Appl. Numer. Math. 217 (2025) 372–389

Contents lists available at ScienceDirect

Applied Numerical Mathematics

journal homepage: www.elsevier.com/locate/apnum

Research Paper

Low-storage exponentially fitted explicit Runge-Kutta methods ☆

I. Higueras [a], J.I. Montijano [b], L. Rández [b,*]

[a] Instituto de Materiales Avanzados y Matemáticas (INAMAT²), Departamento de Estadística, Informática y Matemáticas, Universidad Pública de Navarra, Campus de Arrosadia, 31006 Pamplona, Spain

[b] Instituto Universitario de Matemáticas y Aplicaciones, Departamento de Matemática Aplicada, Facultad de Ciencias, Universidad de Zaragoza, Pedro Cerbuna 12, 50009 Zaragoza, Spain

ARTICLE INFO

Keywords:
Exponential fitting
Low-storage
RK methods
Oscillatory initial value problems

ABSTRACT

In this paper, we study explicit Runge-Kutta (RK) methods for solving high-dimensional systems of ordinary differential equations (ODEs), with oscillatory or periodic solutions, that can be implemented with a few memory registers. We will refer to these schemes as Low-Storage Exponentially Fitted explicit Runge-Kutta methods (LSEFRK).

In order to obtain them, we first study second-order and third-order low-storage (LS) schemes that can be implemented with two memory registers per step of the van der Houwen- and Williamson-type. Next, we construct optimal LSEFRK methods by imposing exponential fitting conditions along with accuracy and stability properties. In this way, new optimal three-stage third-order and five-stage fourth-order LSEFRK schemes are constructed for each type of LS method.

The performance of these new schemes is tested by solving some high-dimensional differential systems with periodic solutions. Comparison with other non-LS exponentially fitted and low-storage non-EF RK methods from the literature shows that the new LSEFRK schemes outperform the efficiency of RK methods that only satisfy either the LS or the EF condition.

1. Introduction

In this paper we study the numerical solution of high-dimensional ($N \gg 1$) Initial Value Problems (IVP) for ODE systems of the form

$$y'(t) = f(y(t)), \quad y(t_0) = y_0 \in R^N, \tag{1}$$

whose solutions are oscillatory or periodic, by using Exponentially Fitted explicit Runge-Kutta (EFRK) schemes. Differential systems with high dimension N arise, for example, after a spatial discretization of some Partial Differential Equations (PDEs). For simplicity, we only study autonomous problems, but the extension of results to the non-autonomous case is straightforward.

☆ Supported by Ministerio de Economía y Competividad (Spain), Project PID2022-141385NB-I00.

* Corresponding author.
E-mail address: randez@unizar.es (L. Rández).

https://doi.org/10.1016/j.apnum.2025.06.017
Received 21 February 2025; Received in revised form 4 June 2025; Accepted 30 June 2025

Available online 5 July 2025

A time step with an explicit s-stage RK method from (t_n, y_n) to (t_{n+1}, y_{n+1}), where $t_{n+1} = t_n + \Delta t$, is computed by:

$$\begin{cases} k_1 = f(y_n), \\ k_2 = f(y_n + \Delta t\, a_{21}\, k_1), \\ \vdots \\ k_s = f(y_n + \Delta t\, (a_{s1}\, k_1 + \ldots + a_{s\, s-1}\, k_{s-1})), \\ y_{n+1} = y_n + \Delta t\, \left(b_1\, k_1 + \ldots + b_s\, k_s\right), \end{cases} \tag{2}$$

where a_{ij}, $(s > i > j \geq 1)$, $b_i, i = 1, \ldots, s$, are real coefficients that define the method represented by the Butcher's tableau

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}, \quad c = A\, e, \quad e = (1, \ldots, 1)^T \in \mathbf{R}^s.$$

As a general rule, the free RK coefficients are chosen taking into consideration different stability and accuracy requirements that depend on the type of problems considered. Furthermore, for high-dimensional problems, the implementation of the RK method must also be taken into account, since this aspect has a great influence on the efficiency of the scheme. For example, a standard implementation of an s-stage RK method requires, in general, the use of $s+1$ memory registers, namely y_n, k_1, k_2, $\ldots$, k_s, of size N to complete each step. Efficiency increases if the scheme can be implemented with fewer memory registers, that is, if it is a low-storage (LS) method. In this paper, we will consider minimal storage methods, that is, methods that can be implemented with two memory registers of dimension N, having at the same time good stability and accuracy properties.

Several authors have studied low-storage methods. Among them, we may mention e.g. Williamson [17], van der Houwen [6], Calvo *et al.*, [2,3] and, more recently, Ketcheson [9]. Thus, it has been proved that, for $s = 4$, there exist no methods in the Williamson and van der Houwen families of order four.

On the other hand, some authors (see, e.g., [1], [13], [14] and the references therein) have designed and constructed numerical methods for solving ODEs with periodic or oscillating solutions. The idea of these methods is to use some known information from the solutions of the IVP (1) to construct, for this class of problems, more accurate and/or efficient schemes than the generic ones for any type of problem. In the seminal papers by Gautschi [4] and Bettis [1], exponentially fitted (EF) linear multistep methods and adapted Runge-Kutta methods, respectively, were proposed for integrating this class of differential problems. The development of EFRK methods is more recent. A detailed description along with an extensive bibliography on this topic can be seen in the book by Ixaru and Vanden Berghe [7].

One way of constructing EFRK methods consist of choosing the coefficients of the scheme so that a set of linearly independent functions, selected taking into account the properties of the solutions of the differential system, are integrated exactly by the numerical scheme. In particular, for differential systems (1) with oscillatory solutions, a relevant class of EF schemes with variable coefficients is the one that considers the set

$$\{1, t, \ldots, t^k, \exp(\pm\lambda t), t\exp(\pm\lambda t), \ldots, t^p \exp(\pm\lambda t)\}, \tag{3}$$

where $\lambda \in i\mathbb{R}$ is a certain frequency, and construct the EF method so that it exactly integrates differential problems whose exact solutions lie in the span of the set (3). These methods are expected to integrate oscillatory problems more accurately than standard schemes based on polynomial functions.

The aim of this paper is to develop optimal low-storage EFRK methods of van der Houwen and Williamson class with $s \leq 5$ stages and maximum order to be used for the numerical integration of systems of differential equations with oscillatory or periodic solutions.

The rest of the paper is organized as follows. In Section 2, we review the basic ideas of the van der Houwen and Williamson low-storage and exponentially fitted Runge-Kutta methods. In Section 3, we construct third- and fourth-order exponentially fitted low-storage methods of van der Houwen and Williamson type. Some numerical experiments showing the efficiency of the new methods compared to others from the literature are given in Section 4. The paper ends with some conclusions in Section 5.

2. Exponentially fitted low-storage RK methods

As we have pointed out above, given an N-dimensional system of ordinary differential equations, the implementation of an s-stage RK method (2) requires, as a general rule, $s+1$ memory registers of size N. If N is large, the efficiency of the RK scheme depends to a large extent on the number of memory registers used in the computations. Consequently, the efficiency of the method can be improved if it can be implemented with only two registers of dimension N and has, at the same time, good stability and accuracy properties. Methods that require only two memory registers are known in the literature as minimum storage schemes.

