  $\mathbb{P}_{q-1}$  or preferably in  $\mathbb{P}_q$ ), and at the same time it has to be easy and/or cheap to compute. In the following we propose two different techniques which satisfy the above requirements.

**3.1. Composition postprocessors,  $\pi_h^{(c)}$ .** Since the kernel is a composition of basic methods in  $\mathcal{G}_l$ , it seems appropriate to approximate the optimal map  $\pi_h$  by a similar composition,  $\pi_h^{(c)}$ , so that the algorithm employed to compute the kernel can also be used. The resulting processed methods are also in  $\mathcal{G}_l$ . The coefficients of the composition  $\pi_h^{(c)}$  are chosen such that (2.9) are verified. It turns out, however, that finding real solutions for the resulting system of polynomial equations can be difficult, and the composition requires typically at least as many basic integrators as the kernel.

As we mentioned in the introduction, if the kernel is time-symmetric and  $\pi_{-h} = \pi_h$ , then the processed method is also time-symmetric, but this equality cannot be satisfied by  $\pi_h^{(c)}$ . Instead we impose that  $\pi_{-h}^{(c)} - \pi_h^{(c)} = \mathcal{O}(h^l), l \geq q$ , for a postprocessor  $\pi_h^{(c)} \in \mathbb{P}_{q-1}$ . In this sense, it is sometimes helpful to construct  $\pi_h^{(c)}$  as a composition of the form  $\pi_h^{(c)} = \omega_h \circ \omega_{-h}$ , or even of the form  $\pi_h^{(c)} = \omega_{-h} \circ \omega_h^2 \circ \omega_{-h}$ . With these special structures many order conditions of the form  $p_{2i-1,j} = 0$  in Theorem 2 are automatically satisfied, but the order conditions (2.9) of even  $k$  must still be enforced, which still account for more equations than the kernel itself. Since the preprocessor

is computed only once, it makes sense to use  $(\pi_h^{(c)})^{-1}$  as an accurate approximation to  $\pi_h^{-1}$ , whereas  $\pi_h^{(c)}$  should be used if intermediate output is not frequently required.

**3.2. Cheap postprocessors,  $\hat{\pi}_h$ .** For a number of problems, the output has to be computed for most steps. It is therefore reasonable to look for an approximation  $\hat{\pi}_h$  to the optimal postprocessor as cheap to compute as possible. Usually  $\hat{\pi}_h$  will be a less accurate approximation to  $\pi_h$  than the composition  $\pi_h^{(c)}$ , but the error  $\hat{\pi}_h(z_n) - \pi_h^{(c)}(z_n)$  is of local character and will eventually be overtaken by the global error of the underlying optimally processed integrator (if the global error grows at least linearly in time) [2].

Here we approximate  $\pi_h$  by reusing intermediate calculations obtained when evaluating the kernel  $\psi_h$  [2], so that the procedure can be considered virtually cost-free. More specifically, if  $z_n \equiv \psi_h^n(\pi_h^{-1}(x_0))$ , then the numerical solution  $x_n = \pi_h(z_n)$  is approximated by the linear combination

$$(3.1) \quad x_n \approx \sum_{i=-\bar{s}}^{\bar{s}} w_i Z_i$$

of intermediate values  $Z_i$  computed when evaluating  $z_n = \psi_h(z_{n-1})$  and  $z_{n+1} = \psi_h(z_n)$ . In particular, we always take  $Z_{-\bar{s}} = z_{n-1}$ ,  $Z_0 = z_n$ , and  $Z_{\bar{s}} = z_{n+1}$ . For the set  $\mathcal{G}_l$  ( $l = 1, 2, 3, 4$ ), the intermediate values  $Z_i$  that we propose for approximating  $x_n$  are the following.

1. In the case of  $\mathcal{G}_1$ , we have  $\bar{s} = 2s$ , and

$$(3.2) \quad Z_{-2s} = z_{n-1}, \quad Z_{2i-1} = \varphi_{\beta_{2i-1}h}^{[a]}(Z_{2i-2}), \quad Z_{2i} = \varphi_{\beta_{2i}h}^{[b]}(Z_{2i-1}),$$

where  $-(s-1) \leq i \leq s$  and  $\beta_{j-2s} = \beta_j$  for each  $j$ .

2. For  $\mathcal{G}_2$ ,

$$(3.3) \quad Z_{-2s} = z_{n-1}, \quad Z_{2i-1} = \chi_{\alpha_{2i-1}h}^*(Z_{2i-2}), \quad Z_{2i} = \chi_{\alpha_{2i}h}(Z_{2i-1}),$$

with  $-(s-1) \leq i \leq s$  and  $\alpha_{j-2s} = \alpha_j$  for each  $j$ .

3. For  $\mathcal{G}_3$  and  $\mathcal{G}_4$ , we take  $\bar{s} = s$ , and

$$(3.4) \quad Z_{-s} = z_{n-1}, \quad Z_i = \mathcal{S}_{\alpha_i h}^{[2]}(Z_{i-1}) \quad (\text{resp., } Z_i = \mathcal{S}_{\alpha_i h}^{[4]}(Z_{i-1})),$$

with  $-(s-1) \leq i \leq s$  and  $\alpha_{j-s} = \alpha_j$  for each  $j$ .

*Remark.* As we will see, the number of conditions to be imposed on the coefficients  $w_i$  so that the cheap postprocessor (3.1) achieves a given accuracy is smaller in  $\mathcal{G}_2$  than in  $\mathcal{G}_1$ . Nevertheless, implementing  $\psi_h \in \mathcal{G}_1$  as an integrator in  $\mathcal{G}_2$  considering the relation between (2.1) and (2.2) is in general more costly because explicitly obtaining the intermediate values  $Z_i$  in (3.3) requires the computation of additional basic flows, which does not in general pay off for the extra accuracy of the postprocessor. An important exception is the case in which  $f_a$  and  $f_b$  come from a partitioned ODE of the form  $q' = f_1(p)$ ,  $p' = f_2(q)$ . Then an integrator in  $\mathcal{G}_1$  can be seen as an explicit partitioned Runge–Kutta method. In that case, there is no practical overhead in implementing the kernel  $\psi_h \in \mathcal{G}_1$  as an integrator in  $\mathcal{G}_2$  because it does not require additional evaluations of  $f_1(p)$  and  $f_2(q)$ , and then it is recommended to do it that way.  $\square$

Each  $Z_i$  can be written as  $Z_i = \phi_h^{(i)}(z_n)$ , where  $\phi_h^{(i)} \in \mathcal{G}_l$  ( $l = 1, 2, 3, 4$ ,  $-\bar{s} \leq i \leq \bar{s}$ ) implies that there exists a series  $F_h^{(i)} = \sum_{k \geq 1} h^k F_k^{(i)} \in \mathcal{L}$  such that formally

$g \circ \phi_h^{(i)} = \exp(F_h^{(i)})[g]$  for each infinitely differentiable map  $g : \mathbb{R}^D \rightarrow \mathbb{R}$  (see [2] for more details). Clearly, (3.1) can be expressed as

$$(3.5) \quad x_n \approx \hat{\pi}_h(z_n), \quad \text{where} \quad \hat{\pi}_h(z) = \sum_{i=-\bar{s}}^{\bar{s}} w_i \phi_h^{(i)}(z).$$

Notice that, in general,  $\hat{\pi}_h \notin \mathcal{G}_l$ . Our goal is to find coefficients  $w_i$  in such a way that  $\hat{\pi}_h \in \mathbb{P}_l$  with  $l$  as large as possible. This is guaranteed for a given  $l \geq 1$  if

$$(3.6) \quad \sum_{i=-\bar{s}}^{\bar{s}} w_i \exp(F_h^{(i)}) = \exp(P_h) + \mathcal{O}(h^{l+1}),$$

where  $P_h = \sum_{k \geq 1} h^k P_k \in \mathcal{L}$  is the vector field corresponding to the optimal postprocessor.

The exponential of a formal series in the graded Lie algebra  $\mathcal{L}$  can be represented as an element of the universal enveloping algebra  $\mathcal{A}$  of  $\mathcal{L}$ . The algebra  $\mathcal{A}$  is also graded, with  $\mathcal{A} = \bigoplus_{k \geq 0} \mathcal{A}_k$ , and  $\mathcal{A}_0 = \text{span}(I)$ . Let us now consider a fixed basis  $\{D_{k,j}\}_{j=1}^{m_k}$  of the homogeneous subspace  $\mathcal{A}_k$  for each  $k \geq 1$ , with  $m_k = \dim \mathcal{A}_k$ . The values of  $m_k$  for  $\mathcal{G}_l$  ( $l = 1, 2, 3, 4$ ) are displayed in Table 4 [2]. Then  $\exp(P_h)$  and  $\exp(F_h^{(i)})$  can be expressed as

$$\exp(P_h) = I + \sum_{k \geq 1} h^k \sum_{j=1}^{m_k} \pi_{k,j} D_{k,j}, \quad \exp(F_h^{(i)}) = I + \sum_{k \geq 1} h^k \sum_{j=1}^{m_k} \phi_{k,j}^{(i)} D_{k,j}$$

for  $-\bar{s} \leq i \leq \bar{s}$ , where  $\pi_{k,j}$  and  $\phi_{k,j}^{(i)}$  are polynomials in the coefficients  $\alpha_j$ ,  $1 \leq j \leq \bar{s}$ . Hence, (3.6) is equivalent to a system of linear equations on the unknowns  $w_i$ , i.e.,

$$(3.7) \quad \sum_{i=-\bar{s}}^{\bar{s}} w_i \phi_{k,j}^{(i)} = \pi_{k,j}, \quad 1 \leq j \leq m_k, \quad 0 \leq k \leq l.$$

In particular,  $\hat{\pi}_h \in \mathbb{P}_0$  requires that  $\sum_{i=-\bar{s}}^{\bar{s}} w_i = 1$ . The total number of equations (3.7) required for  $\hat{\pi}_h \in \mathbb{P}_l$  is  $1 + m_1 + \dots + m_l$  ( $m_0 = 1$ ).

