UNIVERSITY OF CAMBRIDGE

Numerical Analysis Reports

# High Order Numerical Integrators for Differential Equations using Composition and Processing of low Order Methods

**Sergio Blanes**

DAMTP 2000/NA05
May, 2000

Department of Applied Mathematics and Theoretical Physics
Silver Street
Cambridge
England CB3 9EW

# High Order Numerical Integrators for Differential Equations using Composition and Processing of low Order Methods

Sergio Blanes

May 17, 2000

[1]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, England.

**Abstract**

In this paper we show how to build high order integrators for solving ordinary differential equations by composition of low order methods and using the processing technique. From a basic $p$-th order method, $\Psi_p$, one can obtain high order integrators in the processed form $\bar{\Psi}_n = \mathcal{P}\,\mathcal{K}\,\mathcal{P}^{-1}$ $(n > p)$ being both the processor $\mathcal{P}$ and the kernel $\mathcal{K}$ compositions of the basic method. The number of conditions for the kernel is drastically reduced if we compare with a standard composition. The particular case in which $\Psi_p$ is a symmetric scheme of order 2 and 4, respectively, is analysed, and new optimised 6-th and 8-th order integrators are built.

*Keywords:* differential equations; Initial value problems; composition methods; Processing

*AMS classification:* 17B66; 34A50; 65L05

## 1 Introduction

One of the most useful techniques for analytically solving the equation

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}), \qquad \mathbf{y}(0) = \mathbf{y}_0 \in \mathbb{R}^m, \tag{1}$$

is to look for an invertible transformation $\mathbf{z} = \mathcal{P}(\mathbf{y})$ such that the corresponding equation for $\mathbf{z}$,

$$\frac{d\mathbf{z}}{dt} = \mathbf{g}(\mathbf{z}), \qquad \mathbf{z}(0) = \mathcal{P}(\mathbf{y}_0) \in \mathbb{R}^m, \tag{2}$$

is exactly solvable or it is easy to obtain analytical approximations. We will refer to this transformation as the processing technique.

For instance, in classical mechanics, the evolution of an autonomous dynamical system is given by the Hamiltonian equations

$$\begin{aligned}\dot{\mathbf{q}} &= \frac{\partial H(\mathbf{q},\mathbf{p})}{\partial \mathbf{p}} \\ \dot{\mathbf{p}} &= -\frac{\partial H(\mathbf{q},\mathbf{p})}{\partial \mathbf{q}},\end{aligned} \tag{3}$$

where the dot indicates derivative respect to $t$ and $\mathbf{q},\mathbf{p} \in \mathbb{R}^D$ are the canonical coordinates and associated momenta. The scalar function $H : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is the Hamiltonian of the system. An standard technique for exactly solving (3) or to obtain analytical approximations is to look for a canonical transformation [1, 11]

$$\left.\begin{array}{l}\mathbf{Q} = \mathbf{Q}(\mathbf{q},\mathbf{p}) \\ \mathbf{P} = \mathbf{P}(\mathbf{q},\mathbf{p})\end{array}\right\} \qquad \text{or} \qquad (\mathbf{Q},\mathbf{P}) = \mathcal{P}(\mathbf{q},\mathbf{p}), \tag{4}$$

such that $\mathbf{Q},\mathbf{P}$ can be considered as the coordinates and momenta of another Hamiltonian $K(\mathbf{Q},\mathbf{P})$ which is exactly solvable (for example, if we have $K(\mathbf{Q})$ or $K(\mathbf{P})$), or it can be approximately solved (for example, if we have $K(\mathbf{Q},\mathbf{P}) = K_1(\mathbf{P}) + \epsilon K_2(\mathbf{Q},\mathbf{P}) + O(\epsilon^2)$ and $\epsilon \ll 1$). For instance, the Hamilton-Jacobi method and the Perturbative canonical theory are based in this procedure.

However, this useful technique for obtaining analytical results is not an standard procedure for building numerical integrators.

Let us consider $\Psi_{t,\mathbf{f}}$ the exact flow of (1) and $\mathcal{K}_{t,\mathbf{g}}$ the corresponding flow for (2). Then, if the processing technique is used we have

$$\Psi_{t,\mathbf{f}} = \mathcal{P}\,\mathcal{K}_{t,\mathbf{g}}\,\mathcal{P}^{-1}. \tag{5}$$

In this paper we will present new one-step numerical integrators $\hat{\Psi}_h$ for the problem (1) in the form

$$\hat{\Psi}_h = \hat{\mathcal{P}}_h\,\hat{\mathcal{K}}_h\,\hat{\mathcal{P}}_h^{-1}, \tag{6}$$

where $h$ is the time-step and $\hat{\mathcal{K}}_h$, $\hat{\mathcal{P}}_h$ can be considered as numerical methods approximating $\mathcal{K}_{h,\mathbf{g}}$ and $\mathcal{P}$. We will refer to $\hat{\mathcal{K}}_h$ and $\hat{\mathcal{P}}_h$ as the kernel and processor of the method, respectively. Then, $N$ steps can be written in the form

$$\left(\hat{\Psi}_{h,\mathbf{f}}\right)^N = \hat{\mathcal{P}}_h\,\left(\hat{\mathcal{K}}_h\right)^N\,\hat{\mathcal{P}}_h^{-1}. \tag{7}$$

Observe that $\hat{\mathcal{P}}_h$ has to be evaluated only once at the beginning and $\hat{\mathcal{P}}_h^{-1}$ only when the output is desired then, essentially, one can consider the cost of the method as the cost of $\hat{\mathcal{K}}_h$.

The composition (5) was originally considered by Butcher [7] using Runge–Kutta (RK) methods for $\hat{\mathcal{P}}_h$ and $\hat{\mathcal{K}}_h$. A general analysis in this line is presented in [8] but not practical methods are obtained. However, this technique has shown to be very efficient in the context of symplectic integrators for some particular cases: near-integrable systems [26, 16, 4], Runge-Kutta-Nyström (RKN) methods [20, 14, 5] and when the system is separable in two exactly solvable parts in general [2]. In this paper we will show how to obtain high order processed methods by composition of low order methods.

The use of *Geometric Integrators* (GI) have proved very useful for solving (1) when the system has some qualitative properties which are important to retain [21]. For example, the preservation of these properties makes that, in most of the cases, the approximate solutions have smaller error growth than standard integrators. Then, it will be very important than $\hat{\mathcal{K}}_h$ retain these properties and it will be less essential, but still interesting, for $\hat{\mathcal{P}}_h$ to preserve them.

Let us consider now an integrator $S_p(h)$ of order $p$. It is well known that with a composition of this basic method it is possible to obtain high order methods in the form

$$\bar{S}_n(\mathbf{a}) = S_p(a_1 h) \dots S_p(a_k h), \tag{8}$$

where $\bar{S}_n$ is a new integrator of order $n > p$. The vector $\mathbf{a} = (a_1, \dots, a_k)$ characterises the method and the $a_i$ have to solve a number of order conditions; for instance, $\sum_{i=1}^{k} a_i = 1$ for consistency. Observe that if $S_p(h)$ is a GI then the composition will be a GI. In addition, if $S_p(h)$ is a symmetric method with $p$ an even number, and the composition (8) is symmetric $(a_{k+1-i} = a_i, \quad i = 1, 2, \dots)$ then $\bar{S}_n$ will be a symmetric integrator with $n$ an even number.