In [17], Williamson proposed minimum storage methods (that is, two registers of size N) that, in recent decades, have been widely used to solve problems in the context of Computational Aeroacoustics. This class of methods can be implemented in the following way:

Data: $V_1 = 0, \quad U_1 = y_n$
Result: $y_{n+1} = U_{s+1}$
for $j = 1$ **to** s **do**
  $V_{j+1} = \alpha_j V_j + f(U_j) \quad \% \ \alpha_1 = 0$
  $U_{j+1} = U_j + \Delta t \, \beta_j \, V_{j+1}$
end

In this paper we consider the case $s = 3$ stages, which has the following Butcher's tableau

$$\begin{array}{c|ccc} 0 & 0 & & \\ c_2 & \beta_1 & 0 & \\ c_3 & \beta_1 + \beta_2\alpha_2 & \beta_2 & 0 \\ \hline & \alpha_2\alpha_3\beta_3 + \alpha_2\beta_2 + \beta_1 & \alpha_3\beta_3 + \beta_2 & \beta_3 \end{array} \tag{4}$$

Another family of minimum storage methods, referred as $(2R)$-methods, were proposed by van der Houwen in [6]. It has been extensively studied in [8] to obtain optimal methods having in mind semidiscretizations of Navier–Stokes equations. The algorithm for these methods is the following.

Data: $V_1 = 0, \quad U_1 = y_n$
Result: $y_{n+1} = U_{s+1}$
for $j = 1$ **to** s **do**
  $V_{j+1} = f(U_j + \Delta t \, \alpha_j \, V_j) \quad \% \ \alpha_1 = 0$
  $U_{j+1} = U_j + b_j \, \Delta t \, V_{j+1}$
end

and its Butcher's tableau is

$$\begin{array}{c|cccc} 0 & 0 & & & \\ c_2 & b_1 + \alpha_2 & 0 & & \\ c_3 & b_1 & b_2 + \alpha_3 & 0 & \\ \vdots & \vdots & \vdots & & \ddots \\ c_s & b_1 & b_2 & \cdots & b_{s-1} + \alpha_s & 0 \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array} \tag{5}$$

On the other hand, when an autonomous IVPs (1) has periodic or oscillatory solutions, an efficient numerical solution can be obtained by means of EFRK methods. Thus, for a given fixed frequency $\omega \in i\mathbb{R}$, the IVP it can be solved by using an s-stage explicit EFRK defined by:

$$\begin{cases} k_1 = f(y_n), \\ k_2 = f(\gamma_2(\nu) y_n + \Delta t a_{21}(\nu) k_1), \\ \vdots \\ k_s = f\left(\gamma_s(\nu) y_n + \Delta t (a_{s1}(\nu) k_1 + \ldots + a_{s\,s-1}(\nu) k_{s-1})\right), \\ y_{n+1} = y_n + \Delta t \left(b_1(\nu) k_1 + \ldots + b_s(\nu) k_s\right), \end{cases} \tag{6}$$

where $\nu = \omega \Delta t$, and the coefficients $\gamma_i(\nu)$, $a_{ij}(\nu)$, $b_i(\nu)$ are real C^∞ functions in some neighborhood of $\nu = 0$. These coefficients will be selected so that the method (6) integrates exactly IVPs (1) whose exact solutions lie in the span of the set $\{\sinh(\omega t), \cosh(\omega t)\}$. It is necessary to include the additional parameters $\gamma_i(\nu)$ so that each internal stage can be fitted to the linear space $\{\sinh(\omega t), \cosh(\omega t)\}$.

In the case of standard RK methods, we have that $\gamma_i = 1$, $\forall i$ and a_{ij}, b_i are real constants. In order to construct s-stage EFRK schemes fitted to a given linear space of functions $\mathcal{F}$, defined by a basis $\{u_1, \ldots, u_q\}$, we follow the approach of Vanden Berghe et al. [16,7]. Thus, we consider the $s + 1$ linear functionals

$$L_i[u](t) \equiv u(t + c_i \Delta t) - \gamma_i u(t) - \Delta t \sum_{j=1}^{s} a_{ij} u'(t + c_j \Delta t), \ \forall u \in \mathcal{F}, \quad i = 0, 1, \ldots, s, \tag{7}$$

where $c_0 = 1$, and $a_{0j} = b_j$, for $j = 1, \ldots, s$. Then, if we choose the parameters of the method so that all linear functionals L_i vanish for the functions of $\mathcal{F}$, we may ensure that the corresponding RK will integrate exactly all IVPs with solutions in the span of the set $\mathcal{F}$.

Observe that for a given set of constant nodes c_i different between them and a basis $\mathcal{F}$, the conditions $L_i[u](t) = 0$ define $a_{ij} = a_{ij}(h)$, $\gamma_i = \gamma_i(h)$ and therefore the corresponding RK method has coefficients that depend on the step size.

Combining EF methods with low-storage properties, we propose EF Williamson algorithms given by:

Table 1
Third-order conditions.

$b^T e = 1 + \mathcal{O}(\nu^3)$,	$b^T Ae = \frac{1}{2} + \mathcal{O}(\nu^2)$,
$b^T A^2 e = \frac{1}{6} + \mathcal{O}(\nu)$,	$b^T (Ae \cdot Ae)^2 = \frac{1}{3} + \mathcal{O}(\nu)$,
$b^T \delta = \mathcal{O}(\nu^3)$	

Data: $V_1 = 0, \quad U_1 = y_n$
Result: $y_{n+1} = U_{s+1}$
for $j = 1$ **to** s **do**
  $V_{j+1} = \alpha_j(\nu) V_j + f(U_j) \quad \%\ \alpha_1 = 0$
  $U_{j+1} = \gamma_{j+1}(\nu)\, U_j + \Delta t\, \beta_j(\nu)\, V_{j+1}$
end

and for $s = 3$ its Butcher's tableau is given by

$$\begin{array}{c|c|ccc} 0 & 1 & 0 & & \\ c_2 & \gamma_2 & \beta_1 & 0 & \\ c_3 & \gamma_2\gamma_3 & \beta_1\gamma_2 + \beta_2\alpha_2 & \beta_2 & 0 \\ \hline & \gamma_2\gamma_3\gamma_4 & \alpha_2\alpha_3\beta_3 + \alpha_2\beta_2\gamma_3 + \beta_1\gamma_2\gamma_3 & \alpha_3\beta_3 + \beta_2\gamma_3 & \beta_3 \end{array} \equiv \begin{array}{c|c|c} c & \gamma & A \\ \hline & & \widehat{b}^T. \end{array}$$

From the above algorithm, it is easy to check that

$$U_{s+1} = \gamma_{s+1}\gamma_s \cdots \gamma_2\, U_1 + \Delta t \sum_{i=1}^{s} \widehat{b}_i\, k_i,$$

so the RK method exactly integrates the constant solutions of an IVP if and only if $\Pi_{i=2}^{i=s+1} \gamma_i = 1$.

For the above example with $s = 3$, the condition $\gamma_2\gamma_3\gamma_4 = 1$ must be satisfied. Note that in the above scheme, the coefficients b_i, a_{ij} and γ_i depend on ν.

The modified EFRK version for the van der Houwen [6] methods reads as follows:

Data: $V_1 = 0, \quad U_1 = y_n$
Result: $y_{n+1} = U_{s+1}$
for $j = 1$ **to** s **do**
  $V_{j+1} = f(U_j + \Delta t\, \alpha_j(\nu)\, V_j) \quad \%\ \alpha_1 = 0$
  $U_{j+1} = \gamma_{j+1}(\nu)\, U_j + b_j(\nu)\, \Delta t\, V_{j+1}$
end

and their Butcher's tableau is given by

$$\begin{array}{c|c|ccccc} 0 & 1 & 0 & & & & \\ c_2 & \gamma_2 & b_1\gamma_2 + \alpha_2 & 0 & & & \\ c_3 & \gamma_3 & b_1\gamma_3 & b_2\gamma_3 + \alpha_3 & 0 & & \\ \vdots & \vdots & \vdots & & \ddots & & \\ c_s & \gamma_s & b_1\gamma_s & b_2\gamma_s & \cdots & b_{s-1}\gamma_s + \alpha_s & 0 \\ \hline & & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array} \tag{8}$$

We have taken $\gamma_1 = 1$, $\gamma_{s+1} = 1$, $\alpha_1 = 0$ and therefore, for both EF methods (Williamson and van der Houwen), we have at our disposal $3s - 2$ free parameters, namely $\gamma_j, \alpha_j, j = 2, \ldots, s$ and $b_j, j = 1, \ldots, s$.

