

COMPOSITION METHODS FOR DIFFERENTIAL EQUATIONS WITH PROCESSING

S. BLANES[†], F. CASAS[†], AND A. MURUA[‡]

Abstract. We construct numerical integrators for differential equations up to order twelve 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 at virtually cost free.

Key words. Composition methods, geometric integrators, processing technique, cheap post-processor, initial value problems.

AMS subject classifications. 65L05; 65L70; 65D30

1. Introduction. A widely used technique in the geometric integration of ordinary differential equations 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 the basic scheme shares with the exact solution (if these properties are preserved by composition). In close connection with composition yields the splitting idea: if the differential equation can be split into solvable parts, by composition of the corresponding flows it is also possible to build a higher order integrator. These splitting methods are frequently used in celestial mechanics, quantum mechanics, molecular dynamics, accelerator physics and, in general, for numerically solving Hamiltonian systems, 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.

Given the ODE

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

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

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

a one-step numerical integrator for a time step h , $\psi_h : \mathbb{R}^D \rightarrow \mathbb{R}^D$, can be considered as a smooth family of maps with parameter h such that ψ_0 is the identity map. The

[†]Departament de Matemàtiques, Universitat Jaume I, 12071-Castellón, Spain (sblanes@mat.uji.es, Fernando.Casas@uji.es). The work of these authors has been partially supported by Fundació Caixa Castelló-Bancaixa. SB has also been supported by Ministerio de Ciencia y Tecnología (Spain) through a contract in the Pogramme Ramón y Cajal 2001.

[‡]Konputazio Zientziak eta A. A. saila, Informatika Fakultatea, EHU/UPV, Donostia/San Sebastián, Spain (ander@si.ehu.es).

integrator ψ_h is said to be of order q if $\psi_h = \varphi_h + \mathcal{O}(h^{q+1})$ as $h \rightarrow 0$, where φ_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 ψ_h with processing, i.e., algorithms of the form

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

Here ψ_h is referred to as the *kernel* and the map $\pi_h : \mathbb{R}^D \rightarrow \mathbb{R}^D$ as the *post-processor* or *corrector*. The method ψ_h is of *effective order* q if a post-processor π_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 pre-processor π_h^{-1} has only to be done at the beginning of the integration, then the kernel ψ_h acts once per step and finally π_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 π_h , so that ψ_h must fulfill a much reduced set of restrictions [1, 3, 7]. This allows to take kernels of effective order q involving far less evaluations than a conventional integrator of order q . As a result, by considering both the kernel and the post-processor as compositions of basic integrators, highly efficient processed methods have been proposed [25, 3, 1, 15].

To construct a composition method of a given order it is necessary to solve a system of non-linear polynomial equations in the coefficients (the order conditions), whose number and complexity grows very rapidly with the order. As a consequence, it is not a trivial task to find all possible solutions and eventually determine the optimal one. This problem is considerably simplified 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 find very efficient methods. As for the post-processor π_h , if it is constructed as a composition of basic integrators, $\pi_h^{(c)}$, then a large number of equations have to be satisfied by its coefficients. This results in post-processors that are more expensive to evaluate than the kernel ψ_h , so that their use in this form is confined to problems in which intermediate results are not frequently required; otherwise the overall efficiency of the methods is deteriorated.

To overcome this difficulty, in [2] a technique is developed which allows to obtain approximations to the post-processor at virtually cost free and without loss of accuracy. The key point is to replace π_h by a new integrator $\hat{\pi}_h \simeq \pi_h$ obtained from the intermediate stages in the computation of ψ_h . It is recommended to have a very accurate pre-processor π_h^{-1} but, on the contrary, the post-processor can safely be replaced by a cheap approximation satisfying only the order conditions.

In the analysis of composition methods it is useful the notion of adjoint of a given integrator ψ_h [21]. By definition, this is the method ψ_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, then $\psi_{-h} \circ \psi_h = \text{id}$. 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 reduced. The time-symmetry also implies that the map ψ_h inherits additional reversibility properties of the differential equation [13], a fact particularly important in geometric integration,

which aims to discretize the differential equation in such a way that qualitative features of the system are preserved. In the context of processed methods, if the kernel ψ_h is symmetric and the post-processor π_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 post-processors 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 the sequel.

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 exactly computed. Then one can form the following group \mathcal{G}_1 of schemes:

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

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 χ_h and its adjoint χ_h^* , namely

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

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 in both cases equivalent [13]. Methods in \mathcal{G}_2 are more general and can also be used if the system is split in more than two parts, i.e., $x' = f_a + f_b + f_c + \cdots$, just taking $\chi_h = \varphi_h^{[a]} \circ \varphi_h^{[b]} \circ \varphi_h^{[c]} \circ \cdots$ ($\chi_h^* = \cdots \circ \varphi_h^{[c]} \circ \varphi_h^{[b]} \circ \varphi_h^{[a]}$).

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 time-symmetric integrator, then

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

This family of integrators has also 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 time-symmetric integrator can be written in this form; the choice of χ_h is not unique, however).

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

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

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 α_i . More specifically,

1. In \mathcal{G}_1 , if either $\beta_{2s} = 0$, $\beta_{2s-i} = \beta_i$ or $\beta_1 = 0$, $\beta_{2s+1-i} = \beta_{i+1}$. Now the computations done for the last map can be reused for the first map in the following step (this feature is known as FSAL (first same as last) property). Then both maps $\varphi_{\beta;h}^{[a]}$ and $\varphi_{\beta;h}^{[b]}$ 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 $\hat{\psi}_h$, ψ_h and π_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., $\hat{\psi}_h$, ψ_h and π_h are formally the exact 1-flows of the vector fields \hat{F}_h , F_h and P_h , respectively.