Compositions like (8) have been widely used in the literature for building high order integrators [9, 12, 13, 15, 17, 18, 22, 23, 24, 25, 27].

In this paper we will show that it is possible and very efficient to follow a similar strategy but using the processing technique. That means to consider

$$\hat{\mathcal{K}}_h(\mathbf{b}) = S_p(b_1 h) \dots S_p(b_r h) \tag{9}$$

$$\hat{\mathcal{P}}_h(\mathbf{c}) = S_p(c_1 h) \dots S_p(c_s h) \tag{10}$$

in such a way that $\hat{\Psi}_h$ has effective order $n$. We will see that the coefficients $b_i$ in (9) have to solve a considerably smaller number of order conditions than the coefficients $a_i$ in (8) because some of these conditions can be solved by the $c_i$ in (10).

An alternative technique for increasing the order of a method is to use extrapolation. The qualitative properties of the basic method are lost because the linear combinations destroy the group properties but, that happens at higher orders than the order of the final method [3, 10]. This technique shows to be very efficient for increasing the order of GI when the basic method is of order four or higher. In this paper we will consider, as basic, low order methods and, for simplicity, we will take symmetric integrators, but the same idea and technique can be used for non-symmetric basic methods and different orders.

Section 2 introduces some basic concepts on Lie algebras. In Section 3 we consider symmetric basic methods and present the kernel conditions when the kernel is itself a symmetric compositions. Explicit expressions for the kernel and processor are presented for low order methods. In Section 4 we show how to obtain the kernel and processor by composition in such a way that the previous conditions are satisfied. In particular, we present new optimal processed 6-th and 8-th order methods using $S_2$ and $S_4$, respectively. Finally, in Section 5 we present some numerical examples and we will observe that the results completely agree with the theoretically predicted.

# 2 Lie Algebraic Tools

Let us write the eq. (1) in the linear form

$$\mathbf{y}' = \mathbf{F}\mathbf{y} \tag{11}$$

where $\mathbf{F} = \mathbf{f} \cdot \nabla_{\mathbf{y}} = \sum_{i=1}^{m} f_i \frac{\partial}{\partial y_i}$ is the Lie operator associated to $\mathbf{f} = (f_1, \ldots, f_m)$. Then, the solution of (11) takes the formal expression [1]

$$\mathbf{y}(t) = e^{t\mathbf{F}}\mathbf{y}(0), \tag{12}$$

where $e^{t\mathbf{F}}$ is the Lie transformation associated to $\mathbf{f}$. In the same way, a numerical integrator, $\hat{\Psi}_h$, for (11) can be written as

$$\hat{\Psi}_h = e^{h\tilde{\mathbf{F}}}, \tag{13}$$

where $\tilde{\mathbf{F}} = \mathbf{F}_1 + h\mathbf{F}_2 + h^2\mathbf{F}_3 + \ldots$, and $\mathbf{F}_i = \mathbf{f}_i \cdot \nabla_{\mathbf{y}}$. For consistency $\mathbf{F}_1 = \mathbf{F}$.

The operator $\mathbf{F}$ can be considered as an element of a Lie algebra $\mathfrak{g}$ and then $e^{t\mathbf{F}}$ is an element of the Lie group, $\mathfrak{G}$ associated to $\mathfrak{g}$. If (13) is such that $\mathbf{F}_i \in \mathfrak{g}, \;\; i = 2, 3, \ldots$, then $e^{h\tilde{\mathbf{F}}}$ will stay in the same Lie group as the exact solution. In this case, many important qualitative properties of the exact solution are preserved by the numerical integrator [21]. This will be the case if in (8)-(10) the elementary flows belong to the same Lie group as the exact solution, being the methods referred as *Geometric Integrators*.

## 2.1 Composition of Symmetric Methods

Let us consider as the basic method a second order symmetric integrator $S_2(h)$. Then, it is possible to write it as

$$S_2(h) = e^{h\tilde{\mathbf{F}}} = \exp\left(h\mathbf{F}_1 + h^3\mathbf{F}_3 + h^5\mathbf{F}_5 + \ldots\right), \tag{14}$$

with $\mathbf{F}_1 = \mathbf{F}$. At this point the $\mathbf{F}_i$ can be considered as the elements of a graded Lie algebra with grade $i$. Then compositions (8)-(10) can be formally written, making use of the Baker-Campbell-Hausdorff (BCH) formula, in only one exponential $e^C$ with

$$C = \sum_{i=1}^{\infty} h^i \sum_{j=1}^{\nu_i} d_{i,j} E_{i,j} \tag{15}$$

where $d_{i,j}$ are scalar functions depending on the parameters of the method. The $E_{i,j}$ are the elements of a basis of the Lie algebra generated by $\{\mathbf{F}_1, \mathbf{F}_3, \mathbf{F}_5, \ldots\}$. In particular, each $E_{i,j}$ will correspond to $\mathbf{F}_i$ or to a nested commutator of different $\mathbf{F}_j$ of grade $i$, i.e. $E_{i,j} = [\mathbf{F}_k, [\ldots, [\mathbf{F}_l, \mathbf{F}_m] \ldots]], \;\; k + \ldots + l + m = i$, being $\nu_i$ the dimension of the Lie

| $n$ | $\nu_n$ | $S_2$ | | | |
|---|---|---|---|---|---|
| 1 | 1 | $E_{1,1} = \mathbf{F}_1$; | | | |
| 3 | 1 | $E_{3,1} = \mathbf{F}_3$; | | | |
| 4 | 1 | $E_{4,1} = [13]$; | | | |
| 5 | 2 | $E_{5,1} = \mathbf{F}_5$; | $E_{5,2} = [113]$; | | |
| 6 | 2 | $E_{6,1} = [15]$; | $E_{6,2} = [1113]$; | | |
| 7 | 4 | $E_{7,1} = \mathbf{F}_7$; | $E_{7,2} = [115]$; | $E_{7,3} = [11113]$; | $E_{7,4} = [313]$; |
| 8 | 5 | $E_{8,1} = [17]$; | $E_{8,2} = [1115]$; | $E_{8,3} = [111113]$; | |
| | | $E_{8,4} = [1313]$; | $E_{8,5} = [35]$; | | |
| 9 | 8 | $E_{9,1} = \mathbf{F}_9$; | $E_{9,2} = [117]$; | $E_{9,3} = [11115]$; | $E_{9,4} = [1111113]$; |
| | | $E_{9,5} = [11313]$; | $E_{9,6} = [31113]$; | $E_{9,7} = [135]$; | $E_{9,8} = [315]$; |

Table 1: Basis of the Lie algebra generated by $\{\mathbf{F}_1, \mathbf{F}_3, \mathbf{F}_5, \mathbf{F}_7, \mathbf{F}_9\}$ up to order 9 .