Observe that the simplifying assumption $Ae = c$, where $e = (1, \ldots, 1)^T$, is not satisfied (see [11]) for this type of EFRK methods because $\gamma_j(\nu) \neq 1$. However, as $Ae = c + \mathcal{O}(\nu^2)$ these methods have stage-order at least one. Consequently, the number of order conditions is larger than the one required for standard RK methods (see [12]).

In Tables 1 and 2 we show the set of third- and fourth-order conditions, respectively, for the class of EF methods considered in this paper. The dot product $\cdot$ of two vectors represents the component-wise multiplication. Note that, at orders $\{2, 3, 4, 5\}$, there are $\{0, 1, 3, 7\}$ new order conditions, respectively.

In the last row of each table, we give the order conditions relative to the ghost differential elementary (see [12]), where $\delta = \gamma - e = \mathcal{O}(\nu^2)$.

3. Construction of EF low-storage methods

In this section, we present the construction of third- and fourth-order EF low-storage methods. The first part of the section is devoted to van der Houwen methods, and the second part to Williamson ones.

Table 2
Fourth-order conditions.

$b^T e = 1 + \mathcal{O}(\nu^4)$,	$b^T Ae = \frac{1}{2} + \mathcal{O}(\nu^3)$,
$b^T A^2 e = \frac{1}{6} + \mathcal{O}(\nu^2)$,	$b^T (Ae)^2 = \frac{1}{3} + \mathcal{O}(\nu^2)$,
$b^T (Ae)^3 = \frac{1}{4} + \mathcal{O}(\nu)$,	$b^T A^3 e = \frac{1}{24} + \mathcal{O}(\nu)$,
$b^T A(Ae)^2 = \frac{1}{12} + \mathcal{O}(\nu)$,	$b^T (Ae \cdot A^2 e) = \frac{1}{8} + \mathcal{O}(\nu)$
$b^T \delta = \mathcal{O}(\nu^4)$,	$b^T (\delta \cdot Ae) = \mathcal{O}(\nu^3)$, $b^T A\delta = \mathcal{O}(\nu^3)$

3.1. LSEFRK van der Houwen methods

Third-order methods. We first develop an optimal standard van der Houwen method with three stages and third order. Imposing the algebraic conditions to get order three, and expressing the coefficients of the RK tableau in terms of the nodes $c_2 = b_1 + \alpha_2$ and $c_3 = b_1 + b_2 + \alpha_3$, we obtain the following result.

Proposition 3.1. *A three-stage third-order low-storage van der Houwen RK method exists if the real parameters c_2 and c_3 ($c_2 c_3 \neq 0$) satisfy the algebraic relation*

$$g(c_2, c_3) = 6c_2^2 c_3^2 - 6c_2^2 c_3 + 3c_2^2 - 6c_3 c_2^2 + 7c_2 c_3 - 12c_2 + 2c_3^3 - 6c_3 + 4/3 = 0. \tag{9}$$

Proof. There are three families of 3-stage third-order RK methods, namely: Case 1: a (c_2, c_3)-parametric family with $c_2 \neq 0, 2/3, c_3$, and $c_3 \neq 0$; case 2: a one parametric family with $c_2 = 2/3$, $c_3 = 0$; case 3: a one parametric family with $c_2 = c_3 = 2/3$ (see [10, p. 178] for details). With the help of a symbolic calculus program, it is possible to study when a Butcher's tableau can be expressed in the van der Houwen format (5).

1. Case 1: If $c_2 \neq 0, 2/3, c_3$, and $c_3 \neq 0$, there is a (c_2, c_3)-parametric family with

$$a_{21} = c_2, \quad a_{31} = \frac{-3c_3 c_2^2 + 3c_3 c_2 - c_3^2}{c_2 (2 - 3c_2)}, \quad a_{32} = \frac{c_3 (c_3 - c_2)}{c_2 (2 - 3c_2)},$$

and

$$b_1 = \frac{6c_2 c_3 - 3(c_2 + c_3) + 2}{6c_2 c_3}, \quad b_2 = \frac{2 - 3c_3}{6c_2 (c_2 - c_3)}, \quad b_3 = \frac{3c_2 - 2}{6c_3 (c_2 - c_3)}.$$

Then, a_{21} and a_{32} can be written as $b_1 + \alpha_2$ and $b_2 + \alpha_3$ for

$$\begin{aligned} \alpha_2 &= \frac{-6(1 - c_2) c_2 c_3 + 3(c_2 + c_3) - 2}{6c_2 c_3}, \\ \alpha_3 &= \frac{6c_3 (c_2 - c_3)^2 + 9c_2 c_3 - 6(c_2 + c_3) + 4}{6c_2 (3c_2 - 2)(c_2 - c_3)}, \end{aligned} \tag{10}$$

and $a_{31} = b_1$ if and only if c_2 and c_3 satisfy equation (9).

2. Case 2: If $c_2 = 2/3$ and $c_3 = 0$, there is an α-parametric family with

$$a_{21} = \frac{2}{3}, \quad a_{31} = -\frac{1}{4\alpha}, \quad a_{32} = \frac{1}{4\alpha},$$

and

$$b_1 = \frac{1}{4} - \alpha, \quad b_2 = \frac{3}{4}, \quad b_3 = \alpha.$$

Now, a_{21} and a_{32} can be written as $b_1 + \alpha_2$ and $b_2 + \alpha_3$ for

$$\alpha_2 = \frac{1}{12}(12\alpha + 5), \quad \alpha_3 = \frac{1 - 3\alpha}{4\alpha}, \tag{11}$$

and $a_{31} = b_1$ if and only if α is a root of the polynomial $4\alpha^2 - \alpha - 1 = 0$. It can be checked that for $c_2 = 2/3$ and $c_3 = 0$ the equation (9) holds.

3. Case 3: If $c_2 = c_3 = 2/3$, there is an α parametric family

$$a_{21} = \frac{2}{3}, \quad a_{31} = \frac{2}{3} - \frac{1}{4\alpha}, \quad a_{32} = \frac{1}{4\alpha},$$

and

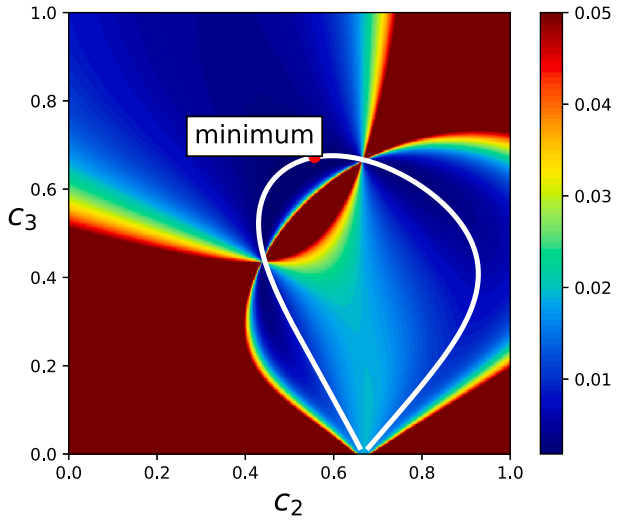

Fig. 1. Contour plot of the $PTLE^2$ and implicit plot of $g(c_2, c_3) = 0$ (white line) for the van der Houwen method.

$$b_1 = \frac{1}{4}, \quad b_2 = \frac{3}{4} - \alpha, \quad b_3 = \alpha.$$

In this family, there is a unique method with van der Houwen structure for

$$\alpha = \frac{4}{15}, \quad \alpha_2 = \frac{5}{12}, \quad \alpha_3 = \frac{4}{15}. \tag{12}$$

It can be checked that for $c_2 = c_3 = 2/3$ the equation (9) holds. □

By imposing $g(c_2, c_3) = 0$, and minimizing the Euclidean norm of the coefficients of the principal term of the local error (PTLE), we obtain the optimal values $c_2 = 0.5567690014617857$, $c_3 = 0.6724325407216719$ with $PTLE^2 = 1.94 \times 10^{-3}$. The absolute stability interval for this three-stage third-order RK is $[-2.51275, 0]$. In Fig. 1 we show the contour plot of the PTLE and the implicit plot of $g(c_2, c_3) = 0$.