If the kernel ψ_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 explicitly \mathcal{L} 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 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 $h^kF_k \in \mathcal{L}_k$ 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. [2]. *An integrator ψ_h has effective order $p \geq q$ if and only if there exist vector fields P_1, \dots, P_{q-1} such that*

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

holds, 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 to \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

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

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

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

THEOREM 2. [2]. *Assume that the chosen basis of each \mathcal{L}_k is such that (2.7) holds. The kernel ψ_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*

$$\begin{aligned} f_{k,i} &= -r_{k,i}, & 1 \leq i \leq l_k := n_k - n_{k-1}^*, & \quad 2 \leq k \leq q, & (2.8) \\ p_{k-1,i} &= -f_{k, l_k + i} - r_{k, l_k + i}, & 1 \leq i \leq n_{k-1}^*, & \quad 2 \leq k \leq q. & (2.9) \end{aligned}$$

If ψ_h is time-symmetric 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 ψ_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 α_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 α .

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 α 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 ψ_h is time-symmetric 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 2.1 for non-symmetric, $s(q)$, and symmetric, $\hat{s}(q)$, kernels.

For a kernel of effective order q (i.e., satisfying eqs. (2.8) for $k \leq q$ but not for $k = q + 1$) one can in principle determine a post-processor such that equations (2.9) hold also for all $k > q$. We refer to that post-processor 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 post-processor 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 ψ_h is of effective order q and the post-processor $\pi_h \in \mathbb{P}_{q-1}$. Note that optimality of a post-processor depends on the chosen basis in each \mathcal{L}_k (see [2] for more details).

TABLE 2.1

Total number of effective order conditions for non-symmetric ($s(q)$) 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 two, 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 2nd order leapfrog scheme. On the other hand, since a general post-processor 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 $p_{1,1} = p_{1,2}$, then the effective order of χ_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 post-processor in \mathcal{G}_1 can be expressed as a composition in \mathcal{G}_2 , the effective order conditions being 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. 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 the 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]

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

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 assuming instead $n_k \sim \rho^k$, a rough estimation of the value of r_q is obtained as follows:

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

and thus *the cost saving of the processing technique increases when ρ 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 2.1 can be obtained through the respective functions $P(T)$ and equation (2.10).

2.5. Order conditions for methods in \mathcal{G}_3 . For the sake of illustration next we 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 post-processor coefficients when the basis of \mathcal{L}_k , $k \leq 11$, is chosen according to (2.7) (see Table 2.2 for the particular basis taken).

TABLE 2.2
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 $F_h = hF_1 + h^3F_3 + h^4F_4 + \cdots$ 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 symmetric kernel, a straightforward computation shows that

$$R_2 = R_3 = R_4 = R_5 = R_6 = 0, \quad R_7 = f_{3,1}p_{4,1}E_{7,2}, \quad R_8 = 0$$

$$R_9 = f_{5,1}p_{4,1}(E_{9,2} - E_{9,4}) + (f_{5,2}p_{4,1} + \frac{1}{2}p_{4,1}^2 + f_{3,1}p_{6,2})E_{9,3} - p_{4,1}(f_{5,2} + \frac{1}{2}p_{4,1})E_{9,6}$$

and similar (but larger) expressions for R_{10} , R_{11} . Equations (2.8)-(2.9) finally lead, after rearranging and simplifying terms, to the order conditions collected in Table 2.3.

TABLE 2.3

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

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

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 pre- and post-processor π_h to ensure that $\hat{\psi}_h$ has the required order. The kernel fixes the accuracy it is possible to reach. This can be attained with the optimal post-processor, defined as the exact 1-flow of an infinite series of vector fields P_h . In practice, one must look for a pre- and post-processor 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 post-processors, $\pi_h^{(c)}$. Since the kernel is a composition of basic methods in \mathcal{G}_l , it seems appropriate to approximate the optimal map π_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 equations (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 post-processor $\pi_h^{(c)} \in \mathbb{P}_{q-1}$. 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 pre-processor is computed only once, it makes sense using $(\pi_h^{(c)})^{-1}$ as an accurate approximation to π_h^{-1} , whereas $\pi_h^{(c)}$ should be used if intermediate output is not frequently required.

3.2. Cheap post-processors, $\hat{\pi}_h$. 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 post-processor as cheap to compute as possible. Usually $\hat{\pi}_h$ will be a less accurate approximation to π_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 π_h by reusing intermediate calculations obtained when evaluating the kernel ψ_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

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

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

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

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

2. For \mathcal{G}_2

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

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

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

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 post-processor (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 post-processor. 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

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

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

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

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

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 3.1 [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 α_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.,

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

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 3.1

Dimensions m_k of \mathcal{A}_k for each class of integrators. If the kernel is time-symmetric, 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	4	8	16	32	64	128	256
\mathcal{G}_2	1	2	4	8	16	32	64	128
\mathcal{G}_3	1	1	2	3	5	8	13	21
\mathcal{G}_4	1	1	1	1	2	3	5	7

For symmetric 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 equations (3.7) are 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

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

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 minimise the difference with the optimal post-processor at higher orders.