TABLE 4

*Dimensions  $m_k$  of  $\mathcal{A}_k$  for each class of integrators. If the kernel is self-adjoint, only the dimensions  $m_{2j}$  are relevant (bold numbers).*

	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$m_6$	$m_7$	$m_8$
$\mathcal{G}_1$	2	<b>4</b>	8	<b>16</b>	32	<b>64</b>	128	<b>256</b>
$\mathcal{G}_2$	1	<b>2</b>	4	<b>8</b>	16	<b>32</b>	64	<b>128</b>
$\mathcal{G}_3$	1	<b>1</b>	2	<b>3</b>	5	<b>8</b>	13	<b>21</b>
$\mathcal{G}_4$	1	<b>1</b>	1	<b>1</b>	2	<b>3</b>	5	<b>7</b>

For self-adjoint composition kernels,  $P_{-h} = P_h$  and  $F_h^{(-i)} = F_{-h}^{(i)}$ , which implies that  $\phi_{k,j}^{(-i)} = (-1)^k \phi_{k,j}^{(i)}$ . The choice  $w_{-i} = w_i$  for all  $i$  in (3.1) then makes sense, as this guarantees that (3.7) is automatically satisfied for odd values of  $k$ . If in addition  $w_0 = 1 - 2 \sum_{i=1}^{\bar{s}} w_i$  (so that  $\hat{\pi}_h \in \mathbb{P}_l$  at least with  $l = 0$ ), (3.7) holds if

$$(3.8) \quad 2 \sum_{i=1}^{\bar{s}} w_i \phi_{2k,j}^{(i)} = \pi_{2k,j}, \quad 1 \leq j \leq m_{2k}, \quad 1 \leq k \leq \frac{l}{2}.$$

Hence, the number of equations that remain to be satisfied by the unknowns  $w_1, \dots, w_{\bar{s}}$  in order that  $\hat{\pi}_h \in \mathbb{P}_{2r+1}$  is  $m_2 + \dots + m_{2r}$ .

When the number of unknowns  $w_i$  is larger than the number of equations (3.7) required so that  $\hat{\pi}_h \in \mathbb{P}_l$  for a given  $l$ , then one can use this freedom to minimize the difference with the optimal postprocessor at higher orders.

**4. New processed methods.** In this section we construct self-adjoint composition kernels in  $\mathcal{G}_l$  satisfying certain optimization criteria and a simplifying choice of the coefficients. For the kernels collected here we also provide in Appendix A the corresponding postprocessor, built according to the procedure of section 3.

**4.1. Construction of the kernel.** For building a specific kernel  $\psi_h$  of effective order  $q$  by composition in  $\mathcal{G}_l$ , one typically has to take as many basic integrators as the total number,  $s(q) = n_q + 1$ , of effective order conditions. As a matter of fact, more stages are frequently required (a) due to the nonexistence of real solutions or (b) to diminish the size of higher ( $> q + 1$ )–order error terms (usually of large magnitude) [5, 13, 15]. In any case, introducing extra parameters makes more difficult the task of finding the numerical values of  $\alpha$  which solve the equations  $N_{k,i}(\alpha) = 0$ . For this reason one usually considers self-adjoint kernels: although this symmetry fixes half of the coefficients, it also solves (in most cases) approximately the same number of equations.

Here the following symmetric compositions in  $\mathcal{G}_l$ ,  $l = 2, 3, 4$  (schemes in  $\mathcal{G}_1$  and  $\mathcal{G}_2$  are studied together), at different orders are considered:

$$\begin{aligned}
 (4.1) \quad \psi_h^{[2]} &= \chi_{\alpha_1 h} \circ \chi_{\alpha_2 h}^* \circ \dots \circ \chi_{\alpha_2 h} \circ \chi_{\alpha_1 h}^*, \\
 (4.2) \quad \psi_h^{[3]} &= \mathcal{S}_{\alpha_1 h}^{[2]} \circ \dots \circ \mathcal{S}_{\alpha_{m-1} h}^{[2]} \circ \mathcal{S}_{\alpha_m h}^{[2]} \circ \mathcal{S}_{\alpha_{m-1} h}^{[2]} \circ \dots \circ \mathcal{S}_{\alpha_1 h}^{[2]}, \\
 (4.3) \quad \psi_h^{[4]} &= \mathcal{S}_{\alpha_1 h}^{[4]} \circ \dots \circ \mathcal{S}_{\alpha_{m-1} h}^{[4]} \circ \mathcal{S}_{\alpha_m h}^{[4]} \circ \mathcal{S}_{\alpha_{m-1} h}^{[4]} \circ \dots \circ \mathcal{S}_{\alpha_1 h}^{[4]},
 \end{aligned}$$

where  $\psi_h^{[2]}$  involves  $2m$  stages (or  $m$  different evaluations of  $\varphi_h^{[a]}$  and  $\varphi_h^{[b]}$  if  $\chi_h = \varphi_h^{[a]} \circ \varphi_h^{[b]}$ ), whereas  $\psi_h^{[3]}$  and  $\psi_h^{[4]}$  have  $(2m - 1)$  stages, and all of them have  $\alpha_1, \dots, \alpha_m$  to solve the effective order conditions.

Among the different solutions, one is interested in those that minimize the  $l_{q+1} = n_{q+1} - n_q^*$  noncorrectable terms at order  $q + 1$ , according to some criterion previously adopted. For instance, one could consider a norm in  $\mathcal{L}_{q+1}$  and use it to devise some objective function of the parameters  $\alpha_i$  to be minimized, aimed at measuring the size of the error. Unfortunately, it is by no means obvious how to characterize the performance of the integrators applied to all initial value problems, since for different problems the dominant error terms are not necessarily the same.

At least two objective functions have been used in the literature:

$$(4.4) \quad E_1(\alpha) = \sum_{i=1}^{\bar{m}} |\alpha_i| \quad \text{and} \quad E_2(\mathbf{d}, \alpha) = \bar{m} \|\mathbf{d} \cdot \mathbf{f}_{q+1}\|^{1/q},$$

where  $\bar{m} = 2m$  or  $2m - 1$  (depending on the set  $\mathcal{G}_l$  considered),  $\alpha_{\bar{m}+1-i} = \alpha_i$ , and  $\mathbf{f}_{q+1} = (f_{q+1,1}, \dots, f_{q+1,l_{q+1}})$  and  $\mathbf{d} = (d_1, \dots, d_{l_{q+1}})$  are weight parameters fixed in advance. Methods with small values of  $E_1$  usually have large stability domains and small error terms. It has been successfully used to find efficient nonprocessed methods in  $\mathcal{G}_3$  up to order 10 [10, 8], although it is not possible to compare the efficiency of schemes with different number of stages. On the other hand, the optimal

weight vector  $\mathbf{d}$  present in  $E_2$  is problem dependent. For methods in  $\mathcal{G}_1$  it is usual to consider  $\mathbf{d}^{(1)} = (1, \dots, 1)$  (we denote  $E_{21} \equiv E_2(\mathbf{d}^{(1)}, \boldsymbol{\alpha})$ ), whereas in  $\mathcal{G}_3$  it is suggested [15] that one take  $\mathbf{d}^{(2)} = (1, 0, \dots, 0)$  (and denote  $E_{22} \equiv E_2(\mathbf{d}^{(2)}, \boldsymbol{\alpha})$ ), since  $f_{q+1,1} = \sum_{i=1}^{\bar{m}} \alpha_i^{q+1}$  is the dominant error term for a number of problems.

Here we adopt the following criterion: we look for methods with small values of  $E_1$  which, in addition, have small values of  $E_{21}$  for schemes in  $\mathcal{G}_1$  and/or  $E_{22}$  for schemes in  $\mathcal{G}_l$ ,  $l = 2, 3, 4$ . Of course, one could propose other vectors  $\mathbf{d}^{(j)}$  and thus different objective functions to minimize.

As we mentioned before, the number of coefficients in the composition is usually larger than the number of order conditions, so that there are  $l$  ( $< m$ ) free parameters  $\alpha_1, \dots, \alpha_l$ . Recently, McLachlan [15] has proposed an idea to find (in a relatively simple way) efficient methods in  $\mathcal{G}_3$ : he takes the free parameters  $(\alpha_1, \dots, \alpha_l)$  equal to the first (and last) coefficient  $\alpha_{l+1}$  of the symmetric composition with the minimum number of stages necessary to satisfy the order conditions. Then, instead of many multidimensional families of solutions, only isolated solutions are obtained which are very easy to analyze. For methods up to effective order 6 in  $\mathcal{G}_3$ , the solutions obtained by applying this *rule of thumb* correspond to simultaneous local minima of  $E_1$  and  $E_{22}$ . For higher orders, this is not necessarily the case, although the corresponding solutions seem to stay very close to the minima.

Here we propose to extend this rule in the following way: we take  $\alpha_1 = \alpha_2 = \dots = \alpha_l \equiv \alpha$  as free parameter and solve the effective order conditions for different values of  $\alpha$ . Notice that  $\alpha = \alpha_{l+1}$  corresponds to McLachlan's rule of thumb. A numerical search with values  $\alpha_1, \dots, \alpha_l$  chosen at random also has been carried out, but the solutions thus found do not lead to smaller values of  $E_1$  and  $E_2$  (see Appendix B for more details). In fact, there exists a strong correlation between  $E_1$  and  $E_2$ , but their absolute minima do not necessarily coincide ( $E_2$  can be zero for some choice of  $\mathbf{d}$ , whereas  $E_1$  is always nonvanishing).

In view of the number of effective order conditions to be satisfied by symmetric kernels in  $\mathcal{G}_l$  and collected in Table 1, it seems reasonable to consider methods up to order 6 in  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , up to order 10 in  $\mathcal{G}_3$ , and up to order 12 in  $\mathcal{G}_4$ . The new methods are denoted by  $P_{\bar{m}q}$ : a processed method of order  $q$  with an  $\bar{m}$ -stage self-adjoint kernel.

**4.2. Kernels in  $\mathcal{G}_1$  and  $\mathcal{G}_2$  up to effective order 6.** Several fourth- and sixth-order schemes in  $\mathcal{G}_1$  (also valid in  $\mathcal{G}_2$ ) are available in the literature, both as standard composition methods [5] and as processed algorithms [3]. These integrators, however, have been designed to be efficient only in  $\mathcal{G}_1$ . In contrast, the methods we present here have been optimized to be used both in  $\mathcal{G}_1$  and  $\mathcal{G}_2$ .

*Fourth-order.* In [3] symmetric kernels with  $m = 3$  and  $m = 4$  are designed (with  $m = 2$  the order conditions do not admit real solutions). We have built kernels with  $m = 5, 6, 7$ , obtaining solutions with very small values of  $E_1$  and  $E_{21}$  and/or  $E_{22}$ . In Table 5 (first column) we present the coefficients of the selected kernel with  $\bar{m} = 2m = 12$  stages.