In order to construct the EF version of the above method, we impose that the final stage integrates exactly the set of functions $\mathcal{F}_0 = \{1, \sinh(\omega t), \cosh(\omega t)\}$, i.e., $L_0[u](t) = 0$ for $u \in \mathcal{F}_0$. These conditions allow us to solve for $b_i(v)$ giving

$$b_1(v) = \frac{v\sinh(v(c_2 - c_3)) - \cosh(c_2 v) + \cosh(v - c_2 v) + \cosh(c_3 v) - \cosh(v - c_3 v)}{v(\sinh(v(c_2 - c_3)) - \sinh(c_2 v) + \sinh(c_3 v))},$$

$$b_2(v) = \frac{\operatorname{csch}\left(c_2 v/2\right)\operatorname{csch}\left(v/2(c_2 - c_3)\right)\left(\sinh\left(c_3 v/2\right) + \sinh\left(v - c_3 v/2\right) - v\cosh\left(\frac{c_3 v}{2}\right)\right)}{2v},$$

$$b_3(v) = \frac{\operatorname{csch}\left(c_3 v/2\right)\operatorname{csch}\left(v(c_2 - c_3)/2\right)\left(v\cosh\left(c_2 v/2\right) - 2\sinh\left(v/2\right)\cosh\left((c_2 - 1)v/2\right)\right)}{2v}.$$

In addition, we impose that each internal stage integrates exactly the set $\mathcal{F}_1 = \{\sinh(\omega t), \cosh(\omega t)\}$, i.e., $L_i[u](t) = 0$ for $u \in \mathcal{F}_0$, $i = 2, 3$, obtaining the other coefficients of the LSEFRK van der Houwen method

$$\alpha_2(v) = \cosh(c_2 v),$$

$$\alpha_3(v) = \frac{\cosh(v(c_2 - c_3))}{\cosh(c_2 v) - b_1(v)v\sinh(c_2 v)},$$

$$\gamma_2(v) = \frac{\sinh(c_2 v)}{v} - b_1(v)\cosh(c_2 v),$$

$$\gamma_3(v) = \frac{b_1(v)v\cosh(c_3 v) + b_2(v)v\cosh(v(c_2 - c_3)) - \sinh(c_3 v)}{b_1(v)v^2\sinh(c_2 v) - v\cosh(c_2 v)}.$$

By using the optimal values of the parameters c_2 and c_3 we obtain an optimized third-order EFRK method where the Taylor expansion of the coefficients is given by

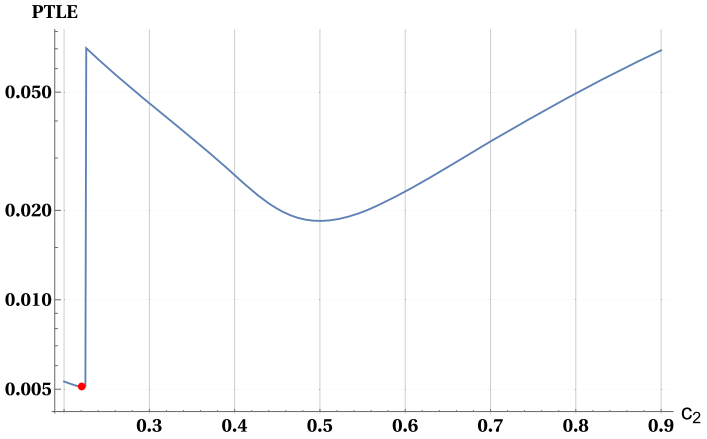

Fig. 2. Plot of the $PTLE$ for the five-stage fourth-order van der Houwen method. The minimum is located at $c_2 = 0.22231917805432730040$ with value $PTLE = 5.09 \times 10^{-3}$.

$$
\begin{aligned}
b_1(v) =& 0.24873062157793833146 - 0.00361897876281062222 69\, v^2 + \\
& 0.00005191139455381690685\, v^4 - 6.85307034739150588\, 10^{-7} v^6 + O(v^8) \\
b_2(v) =& 0.04476758131275615 6932 - 0.01853052296636087 4502\, v^2 - \\
& 0.00004345308072329284 22\, v^4 - 3.7739196003284196\, 10^{-7} v^6 + O(v^8) \\
\alpha_2(v) =& 0.70650179710930551160 + 0.02214950172917149 6729\, v^2 - \\
& 8.4583138305240647\, 10^{-6} v^4 + 1.06269899477199254\, 10^{-6} v^6 + O(v^8) \\
\alpha_3(v) =& 0.30803837988384736853 - 0.00616760780057500 9236\, v^2 - \\
& 0.00004103497466471072 45\, v^4 + 1.2947479917536500\, 10^{-7} v^6 + O(v^8) \\
b_3(v) =& 0.37893433783097741160 + 0.01003513702979541372\, v^2 + \\
& 0.00022490087039709 13\, v^4 + 6.5188239484918\, 10^{-6} v^6 + O(v^8) \\
\gamma_2(v) =& 1 + 0.15499586049437696347\, v^2 + \\
& 0.00400395279506539424656\, v^4 + 0.00004137307392333510 41462\, v^6 + O(v^8) \\
\gamma_3(v) =& 1 - 0.00982133335283973326615\, v^2 + \\
& 0.00130561602216285080\, v^4 - 0.00000838469328026526\, v^6 + O(v^8)
\end{aligned}
\tag{13}
$$

Fourth-order five-stage methods. Next, we derive the optimal low-storage RK method with five stages and order four of van der Houwen type. We have at our disposal nine parameters, namely b_1, b_i, $\alpha_i, i = 2, \ldots, 5$; by solving the eight order conditions required to reach fourth order, we get a one parameter family of methods depending on the free parameter c_2.

In Fig. 2 we show the plot of the PTLE for the standard five-stage fourth-order van der Houwen method.

The above RK method has a stability interval $I = [-4.85141, 0]$ and the Euclidean norm of the principal term of the local error is 5.09×10^{-3}.

In order to obtain the LSEFRK version of the above method, we proceed as in the third-order case. Imposing that the numerical solution fits to $\{1, t, \sinh(\omega t), \cosh(\omega t)\}$ and the internal stages integrate exactly the set $\{\sinh(\omega t), \cosh(\omega t)\}$, we obtain the coefficients $\gamma_i(v), b_i(v), \alpha_i(v)$. In Appendix A, the Taylor series up to order eight of these coefficients are provided with twenty-digit precision. Note that for the value $v = 0$ we obtain a standard fourth-order RK method.

3.2. LSEFRK Williamson methods

For third-order three-stage methods, by imposing the third-order conditions, we can state the following result:

Proposition 3.2. *A three-stage third-order low-storage Williamson RK method exists if the real parameters α_2, β_1 $(\alpha_2 \beta_1 \neq 0)$ satisfy the algebraic relation*

$$
\hat{g}(\alpha_2, \beta_1) = 1 + \alpha_2^2 - 6\beta_1 + 15\beta_1^2 - 24\beta_1^3 + 18\beta_1^4 + \alpha_2(2 - 6\beta_1 + 3\beta_1^2 + 6\beta_1^3) = 0. \tag{14}
$$

Proof. The proof is similar to that of Proposition 3.1. For each case, we study when a Butcher's tableau can be expressed in the Williamson format (4).