4. New processed methods. In this section we construct time-symmetric 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 post-processor, built according to the procedure of section 3.

4.1. Construction of the kernel. For building a specific kernel ψ_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 non-existence 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 α which solve the equations $N_{k,i}(\alpha) = 0$. For this reason one usually considers time-symmetric 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

$$\psi_h^{[2]} = \chi_{\alpha_1 h} \circ \chi_{\alpha_2 h}^* \circ \cdots \circ \chi_{\alpha_2 h} \circ \chi_{\alpha_1 h}^* \quad (4.1)$$

$$\psi_h^{[3]} = \mathcal{S}_{\alpha_1 h}^{[2]} \circ \cdots \circ \mathcal{S}_{\alpha_{m-1} h}^{[2]} \circ \mathcal{S}_{\alpha_m h}^{[2]} \circ \mathcal{S}_{\alpha_{m-1} h}^{[2]} \circ \cdots \circ \mathcal{S}_{\alpha_1 h}^{[2]} \quad (4.2)$$

$$\psi_h^{[4]} = \mathcal{S}_{\alpha_1 h}^{[4]} \circ \cdots \circ \mathcal{S}_{\alpha_{m-1} h}^{[4]} \circ \mathcal{S}_{\alpha_m h}^{[4]} \circ \mathcal{S}_{\alpha_{m-1} h}^{[4]} \circ \cdots \circ \mathcal{S}_{\alpha_1 h}^{[4]}, \quad (4.3)$$

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 $l_{q+1} = n_{q+1} - n_q^*$ non-correctable 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 α_i to be minimized, aimed to measure 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:

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

where $\bar{m} = 2m$ or $2m-1$ (depending on the set \mathcal{G}_l considered), $\alpha_{\bar{m}+1-i} = \alpha_i$, $\mathbf{f}_{q+1} = (f_{q+1,1}, \dots, f_{q+1,l_{q+1}})$ 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 non-processed 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] to 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 α_{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 six 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 seems 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 α . 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 has also 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 election of \mathbf{d} , whereas E_1 is always non-vanishing).

In view of the number of effective order conditions to be satisfied by symmetric kernels in \mathcal{G}_l and collected in Table 2.1, 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} -stages symmetric kernel.

4.2. Kernels in \mathcal{G}_1 and \mathcal{G}_2 up to effective order 6. Several 4th and 6th-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 4.1 (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 involving up to $m = 10$. The most efficient methods found have $m = 9$ and $m = 10$ and their coefficients are collected in Table 4.1 (second and third column). 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 4.1

Coefficients in (4.1) for symmetric kernels in \mathcal{G}_1 and \mathcal{G}_2 . These kernels can be used with the corresponding post-processors of Table A.1.

P_{64}	P_{96}	P_{106}
$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6$
$\alpha_4 = 0.1341940158122142$	$\alpha_5 = 0.1106570871853300$	$\alpha_6 = 0.1008383784517379$
$\alpha_5 = -\alpha_4 - 0.18$	$\alpha_6 = -0.2854111127287940$	$\alpha_7 = 0.1008384231345842$
$\alpha_6 = 1/2 - (\alpha_1 + \dots + \alpha_5)$	$\alpha_7 = 0.2138498496192465$	$\alpha_8 = -0.2387378577407101$
$\alpha_{13-i} = \alpha_i, \quad i = 1, \dots, 6$	$\alpha_8 = -0.3402583791791715$	$\alpha_9 = -0.2387378757998321$
	$\alpha_9 = 1/2 - (\alpha_1 + \dots + \alpha_8)$	$\alpha_{10} = 1/2 - (\alpha_1 + \dots + \alpha_9)$
	$\alpha_{19-i} = \alpha_i, \quad i = i, \dots, 9$	$\alpha_{21-i} = \alpha_i, \quad i = i, \dots, 10$

4.3. Kernels in \mathcal{G}_3 up to effective order 10. Non-processed 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 $(2m - 1)$ -stages 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 non-correctable 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 to take the 11 stages kernel given in [15] (P_{116}), and the 13 stages kernel (P_{136}) which, in addition, satisfies $f_{7,2} = 0$ (see Table 4.2).

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 $l = 2$ (P₁₃₈) and the other with $l = 5$ (P₁₉₈). The coefficients are collected in Table 4.2. 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 P₁₉₁₀ and P₂₃₁₀ whose coefficients are collected in Table 4.2 have been selected.

TABLE 4.2

Coefficients in (4.2) for symmetric kernels in \mathcal{G}_3 . They have can be used with the respective post-processors collected in Table A.2.