*Sixth-order.* With  $m = 5$  it is possible to solve all the order conditions [3], but the resulting schemes are not very efficient. We have considered kernels with  $m$  up to 10. The most efficient methods found have  $m = 9$  and  $m = 10$  and their coefficients are collected in Table 5 (second and third columns). These methods are discussed in more detail in Appendix B. We should remark that, if  $\chi_h = \varphi_h^{[a]} \circ \varphi_h^{[b]}$ , these kernels require  $m$  evaluations of  $\varphi_h^{[a]}$  and  $\varphi_h^{[b]}$  (due to the FSAL property).

TABLE 5

Coefficients in (4.1) for symmetric kernels in  $\mathcal{G}_1$  and  $\mathcal{G}_2$ . These kernels can be used with the corresponding postprocessors of Table 8.

P <sub>6</sub> 4	P <sub>9</sub> 6	P <sub>10</sub> 6
$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6$
$\alpha_4 = 0.1341940158122142$	$\alpha_5 = 0.1106570871853300$	$\alpha_6 = 0.1008383784517379$
$\alpha_5 = -\alpha_4 - 0.18$	$\alpha_6 = -0.2854111127287940$	$\alpha_7 = 0.1008384231345842$
$\alpha_6 = 1/2 - (\alpha_1 + \dots + \alpha_5)$	$\alpha_7 = 0.2138498496192465$	$\alpha_8 = -0.2387378577407101$
$\alpha_{13-i} = \alpha_i, \quad i = 1, \dots, 6$	$\alpha_8 = -0.3402583791791715$	$\alpha_9 = -0.2387378757998321$
	$\alpha_9 = 1/2 - (\alpha_1 + \dots + \alpha_8)$	$\alpha_{10} = 1/2 - (\alpha_1 + \dots + \alpha_9)$
	$\alpha_{19-i} = \alpha_i, \quad i = i, \dots, 9$	$\alpha_{21-i} = \alpha_i, \quad i = i, \dots, 10$

**4.3. Kernels in  $\mathcal{G}_3$  up to effective order 10.** Nonprocessed methods up to order 10 are available in [26, 23, 13, 10, 8] and up to order 6 using processing in [1, 15]. Based on the symmetric  $(2m - 1)$ -stage composition (4.2) we present the following kernels.

*Sixth-order.* With  $m = 3$  (5 stages) the effective order conditions  $f_{1,1} = 1$ ,  $f_{3,1} = f_{5,1} = 0$  possess two real solutions, but both of them lead to high values of the noncorrectable error terms  $f_{7,1}$  and  $f_{7,2}$  [1]. With  $l = 1, \dots, 5$  free parameters we have obtained a number of methods with small values of  $E_1$  and  $E_{22}$ . We propose taking the 11-stage kernel given in [15] (P<sub>11</sub>6) and the 13-stage kernel (P<sub>13</sub>6) which, in addition, satisfies  $f_{7,2} = 0$  (see Table 6).

*Eighth-order.* With  $m \geq 5$  (at least 9 stages) one has enough variables to achieve effective order 8, although the solutions obtained give poorly efficient schemes. Additional stages are considered by taking up to  $l = 5$  free parameters. We choose two solutions, one with  $l = 2$  (P<sub>13</sub>8) and the other with  $l = 5$  (P<sub>19</sub>8). The coefficients are collected in Table 6. The solution with  $l = 2$  does not follow exactly the rule of thumb, but it is quite close to it ( $\alpha_1 = \alpha_2 \simeq \alpha_5$ ).

*Tenth-order.* If  $m \geq 8$  (at least 15 stages) one has enough parameters to achieve effective order 10. Kernels with  $l = 0, \dots, 5$  free parameters have been analyzed, and the schemes P<sub>19</sub>10 and P<sub>23</sub>10 whose coefficients are collected in Table 6 have been selected.

**4.4. Kernels in  $\mathcal{G}_4$  up to effective order 12.** Next we consider the symmetric  $(2m - 1)$ -stage composition (4.3). The numerical search for methods of order  $q$  in  $\mathcal{G}_3$  and of order  $q + 2$  in  $\mathcal{G}_4$  is closely related because the corresponding equations are very similar. In fact, the number of order conditions is smaller in  $\mathcal{G}_4$ , and the numerical search is easier. Now the methods chosen exactly follow the rule of thumb.

*Eighth-order.* If  $m \geq 3$  (at least 5 stages) one has enough variables to get effective order 8. With  $m = 3$ , solutions with large values of the noncorrectable leading error term  $f_{9,1}$  are obtained [1]. The coefficients of a 9-stage kernel (P<sub>9</sub>8) are collected in Table 7. Kernels with more stages have also been analyzed, but their performance is not clearly superior.

*Tenth-order.* At least 7 stages ( $m \geq 4$ ) are necessary to achieve effective order 10. Kernels with up to  $l = 5$  free parameters have been considered, and the kernel P<sub>13</sub>10 with coefficients given in Table 7 is finally selected.

*Twelfth-order.* If  $m \geq 6$  (at least 11 stages) a twelfth-order integrator can be obtained. Kernels with up to  $l = 6$  free parameters have been analyzed. The coefficients for the composition with 19 stages, P<sub>19</sub>12, are collected in Table 7.

TABLE 6

Coefficients in (4.2) for symmetric kernels in  $\mathcal{G}_3$ . They can be used with the respective postprocessors collected in Table 9.

P <sub>116</sub>	P <sub>136</sub>	P <sub>198</sub>
$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$	$\alpha_1 = \dots = \alpha_6$
$\alpha_4 = 0.1705768865009222157$	$\alpha_4 = 0.125696288720106$	$\alpha_6 = 0.09155941827296$
$\alpha_5 = -0.423366140892658048$	$\alpha_5 = 0.148070660114965$	$\alpha_7 = 0.36968952549113$
$\alpha_6 = 1 - 2(\alpha_1 + \dots + \alpha_5)$	$\alpha_6 = -0.350856370823828$	$\alpha_8 = 0.06866857653282$
$\alpha_{12-i} = \alpha_i, \quad i = 1, \dots, 5$	$\alpha_7 = 1 - 2(\alpha_1 + \dots + \alpha_6)$	$\alpha_9 = -0.28931413259236$
	$\alpha_{14-i} = \alpha_i, \quad i = 1, \dots, 6$	$\alpha_{10} = 1 - 2(\alpha_1 + \dots + \alpha_9)$
		$\alpha_{20-i} = \alpha_i, \quad i = 1, \dots, 9$
P <sub>138</sub>	P <sub>2310</sub>	P <sub>1910</sub>
$\alpha_1 = \alpha_2$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5$	$\alpha_1 = \alpha_2 = \alpha_3$
$\alpha_2 = 0.168$	$\alpha_5 = 0.121657748919383$	$\alpha_3 = 0.16176042393895$
$\alpha_3 = 0.585550530805562$	$\alpha_6 = -0.511318780154828$	$\alpha_4 = -0.71963383963697$
$\alpha_4 = -0.460090457516872$	$\alpha_7 = -0.172858614884985$	$\alpha_5 = 0.79594876856276$
$\alpha_5 = 0.172863148729731$	$\alpha_8 = 0.123016258833066$	$\alpha_6 = 0.59733925980951$
$\alpha_6 = 0.179664539695039$	$\alpha_9 = 0.441503951671565$	$\alpha_7 = 0.082347969317011$
$\alpha_7 = 1 - 2(\alpha_1 + \dots + \alpha_6)$	$\alpha_{10} = -0.327071324165477$	$\alpha_8 = -0.43345109677776$
$\alpha_{14-i} = \alpha_i, \quad i = 1, \dots, 6$	$\alpha_{11} = 0.070952700957766$	$\alpha_9 = 0.10313406454059$
	$\alpha_{12} = 1 - 2(\alpha_1 + \dots + \alpha_{11})$	$\alpha_{10} = 1 - 2(\alpha_1 + \dots + \alpha_9)$
	$\alpha_{24-i} = \alpha_i, \quad i = 1, \dots, 11$	$\alpha_{20-i} = \alpha_i, \quad i = 1, \dots, 9$

TABLE 7

Coefficients in (4.3) for symmetric kernels in  $\mathcal{G}_4$ . The corresponding processed integrator is formed with the postprocessors of Table 10.

P <sub>98</sub>	P <sub>1310</sub>	P <sub>1912</sub>
$\alpha_1 = \alpha_2 = \alpha_3$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5$
$\alpha_3 = 0.2233380451446240$	$\alpha_4 = 0.1578763989460225$	$\alpha_5 = 0.1008183703667023$
$\alpha_4 = -0.3739874130116841$	$\alpha_5 = -0.3010347145730912$	$\alpha_6 = 0.2659998884940344$
$\alpha_5 = 1 - 2(\alpha_1 + \dots + \alpha_4)$	$\alpha_6 = 0.3577728201055250$	$\alpha_7 = -0.2142408955837595$
$\alpha_{10-i} = \alpha_i, \quad i = 1, \dots, 4$	$\alpha_7 = 1 - 2(\alpha_1 + \dots + \alpha_6)$	$\alpha_8 = -0.2968630395353906$
	$\alpha_{14-i} = \alpha_i, \quad i = 1, \dots, 6$	$\alpha_9 = 0.08741046298860494$
		$\alpha_{10} = 1 - 2(\alpha_1 + \dots + \alpha_9)$
		$\alpha_{20-i} = \alpha_i, \quad i = 1, \dots, 9$

**4.5. On the accuracy of the coefficients.** The coefficients of the previous kernels have been obtained with only 15 digits, even when the order of the scheme is 10 or 12. The error introduced due to the accuracy of the coefficients is propagated along the integration, and thus one could think that a 15-digit precision is not sufficient for carrying out very accurate computations. It turns out, however, that this is not observed in practice. To explain this phenomenon notice that, since the kernel is self-adjoint, the asymptotic expansion of the error contains only odd powers of  $h$ :  $hf_{1,i_1}$ ,  $h^3f_{3,i_3}$ ,  $h^5f_{5,i_5}$ ,  $\dots$ , and the step size  $h$  has to be small enough to guarantee convergence and sufficiently high accuracy. Now the consistency conditions for  $f_{1,i_1}$  are exactly satisfied (up to roundoff), and the error in the lowest-order conditions  $N_{3,i}(\boldsymbol{\alpha}) = 0$  and  $N_{5,i}(\boldsymbol{\alpha}) = 0$  is approximately  $h^3 10^{-15}$  and  $h^5 10^{-15}$ , respectively, which is sufficiently accurate for most problems. If, for a given method, one is interested in obtaining more precise coefficients, the process illustrated by the following example can be applied.