| $n$ | $\nu_n$ | $S_4$ | | |
|---|---|---|---|---|
| 1 | 1 | $E_{1,1} = \mathbf{F}_1$; | | |
| 5 | 1 | $E_{5,1} = \mathbf{F}_5$; | | |
| 6 | 1 | $E_{6,1} = [15]$; | | |
| 7 | 2 | $E_{7,1} = \mathbf{F}_7$; | $E_{7,2} = [115]$; | |
| 8 | 2 | $E_{8,1} = [17]$; | $E_{8,2} = [1115]$; | |
| 9 | 3 | $E_{9,1} = \mathbf{F}_9$; | $E_{9,2} = [117]$; | $E_{9,3} = [1115]$; |
| 10 | 3 | $E_{10,1} = [19]$; | $E_{10,2} = [1117]$; | $E_{10,3} = [11115]$; |
| 11 | 5 | $E_{11,1} = \mathbf{F}_{11}$; | $E_{11,2} = [119]$; | $E_{11,3} = [11117]$; |
| | | $E_{11,4} = [1111115]$; | $E_{11,5} = [515]$; | |

Table 2: Basis of the Lie algebra generated by $\{\mathbf{F}_1, \mathbf{F}_5, \mathbf{F}_7, \mathbf{F}_9, \mathbf{F}_{11}\}$ up to order 11 .

subalgebra of degree $i$. In Table 1 we present a basis up to order 9, where the notation $[i \ldots jk] \equiv [\mathbf{F}_i, [\ldots, [\mathbf{F}_j, \mathbf{F}_k] \ldots]]$ has been used.

If we consider a 4-th order symmetric integrator $S_4$ as the basic method, then

$$S_4(h) = e^{h\tilde{\mathbf{F}}} = \exp\left(h\mathbf{F}_1 + h^5\mathbf{F}_5 + h^7\mathbf{F}_7 + \ldots\right), \tag{16}$$

and we have to consider the Lie algebra generated by $\{\mathbf{F}_1, \mathbf{F}_5, \mathbf{F}_7, \ldots\}$. It corresponds to the previous algebra taking $\mathbf{F}_3 = 0$ so, a considerably simplification happens, allowing us to analyse easily high order methods. A basis up to order 11 is given in Table 2.

Compositions using $S_4$ as basic integrator can be useful for obtaining high order methods mainly for two reasons:

$i$) The reduced number of equations allows a deep analysis of the numerical solutions.

$ii$) It is possible to use a large quantity of different methods $S_4$ specially tailored for particular problems, i.e. RKN methods.

## 3 Kernel and Processor conditions

In the same way as in [2], and for simplicity, it is possible to consider for $\hat{\mathcal{K}}_h$ a symmetric composition, so

$$\hat{\mathcal{K}}_h^{-1} = \hat{\mathcal{K}}_{-h} \tag{17}$$

and for the processor we can take an scheme with only even powers of $h$

$$\hat{\mathcal{P}}_{-h} = \hat{\mathcal{P}}_h. \tag{18}$$

From (14) and (16) we see that $S_i(-h) = S_i^{-1}(h), \quad i = 2,4$ so, the composition (10) can not satisfy condition (18) exactly. However, it is possible to preserve this condition up to $h^s$ with $s$ as high as desired.

Formally we can write

$$\begin{aligned} \hat{\mathcal{K}}_h &= \exp\left( \sum_{i=1}^{\infty} h^{2i-1} \sum_{j=1}^{\nu_i} k_{2i-1,j} E_{2i-1,j} \right) \equiv e^{hK} & (19) \\ \hat{\mathcal{P}}_h &= \exp\left( \sum_{i=1}^{\infty} h^{2i} \sum_{j=1}^{\nu_i} p_{2i,j} E_{2i,j} \right) \equiv e^{P}, & (20) \end{aligned}$$

and using the BCH formula

$$\hat{\Psi}_h = \hat{\mathcal{P}}_h \, \hat{\mathcal{K}}_h \, \hat{\mathcal{P}}_h^{-1} = e^P e^{hK} e^{-P} = e^{h\mathcal{F}} \tag{21}$$

with

$$\begin{aligned} h\mathcal{F} &= e^P hK e^{-P} = e^{[ad_P, \cdot]} hK = hK + [P, hK] + \frac{1}{2}[P,[P,hK]] + \dots \\ &= \sum_{i=1}^{\infty} h^{2i-1} \sum_{j=1}^{\nu_i} f_{2i-1,j} E_{2i-1,j} & (22) \end{aligned}$$

where the $f_{i,j}$ are polynomial functions of $k_{r,s}, \;\; r \leq i$ and $p_{r,s}, \;\; r \leq i-1$. Observe that $h\mathcal{F}$ does not contain even powers of $h$. The integrator $\hat{\Psi}_h$ will be of order $n$ if $\mathcal{F} = \mathbf{F} + O(h^n)$ or, equivalently, if

$$\begin{aligned} & f_{1,1} = k_{1,1} = 1, \\ & f_{2i-1,j}(\mathbf{k}, \mathbf{p}) = 0, \qquad i = 2, \dots \frac{n}{2}, \quad j = 1 \dots, \nu_i, & (23) \end{aligned}$$

where $\mathbf{k} = (k_{1,1}, \dots, k_{n,\nu_n})$ and $\mathbf{p} = (p_{1,1}, \dots, p_{n-1,\nu_{n-1}})$. The total number of order conditions is

$$N_f = \sum_{i=1}^{n/2} \nu_{2i-1}, \tag{24}$$

| $n$ | $S_2$ no-proc | $S_2$ proc | $S_4$ no-proc | $S_4$ proc |
|---|---|---|---|---|
| 2 | 1 | 1 | 1 | 1 |
| 4 | 2 | 2 | - | - |
| 6 | 4 | 3 | 2 | 2 |
| 8 | 8 | 5 | 4 | 3 |
| 10 | 16 | 8 | 7 | 4 |
| 12 | 34 | 14 | 12 | 6 |
| 14 | | | 21 | 8 |

Table 3: Order conditions for symmetric compositions of the basic methods $S_2$ and $S_4$ for a kernel and for a non-processed composition.

which agree with the number of order conditions for a non-processed symmetric composition. The main difference is that now we have a number of variables $p_{i,j}$ which can be used for solving some of the conditions.

**Theorem**: Given (19) and (20) as the kernel and processor of an $n$-th order integrator ($n$ even) then, the number of necessary conditions to be satisfied by the kernel is given by

$$N_k = \sum_{i=1}^{n/2} \nu_{2i-1} - \sum_{i=1}^{n/2-1} \nu_{2i} = \sum_{i=0}^{n/2-1} (\nu_{2i+1} - \nu_{2i}). \tag{25}$$

PROOF: in a similar way to [2] it is possible to eliminate the $\sum_{i=1}^{n/2-1} \nu_{2i}$ variables $p_{i,j}$ from (23) solving the same number of equations. The other $N_k$ conditions will only depend on the kernel coefficients, being the kernel conditions. □

In Table 3 we present the number of order conditions for a non-processed composition as well as for the kernel, both when considering as the basic method $S_2$ and $S_4$. From this table it is clear the great saving in the number of evaluations per step when the processing technique is used for high order methods.