P ₁₁₆	P ₁₃₆	P ₁₉₈
$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$	$\alpha_1 = \dots = \alpha_6$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$
$\alpha_4 = 0.1705768865009222157$	$\alpha_4 = 0.125696288720106$	$\alpha_6 = 0.09155941827296$
$\alpha_5 = -0.423366140892658048$	$\alpha_5 = 0.148070660114965$	$\alpha_7 = 0.36968952549113$
$\alpha_6 = 1 - 2(\alpha_1 + \dots + \alpha_5)$	$\alpha_6 = -0.350856370823828$	$\alpha_8 = 0.06866857653282$
$\alpha_{12-i} = \alpha_i, \quad i = 1, \dots, 5$	$\alpha_7 = 1 - 2(\alpha_1 + \dots + \alpha_6)$	$\alpha_9 = -0.28931413259236$
	$\alpha_{14-i} = \alpha_i, \quad i = 1, \dots, 7$	$\alpha_{10} = 1 - 2(\alpha_1 + \dots + \alpha_9)$
		$\alpha_{20-i} = \alpha_i, \quad i = 1, \dots, 9$
P ₁₃₈	P ₂₃₁₀	P ₁₉₁₀
$\alpha_1 = \alpha_2$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5$	$\alpha_1 = \alpha_2 = \alpha_3$
$\alpha_2 = 0.168$	$\alpha_5 = 0.121657748919383$	$\alpha_3 = 0.16176042393895$
$\alpha_3 = 0.585550530805562$	$\alpha_6 = -0.511318780154828$	$\alpha_4 = -0.71963383963697$
$\alpha_4 = -0.460090457516872$	$\alpha_7 = -0.172858614884985$	$\alpha_4 = -0.71963383963697$
$\alpha_5 = 0.172863148729731$	$\alpha_8 = 0.123016258833066$	$\alpha_6 = 0.59733925980951$
$\alpha_6 = 0.179664539695039$	$\alpha_9 = 0.441503951671565$	$\alpha_7 = 0.082347969317011$
$\alpha_7 = 1 - 2(\alpha_1 + \dots + \alpha_6)$	$\alpha_{10} = -0.327071324165477$	$\alpha_8 = -0.43345109677776$
$\alpha_{14-i} = \alpha_i, \quad i = 1, \dots, 7$	$\alpha_{11} = 0.070952700957766$	$\alpha_9 = 0.10313406454059$
	$\alpha_{12} = 1 - 2(\alpha_1 + \dots + \alpha_{11})$	$\alpha_{10} = 1 - 2(\alpha_1 + \dots + \alpha_9)$
	$\alpha_{24-i} = \alpha_i, \quad i = 1, \dots, 11$	$\alpha_{20-i} = \alpha_i, \quad i = 1, \dots, 9$

4.4. Kernels in \mathcal{G}_4 up to effective order 12. Next we consider the symmetric $(2m - 1)$ -stages 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 follow exactly 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 non-correctable leading error term $f_{9,1}$ are obtained [1]. The coefficients of a 9 stages kernel (P₉₈) are collected in Table 4.3. 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 P₁₃₁₀ with coefficients given in Table 4.3 is finally selected.

Twelfth-order. If $m \geq 6$ (at least 11 stages) a 12th-order integrator can be obtained. Kernels with up to $l = 6$ free parameters have been analyzed. The coefficients for the composition with 19 stages, P₁₉₁₂, are collected in Table 4.3.

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

TABLE 4.3

Coefficients in (4.3) for symmetric kernels in \mathcal{G}_4 . The corresponding processed integrator is formed with the post-processors of Table A.3.

P ₉ 8	P ₁₃ 10	P ₁₉ 12
$\alpha_1 = \alpha_2 = \alpha_3$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$	$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \alpha_5$
$\alpha_3 = 0.2233380451446240$	$\alpha_4 = 0.1578763989460225$	$\alpha_5 = 0.1008183703667023$
$\alpha_4 = -0.3739874130116841$	$\alpha_5 = -0.3010347145730912$	$\alpha_6 = 0.2659998884940344$
$\alpha_5 = 1 - 2(\alpha_1 + \dots + \alpha_4)$	$\alpha_6 = 0.3577728201055250$	$\alpha_7 = -0.2142408955837595$
$\alpha_{10-i} = \alpha_i, \quad i = 1, \dots, 4$	$\alpha_7 = 1 - 2(\alpha_1 + \dots + \alpha_6)$	$\alpha_8 = -0.2968630395353906$
	$\alpha_{14-i} = \alpha_i, \quad i = 1, \dots, 6$	$\alpha_9 = 0.08741046298860494$
		$\alpha_{10} = 1 - 2(\alpha_1 + \dots + \alpha_9)$
		$\alpha_{20-i} = \alpha_i, \quad i = 1, \dots, 9$

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 digits 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 symmetric, the asymptotic expansion of the error only contains 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 round off) and the error in the lowest order conditions $N_{3,i}(\alpha) = 0$ and $N_{5,i}(\alpha) = 0$ are approximately h^310^{-15} and h^510^{-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 P₂₃10 in \mathcal{G}_3 are desired with higher accuracy. We show how to increase the actual precision from h^310^{-15} to h^510^{-15} just by adding some extra digits to one of the coefficients α_j .

First we consider the lowest effective order conditions $f_{1,1} = \sum_i \alpha_i = 1$, $N_{3,1}(\alpha) = f_{3,1} = \sum_i \alpha_i^3 = 0$. Let us fix all values of α_i from Table 4.2 except one, say α_{11} (as well as α_{12} to ensure that condition $f_{1,1} = 1$ is exactly satisfied). Next, solve equation $N_{3,1}(\alpha) = 0$ 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 $h^310^{-25} + h^510^{-15} \sim h^510^{-15}$. This process can also be used to increase the accuracy to h^710^{-15} just by taking two parameters to solve $N_{3,1}(\alpha) = 0$ and $N_{5,1}(\alpha) = f_{5,1} = 0$.