*Example.* Suppose that the coefficients of the kernel for the processed method P<sub>2310</sub> in  $\mathcal{G}_3$  are desired with higher accuracy. We show how to increase the actual precision from  $h^3 10^{-15}$  to  $h^5 10^{-15}$  just by adding some extra digits to one of the coefficients  $\alpha_j$ .

First we consider the lowest effective order conditions  $f_{1,1} = \sum_i \alpha_i = 1$ ,  $N_{3,1}(\boldsymbol{\alpha}) = f_{3,1} = \sum_i \alpha_i^3 = 0$ . Let us fix all values of  $\alpha_i$  from Table 6 except one, say  $\alpha_{11}$  (as well as  $\alpha_{12}$  to ensure that condition  $f_{1,1} = 1$  is exactly satisfied). Next, solve  $N_{3,1}(\boldsymbol{\alpha}) = 0$  with as many digits as desired. For example, with 25 digits one gets  $\alpha_{11} = 0.0709527009577666581583926$ , and the error of the method due to the accuracy of the coefficients is  $h^3 10^{-25} + h^5 10^{-15} \sim h^5 10^{-15}$ . This process can also be used to increase the accuracy to  $h^7 10^{-15}$  just by taking two parameters to solve  $N_{3,1}(\boldsymbol{\alpha}) = 0$  and  $N_{5,1}(\boldsymbol{\alpha}) = f_{5,1} = 0$ .

**5. Numerical examples.** In this section we test the performance of the new processed methods on some numerical examples. A rigorous test is very difficult because the relative performance between the methods usually depend on the particular problem analyzed. If the vector field considered involves some parameters, this relative performance may depend also on them as well as on the particular splitting or basic method used for the integration. It may also depend on the initial conditions and even on the final time at which the comparison is done (although this dependence can be minimized taking an average of the error along the integration). For this reason our experiments involve several problems depending on parameters and a comparison of the processed integrators with other schemes available in the literature for different values of parameters, initial conditions, and splitting or basic methods used. In particular, we compare with the following methods (the order, number of stages, and the class  $\mathcal{G}_l$  to which they belong is indicated):

- *Order 4.*  
 $\mathcal{G}_1, \mathcal{G}_2$  (BM<sub>6</sub>4): the 6-stage method (optimized in  $\mathcal{G}_1$ ) given in [5];  
 $\mathcal{G}_3$  (Y<sub>3</sub>4): the well-known 3-stage method of [26] (only used as a reference);  
 $\mathcal{G}_3$  (S<sub>5</sub>4): the 5-stage method of [22].
- *Order 6.*  
 $\mathcal{G}_1, \mathcal{G}_2$  (BM<sub>10</sub>6): the 10-stage method (optimized in  $\mathcal{G}_1$ ) given in [5];  
 $\mathcal{G}_3$  (M<sub>9</sub>6): the optimized 9-stage method ( $S, m = 9$ ) of [13].
- *Order 8.*  
 $\mathcal{G}_3$  (M<sub>17</sub>8): the 17-stage method ( $S, m = 17$ ) of [13];  
 $\mathcal{G}_4$  (B<sub>7</sub>8): the 7-stage method of [1].
- *Order 10.*  
 $\mathcal{G}_3$  (R<sub>33</sub>10): the 33-stage method (s33odr10c) of [10];  
 $\mathcal{G}_3$  (G<sub>33</sub>10): the 33-stage method of [8].  
 $\mathcal{G}_4$  (O<sub>13</sub>10): the 13-stage method of [20].
- *Order 12.*  
 $\mathcal{G}_4$  (O<sub>23</sub>12): the 23-stage method of [20].

In the comparison, the computational cost (or the number of evaluations) of the kernel is taken as the cost of the processed method. All computations are always started with the most accurate composition preprocessor  $(\pi_h^{(c)})^{-1}$  available. For post-processing we take  $\pi_h^{(c)}$ , although very similar results are obtained when the alternative cheap postprocessor  $\hat{\pi}_h$  is used instead, in agreement with the results exposed in [2].

*Example 1.* We consider the Lotka–Volterra system

$$(5.1) \quad u' = u(v - 2), \quad v' = v(1 - u),$$

which has as first integral  $I(u, v) = \ln(uv^2) - (u + v)$ . For simplicity, we split the vector field as  $f = f_a + f_b$ , with  $f_a = (u(v - 2), 0)$  and  $f_b = (0, v(1 - u))$ , which allows us to consider explicit methods which preserve the Poisson structure. This implies, in particular, that the first integral  $I(u, v)$  is approximately preserved (for in that case, splitting



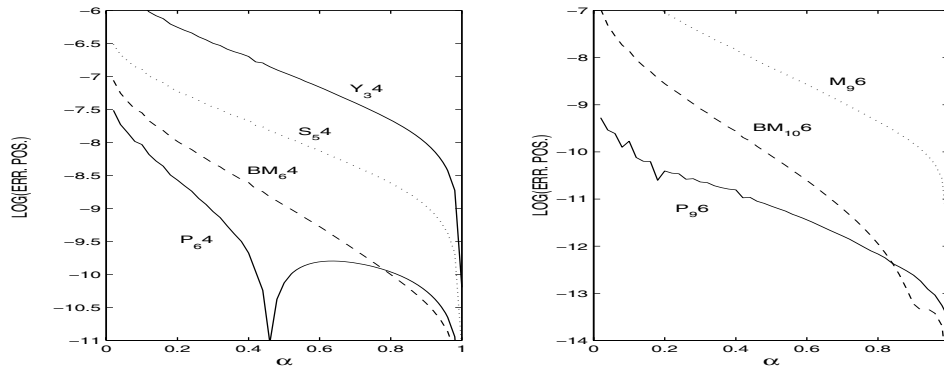


FIG. 1. Average relative error in position for the methods in  $\mathcal{G}_1$  ( $\mathcal{G}_2$ ) and for different initial conditions:  $(u_0, v_0) = (\alpha, 2\alpha)$ ,  $0 < \alpha < 1$ . The methods  $Y_{3,4}$ ,  $S_{5,4}$ , and  $M_{9,6}$ , belonging to  $\mathcal{G}_3$ , are included as reference since they are frequently used. Time steps used are  $h = m/270$  for the fourth-order methods denoted by  $X_m4$  with  $X=Y, S, BM, P$ , so that the four methods require the same number of function evaluations. Similarly, for the sixth-order methods  $X_m6$  with  $X=M, BM, P$ , we have taken  $h = m/180$ .

methods are, with an appropriate change of variables, symplectic methods applied to a Hamiltonian system with Hamiltonian function  $H(p, q) := I(u(p, q), v(p, q))$ . The problem can be integrated by schemes in  $\mathcal{G}_1/\mathcal{G}_2$ . In the region  $0 < u, v$  the system has periodic trajectories around  $(u, v) = (1, 2)$ . We take as initial conditions  $(u_0, v_0) = (\alpha, 2\alpha)$  for different values of  $\alpha \in (0, 1)$ , and integrate for  $t \in [0, 50]$ . In this way all periodic trajectories are considered. For each initial condition we compute the error at  $t = 1, 2, \dots, 50$  and measure the average relative error in phase space. The time step is chosen in such a way that (i) all the methods of the same order in the same figure require approximately the same number of function evaluations and (ii) the relative performance between them does not change qualitatively when  $h$  is reduced. Figure 1 shows the results obtained for the fourth- and sixth-order methods in  $\mathcal{G}_1$  and  $\mathcal{G}_2$ . We observe that the relative performance between the methods highly depend on the initial conditions but, in general, the new processed methods are more efficient.

To compare the performance of the methods in  $\mathcal{G}_3$  we take  $\mathcal{S}_h^{[2]} = \varphi_{h/2}^{[a]} \circ \varphi_h^{[b]} \circ \varphi_{h/2}^{[a]}$  as the basic self-adjoint second-order integrator. The corresponding results are shown in Figure 2. Again, the processed methods are clearly superior. When the symmetric second-order implicit midpoint rule is taken as  $\mathcal{S}_h^{[2]}$ , we have observed that the results of the schemes are more sensitive to initial conditions giving, in general, more accurate results when  $\alpha$  is close to 1. Then, depending on the value of  $\alpha$  and the qualitative properties which we desire to preserve, one has to choose the most appropriate basic integrator. However, what it is important for our analysis is that the relative performance of the compared methods does not change considerably.

Finally, to compare the behavior of methods of different orders we plot efficiency diagrams. For this test we fix  $\alpha$  to a representative value of the performance of the methods: in view of Figures 1 and 2, we take  $\alpha = 1/3$ . Figure 3 shows the average relative error in phase space as a function of the number of function evaluations for different time steps. The high efficiency achieved by the processed methods  $P_{9,6}$  and  $P_{13,8}$  is apparent.

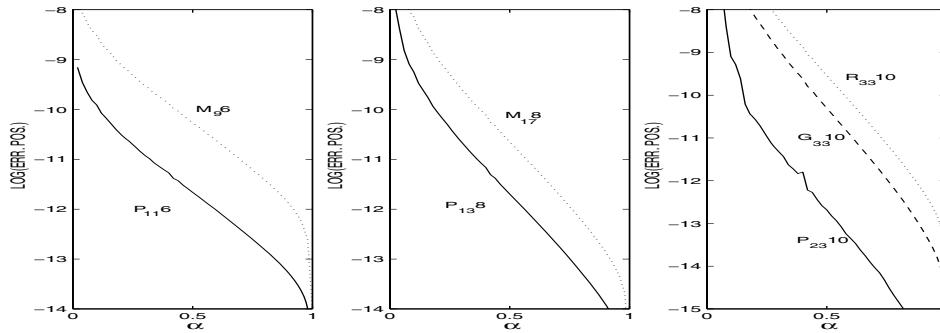


FIG. 2. Same as Figure 1 but for methods in  $\mathcal{G}_3$  taking  $h = m/396$  for  $X_m6$ ,  $h = m/(13 \times 17)$  for  $X_m8$  ( $X=M, P$ ),  $h = 1/8$  for  $G_{3310}$ ,  $R_{3310}$ , and  $h = 1/12$  for  $P_{2310}$ , so that all methods separately in each plot require, approximately, the same number of function evaluations.