Observe that the variables $p_{2i,j}, \quad i < n/2-1$ have been used for solving the order conditions but, we still have the $\nu_n$ variables $p_{n,j}, \quad j = 1, \ldots, \nu_n$, which appear in the $\nu_{n+1}$ error functions at order $h^{n+1}, \quad f_{n+1,j}, \quad j = 1, \ldots, \nu_{n+1}$, and can be used for minimising their value [2]. This optimisation procedure will be particularly simple in some cases.

## 3.1 Examples

In order to illustrate how to obtain the kernel conditions, how to get the conditions for the processor and to minimise the error coefficients, we will apply this technique for obtaining an optimal 6-th order integrator using $S_2$ as the basic methods and an 8-th order integrators using $S_4$.

**Optimal 6-th order integrator with** $S_2$: let us consider the kernel and processor up to order 7:

$$\begin{aligned} hK &= hk_{1,1}E_{1,1} + h^3k_{3,1}E_{3,1} + h^5\left(k_{5,1}E_{5,1} + k_{5,2}E_{5,2}\right) + \\ &\quad h^7\left(k_{7,1}E_{7,1} + k_{7,2}E_{7,2} + k_{7,3}E_{7,3} + k_{7,4}E_{7,4}\right) \qquad (26) \\ P &= h^4p_{4,1}E_{4,1} + h^6\left(p_{6,1}E_{6,1} + p_{6,2}E_{6,2}\right) \qquad (27) \end{aligned}$$

where the $E_{i,j}$ correspond to the basis in Table 1. After a simple algebraic manipulation we have

$$\begin{aligned} h\mathcal{F} &= e^P hK e^{-P} = hK + [P, hK] + O(h^8) \\ &= hk_{1,1}E_{1,1} + h^3k_{3,1}E_{3,1} + h^5\left(k_{5,1}E_{5,1} + (k_{5,2} - p_{4,1})E_{5,2}\right) + \qquad (28) \\ &\quad h^7\left(k_{7,1}E_{7,1} + (k_{7,2} - p_{6,1})E_{7,2} + (k_{7,3} - p_{6,2})E_{7,3} + (k_{7,4} - p_{4,1}k_{3,1})E_{7,4}\right), \end{aligned}$$

then, the order conditions are

$$\begin{aligned} f_{1,1} &= k_{1,1} = 1 \qquad & f_{3,1} &= k_{3,1} = 0 \\ f_{5,1} &= k_{5,1} = 0 \qquad & f_{5,2} &= k_{5,2} - p_{4,1} = 0, \end{aligned}$$

so, the 3 kernel conditions at order 6 we see from Table 3 correspond to: $k_{1,1} = 1,\ k_{3,1} = 0,\ k_{5,1} = 0$, and the processor condition is $p_{4,1} = k_{5,2}$. Substituting these results into the error functions we have

$$\begin{aligned} f_{7,1} &= k_{7,1} \qquad & f_{7,2} &= k_{7,2} - p_{6,1} \\ f_{7,3} &= k_{7,3} - p_{6,2} \qquad & f_{7,4} &= k_{7,4}, \end{aligned}$$

and taking $p_{6,1} = k_{7,2}$ and $p_{6,2} = k_{7,3}$ we obtain $f_{7,2} = f_{7,3} = 0$, corresponding to the optimisation with the processor previously mentioned. Now, $f_{7,1}$ and $f_{7,4}$ depend only on $k_{i,j}$. Then, the conditions $k_{7,1} = 0$ and $k_{7,4} = 0$ will correspond to the two new kernel conditions if we want to obtain an 8-th order integrator. In this case, more conditions on the processor can be used for optimisation.

**Optimal 8-th order integrator with** $S_4$: following the same procedure we can write, keeping terms up to order $h^9$

$$\begin{aligned} hK &= hk_{1,1}E_{1,1} + h^5k_{5,1}E_{5,1} + h^7\left(k_{7,1}E_{7,1} + k_{7,2}E_{7,2}\right) \\ &\quad h^9\left(k_{9,1}E_{9,1} + k_{9,2}E_{9,2} + k_{9,3}E_{9,3}\right) \qquad (29) \\ P &= h^6p_{6,1}E_{6,1} + h^8\left(p_{8,1}E_{8,1} + p_{8,2}E_{8,2}\right), \qquad (30) \end{aligned}$$

where now the $E_{i,j}$ correspond to the basis in Table 2. From the order conditions we have the kernel and processor conditions:

$$\begin{aligned} k_{1,1} &= 1 \qquad & k_{5,1} &= k_{7,1} = 0 \\ p_{6,1} &= k_{7,2}, \end{aligned}$$

and considering $p_{8,1} = k_{9,2}$ and $p_{8,2} = k_{9,3}$ we have for the error functions

$$f_{9,1} = k_{9,1}; \qquad f_{9,2} = f_{9,3} = 0, \qquad (31)$$

being $k_{9,1} = 0$ the only new kernel condition for 10-th order methods.

# 4 Obtaining the kernel and processor by composition

Using the analysis of the previous section we can obtain the conditions to be satisfied by the coefficients $k_{i,j}$ and $p_{i,j}$. From (9) and (10), and using the BCH formula we have:

$$S_p(b_1 h) \ldots S_p(b_r h) = e^{hK(\mathbf{b})} \tag{32}$$

$$S_p(c_1 h) \ldots S_p(c_s h) = e^{P(\mathbf{c})} \tag{33}$$

with $\mathbf{b} = (b_1, \ldots, b_r)$ and $\mathbf{c} = (c_1, \ldots, c_s)$ being the unknowns. The $k_{i,j}$ and $p_{i,j}$ are polynomial functions of $b_k$ and $c_k$, respectively. Here, we have to evaluate $\mathbf{b}$ such that the $k_{i,j}$ satisfy the kernel conditions. Once the kernel is chosen we have to evaluate $\mathbf{c}$ in such a way that the $p_{i,j}$ satisfy the required conditions for this particular kernel.

Observe that the computational cost of the method as well as the leading error term will depend exclusively on $\mathbf{b}$. At the same time, we can find many solutions for (32). Among all these solutions we will choose the one which minimise the error terns. There is a problem at this point. The leading error term of an $n$-th order method is given by

$$E_r^{(n)} = \sum_{j=1}^{\nu_{n+1}} f_{n+1,j} E_{n+1,j} \tag{34}$$

where (after the optimisation with the processor parameters) the $f_{n+1,j}$ depend only on the kernel coefficients. We are interested in a solution which minimises $\|E_r^{(n)}\|$ for a given norm but, the relative value of the different elements $\|E_{n+1,j}\|$ is strongly dependent on the basic method used. For instance, the error of the 6-th order method presented in the last section is given by

$$E_r^{(6)} = k_{7,1}(\mathbf{b})\mathbf{F}_7 + k_{7,4}(\mathbf{b})[\mathbf{F}_3, [\mathbf{F}_1, \mathbf{F}_3]] \tag{35}$$

where $k_{7,1}$ and $k_{7,4}$ depend on the particular solution $\mathbf{b}$ for the kernel conditions and $\|\mathbf{F}_7\|$, $\|[\mathbf{F}_3, [\mathbf{F}_1, \mathbf{F}_3]]\|$ depend on the basic method chosen. On the other hand, for the processor we only need a solution which satisfy the required conditions.