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 initial conditions. It may depend 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 and initial conditions. 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):

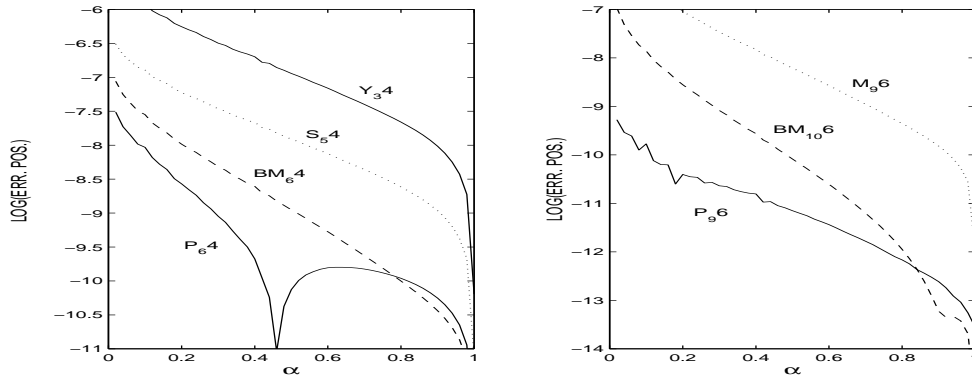


FIG. 5.1. Average relative error in position for the methods in \mathcal{G}_1 (\mathcal{G}_2) and for different initial conditions: $(u_0, v_0) = (\alpha, 2\alpha)$. The methods Y_{34} , S_{54} and M_{96} , belonging to \mathcal{G}_3 , are included as reference since they are frequently used.

- **Order 4.**

$\mathcal{G}_1, \mathcal{G}_2$ (BM_{64}): the 6 stages method (optimized in \mathcal{G}_1) given in [5];

\mathcal{G}_3 (Y_{34}): the well known 3 stages method of [26] (only used as a reference);

\mathcal{G}_3 (S_{54}): the 5 stages method of [22].

- **Order 6.**

$\mathcal{G}_1, \mathcal{G}_2$ (BM_{106}): the 10 stages method (optimized in \mathcal{G}_1) given in [5];

\mathcal{G}_3 (M_{96}): the optimized 9 stages method ($S, m = 9$) of [13].

- **Order 8.**

\mathcal{G}_3 (M_{178}): the 17 stages method ($S, m = 17$) of [13];

\mathcal{G}_4 (B_78): the 7 stages method of [1].

- **Order 10.**

\mathcal{G}_3 (R_{3310}): the 33 stages method (s33odr10c) of [10];

\mathcal{G}_3 (G_{3310}): the 33 stages method of [8].

\mathcal{G}_4 (O_{1310}): the 13 stages method of [20].

- **Order 12.**

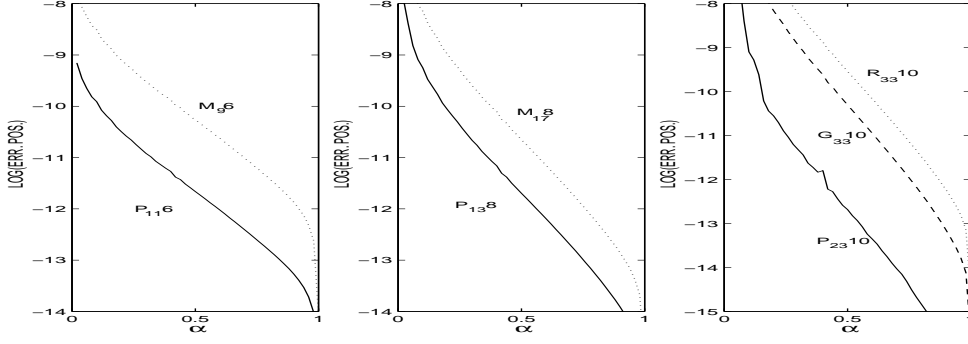
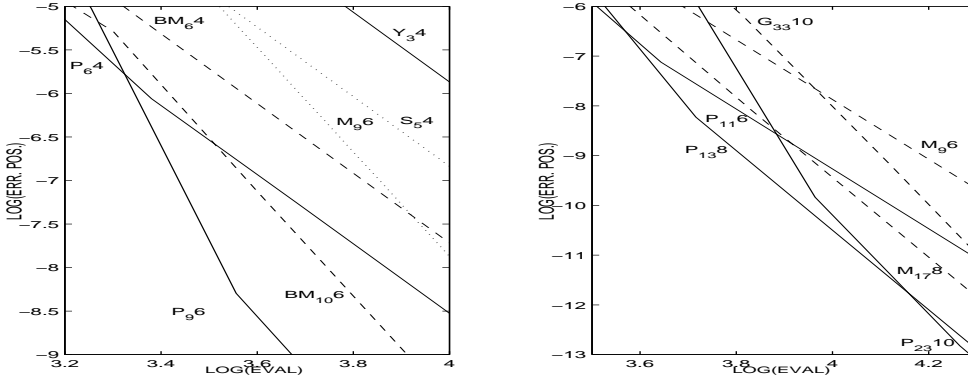
\mathcal{G}_4 (O_{2312}): the 23 stages method of [20].

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 pre-processor $(\pi_h^{(c)})^{-1}$ available. For post-processing we take $\pi_h^{(c)}$, although very similar results are obtained when the alternative cheap post-processor $\hat{\pi}_h$ is used instead, in agreement with the results exposed in [2].

Example 1. We consider the Lotka-Volterra system

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

which has as first integral $I(u, v) = \ln(uv^2) - (u + v)$. We split the vector field as $f = f_a + f_b$, with $f_a = (u(v - 2), 0)$ and $f_b = (0, v(1 - u))$. 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

FIG. 5.2. Same as figure 5.1 but for methods in \mathcal{G}_3 .FIG. 5.3. Average error in phase space versus the number of evaluations for the methods of Fig. 5.1 and 5.2 taking different time steps and $\alpha = 1/3$.