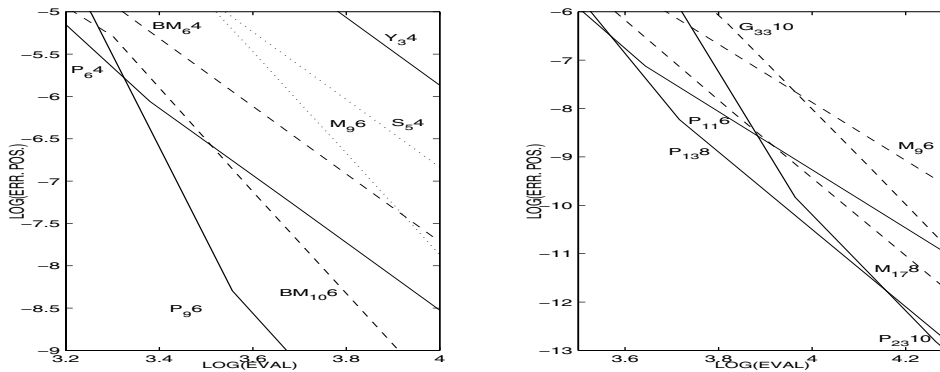


FIG. 3. Average error in phase space versus the number of evaluations for the methods of Figures 1 and 2 taking different time steps and  $\alpha = 1/3$ .

*Example 2.* Let us consider now the ABC-flow [8, 16], with equations

$$(5.2) \quad x' = B \cos y + C \sin z, \quad y' = C \cos z + A \sin x, \quad z' = A \cos x + B \sin y,$$

which has been studied as a model volume-preserving three-dimensional flow. The vector field  $f$  is separable into three solvable parts, namely

$$f = f_a + f_b + f_c = A(0, \sin x, \cos x) + B(\cos y, 0, \sin y) + C(\sin z, \cos z, 0).$$

For  $B = C = 1$ ,  $A = \alpha$  we take as initial condition  $(x_0, y_0, z_0) = (3.14, 2.77, 0)$  and integrate until  $t = 20$  for different values of  $\alpha \in [0, 1]$ . The  $\alpha$ -dependent vector field can be used to test integrators in  $\mathcal{G}_l$ ,  $l = 2, 3, 4$ . We choose the following explicit basic methods: (i) in  $\mathcal{G}_2$ , the first-order scheme  $\chi_h = \varphi_h^{[a]} \circ \varphi_h^{[b]} \circ \varphi_h^{[c]}$  and its adjoint  $\chi_h^* = \varphi_h^{[c]} \circ \varphi_h^{[b]} \circ \varphi_h^{[a]}$ ; (ii) in  $\mathcal{G}_3$ , the symmetric composition  $\mathcal{S}_h^{[2]} = \chi_{h/2} \circ \chi_{h/2}^*$ ; and (iii) in  $\mathcal{G}_4$ , the well-known 3-stage fourth-order scheme  $\mathcal{S}_h^{[4]} = \mathcal{S}_{\alpha_1 h}^{[2]} \circ \mathcal{S}_{\alpha_2 h}^{[2]} \circ \mathcal{S}_{\alpha_1 h}^{[2]}$ , with  $\alpha_1 = 1/(2 - 2^{1/3})$  and  $\alpha_2 = 1 - 2\alpha_1$ . Obviously, more efficient self-adjoint fourth-order basic methods  $\mathcal{S}_h^{[4]}$  could improve the accuracy of the results, but at this stage, we are mainly interested in the relative performance of the different integrators, and this does not change considerably with the basic method.

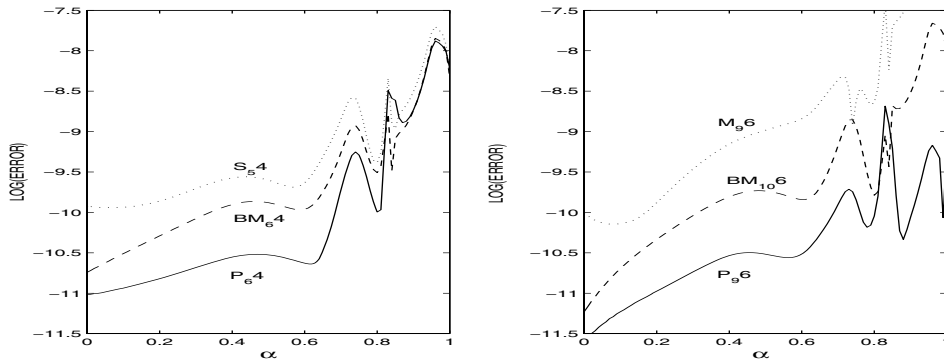


FIG. 4. Error in position for the ABC-flow with  $\mathcal{B} = \mathcal{C} = 1$  at the same computational cost for the fourth- (left) and sixth-order methods (right) in  $\mathcal{G}_2$  taking different values of the parameter  $A = \alpha$ ,  $0 \leq \alpha \leq 1$ , and initial condition  $(x_0, y_0, z_0) = (3.14, 2.77, 0)$ .

The time step size used for each method is such that all the integrators of the same order appearing in the same figure require approximately the same number of evaluations, and the results have the expected asymptotic behavior corresponding to the respective order. Figure 4 shows the results obtained with the fourth- and sixth-order methods in  $\mathcal{G}_2$ , whereas Figure 5 corresponds to the sixth-, eighth-, and tenth-order methods in  $\mathcal{G}_3$ . We have repeated the experiment with different initial conditions, and the relative performance of the methods also changes. However, we have observed a superiority of the processed integrators which, on average, is well represented by Figures 4 and 5.

On the other hand, since we have taken  $\mathcal{B} = \mathcal{C}$ , the system has a reversing symmetry group with 16 elements [16], not preserved by the preceding splitting. However, if we consider

$$(5.3) \quad f = f_a + f_b = \mathcal{B}(\cos y + \sin z, 0, 0) + (0, \mathcal{B} \cos z + \mathcal{A} \sin x, \mathcal{A} \cos x + \mathcal{B} \sin y),$$

this splitting does preserve the reversing symmetry group. Notice that  $f_a$  is explicitly integrable, but  $f_b$  is not. We can integrate  $f_b$ ,  $\varphi_h^{[b]}$ , with the midpoint rule,  $\tilde{\varphi}_h^{[b]}$ , which preserves all the properties [16]. We repeated the experiments shown in Figure 5 using as the basic method  $\mathcal{S}_h^{[2]} = \varphi_{h/2}^{[a]} \circ \tilde{\varphi}_h^{[b]} \circ \varphi_{h/2}^{[a]}$  and the same time steps. The midpoint rule was implemented with the Newton method (three iterations usually gave enough accuracy). The results are shown in Figure 6, where we observe that the relative performance between the methods change, but the processed methods are more accurate.

Finally, Figure 7 shows the results obtained by the methods in  $\mathcal{G}_4$  using the explicit methods. Now the processed integrators are always between three and four orders of magnitude more accurate at the same cost, in agreement with the size of the leading order error coefficients.

*Example 3.* Finally, to illustrate the interest of using methods in  $\mathcal{G}_4$ , we consider the two-body gravitational problem. The corresponding evolution equation may be written as

$$(5.4) \quad x_i'' = -\frac{x_i}{(x_1^2 + x_2^2)^{3/2}}, \quad i = 1, 2,$$

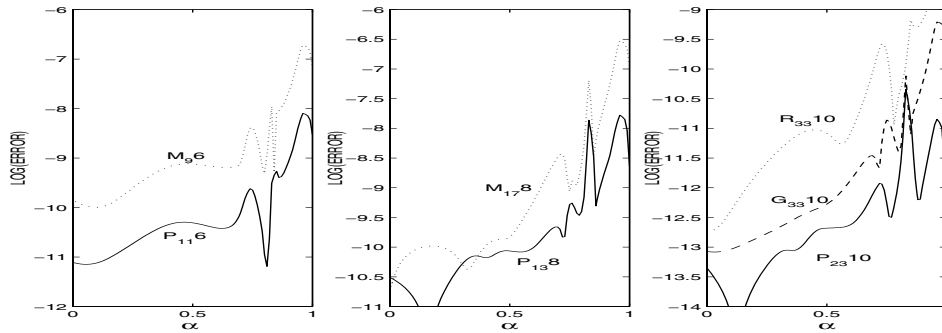


FIG. 5. Same as Figure 4 for the sixth-, eighth-, and tenth-order methods in  $\mathcal{G}_3$  taking  $h = m/99$  for  $X_m6$ ,  $h = 4m/(13 \times 17)$  for  $X_m8$  ( $X=M, P$ ),  $h = 5/23$  for  $G_{33}10$ ,  $R_{33}10$ , and  $h = 5/33$  for  $P_{23}10$ .

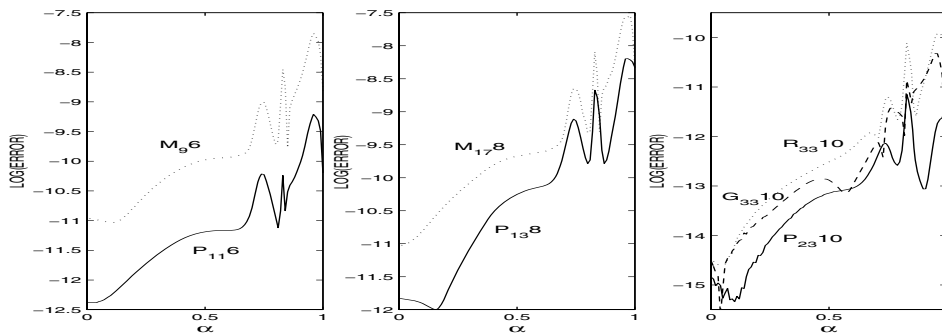


FIG. 6. Same as Figure 4 for the sixth-, eighth-, and tenth-order methods in  $\mathcal{G}_3$  for the splitting given in (5.3) and using the implicit midpoint rule to solve the equation for  $f_b$ .

and we take as initial condition  $x_1(0) = 1 - e$ ,  $x_2(0) = x'_1(0) = 0$ ,  $x'_2(0) = [(1+e)/(1-e)]^{1/2}$ , which produces an orbit with eccentricity  $e$ . With the value  $e = 1/4$ , the orbit is determined numerically for 100 periods and the mean error in positions is computed. This problem is suited to integration by symplectic Runge–Kutta–Nyström methods [21], and moreover, modified potentials can also be used [4, 12]. In particular, Koseleff [11] designed an efficient self-adjoint fourth-order scheme, which can be used as the basic method:

$$(5.5) \quad S_h^{[4]} = \varphi_{h/6}^{[b]} \circ \varphi_{h/2}^{[a]} \circ \varphi_{2h/3, h^3/72}^{[b,c]} \circ \varphi_{h/2}^{[a]} \circ \varphi_{h/6}^{[b]}.$$

Here

$$(5.6) \quad \varphi_{\alpha h}^{[a]}(x_i) = x_i + \alpha h x'_i, \quad \varphi_{\beta h, \gamma h^3}^{[b,c]}(x'_i) = x'_i - x_i G(\beta + \hat{\gamma} G), \quad i = 1, 2,$$

with  $\hat{\gamma} = 4\gamma h$ ,  $G = h/(x_1^2 + x_2^2)^{3/2}$ , and  $\varphi_{\beta h}^{[b]}$  corresponds to taking  $\gamma = 0$  in the modified flow  $\varphi_{\beta h, \gamma h^3}^{[b,c]}$ . Notice that the cost of evaluating  $\varphi_{\beta h, \gamma h^3}^{[b,c]}$  instead of  $\varphi_{\beta h}^{[b]}$  increases only by one addition and one multiplication, so that it can be neglected.