From Table 3 it is clear that the processing technique will be very useful for obtaining high order methods because the reduction on the number of conditions. However, in order to illustrate this procedure we will consider 6-th order methods using $S_2$ and 8-th order methods using $S_4$ as in the previous section, where the saving versus non-processed methods is minimum. We will show explicitly how to get an optimal kernel and how to obtain a processor for this particular kernel.

As mentioned, the polynomial functions $k_{i,j}(\mathbf{b})$ and $p_{i,j}(\mathbf{d})$ can be obtained using the BCH formula. If the kernel is a symmetric composition using $S_2$, we can take the recurrence formulae presented in [27] based on the symmetric BCH formula or, alternatively, to consider the expressions presented in [23]. If $S_4$ is used, small modifications on the previous expressions are necessary.

For the processor we will consider the BCH formula

$$
\begin{aligned}
e^X e^Y &= \exp\left(X + Y + \frac{1}{2}[X,Y] + \frac{1}{12}([X,[X,Y]] + [Y,[Y,X]]) + \right. \\
&\qquad \left. \frac{1}{24}[X,[Y,[Y,X]]] + \dots \right) .
\end{aligned} \tag{36}
$$

Considering the general term

$$
Q(\alpha^{(k)}) = \exp\left(\sum_{i=1}^{\infty} h^i \sum_{j=1}^{\nu_i} \alpha_{i,j}^{(k)} E_{i,j}\right) \tag{37}
$$

we are interested in the recurrence expression

$$
Q(\alpha^{(k+1)}) = S_p(xh) Q(\alpha^{(k)}) \tag{38}
$$

for $p = 2, 4$. Using (36) and $S_2$, and after some algebra we have, up to order 6

$$
\begin{aligned}
\alpha_{1,1}^{(k+1)} &= \alpha_{1,1}^{(k)} + x \\
\alpha_{3,1}^{(k+1)} &= \alpha_{3,1}^{(k)} + x^3 \\
\alpha_{4,1}^{(k+1)} &= \alpha_{4,1}^{(k)} + \frac{1}{2}\left(x\alpha_{3,1}^{(k)} - x^3\alpha_{1,1}^{(k)}\right) \\
\alpha_{5,1}^{(k+1)} &= \alpha_{5,1}^{(k)} + x^5 \\
\alpha_{5,2}^{(k+1)} &= \alpha_{5,2}^{(k)} + \frac{1}{2}x\alpha_{4,1}^{(k)} + \frac{1}{12}\left(x^2\alpha_{3,1}^{(k)} - x^4\alpha_{1,1}^{(k)} + x^3(\alpha_{1,1}^{(k)})^2 - x\alpha_{1,1}^{(k)}\alpha_{3,1}^{(k)}\right) \\
\alpha_{6,1}^{(k+1)} &= \alpha_{6,1}^{(k)} + \frac{1}{2}\left(x\alpha_{5,1}^{(k)} - x^5\alpha_{1,1}^{(k)}\right) \\
\alpha_{6,2}^{(k+1)} &= \alpha_{6,2}^{(k)} + \frac{1}{2}x\alpha_{5,2}^{(k)} + \frac{1}{12}\left(x^2\alpha_{4,1}^{(k)} - x\alpha_{1,1}^{(k)}\alpha_{4,1}^{(k)}\right) + \frac{1}{24}\left(x^4(\alpha_{1,1}^{(k)})^2 - x^2\alpha_{1,1}^{(k)}\alpha_{3,1}^{(k)}\right) .
\end{aligned} \tag{39}
$$

If we take $S_4$ then

$$
\begin{aligned}
\alpha_{1,1}^{(k+1)} &= \alpha_{1,1}^{(k)} + x \\
\alpha_{5,1}^{(k+1)} &= \alpha_{5,1}^{(k)} + x^5 \\
\alpha_{6,1}^{(k+1)} &= \alpha_{6,1}^{(k)} + \frac{1}{2}\left(x\alpha_{5,1}^{(k)} - x^5\alpha_{1,1}^{(k)}\right) \\
\alpha_{7,1}^{(k+1)} &= \alpha_{7,1}^{(k)} + x^7 \\
\alpha_{7,2}^{(k+1)} &= \alpha_{7,2}^{(k)} + \frac{1}{2}x\alpha_{6,1}^{(k)} + \frac{1}{12}\left(x^2\alpha_{5,1}^{(k)} - x^6\alpha_{1,1}^{(k)} + x^5(\alpha_{1,1}^{(k)})^2 - x\alpha_{1,1}^{(k)}\alpha_{5,1}^{(k)}\right) \\
\alpha_{8,1}^{(k+1)} &= \alpha_{8,1}^{(k)} + \frac{1}{2}\left(x\alpha_{7,1}^{(k)} - x^7\alpha_{1,1}^{(k)}\right) \\
\alpha_{8,2}^{(k+1)} &= \alpha_{8,2}^{(k)} + \frac{1}{2}x\alpha_{7,2}^{(k)} + \frac{1}{12}\left(x^2\alpha_{6,1}^{(k)} - x\alpha_{1,1}^{(k)}\alpha_{6,1}^{(k)}\right) + \frac{1}{24}\left(x^6(\alpha_{1,1}^{(k)})^2 - x^2\alpha_{1,1}^{(k)}\alpha_{5,1}^{(k)}\right) .
\end{aligned} \tag{40}
$$

Considering (33) and taking $\alpha_{i,j}^{(0)} = 0$ as initial conditions we will have that $p_{i,j}(\mathbf{c}) = \alpha_{i,j}^{(s)}$.

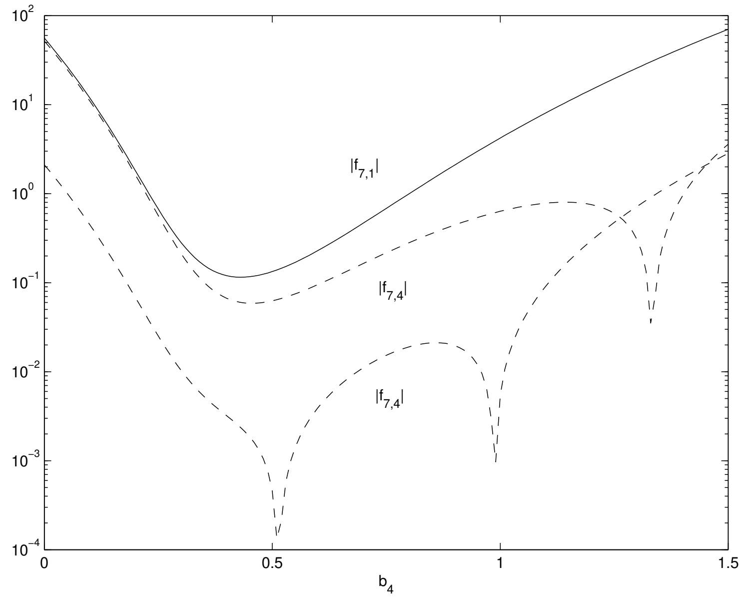


Figure 1: Error functions $|f_{7,1}|$ and $|f_{7,4}|$ versus $b_4$, for the seven-stages processed method using $S_2$.