1. Case 1: If $c_2 \neq 0, 2/3, c_3$, and $c_3 \neq 0$, there is a (c_2, c_3)-parametric family with

$$\beta_1 = c_2, \qquad \beta_2 = \frac{c_3\left(c_2 - c_3\right)}{c_2\left(3c_2 - 2\right)}, \qquad \beta_3 = \frac{3c_2 - 2}{6c_3\left(c_2 - c_3\right)},$$

and

$$\alpha_2 = \frac{-3c_2^2 + 2c_2 - c_3}{c_3},$$
$$\alpha_3 = \frac{c_3\left(-6c_3c_2^2 - \left(-12c_3^2 + 9c_3 - 6\right)c_2 - 6c_3^3 + 6c_3 - 4\right)}{c_2\left(3c_2 - 2\right)^2},$$

where β_1 and α_2 must satisfy equation (14).
2. Case 2: If $c_2 = 2/3$ and $c_3 = 0$, there is a unique method, namely

$$\alpha = -\frac{1}{3}, \quad \beta_1 = \frac{2}{3}, \quad \beta_2 = -\frac{3}{4}, \quad \beta_3 = -\frac{1}{3}, \quad \alpha_2 = -\frac{1}{9}, \quad \alpha_3 = -\frac{9}{2}.$$

It can be checked that β_1 and α_2 satisfy equation (14).
3. Case 3: If $c_2 = c_3 = 2/3$, there is a unique method, namely

$$\alpha = \frac{1}{3}, \quad \beta_1 = \frac{2}{3}, \quad \beta_2 = \frac{3}{4}, \quad \beta_3 = \frac{1}{3}, \quad \alpha_2 = -1, \quad \alpha_3 = -1.$$

It can be checked that β_1 and α_2 satisfy equation (14). □

Taking into account the relation $\hat{g}(\alpha_2, \beta_1) = 0$ and minimizing the principal term of the local error (PTLE), we get the optimal values

$$\alpha_2 = -0.73421135582465879426, \ \beta_1 = 0.53917676636979229137,$$

with $PTLE^2 = 1.78 \times 10^{-3}$. The absolute stability interval for this three-stage third-order RK is $[-2.51275, 0]$. In Fig. 3 we show the contour plot of the PTLE and the implicit plot of $\hat{g}(\alpha_2, \beta_1) = 0$. For the optimal values of β_1 and α_2, the corresponding values of the nodes of the RK are $c2 = \beta_1$ and $c_3 = 0.77587504613095876934$.

Like in the case of van der Houwen methods, we impose the exponential fitting conditions. For the second stage, we get:

$$\beta_1 = \sinh(c_2 v)/v, \quad \gamma_2 = \cosh(c_2 v)\,.$$

For the third stage it is necessary to solve the system

$$\begin{aligned} &-\beta_2 v \sinh(c_2 v) + \cosh(c_3 v) - \gamma_2\gamma_3 = 0\,, \\ &-\alpha_2\beta_2 - \beta_1\gamma_3 - \beta_2 \cosh(c_2 v) + \frac{\sinh(c_3 v)}{v} = 0\,. \end{aligned} \tag{15}$$

For the final stage, the equations are:

$$\begin{aligned} &\gamma_3(\gamma_2(\alpha_2\alpha_3\beta_3 + \alpha_3\beta_3 + \beta_3 - 1) + \beta_1) + (\alpha_2 + 1)\beta_2 = 0\,, \\ &-v\left(\sinh(c_2 v)\left(\alpha_3\beta_3 + \frac{\beta_2}{\gamma_2\gamma_3}\right) + \beta_3 \sinh(c_3 v)\right) + \cosh(v) - 1 = 0\,, \\ &\gamma_2\gamma_3 \sinh(v) = v(\alpha_2\alpha_3\beta_3\gamma_2\gamma_3 + \alpha_2\beta_2 + \cosh(c_2 v)(\alpha_3\beta_3\gamma_2\gamma_3 + \beta_2) + \beta_1\gamma_3 + \beta_3\gamma_2\gamma_3 \cosh(c_3 v))\,. \end{aligned} \tag{16}$$

Solving the five equations in (15)–(16), for the values of the coefficients β_1 and α_2 of the optimal standard method, we obtain the third-order EFRK whose coefficients are given by the following expansions:

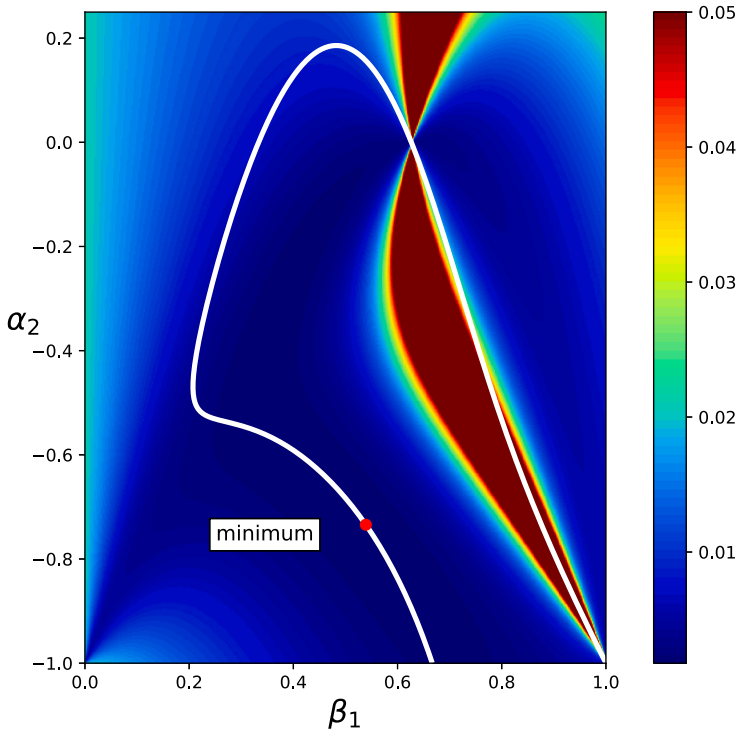

Fig. 3. Contour plot of the $PTLE^2$ and the implicit plot of $\hat{g}(\alpha_2, \beta_1) = 0$ (white line) for the Williamson method.

$$
\begin{aligned}
c_2 &= 0.5391767663697922 9137\\
c_3 &= 0.7758750461309587 6934\\
\beta_1(v) &= 0.539176766369792291 + 0.026124155426404284534055\, v^2 +\\
&\quad 0.000379729732053137 8385\, v^4 + 2.6283769625243170028\, 10^{-6} v^6 + O(v^8)\\
\beta_2(v) &= 0.8905507626014911 06158 + 0.004380999953957267 31344\, v^2 +\\
&\quad 0.000142943255140789142634\, v^4 - 1.0915762879538515 9531 \times 10^{-6} v^6 +\\
&\quad 4.872771579437670677 67\times 10^{-8} v^8 + O(v^8)\\
\beta_3(v) &= 0.347103425736609543037 + 0.007082753901921118 54465\, v^2 -\\
&\quad 0.000046520056575514 8088868\, v^4 + 8.787045542755305 03569 \times 10^{-7} v^6 -\\
&\quad 1.326023749124785799 20\times 10^{-8} v^8\\
\alpha_2(v) &= -0.734211355824658794258 + 0.107896194768588018121\, v^2 -\\
&\quad 0.013139383090113253 8709\, v^4 + 0.001555346040924732 82591\, v^6 -\\
&\quad 0.000183369642283302 811147\, v^8 + O(v^8)\\
\alpha_3(v) &= -1.333011114312550 86975 - 0.455944677921104940896\, v^2 -\\
&\quad 0.102945640386477642171\, v^4 - 0.023584448286987648 7818\, v^6 -\\
&\quad 0.005420002145506150 82644\, v^8 + O(v^8)\\
\gamma_2(v) &= 1 + 0.145355792696492789 20\, v^2 +\\
&\quad 0.003521384411737631 10970\, v^4 + 0.000034123574837146 4212742\, v^6 +\\
&\quad 1.771449739318397471 87\times 10^{-7} v^8 + O(v^8)\\
\gamma_3(v) &= 1 - 0.324529029559758619102\, v^2 +\\
&\quad 0.033123038062006344 1470\, v^4 - 0.003932664807969356 28762\, v^6 +\\
&\quad 0.000462000557975788 049956\, v^8 + O(v^8)
\end{aligned}
\tag{17}
$$

Fourth-order five-stage methods: For the derivation of fourth-order five-stage low-storage EFRK methods, we first derive a standard fourth-order method by minimizing the Euclidean norm of the PTLE. Having in mind that there exists a family of fourth-order methods depending on the parameter β_1, we try to compute $PTLE$ against β_1, but in this case, we obtain several methods for every value of β_1.