We compare the new processed methods in  $\mathcal{G}_4$ , taking (5.5) as the basic integrator with the nonprocessed schemes  $M_{17}8$  and  $G_{33}10$  in  $\mathcal{G}_3$  and using as basic second-order symmetric integrator the well-known Störmer–Verlet/leapfrog method. Figure

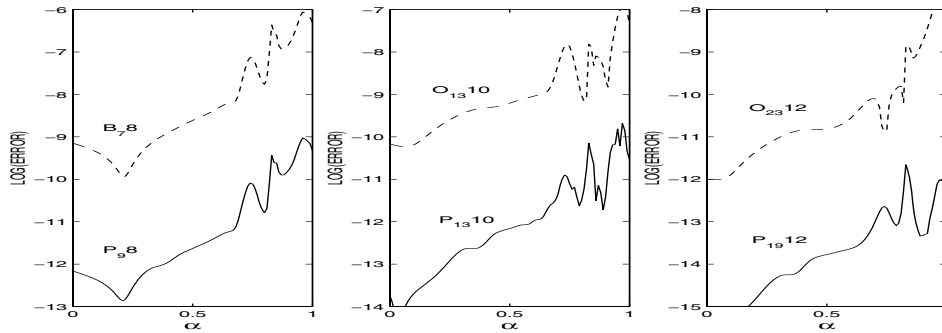


FIG. 7. Same as Figure 4 for the eighth-, tenth-, and twelfth-order methods in  $\mathcal{G}_4$ .

8 shows the average relative error in position versus the number of  $\varphi_{\beta h}^{[b]}$  evaluations (one evaluation of (5.5) is considered to involve two  $\varphi_{\beta h}^{[b]}$  evaluations due to the FSAL property). It is worth noticing the excellent performance of the processed methods  $P_{9,8}$  and  $P_{13,10}$ . In particular, for the step sizes considered in the figure  $P_{13,10}$  behaves as a twelfth-order integrator: in fact, one has to take a much smaller  $h$  to recover the asymptotic behavior expected for a tenth-order method. A similar pattern can be observed in the diagram corresponding to  $P_{9,8}$ .

This example clearly shows that in certain circumstances and depending on the particular structure of the system at hand, it can be advantageous to design an efficient fourth-order scheme  $\mathcal{S}_h^{[4]}$  and then build by composition higher-order integrators based on it. Obviously, the performance of the methods highly depend on the particular fourth-order basic scheme and a more detailed analysis of this could even improve the performance observed in Figure 8.

We must also remark that after 100 revolutions it is still possible to obtain an accuracy of 18 digits. However, the coefficients for the numerical methods in Table 7 are given with only 15 digits. This is so because the lowest-order condition is multiplied by  $h^5$ , which is also a small number. In other words, the comments of section 4.5 are valid here.

**6. Conclusions.** In this paper we have constructed numerical integrators for differential equations up to order twelve by composition of basic integrators and using the processing technique. Four families of integrators are considered: (i) fourth- and sixth-order composition methods for a system separable into two solvable parts or (ii) using as basic methods a first-order integrator and its adjoint; (iii) sixth-, eighth-, and tenth-order composition methods using second-order self-adjoint methods; and (iv) eighth-, tenth-, and twelfth-order composition methods using fourth-order self-adjoint methods.

Our starting point is a theoretical analysis of the processing technique for the cases considered here, which allows us to obtain in a relatively simple way the number and explicit form of the order conditions for the kernel. The analysis clearly shows that the processing technique is a very promising procedure to obtain efficient methods when implemented with constant time step. In that case, the cost of the postprocessor can be neglected even if the output has to be frequently computed since, if desired, it can be approximated using intermediate stages of the kernel.

Next, we have carried out a numerical search of the coefficients of different com-

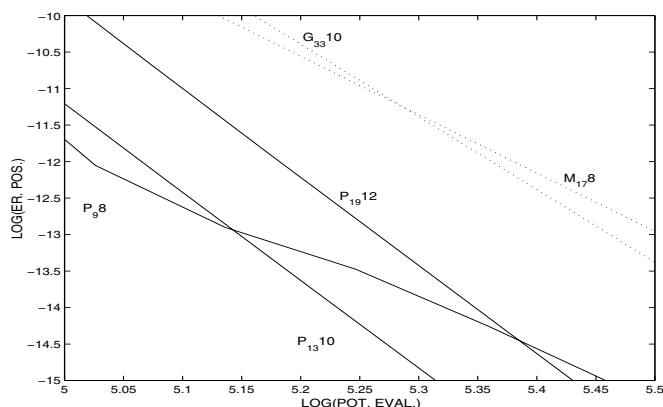


FIG. 8. Average error in position versus number of evaluations.  $P_{98}$ ,  $P_{1310}$ , and  $P_{1912}$  are processed methods in  $\mathcal{G}_4$  using (5.5) as the basic method (considered as two evaluations of the potential).  $M_{178}$  and  $G_{3310}$  are methods in  $\mathcal{G}_3$  using the Störmer–Verlet leapfrog method (one evaluation of the potential).

positions to get efficient kernels for the cases considered. Compositions with more stages than strictly required to solve the order conditions are considered for optimization purposes by following, mainly, the rule of thumb proposed in [15]. In this way, the higher accuracy obtained compensates the increase in the computational cost of the methods.

A large number of numerical experiments have been carried out to compare the performance of the new processed methods with other well-established nonprocessed methods, and some representative results are shown. We have considered different initial conditions, values of the parameters, basic methods, and splittings to better appreciate the performance of the methods.

We think that some comments about the choice and applicability of the different methods proposed here could be of interest for the potential user. If a given problem can be integrated using methods in  $\mathcal{G}_1$ , then it is clear that it can also be integrated with methods in  $\mathcal{G}_2$ ,  $\mathcal{G}_3$ , and  $\mathcal{G}_4$ , but for the same order, methods in  $\mathcal{G}_1$  and  $\mathcal{G}_2$  are more efficient than schemes in  $\mathcal{G}_3$  and  $\mathcal{G}_4$ . Thus, if one is interested in fourth- or sixth-order methods, it is, in general, preferable to consider methods in  $\mathcal{G}_1$  or  $\mathcal{G}_2$ ; for orders 8 or 10 we recommend methods in  $\mathcal{G}_3$ , and only for higher-order methods or when a very efficient fourth-order scheme is available do we recommend methods in  $\mathcal{G}_4$ . The same comments are valid for a problem which can be integrated in  $\mathcal{G}_2$  but not in  $\mathcal{G}_1$ .

Although the processing technique has become, during the past few years, a standard technique for constructing efficient high-order composition integrators, there are still some issues which need to be clarified.

- Which is the most appropriate objective function to minimize leading to the most efficient methods for a given class of problems?
- We have considered self-adjoint kernels with more stages than necessary to solve the effective order conditions. On the other hand, for nonsymmetric kernels such a number of stages would contain a larger number of free parameters. The question is, With such assumptions, could other interesting solutions not contemplated in this work exist?

- Could there exist other possible symmetries for the coefficients and different *rules of thumb* which would allow one to obtain more efficient methods?
- How can one build specific fourth-order methods for a given problem that when used as a basic scheme in  $\mathcal{G}_4$  would lead to very efficient higher-order composition methods?
- Would it be possible to construct more efficient composition methods for particular problems of physical relevance for which the Lie algebra involved presents a simpler structure than those analyzed in this work?

**Appendix A. Construction of specific postprocessors.** Next we present several composition and cheap postprocessors for some of the kernels collected in section 4. The resulting integrators provide the most efficient results on different test examples (analyzed in section 5).

With respect to composition postprocessors, in accordance with the comments in section 4.1, we take  $\pi_h^{(c)} = w_h^{[i]} \circ w_{-h}^{[i]}$ ,  $i = 2, 3, 4$ , for the composition postprocessor, with

$$(A.1) \quad w_h^{[2]} = \chi_{\gamma_{2r}h} \circ \chi_{\gamma_{2r-1}h}^* \circ \cdots \circ \chi_{\gamma_2h} \circ \chi_{\gamma_1h}^*,$$

$$(A.2) \quad w_h^{[3]} = \mathcal{S}_{\gamma_rh}^{[2]} \circ \cdots \circ \mathcal{S}_{\gamma_1h}^{[2]},$$

$$(A.3) \quad w_h^{[4]} = \mathcal{S}_{\gamma_rh}^{[4]} \circ \cdots \circ \mathcal{S}_{\gamma_1h}^{[4]}$$

and require that  $\pi_h^{(c)} \in \mathbb{P}_{q-1}$ , although in some cases a more accurate composition  $\pi_h^{(c)} \in \mathbb{P}_q$  is also built, particularly when the number of stages required is not too large. On the other hand, the cheap postprocessors are constructed by following the procedure outlined in section 3.2.