Let us consider $\mathbf{b} = (b_3, b_2, b_1, b_2, b_3)$ for the five-stages kernel of a 6-th order integrator. The kernel conditions are

$$k_{1,1} = b_1 + 2(b_2 + b_3) = 1; \quad k_{3,1} = b_1^3 + 2(b_2^3 + b_3^3) = 0; \quad k_{5,1} = b_1^5 + 2(b_2^5 + b_3^5) = 0.$$

The two real solutions give for the error functions: $f_{7,1} = 56.03$ in both cases and $f_{7,1} = 2.112, -51.99$. These large error functions can be considerably reduced considering a seven-stages composition, $\mathbf{b} = (b_4, b_3, b_2, b_1, b_2, b_3, b_4)$, with $b_4$ as a free parameter. The computational cost increases from 5 to 7 stages so, it will be interesting only if the error functions are reduced at least by a factor $(7/5)^6 = 7.529...$

In Fig. 1 we present the values of $|f_{7,1}|$ and $|f_{7,4}|$ versus $b_4$. We observe that the value of $|f_{7,1}|$ at the minimum is more than two orders of magnitude smaller than for $b_4 = 0$. On the other hand, the two different solutions for $|f_{7,4}|$ can be cancelled at several points. Observe that one of the zeros of $|f_{7,4}|$ is very close to the minimum of $|f_{7,1}|$. Then, we will chose the closest value of $b_4$ to the minimum of $|f_{7,1}|$ such that $f_{7,4} = 0$. The solution is given in Table 4.

Finally, we have to find a processor which satisfy all the the required conditions. For example, we will take $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, -c_1, -c_2, -c_3, -c_4, -c_5)$, where a solution is given in Table 4. For this solution, the processor have not odd powers of $h$ up to order 9. That means for the whole method we have $h\mathcal{F} = h\mathbf{F} + h^7 k_{7,1} E_{7,1} + O(h^9)$ and it will work in practice as a symmetric integrator.

In order to better appreciate the efficiency of the new method, we present in Table 5

| 6-th Order:$S_2$ | |
|---|---|
| | $c_5 = 0.375012038697862$ |
| $b_4 = 0.513910778424374$ | $c_4 = 0.384998538774070$ |
| $b_3 = 0.364193022833858$ | $c_3 = -0.074332422810238$ |
| $b_2 = -0.867423280969274$ | $c_2 = -0.461165940466494$ |
| $b_1 = 1 - 2(b_2 + b_3 + b_4)$ | $c_1 = -(c_2 + c_3 + c_4 + c_5)$ |
| 8-th Order:$S_4$ | |
| | $c_5 = 0.1$ |
| $b_4 = 0.3836$ | $c_4 = 0.153884390967272$ |
| $b_3 = 0.38378409898601552832$ | $c_3 = 0.295715027608753$ |
| $b_2 = -0.58571608011635309034$ | $c_2 = -0.182295174329697$ |
| $b_1 = 1 - 2(b_2 + b_3 + b_4)$ | $c_1 = -(c_2 + c_3 + c_4 + c_5)$ |

Table 4: Coefficients for the new processed 6-th and 8-th order methods with kernel $\mathbf{b} = (b_4, b_3, b_2, b_1, b_2, b_3, b_4)$ and processor $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, -c_1, -c_2, -c_3, -c_4, -c_5)$, using $S_2$ and $S_4$, respectively.

| method-stages | $f^*_{7,1}$ | $f^*_{7,2}$ | $f^*_{7,3}$ | $f^*_{7,4}$ |
|---|---|---|---|---|
| Yos-7 | 0.88839 | 0.02987 | 0.0001961 | −0.01798 |
| McL-9 | 0.40799 | 0.01254 | 0.0001012 | −0.02451 |
| Proc-5 | 7.44179 | 0 | 0 | 0.28046 |
| Proc-7 | 0.14135 | 0 | 0 | 0 |

Table 5: Error terms for the non-processed and processed methods. The numerical values correspond to $f^*_{7,i} = (m/7)^6 f_{7,i}$, where $m$ is the number of stages per step for each method.

the error terms of the most efficient 6-th order methods we found in the literature which take (8) using $S_2$. These coefficients are normalised in order to take into account the different number of stages. The methods are: the best 7-stages of [27] and the optimised 9-stages of [15]. We must say that in spite of the small values of $f_{7,i}$, $i = 2, 3, 4$, their corresponding $\|E_{7,i}\|$ usually take greater values than $\|E_{7,1}\|$. This is the case, for instance, if we consider the well known leapfrog method as $S_2$ in a separable problem. From the results of this table we already appreciate the benefits of using the processing technique.

In the same way, for an 8-th order integrator we can consider $\mathbf{b} = (b_3, b_2, b_1, b_2, b_3)$ and $S_4$ as the basic method. Now, we have only one error term, $f_{9,1}$. The two real solutions give $f_{9,1} = 8.22\ldots$ This large value can be considerably reduced considering, as previously, $\mathbf{b} = (b_4, b_3, b_2, b_1, b_2, b_3, b_4)$ which has one free parameter. It seems interesting to minimise $f_{9,1}$ as well as $\mu = \sum_i |b_i|$ in order the higher order terms do not take extremely large values. In Fig. 2 we represent $|f_{9,1}|$ and $\mu$ versus $b_4$ as well as $|f_{11,1}|$ as representative of higher order error terms. All of them have the minimum at the same point. Observe

| method-stages | $f^*_{9,1}$ | $f^*_{9,2}$ | $f^*_{9,3}$ | $f^*_{11,1}$ |
|---|---|---|---|---|
| no-proc-7 | 0.270047 | 0.0100179 | 0.0000904 | 0.88511 |
| Proc-5 | 0.556848 | 0 | 0 | 1.99022 |
| Proc-7 | 0.0016815 | 0 | 0 | 0.001506 |

Table 6: Error terms for the non-processed and processed methods. The numerical values correspond to $f^*_{9,i} = (m/7)^8 f_{9,i}$, where $m$ is the number of stages per step for each method, and $f^*_{11,1} = (m/7)^{10} f_{11,1}$.

that the value of the error functions is reduced around four orders of magnitude at the minimum if we compare with their values at $b_4 = 0$.

A solution for $b_4$ near the minimum is given in Table 4. For the processor we can take $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5, -c_1, -c_2, -c_3, -c_4, -c_5)$ and one solution is given in Table 4. Again, with this processor we have that $h\mathcal{F} = h\mathbf{F} + h^9 k_{9,1} E_{9,1} + O(h^{11})$ and the method is, for practical purposes equivalent to a symmetric integrator.

In order to appreciate the improvement obtained using the processing technique, let us consider a non-processed 8-th order composition. It requires at least seven stages, i.e. $\mathbf{a} = (a_4, a_3, a_2, a_1, a_2, a_3, a_4)$. We found everal solutions and the one with smaller error coefficients is:

$$\begin{aligned} a_4 &= 0.846121147469682, & a_3 &= 0.158012845800852 \\ a_2 &= -1.09020666054393, & a_1 &= 1 - 2(a_2 + a_3 + a_4) \end{aligned}$$

where the error terms are given in Table 4. This method has exactly the same computational cost as the seven-stages processed method but, the error coefficients are more than two orders of magnitude greater.