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 evaluations and (ii) the relative performance between them does not change qualitatively when h is reduced. Figure 5.1 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 symmetric second order integrator. The corresponding results are shown in Figure 5.2. Again, the processed methods are clearly superior. When the symmetric second order implicit midpoint method is taken as $\mathcal{S}_h^{[2]}$, we have observed that the results of the schemes are more sensitive to initial conditions, but their relative performance does not change considerably.

Finally, to compare the behavior of methods of different orders we plot efficiency diagrams. For this test we fix α to a representative value of the performance of the methods: in view of Figures 5.1 and 5.2, we take $\alpha = 1/3$. Figure 5.3 shows the average relative error in phase space as a function of the number of evaluations for different time steps. The high efficiency achieved by the processed methods P_{96} and P_{138} is apparent.

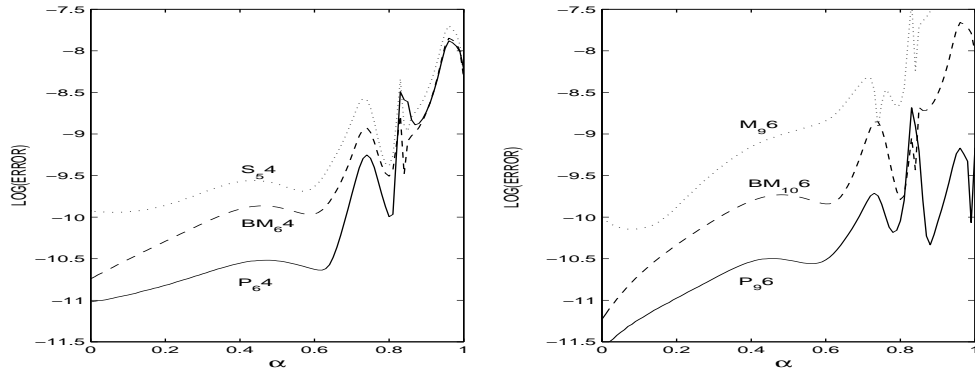


FIG. 5.4. Error in position for the ABC-flow at the same computational cost for the fourth- (left) and sixth-order methods (right) in \mathcal{G}_2 taking different values of the parameter $A = \alpha$.

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

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

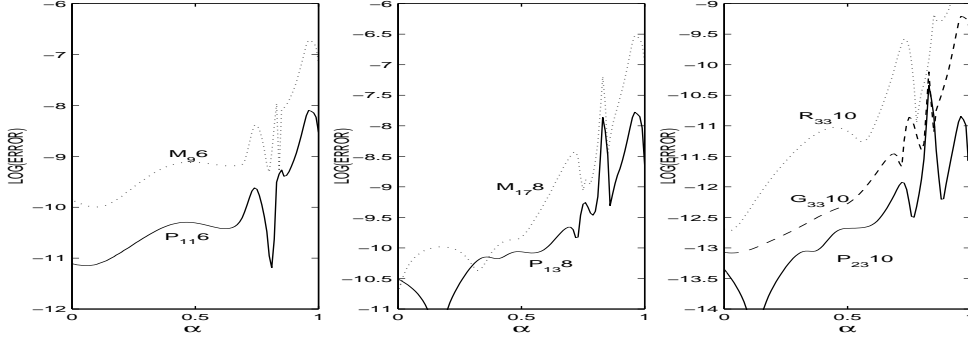
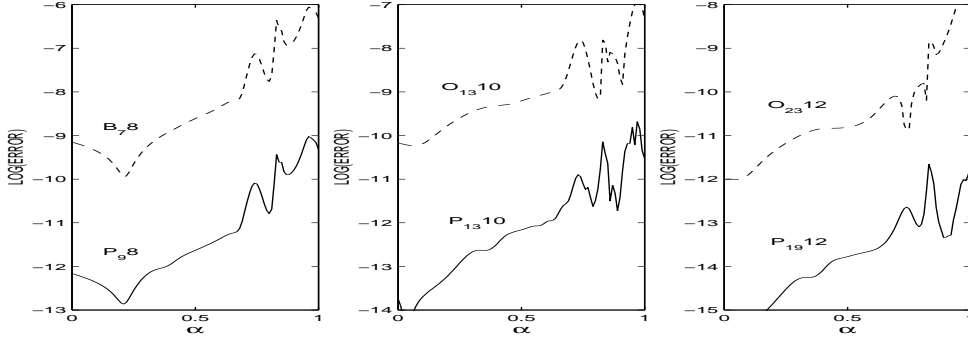
and whose vector field f is separable in three solvable parts, namely

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

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 α -dependent vector field can be used to test integrators in \mathcal{G}_l , $l = 2, 3, 4$. We choose the following 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 stages 4th-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 time-symmetric 4th-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.