*Postprocessors in  $\mathcal{G}_1$  and  $\mathcal{G}_2$ .* Condition  $p_{1,1} = \sum_i \gamma_i = 0$  produces a composition  $\pi_h^{(c)} = w_h^{[2]} \circ w_{-h}^{[2]} \in \mathbb{P}_2$ , whereas 4 and 15 order conditions have to be satisfied if  $\pi_h^{(c)} \in \mathbb{P}_j$ ,  $j = 4, 6$ , respectively. In Table 8 we give coefficients for  $\pi_h^{(c)}$  in  $\mathbb{P}_4$  (P<sub>6</sub>4). Since the kernel P<sub>9</sub>6 of effective order 6 is the most efficient in practice, we present two possible postprocessors: one in  $\mathbb{P}_5$  ( $r = 4$ ) and a more accurate one in  $\mathbb{P}_6$  ( $r = 9$ ), which is recommended for use in both  $\mathcal{G}_1$  and  $\mathcal{G}_2$ . Finally, we provide a cheap postprocessor  $\hat{\pi}_h \in \mathbb{P}_3$  for P<sub>6</sub>4.

*Postprocessors in  $\mathcal{G}_3$ .* For a sixth-order processed integrator it is required that  $\pi_h \in \mathbb{P}_5$  and, according to Table 3, this is accomplished if  $p_{4,1} = -f_{5,2}$  and  $p_{1,1} = p_{3,1} = p_{5,1} = p_{5,2} = 0$ . In fact, with  $\pi_h^{(c)} = w_h^{[3]} \circ w_{-h}^{[3]}$ ,  $w_h^{[3]}$  given by (A.2) and satisfying  $\sum_{i=1}^r \gamma_i = 0$ , we need only to enforce  $p_{4,1} = -f_{5,2}$ . If, in addition,  $p_{6,1} = -f_{7,3}$  and  $p_{6,2} = -f_{7,4}$  hold, then  $\pi_h^{(c)} \in \mathbb{P}_6$ , whereas if condition  $\sum_{i=1}^r \gamma_i^3 = 0$  is satisfied, then  $\pi_h^{(c)} \in \mathbb{P}_7$ : this is, in fact, the composition postprocessor collected in Table 9 for P<sub>11</sub>6 and P<sub>13</sub>6, preserving time-symmetry up to  $\mathcal{O}(h^9)$ . Taking into account the conditions on  $p_{8,i}$ ,  $i = 1, \dots, 5$ , one gets  $\pi_h^{(c)} \in \mathbb{P}_8$ , and so on. For P<sub>13</sub>8 (which works in practice more efficiently than P<sub>19</sub>8) we provide a composition postprocessor in  $\mathbb{P}_8$  and a 24-stage  $\pi_h^{(c)} \in \mathbb{P}_9$  for P<sub>23</sub>10. One could get a postprocessor in  $\mathbb{P}_{10}$  for the tenth-order case, but this would require 11 new conditions to be satisfied by  $w_h^{[3]}$ , giving a 46-stage postprocessor. We also present the coefficients of cheap processors  $\hat{\pi}_h \in \mathbb{P}_{q-1}$ ,  $q = 6, 8$ , valid for P<sub>11</sub>6 and P<sub>13</sub>8, respectively. A similar  $\hat{\pi}_h$  could be obtained for P<sub>23</sub>10, but from Table 4, it would need at least 33 coefficients  $w_i$ .

TABLE 8  
 Postprocessor coefficients in (A.1) and (3.5) for some kernels in  $\mathcal{G}_1/\mathcal{G}_2$ .

$P_{64} : \pi_h^{(c)} \in \mathbb{P}_4$	$P_{64} : \hat{\pi}_h \in \mathbb{P}_3$	$P_{96} : \pi_h^{(c)} \in \mathbb{P}_5$
$\gamma_1 = 0$	$w_0 = 1 - 2(w_1 + w_5 + w_6 + w_7)$	$\gamma_1 = 0.01012220322650864$
$\gamma_2 = -(\gamma_3 + \dots + \gamma_6)$	$w_1 = 0.46640472356735$	$\gamma_2 = -0.2840923693556972$
$\gamma_3 = -0.1612056894758833$	$w_5 = -0.02125258839849$	$\gamma_3 = -0.0739474688028612$
$\gamma_4 = -0.2694936673582758$	$w_6 = -0.04899563905006$	$\gamma_4 = 0.00007025760293332$
$\gamma_5 = 0.2736158718483377$	$w_7 = 0.00811211574986$	$\gamma_5 = -0.02385471740106014$
$\gamma_6 = 0.1832420262145362$		$\gamma_6 = 0.08878308313058653$
		$\gamma_7 = -(\gamma_1 + \dots + \gamma_6)$
		$\gamma_8 = 0$
$P_{96} : \pi_h^{(c)} \in \mathbb{P}_6$		
$\gamma_1 = -0.2013206758948216$	$\gamma_2 = -0.1281761283096599$	$\gamma_3 = 0.0570146336015926$
$\gamma_4 = -0.0359500915398769$	$\gamma_5 = 0.0498478378426457$	$\gamma_6 = 0.1725721056066613$
$\gamma_7 = -0.0654901907171583$	$\gamma_8 = -0.0576378685707717$	$\gamma_9 = 0.1817517334193077$
$\gamma_{10} = 0.1672419181837143$	$\gamma_{11} = 0.2111126647112377$	$\gamma_{12} = -0.0733409487114027$
$\gamma_{13} = -0.1420023272628973$	$\gamma_{14} = 0.1215976276874689$	$\gamma_{15} = 0.0329874370062511$
$\gamma_{16} = -0.2361731625865831$	$\gamma_{17} = -(\gamma_1 + \dots + \gamma_{16})$	$\gamma_{18} = 0$

TABLE 9  
 Postprocessor coefficients in (A.2) and (3.5) for some kernels in  $\mathcal{G}_3$ .

$P_{116} : \pi_h^{(c)} \in \mathbb{P}_6$	$P_{116} : \hat{\pi}_h \in \mathbb{P}_5$	$P_{136} : \pi_h^{(c)} \in \mathbb{P}_6$
$\gamma_6 = -(\gamma_1 + \dots + \gamma_5)$	$w_0 = 1 - 2(w_1 + w_5 + w_6 + w_7)$	$\gamma_6 = 0.1$
$\gamma_5 = -0.20621953139126$	$w_1 = 0.35601475536028$	$\gamma_5 = 0.225080298761176$
$\gamma_4 = 0.23651387483203$	$w_5 = 0.12246549694690$	$\gamma_4 = 0.191244694511161$
$\gamma_3 = 0.09086982276241$	$w_6 = 0.00415291514453$	$\gamma_3 = -0.212763792194890$
$\gamma_2 = 0.24687306977659$	$w_7 = -0.20658995116781$	$\gamma_2 = -0.09660157306582295$
$\gamma_1 = -0.1$		$\gamma_1 = -(\gamma_2 + \dots + \gamma_6)$
$P_{138} : \pi_h^{(c)} \in \mathbb{P}_8$	$P_{138} : \hat{\pi}_h \in \mathbb{P}_7$	$P_{2310} : \pi_h^{(c)} \in \mathbb{P}_9$
$\gamma_{10} = -(\gamma_1 + \dots + \gamma_9)$	$w_0 = 1 - 2(w_1 + \dots + w_{12})$	$\gamma_{12} = -(\gamma_1 + \dots + \gamma_{11})$
$\gamma_9 = -0.008488123494574411$	$w_1 = -3.6976426586421067$	$\gamma_{11} = 0.4727142080578221$
$\gamma_8 = -0.337188967354338$	$w_2 = 1.0615669344875514$	$\gamma_{10} = 0.01344750613191108$
$\gamma_7 = 0.333987768164597$	$w_3 = 0.040377839731292050$	$\gamma_9 = -0.4637104712987078$
$\gamma_6 = 0.588351189003849$	$w_4 = 0.0830491660507623$	$\gamma_8 = -0.3045590922565247$
$\gamma_5 = 0.162324207599241$	$w_5 = -0.0221811460897851$	$\gamma_7 = -0.5382945821834320$
$\gamma_4 = -0.511744926116413$	$w_6 = -0.1398573630328631$	$\gamma_6 = -0.1899795533199732$
$\gamma_3 = 0.236885952363384$	$w_7 = -0.0074999124845547055$	$\gamma_5 = 0.1548256472553489$
$\gamma_2 = 0.598212975943381$	$w_8 = 0.21992320817724267$	$\gamma_4 = -0.3594148033156072$
$\gamma_1 = -0.543415765371656$	$w_9 = -0.21401705459232256$	$\gamma_3 = 0.3430345669677392$
	$w_{10} = -0.014339878804936956$	$\gamma_2 = 0.5334030283695922$
	$w_{11} = 0.09819025594252939$	$\gamma_1 = 0.009116042043427756$
	$w_{12} = 0.034452779507214946$	

Postprocessors in  $\mathcal{G}_4$ . In Table 10 we present accurate composition postprocessors  $\pi_h^{(c)} = w_h^{[4]} \circ w_{-h}^{[4]}$  belonging to  $\mathbb{P}_8$  and  $\mathbb{P}_{10}$  for  $P_{98}$  and  $P_{1310}$ , respectively, whereas the postprocessor for the twelfth-order method belongs to  $\mathbb{P}_{11}$ .

**Appendix B.** Here we illustrate the validity of the criteria adopted in section 4 for the choice of the particular kernels collected there. We fix our attention, in particular, on the compositions of effective order 6 in  $\mathcal{G}_2$  and effective order 8 in  $\mathcal{G}_3$ . In the first case we take the symmetric composition (4.1) with the minimum number of maps required to satisfy the effective order conditions ( $m = 5$ )  $\psi(m)$ , and then we introduce  $l \leq 5$  additional maps with free parameters  $\alpha_1, \dots, \alpha_l$  in the composition.



TABLE 10  
Postprocessor coefficients for kernels in  $\mathcal{G}_4$ .