Table 3
Comparison of third-order methods. ASI is the absolute stability interval and $\|\cdot\|_2$ is the Euclidean norm of the principal term of the local error.

Method	stages	$\|\cdot\|_2$	ASI
VdH3 (van der Houwen)	3	4.41×10^{-2}	−2.51275
Will3 (Williamson)	3	5.02×10^{-2}	−2.51275
RK3	3	5.89×10^{-2}	−2.51275

After an optimization process taking into account a balance between absolute stability, length of the absolute stability interval, and accuracy requirements, we select the coefficient $\beta_1 = 0.141871118\ldots$ because for this value $c_5 = 1$, which is near the optimal value. So, the principal term of the local error is $PTLE^2 = 2.72\times10^{-5}$ and the absolute stability interval $[-4.19877, 0]$.

In order to derive the coefficients of the EF version of the optimal standard method, we impose that all internal stages are fitted to the set of functions $\{\sinh(\omega t), \cosh(\omega t)\}$ and the final stage to $\{1, t, \sinh(\omega t), \cosh(\omega t)\}$. It is an easy task for the second stage but, for the rest of the stages, it is very hard to get a solution analytically.

So, we set each coefficient of the RK method as a Taylor series only with even powers, for instance,

$$\beta_2(\nu) = \beta_{20} + \beta_{22}\nu^2 + \beta_{24}\nu^4 + \ldots,$$

and we extract the Taylor's expansion of the terms substituting in (7).

In Appendix B, the Taylor series in even powers of the coefficients $\alpha_i(\nu)$, $\beta_i(\nu)$, $\gamma_i(\nu)$ are provided with twenty-digit precision up to eighth order. Note that for $\nu = 0$ we obtain a standard fourth-order RK method.

4. Numerical examples

In this section we perform some numerical experiments to test the efficiency of the new integrators obtained in the previous sections when they are used to solve numerically some oscillatory systems. The new third- and fourth-order schemes are compared with the well-known classical three-stage third-order RK method, and the standard fourth-order schemes developed by Calvo et al. [3] and Ketcheson [9], respectively.

For numerical comparisons, we consider the maximum global error in the solution over the entire integration interval versus the computational cost measured by the number of steps required by each method.

All experiments are conducted on Python 3, and this study runs the experiments on a PC with a processor Intel(R) Core(TM) i7, 32 GB RAM, and Ubuntu 24.

4.1. Third-order methods

Here, we present the results obtained with the following third-order methods:

- **EFvdH3**: The exponentially fitted van der Houwen method (13) developed in the section 3.1,
- **VdH3**: The *standard* version of the above method,
- **EFWill3**: The exponentially fitted Williamson 3-stage, third-order method (17) developed in the section 3.2,
- **Will3**: The *standard* version of the above method,
- **RK3**: The 3-stage, third-order RK formula based on the Simpson's rule:

$$\begin{array}{c|ccc} 0 & 0 & & \\ 1/2 & 1/2 & 0 & \\ 1 & -1 & 2 & 0 \\ \hline & 1/6 & 4/6 & 1/6 \end{array}$$

In Table 3 we give the main properties of the standard RK methods used in the numerical tests.

Next, we use the above methods to solve Duffing's equation and a perturbed Kepler problem.

Duffing's equation. We consider the IVP:

$$y''(t) + (\lambda^2 + k^2)y(t) = 2k^2 y(t)^3, \quad t \in [0, 40],$$
$$y(0) = 0, \quad y'(0) = \lambda,$$

where $\lambda = 5$ and $k = 0.035$. The exact solution is

$$y(t) = \mathrm{sn}\left(\lambda t, (\tfrac{k}{\lambda})^2\right),$$

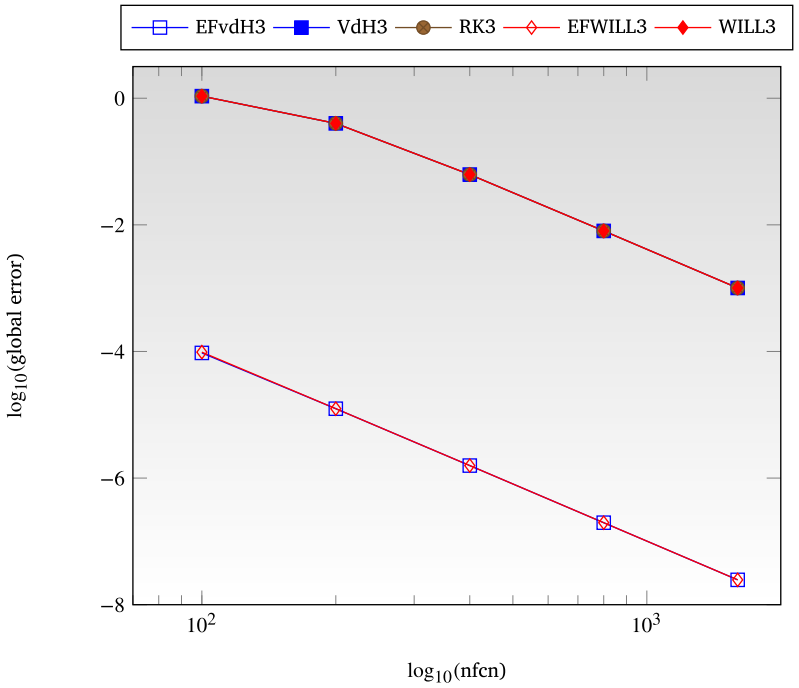

Fig. 4. Efficiency of third-order methods for the Duffing's equation.

where sn denotes the elliptic Jacobi function. We choose the fitting frequency $\omega = 5i$ in the EF methods and compute the numerical solution with time steps $\Delta t = 1/(5 \times 2^m)$, $m = 0, \dots, 4$.

In Fig. 4 we show the efficiency of the third-order methods considered in this section. We obtain that the exponentially fitted van der Houwen (13) and Williamson (17) methods constructed in this paper have the same efficiency and outperform the rest of third-order methods under consideration. Also, the efficiency of the non-fitted schemes is the same for this problem.

Perturbed Kepler's problem. Given the perturbed Hamiltonian function

$$H(p,q) = \frac{1}{2}\left(p_1^2 + p_2^2\right) - \left(q_1^2 + q_2^2\right)^{-1/2} - (2\varepsilon + \varepsilon^2)/(3\left(q_1^2 + q_2^2\right)^{3/2}),$$

we consider the differential system associated to it with initial conditions:

$$q_1(0) = 1, \quad q_2(0) = 0, \quad p_1(0) = 0, \quad p_2(0) = 1 + \varepsilon,$$

where $0 < \varepsilon \ll 1$. The exact solution is given by

$$q_1(t) = \cos(t + \varepsilon t), \quad q_2(t) = \sin(t + \varepsilon t), \quad p_i(t) = q_i'(t), \ i = 1, 2.$$

The numerical results have been computed for $\varepsilon = 10^{-3}$ and the fitting frequency $\omega = i$. We have used the integration steps $\Delta t = \frac{\pi}{10 \times 2^m}$, $m = 0, \dots, 5$, and the problem is integrated up to $t_{end} = 10\pi$.