From the results of this section and Table 3 we expect that higher order processed methods will be considerably more efficient than the corresponding non-processed integrators. However, higher order methods are technically more involved both for getting the kernel and optimal processor conditions as well as due to the numerical implementation for obtaining the best solution. This problem is under consideration and the results will be published elsewhere.

# 5 Numerical Experiments

In order to illustrate the efficiency of the new processed methods we will compare their performances versus standard integrators (of the same family) on the following two examples:

**Example 1: Volterra-Lotka problem.** Let us consider the simple differential equations

$$\begin{aligned} \dot{u} &= u(v-2) \\ \dot{v} &= v(1-u) \end{aligned} \qquad (41)$$

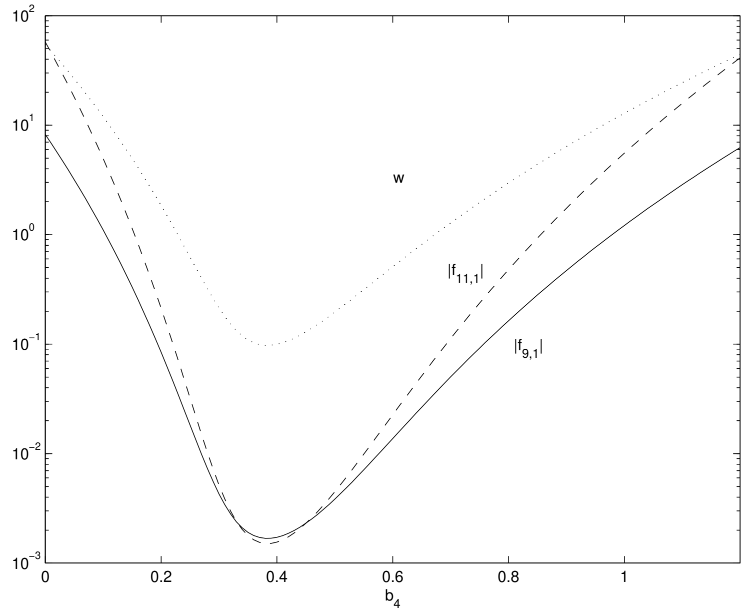


Figure 2: Absolute value of the coefficients $w$ and error functions $|f_{9,1}|$ and $|f_{11,1}|$ versus $b_4$. In order to appreciate the position of the minimum for the coefficients of the method we display $w = (\mu^2/20)^4$ with $\mu = \sum_i |b_i|$.

which corresponds to (1) with $\mathbf{y} = (u, v)$ and $\mathbf{f}(\mathbf{y}) = (u(v-2), v(1-u))$. The system has the first integral: $I(u, v) = \ln(u) - u + 2\ln(v) - v = Const.$ Several second order symmetric methods can be used as the basic scheme $S_2$. The following two methods will be considered:

*i) Implicit midpoint rule:*

$$\mathbf{y}_{k+1} = S_2\mathbf{y}_k = \mathbf{y}_k + h\mathbf{f}\left(\frac{\mathbf{y}_k + \mathbf{y}_{k+1}}{2}\right) \tag{42}$$

where $\mathbf{y}_k$ is an approximation to the exact solution $\mathbf{y}(kh)$.

*ii) Leapfrog method.* The vector field is separable in two exactly solvable parts $\mathbf{f} = \mathbf{f}_A + \mathbf{f}_B$ with $\mathbf{f}_A = (u(v-2), 0)$ and $\mathbf{f}_B = (0, v(1-u))$. If we denote by $e^{tA}$ and $e^{tB}$ the exact flows of $\dot{\mathbf{y}}_A = \mathbf{f}_A$ and $\dot{\mathbf{y}}_B = \mathbf{f}_B$, respectively, we have that

$$S_2 = e^{\frac{h}{2}A}\, e^{hB}\, e^{\frac{h}{2}A} \tag{43}$$

correspond to the well known second order leapfrog method, and it is symmetric.

The 6-th order numerical integrators considered are: the best seven-stages of [27] (NP-7) and the new seven-stages processed method (P-7). The result of the numerical experiments is shown in Fig. 3 considering as basic methods (42) and (43). As initial values we choose $(u_0, v_0) = (1, 1)$ at $t = 0$, and we integrated up to $t = 100$. The figures show the average error in the first integral $|(I(u, v) - I(u_0, v_0))|$, versus the number of evaluations

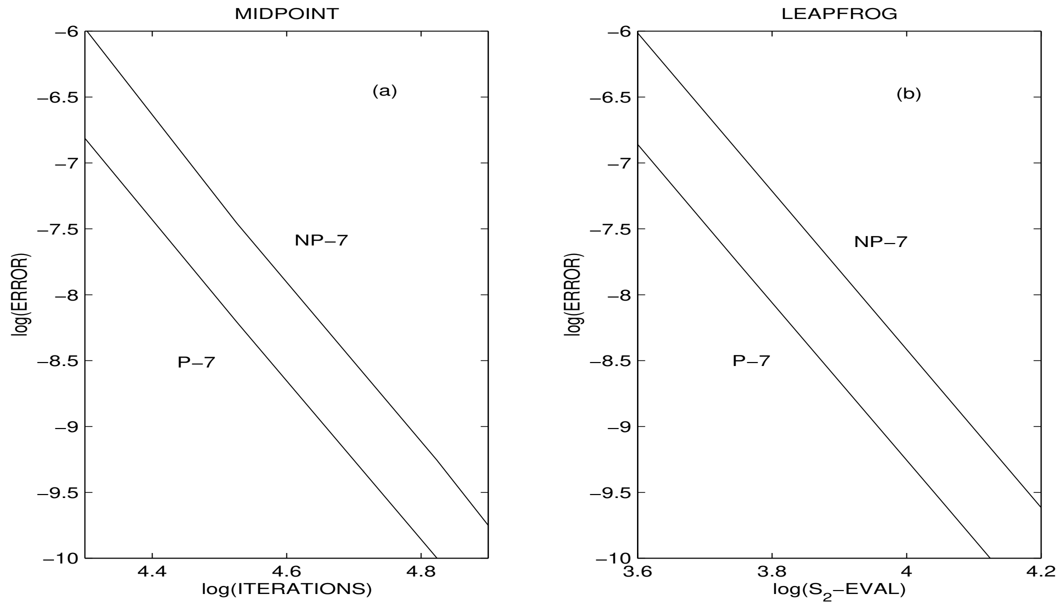


Figure 3: (a) Average error for the non-processed (NP-7) and processed (P-7) seven stages 6-th order methods: (a) vs. the number of iterations when the implicit midpoint rule is used, and (b) vs. number of $S_2$ evaluations where $S_2$ is the leap-frog.

in the Newton iteration which was used in the implementation of the midpoint rule, Fig. 3(a), and versus the number of $S_2$ evaluations in the leapfrog case, Fig. 3(b), for different time-steps. The figures agree with the results presented in Table 5. The nine-stages McL-9 stays between the two previous methods, again in agreement with the error terms in Table 5. We must say that with nine stages it could be possible to get a high order processed method or a method with smaller error coefficients.