$P_{98} : \pi_h^{(c)} \in \mathbb{P}_8$	$P_{1310} : \pi_h^{(c)} \in \mathbb{P}_{10}$	$P_{1912} : \pi_h^{(c)} \in \mathbb{P}_{11}$
$\gamma_5 = -(\gamma_1 + \dots + \gamma_4)$	$\gamma_7 = -(\gamma_1 + \dots + \gamma_6)$	$\gamma_8 = -(\gamma_1 + \dots + \gamma_7)$
$\gamma_4 = -0.2196648965658254$	$\gamma_6 = 0.2834714107596056$	$\gamma_7 = 0.2505061593209141$
$\gamma_3 = 0.2405373742563472$	$\gamma_5 = -0.2553426863586816$	$\gamma_6 = -0.01365235094631587$
$\gamma_2 = 0.1406336264566169$	$\gamma_4 = -0.03297486475329144$	$\gamma_5 = 0.1795942654148864$
$\gamma_1 = 0.1$	$\gamma_3 = 0.2298746411002190$	$\gamma_4 = -0.1860801553027685$
	$\gamma_2 = -0.2869384247718548$	$\gamma_3 = -0.2526234033672912$
	$\gamma_1 = -0.2110578773704694$	$\gamma_2 = 0.01093309142620025$
		$\gamma_1 = -0.1873261189973930$

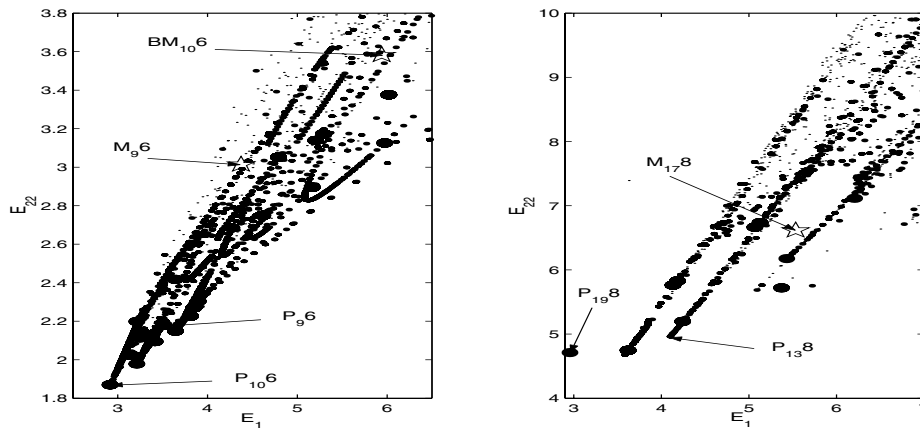


FIG. 9. Values of  $E_1$  and  $E_{22}$  for self-adjoint kernels using  $l$  free parameters when they are chosen randomly (small dots), applying the generalized rule of thumb (medium dots) and using McLachlan's rule of thumb (big dots). Only the results for the kernels of effective order 6 in  $\mathcal{G}_1/\mathcal{G}_2$  with  $l = 3, 4, 5$  (left) and the results for the kernels of effective order 8 in  $\mathcal{G}_3$  with  $l = 1, 2, 3$  (right) are shown.

For instance, when  $l = 4$  the kernel has the form

$$\psi_h^{[2]} = \chi_{\alpha_1 h} \circ \chi_{\alpha_2 h}^* \circ \chi_{\alpha_3 h} \circ \chi_{\alpha_4 h}^* \circ \psi(m) \circ \chi_{\alpha_4 h} \circ \chi_{\alpha_3 h}^* \circ \chi_{\alpha_2 h} \circ \chi_{\alpha_1 h}^*.$$

Next, for each  $l$  and given values of  $\alpha_1, \dots, \alpha_l$ , we solve the effective order conditions and compute the objective functions  $E_1$  and  $E_{22}$ . In Figure 9 (left) we collect the results obtained when (i) the free parameters  $\alpha_1, \dots, \alpha_l$  are chosen at random (small dots); (ii)  $\alpha_1 = \dots = \alpha_l = \alpha$  for different values of  $\alpha$  (medium dots); and (iii) applying McLachlan's rule of thumb, i.e.,  $\alpha_1 = \dots = \alpha_l = \alpha_{l+1}$  (big dots). For clarity, only the results with  $l = 3, 4, 5$  and a small fraction of points in (i) are shown. It is worth noticing that a random selection of  $\alpha_1, \dots, \alpha_l$  does not lead to smaller values of  $E_1$  and  $E_{22}$  than those obtained by applying the rule of thumb or its generalization. In fact, the medium dots lie on curves, whereas the big dots are located in the neighborhood of their minima. For comparison, we also include the values obtained with the nonprocessed 10-stage method  $BM_{106}$  (optimized as a method in  $\mathcal{G}_1$ ) given in [5] when  $\chi_h = \varphi_h^{[a]} \circ \varphi_h^{[b]}$  and the optimized method  $M_{96}$  with  $S_h^{[2]} = \chi_{h/2} \circ \chi_{h/2}^*$  (stars).

This figure should be interpreted with caution: from the obtained values of  $E_1$  and  $E_{22}$ ,  $M_{96}$  should be more efficient than  $BM_{106}$ , but in practice it is just the opposite.

In fact, the value of  $E_{21}$  (using a basis of  $\mathcal{L}$  in  $\mathcal{G}_1$ ) is smaller with  $\text{BM}_{106}$ . This example indicates some of the difficulties involved in the choice of the best objective functions for methods in  $\mathcal{G}_1$  and  $\mathcal{G}_2$ . Among the different solutions in the lower left corner of the figure we have selected two of them:  $\text{P}_{106}$  (providing the minimum of  $E_1$  and  $E_{22}$ ) and  $\text{P}_{96}$  (with the smallest value of  $E_{21}$ ). In practice, the kernel  $\text{P}_{96}$  shows the best performance.

In Figure 9 (right) we show the results obtained with the kernel of effective order 8 in  $\mathcal{G}_3$ . Now  $l = 1, 2, 3$  free parameters are considered. We also include the values given by the 17-stage method  $\text{M}_{178}$  and the 19-stage kernel  $\text{P}_{198}$  of Table 6.

**Acknowledgment.** The authors would like to thank the referees for their careful reading of the manuscript and their comments, which helped to improve this paper a great deal.

#### REFERENCES

- [1] S. BLANES, *High order numerical integrators for differential equations using composition and processing of low order methods*, Appl. Numer. Math., 37 (2001), pp. 289–306.
- [2] S. BLANES, F. CASAS, AND A. MURUA, *On the numerical integration of ordinary differential equations by processed methods*, SIAM J. Numer. Anal., 42 (2004), pp. 531–552.
- [3] S. BLANES, F. CASAS, AND J. ROS, *Symplectic integration with processing: A general study*, SIAM J. Sci. Comput., 21 (1999), pp. 711–727.
- [4] S. BLANES, F. CASAS, AND J. ROS, *High-order Runge–Kutta–Nyström geometric integrators with processing*, Appl. Numer. Math., 39 (2001), pp. 245–259.
- [5] S. BLANES AND P. C. MOAN, *Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods*, J. Comput. Appl. Math., 142 (2002), pp. 313–330.
- [6] J. BUTCHER, *The effective order of Runge–Kutta methods*, in Conference on the Numerical Solution of Differential Equations, Lecture Notes in Math. 109, Springer-Verlag, Berlin, 1969, pp. 133–139.
- [7] J. BUTCHER AND J. M. SANZ-SERNA, *The number of conditions for a Runge–Kutta method to have effective order  $p$* , Appl. Numer. Math., 22 (1996), pp. 103–111.
- [8] E. HAIRER, C. LUBICH, AND G. WANNER, *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer Ser. Comput. Math. 31, Springer-Verlag, Berlin, 2002.
- [9] S.-J. KANG AND M.-H. KIM, *Free Lie algebras, generalized Witt formula, and the denominator identity*, J. Algebra, 183 (1996), pp. 560–594.
- [10] W. KAHAN AND R. C. LI, *Composition constants for raising the order of unconventional schemes for ordinary differential equations*, Math. Comp., 66 (1997), pp. 1089–1099.
- [11] P. V. KOSELEFF, *Formal Calculus for Lie Methods in Hamiltonian Mechanics*, Ph.D. thesis, Lawrence Berkeley Laboratory LBID-2030, Tech. report UC-405, Berkeley, CA, 1994.
- [12] M. A. LÓPEZ-MARCOS, J. M. SANZ-SERNA, AND R. D. SKEEL, *Explicit symplectic integrators using Hessian-vector products*, SIAM J. Sci. Comput., 18 (1997), pp. 223–238.
- [13] R. I. MCLACHLAN, *On the numerical integration of ordinary differential equations by symmetric composition methods*, SIAM J. Sci. Comput., 16 (1995), pp. 151–168.
- [14] R. I. MCLACHLAN, *More on symplectic correctors*, in Integration Algorithms and Classical Mechanics, Fields Inst. Commun. 10, J. E. Marsden, G. W. Patrick, and W. F. Shadwick, eds., AMS, Providence, RI, 1996, pp. 141–149.
- [15] R. I. MCLACHLAN, *Families of high order composition methods*, Numer. Algorithms, 31 (2002), pp. 233–246.
- [16] R. I. MCLACHLAN AND R. QUISPÉL, *Splitting methods*, Acta Numer., 11 (2002), pp. 341–434.
- [17] H. MUNTJE-KAAS AND B. OWREN, *Computations in a free Lie algebra*, R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci., 357 (1999), pp. 957–981.
- [18] A. MURUA AND J. M. SANZ-SERNA, *Order conditions for numerical integrators obtained by composing simpler integrators*, R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci., 357 (1999), pp. 1079–1100.
- [19] P. J. OLVER, *Applications of Lie Groups to Differential Equations*, 2nd ed., Springer-Verlag, New York, 1993.

- [20] I. P. OMEL'YAN, I. M. MRYGLOD, AND R. FOLK, *Construction of high-order force-gradient algorithms for integration of motion in classical and quantum systems*, Phys. Rev. E, 66 (2002), 026701.
- [21] J. M. SANZ-SERNA AND M. P. CALVO, *Numerical Hamiltonian Problems*, Chapman and Hall, London, 1994.
- [22] M. SUZUKI, *General theory of fractal path integrals with applications to many-body and statistical physics*, J. Math. Phys., 32 (1991), pp. 400–407.
- [23] M. SUZUKI, *Quantum Monte Carlo methods and general decomposition theory of exponential operators and symplectic integrators*, Phys. A, 205 (1994), pp. 65–79.
- [24] V. S. VARADARAJAN, *Lie Algebras and Their Representations*, Springer-Verlag, New York, 1984.
- [25] J. WISDOM, M. HOLMAN, AND J. TOUMA, *Symplectic correctors*, in *Integration Algorithms and Classical Mechanics*, Fields Inst. Commun. 10, J. E. Marsden, G. W. Patrick, and W. F. Shadwick, eds., AMS, Providence, RI, 1996, pp. 217–244.
- [26] H. YOSHIDA, *Construction of higher order symplectic integrators*, Phys. Lett. A, 150 (1990), pp. 262–268.