The efficiency plots for the third-order methods are shown in Fig. 5. We observe that the most efficient method turns out to be the EF Williamson method (17), and it is followed by the van der Houwen method (13). Both of them outperform the standard Williamson and van der Houwen methods.

4.2. Fourth-order methods

In this section we test the following fourth-order methods:

- **EFvdH5**: The EF van der Houwen (18) method developed in section 3.1,
- **VdH5**: The *standard* version of the method EFVdH5,
- **EFWill5**: The EF Williamson (19) method developed in the section 3.2,
- **Will5**: The *standard* version of the method EFWill5,
- **Cal64**: The 6-stage fourth-order low-storage method in article by Calvo et al. [3],
- **Ket64**: The 6-stage fourth-order low-storage method in the article by Ketcheson [9].

In Table 4 we show the main properties of the standard RK methods used in the numerical comparisons.

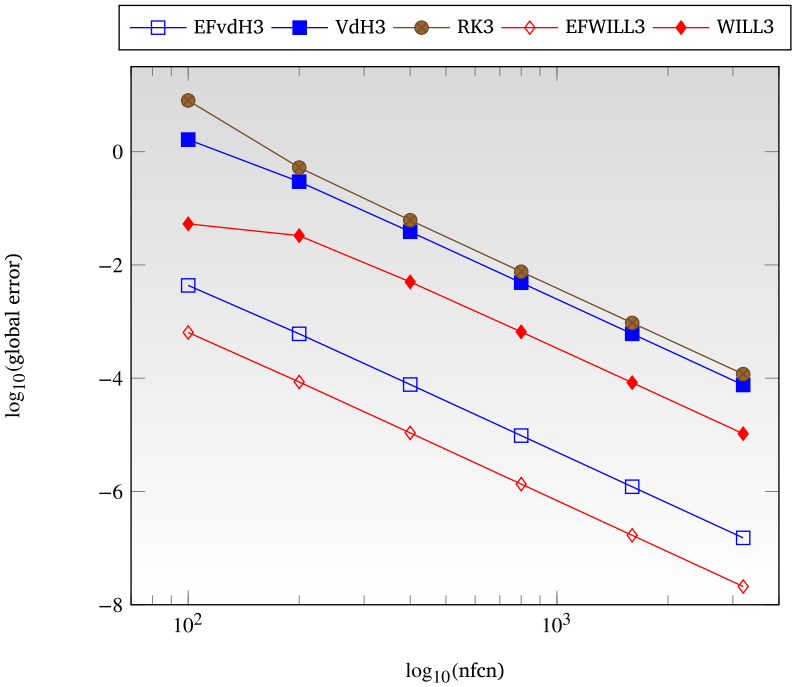

Fig. 5. Efficiency of third-order methods for the perturbed Kepler's equation.

Table 4
Comparison of fourth-order methods. ASI is the absolute stability interval and $\|\cdot\|_2$ is the Euclidean norm of the principal term of the local error.

Method	stages	$\|\cdot\|_2$	ASI
VdH5 (van der Houwen)	5	5.11×10^{-3}	−4.9016
Will5 (Williamson)	5	5.22×10^{-3}	−4.1987
Ket64[a]	6	4.16×10^{-3}	−6.3008
Cal64	6	2.86×10^{-3}	−2.1261

[a] In this method $c_6 > 1$.

We check these fourth-order methods on the following IVPs:

Duffing's equation. We consider again the Duffing's equation used in Section 4.1 and the integration steps $\Delta t = 1/(5\times 2^m)$, $m = 1,\dots,5$. In Fig. 6 we show the efficiency plots for this problem. We again observe that the EF van der Houwen (18) and Williamson (19) methods derived in this paper outperform the rest of methods.

Perturbed Kepler's problem. We use integration steps $\Delta t = 1/(10\pi\times 2^m)$, $m = 0,\dots,4$ and its efficiency plot is depicted in Fig. 7. Again, the EF methods are the best ones.

Inhomogeneous linear system. We consider the wave equation [15] given by

$$\frac{\partial^2 x}{\partial t^2} = 4\frac{\partial^2 x}{\partial r^2} + \sin(t)\cos\left(\frac{\pi x}{100}\right), \quad 0\le r\le 100, \quad t\in[0,40\pi],$$
$$\frac{\partial x}{\partial r}(t,0) = \frac{\partial x}{\partial r}(t,100) = 0,$$
$$x(0,r) = 0, \quad \frac{\partial x}{\partial t}(0,r) = \frac{100^2}{4\pi^2 - 100^2}\cos\left(\frac{\pi x}{100}\right),$$

whose exact solution is

$$x(t,r) = \frac{100^2}{4\pi^2 - 100^2}\sin(t)\cos\left(\frac{\pi x}{100}\right).$$

The spatial derivative $\partial^2 x/\partial r^2$ is discretized with fourth-order symmetric differences and one-sided finite differences of the same order at the internal and boundary nodes, respectively. We set the spatial step $\Delta r = 100/n$, with $n = 20$ obtaining the following linear system with constant coefficients

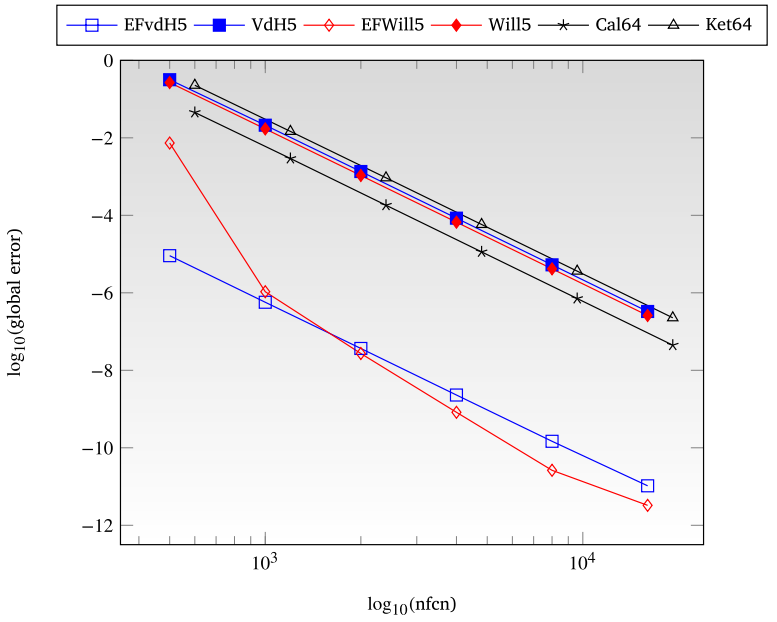

Fig. 6. Efficiency of fourth-order methods for the Duffing's problem.

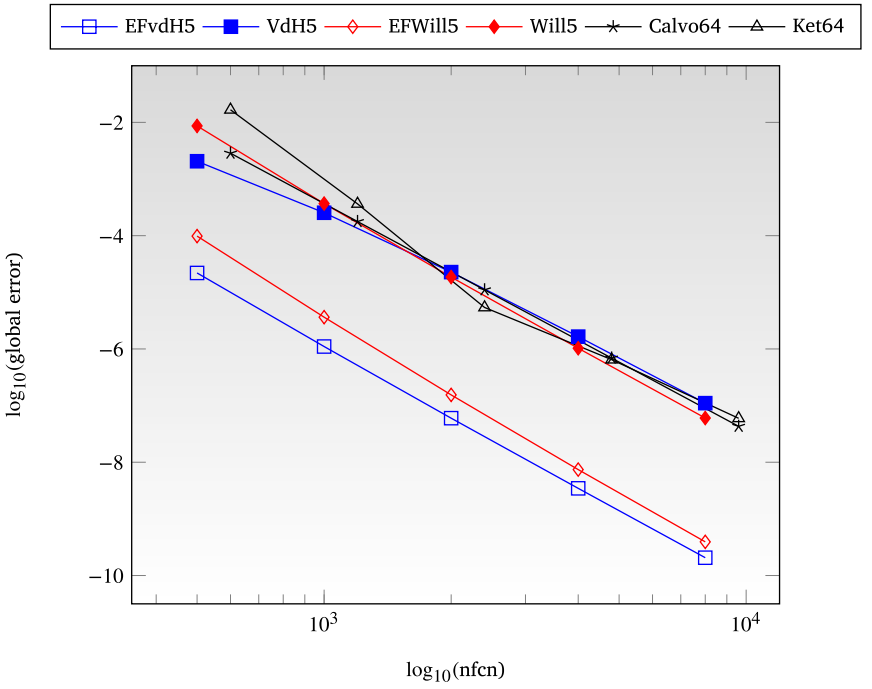

Fig. 7. Efficiency of fourth-order methods for the Perturbed Kepler problem.