**Example 2: Kepler problem**. Let us consider the Hamilton equations (3) associated to the Hamiltonian

$$H(\mathbf{q},\mathbf{p}) = \frac{1}{2}\left(p_x^2+p_y^2\right) - \frac{1}{\sqrt{q_x^2+q_y^2}}, \tag{44}$$

where $\mathbf{q}=(q_x,q_y)$ and $\mathbf{p}=(p_x,p_y)$. We take as initial conditions $p_x=0,\ p_y=\sqrt{\frac{1+e}{1-e}},\ q_x=1-e,\ q_y=0$, and eccentricity $e=0.5$. These correspond to an orbit of period $2\pi$ and energy $-1/2$. The Hamiltonian is separable in the kinetic and potential energy. For $S_4$ we will consider the three-stages method of [27]. Many symmetric fourth-order methods, specially tailored for RKN problems, can be used as $S_4$ [6]. The relative efficiency between the 8-th order methods has to be nearly independent of $S_4$ but, using more efficient RKN algorithms we reach machine accuracy before clearly appreciate this result. The 8-th order numerical integrators considered correspond to the seven-stages processed (P-7) and non-processed (NP-7).

We integrated the system along 10 periods and measured the average error in position

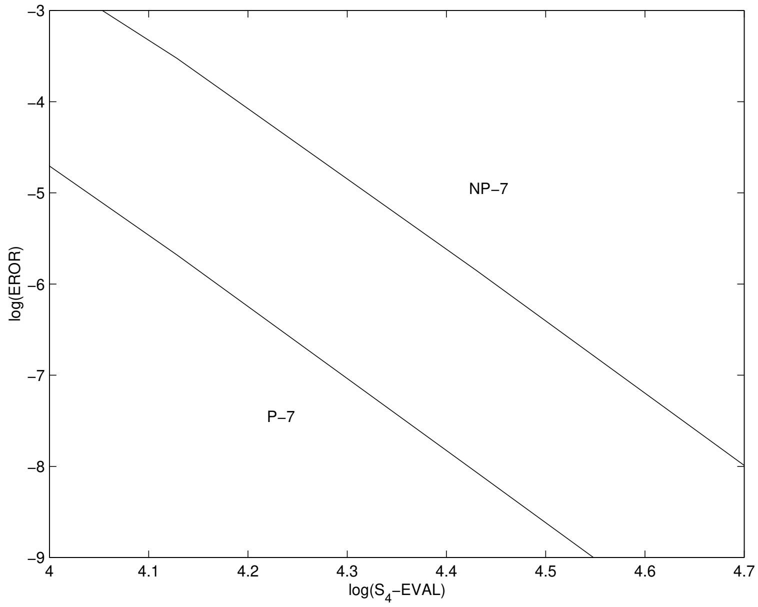


Figure 4: (a) Average error in position of the non-processed (NP-7) and processed (P-7) seven stages 8-th order methods vs. number of $S_4$ evaluations.

versus the number of evaluations of $S_4$. Fig 4 shows the results obtained which perfectly agree with the values of the error terms presented in Table 6.

# 6 Conclusions

We have shown that it is possible and very efficient to increase the order of a basic method using both the composition and the processing techniques. This efficiency comes because the kernel has to satisfy a considerable smaller number of conditions than a non-processed composition. The processor is used for solving some of the order conditions as well as for minimising the leading error terms.

If the basic method is symmetric and a symmetric compositions for the kernel is used, the number of kernel conditions is given. In this case, the whole method will be equivalent to a symmetric integrator up to a given order if the processor is chosen properly.

From the analysis of the paper it seems clear that the relative efficiency of the processed technique versus the standard composition increases with the order. For a 6-th and 8-th order methods using $S_2$ and $S_4$, respectively, we show explicitely that the improvement is already considerable so, higher order methods are very promising. These methods are technically more involved and are, at this moment, under consideration.

## Acknowledgements

The author acknowledges the Ministerio de Educación y Cultura (Spain) for a post-doctoral fellowship and the University of Cambridge for its hospitality.

## References

[1] V.I. Arnold, *Mathematical Methods of Classical Mechanics*, 2nd Ed., Springer, New York, (1989).

[2] S. Blanes, F. Casas and J. Ros, Symplectic integrators with processing: a general study, *SIAM J. Sci. Comput.*, **21** (1999), pp. 711-727.

[3] S. Blanes, F. Casas and J. Ros, Extrapolation of symplectic integrators, *Celest. Mech.* **75** (1999), pp. 149-161.

[4] S. Blanes, F. Casas and J. Ros, Processing symplectic methods for near-integrable Hamiltonian systems, Preprint (1998).

[5] S. Blanes, F. Casas and J. Ros, High-order Runge-Kutta-Nyström geometric integrators with processing, to appear.

[6] S. Blanes and P.C. Moan, Splitting Methods for non-autonomous differential Equation, DAMTP tech. report 1999/NA21, University of Cambridge (1999).

[7] 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

[8] J. Butcher y 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.

[9] M.P. Calvo y J.M. Sanz-Serna, High-order symplectic Runge-kutta-Nyström methods, *SIAM J. Sci. Comput.*, **14** (1993), pp. 1237-1252.

[10] R.P.K. Chan and A. Murua, Extrapolation of symplectic methods for Hamiltonian problems, *Appl. Numer. Math.* **34** (2000), pp. 189-205.

[11] H. Goldstein, *Classical Mechanics*, 2nd edition, Addison-Wesley, Massachusetts, 1983.

[12] A. Iserles, Composite exponential Approximations, *Math. Comp.* **38** (1982), pp.99-112.

[13] A. Iserles, Composite methods for numerical solutions of stiff systems of ODE's, *SIAM J. Num. Anal.* **21** (1984), pp. 340-351.

[14] 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.

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

[16] 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.

[17] 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.

[18] Qin Meng-Zhao and Zhu Wen-Jie, Construction of higher orther symplectic schemes by composition, *Computing*, **47** (1992),pp. 309-321.

[19] S. Reich, Symplectic integration of constrained Hamiltonian systems by composition methods, *SIAM J. Numer. Anal.* **33** (1996), pp. 475-491.

[20] G. Rowlands, A numerical algorithm for Hamiltonian systems, *J. Comput. Phys.*, **97** (1991), pp. 235-239.

[21] J.M. Sanz-Serna and M.P. Calvo, *Numerical Hamiltonian Problems* (Chapman & Hall, London, 1994).

[22] M. Suzuki, 'Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations', Phys. Lett. A **146** (1990), pp. 319-323

[23] M. Suzuki, 'General theory of fractal path integrals with applications to many-body and statistical physics', J. Math. Phys. **32** (1991), pp. 400-407

[24] M. Suzuki, 'General theory of higher-order decomposition of exponential operators and symplectic integrators', Phys. Lett. A **165** (1992), pp. 387-395

[25] Ch. Tsitouras, 'A tenth order symplectic Runge-Kutta-Nyström method', Celest. Mech. **74** (1999), pp. 223-230.

[26] J. Wisdom, M. Holman, y 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.

[27] H. Yoshida, Construction of higher order symplectic integrators, *Phys. Lett. A*, **150** (1990), pp. 262-268.