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 5.4 shows the results obtained with the fourth- and sixth-order methods in \mathcal{G}_2 , whereas Figure 5.5 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 change. However, we have observed a superiority of the processed integrators which, in average, is well represented by Figures 5.4 and 5.5. Finally, Figure 5.6 shows the results obtained by the methods in \mathcal{G}_4 . 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

FIG. 5.5. Same as Fig. 5.4 for the sixth- eighth- and tenth-order methods in \mathcal{G}_3 .FIG. 5.6. Same as Fig. 5.4 for the eighth-, tenth- and twelfth-order methods in \mathcal{G}_4 .

written as

$$x_i'' = -\frac{x_i}{(x_1^2 + x_2^2)^{3/2}}, \quad i = 1, 2 \quad (5.3)$$

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 suitable to be integrated with symplectic Runge–Kutta–Nyström methods [21] and also modified potentials can be used [4, 12]. In particular, Koseleff [11] designed an efficient time-symmetric 4th-order scheme, which can be used as the basic method:

$$\mathcal{S}_h^{[4]} = \varphi_{h/6}^{[b]} \circ \varphi_{h/2}^{[a]} \circ \varphi_{2h/3, h^3/72}^{[b,c]} \circ \varphi_{h/2}^{[a]} \circ \varphi_{h/6}^{[b]}. \quad (5.4)$$

Here

$$\varphi_{\alpha h}^{[a]}(x_i) = x_i + \alpha h x_i', \quad \varphi_{\beta h, \gamma h^3}^{[b,c]}(x_i') = x_i' - x_i G(\beta + \hat{\gamma} G), \quad i = 1, 2 \quad (5.5)$$

with $\hat{\gamma} = 4\gamma h$, $G = h/(x_1^2 + x_2^2)^{3/2}$ and $\varphi_{\beta h}^{[b]}$ corresponds to take $\gamma = 0$ 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 in one addition and one multiplication, so that it can be neglected.

We compare the new processed methods in \mathcal{G}_4 taking (5.4) as the basic integrator with the non processed schemes $M_{17,8}$ and $G_{33,10}$ in \mathcal{G}_3 using as basic second order

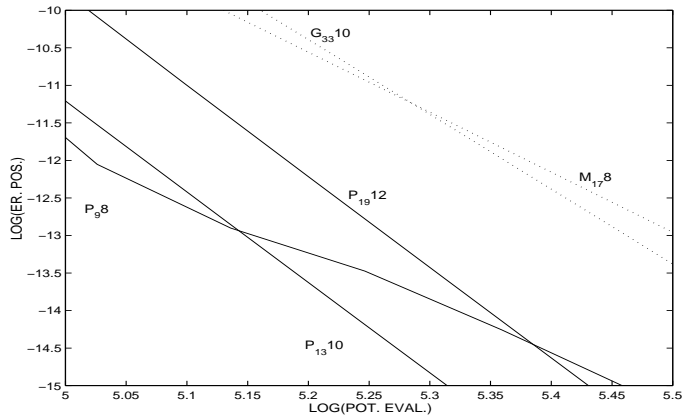


FIG. 5.7. Average error in position versus number of evaluations. $P_{9,8}$, $P_{13,10}$ and $P_{19,12}$ are processed methods in \mathcal{G}_4 using (5.4) as the basic method (considered as two evaluations of the potential). $M_{17,8}$ and $G_{33,10}$ are methods in \mathcal{G}_3 using the Störmet/Verlet/Leap-Frog method (one evaluations of the potential).

symmetric integrator the well known Störmet/Verlet/leap-frog method. Figure 5.7 shows the average relative error in position versus the number of $\varphi_{\beta h}^{[b]}$ evaluations (one evaluation of (5.4) 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, $P_{13,10}$ behaves for the step sizes considered in the figure as a 12th-order integrator: in fact, one has to take a much smaller h to recover the asymptotic behavior expected for a 10th-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 4th-order scheme $\mathcal{S}_h^{[4]}$ and then to 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 5.7.

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 4.3 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.

Appendix A. Construction of specific post-processors.

Next we present several composition and cheap post-processors 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 post-processors, 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 post-processor,

with

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

$$w_h^{[3]} = \mathcal{S}_{\gamma_{2r}h}^{[2]} \circ \cdots \circ \mathcal{S}_{\gamma_1h}^{[2]} \quad (\text{A.2})$$

$$w_h^{[4]} = \mathcal{S}_{\gamma_{2r}h}^{[4]} \circ \cdots \circ \mathcal{S}_{\gamma_1h}^{[4]} \quad (\text{A.3})$$

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 post-processors are constructed by following the procedure outlined in section 3.2.

Post-processors 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 A.1 we give coefficients for $\pi_h^{(c)}$ in \mathbb{P}_4 (P₆₄). Since the kernel P₉₆ of effective order 6 is the most efficient in practice, we present two possible post-processors: one in \mathbb{P}_5 ($r = 4$) and a more accurate one in \mathbb{P}_6 ($r = 9$), which is recommended for use both in \mathcal{G}_1 and \mathcal{G}_2 . Finally, we provide a cheap post-processor $\hat{\pi}_h \in \mathbb{P}_3$ for P₆₄.

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

Post-processors in \mathcal{G}_3 . For a 6th-order processed integrator it is required that $\pi_h \in \mathbb{P}_5$ and, according to Table 2.3, 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 only need 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 post-processor collected in Table A.2 for P₁₁₆ and P₁₃₆, 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 P₁₃₈ (which works in practice more efficiently than P₁₉₈) we provide a composition post-processor in \mathbb{P}_8 and a 24 stages $\pi_h^{(c)} \in \mathbb{P}_9$ for P₂₃₁₀. One could get a post-processor in \mathbb{P}_{10} for the 10th-order case, but this would require 11 new conditions to be satisfied by $w_h^{[3]}$, giving a 46 stages post-processor. We also present the coefficients of cheap

processors $\hat{\pi}_h \in \mathbb{P}_{q-1}$, $q = 6, 8$, valid for P₁₁₆ and P₁₃₈, respectively. A similar $\hat{\pi}_h$ could be obtained for P₂₃₁₀ but, from Table 3.1, it would need at least 33 coefficients w_i .

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

Post-processors in \mathcal{G}_4 . In Table A.3 we present accurate composition post-processors $\pi_h^{(c)} = w_h^{[4]} \circ w_{-h}^{[4]}$ belonging to \mathbb{P}_8 and \mathbb{P}_{10} for P₉₈ and P₁₃₁₀, respectively, whereas the post-processor for the 12th-order method belongs to \mathbb{P}_{11} .

TABLE A.3
Post-processor 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 : $\pi_h^{(c)} \in \mathbb{P}_{11}$
$\gamma_1 = -(\gamma_2 + \dots + \gamma_5)$	$\gamma_1 = -(\gamma_2 + \dots + \gamma_7)$	$\gamma_1 = -(\gamma_2 + \dots + \gamma_8)$
$\gamma_2 = -0.2196648965658254$	$\gamma_2 = 0.2834714107596056$	$\gamma_2 = 0.2505061593209141$
$\gamma_3 = 0.2405373742563472$	$\gamma_3 = -0.2553426863586816$	$\gamma_3 = -0.01365235094631587$
$\gamma_4 = 0.1406336264566169$	$\gamma_4 = -0.03297486475329144$	$\gamma_4 = 0.1795942654148864$
$\gamma_5 = 0.1$	$\gamma_5 = 0.2298746411002190$	$\gamma_5 = -0.1860801553027685$
	$\gamma_6 = -0.2869384247718548$	$\gamma_6 = -0.2526234033672912$
	$\gamma_7 = -0.2110578773704694$	$\gamma_7 = 0.01093309142620025$
		$\gamma_8 = -0.1873261189973930$

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 introduce $l \leq 5$ additional maps with free parameters $\alpha_1, \dots, \alpha_l$ in the composition. For instance, when $l = 4$ the kernel has the form

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

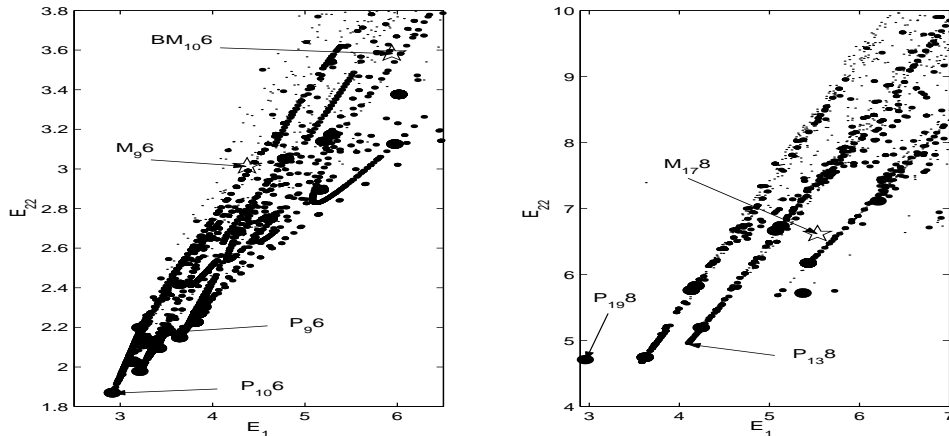


FIG. B.1. Values of E_1 and E_{22} for symmetric kernels using l free parameters when they are chosen randomly (small dots), applying the generalized rule of thumb (medium dots) and using the rule of thumb (big dots). Only the results for the kernels of effective order six in $\mathcal{G}_1/\mathcal{G}_2$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

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 B.1a 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 α (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 non-processed 10 stages 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 BM_{106} . This example indicates some of the difficulties involved in the election 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: P_{106} (providing the minimum of E_1 and E_{22}) and P_{96} (with the smallest value of E_{21}). In practice, the kernel P_{96} shows the best performance.

In Figure B.1b 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 stages method M_{178} and the 19 stages kernel P_{198} of Table 4.2.

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. Accepted for publication in *SIAM J. Numer. Anal.* Preprint available at: <http://www.focm.net/gi/gips/>
- [3] S. Blanes, F. Casas, and J. Ros, Symplectic integrators 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., Vol. 109, Springer, 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*, Vol. 10, J.E. Marsden, G.W. Patrick, and W.F. Shadwick, eds., American Mathematical Society, Providence, R.I. (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. Quispel, Splitting methods, *Acta Numerica*, 11 (2002), pp. 341–434.
- [17] H. Munthe-Kaas and B. Owren, Computations in a free Lie algebra, *Philosophical Trans. Royal Soc. A*, 357 (1999), pp. 957–981.
- [18] A. Murua and J.M. Sanz-Serna, Order conditions for numerical integrators obtained by composing simpler integrators, *Philosophical Trans. Royal Soc. A*, 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. Omelyan, I.M. Mryglod, and R. Folk, On the 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 & 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, *Physica 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*, Vol. 10, J.E. Marsden, G.W. Patrick, and W.F. Shadwick, eds., American Mathematical Society, Providence, R.I. (1996), pp. 217–244.
- [26] H. Yoshida, Construction of higher order symplectic integrators, *Phys. Lett. A*, 150 (1990), pp. 262–268.