$$\begin{bmatrix} x_1''(t) \\ x_2''(t) \\ \vdots \\ x_{n+1}''(t) \end{bmatrix} = \frac{4}{(\Delta r)^2} \begin{bmatrix} -\frac{415}{72} & 8 & -3 & \frac{8}{9} & -\frac{1}{8} & 0 & \cdots & 0 \\ \frac{257}{144} & -\frac{10}{3} & \frac{7}{4} & -\frac{2}{9} & \frac{1}{48} & 0 & \cdots & \vdots \\ -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} & & & \vdots \\ 0 & & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & & -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} \\ \vdots & \cdots & 0 & \frac{1}{48} & -\frac{2}{9} & \frac{7}{4} & -\frac{10}{3} & \frac{257}{144} \\ 0 & \cdots & 0 & -\frac{1}{8} & \frac{8}{9} & -3 & 8 & -\frac{415}{72} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n+1}(t) \end{bmatrix}$$

Table 5
CPU times for the five-stage Williamson Runge-Kutta implemented as a standard RK (CPU_1) and with a two-register low-storage implementation (CPU_2).

n	CPU_1	CPU_2	ratio
500000	32.72	27.27	1.199
1000000	66.97	53.61	1.249
2000000	145.61	121.00	1.203
4000000	308.28	250.40	1.231

$$+\sin(t)\begin{bmatrix}\cos\left(\frac{0\cdot\Delta r}{100}\cdot\pi\right)\\ \cos\left(\frac{1\cdot\Delta r}{100}\cdot\pi\right)\\ \vdots\\ \cos\left(\frac{n\cdot\Delta r}{100}\cdot\pi\right)\end{bmatrix},$$

where

$$x_1(t)\approx x(t,0),\quad x_2(t)\approx x(t,\Delta r),\quad \dots\quad x_{21}(t)\approx x(t,20\Delta r)\,.$$

In order to solve the above problem, we reduce it to first-order by doubling its dimension. The stepsize time used in the numerical experiments is $h=4\pi/(3\times 2^i)$, $i=1,\dots,4$, and the fitting frequency is $w=i$. To compute the global error in each step, we have used the code DOPRI853 [5] at stringent tolerance.

Fermi-Pasta-Ulam problem. The Hamiltonian function for this problem is given by

$$H=\frac{1}{2}\sum_{i=1}^{n}(u_i'^2+v_i'^2+\lambda^2 v_i^2)+\frac{1}{4}(u_1-v_1)^4$$
$$+\frac{1}{4}\sum_{i=1}^{n-1}(u_{i+1}-v_{i+1}-u_i-v_i)^4+\frac{1}{4}(u_n+v_n)^4,$$

where $u_{n+1}=v_{n+1}=0$. We consider zero initial conditions except for the variables

$$u_1(0)=1,\ u_1'(0)=1,\ v_1(0)=1/\lambda,\ v_1'(0)=1.$$

The numerical results have been computed with the integration steps $\Delta t=1/(25\times 2^m)$, $m=1,\dots,4$. We take the parameter values $\lambda=50$, the fitting frequency $\omega=i\lambda$ and $n=100$, obtaining a system of dimension $N=400$. The problem is integrated up to $t_{end}=100$.

In order to check the performance of the fourth-order methods, we consider the maximum error in the Hamiltonian over the entire interval versus the computational cost measured by the number of function evaluations required by each method.

In Figs. 7, 8 and 9 we show the efficiency plots of all fourth-order methods for the perturbed Kepler, the Fermi-Pasta-Ulam and the inhomogeneous linear system problems respectively. Once more, we observe the gain obtained when EF methods are used. For both problems, the EF van der Houwen method is the best method among the considered fourth-order ones considered in this section.

To show the efficiency of low-storage schemes, we compare two different implementations of the five-stage Williamson method (see section 2). The first uses only two registers, while the second follows a standard Runge-Kutta approach, allocating a separate register for each stage. For this comparison, we consider again the Fermi-Pasta-Ulam problem with parameters $t_{end}=1$, $\lambda=50$, $h=10^{-2}$. Table 5 presents the problem size, the CPU times (seconds) for each implementation, and the performance ratio between them. Since we use the same method but with different implementation, the error is the same for both methods. Similar results are obtained with the van der Houwen method (see Section 2). We can observe that for the size of this problem, the efficiency in the low-storage implementation is about 20%.

5. Conclusions

In this paper, we have constructed four new low-storage EF methods of van der Houwen- and Williamson-type with algebraic orders three and four, and some numerical experiments have been carried out on some representative differential problems. Taking into account the results obtained, we can conclude that the new methods are more efficient than the corresponding standard ones and, for this class of problems, perform better than the schemes constructed by Calvo et al. [3] and Ketcheson [9]. Additionally, they are better suited for solving high-dimensional oscillatory problems whose frequency is approximately known in advance.

CRediT authorship contribution statement

I. Higueras: Writing – review & editing, Investigation, Writing – original draft. **J.I. Montijano:** Writing – original draft, Writing – review & editing, Investigation. **L. Rández:** Writing – original draft, Funding acquisition, Writing – review & editing, Investigation.

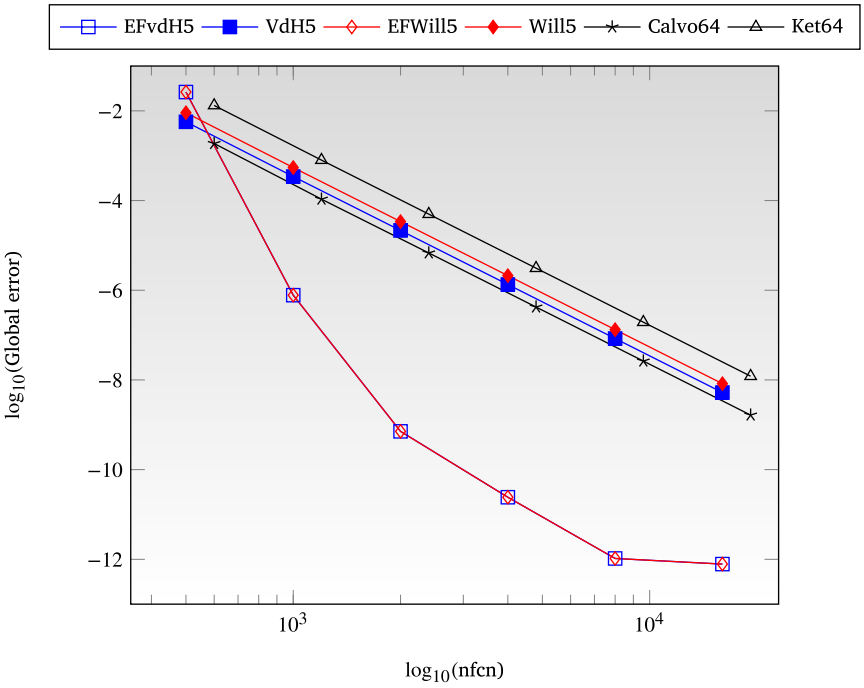

Fig. 8. Efficiency of fourth-order methods for the inhomogeneous linear system.

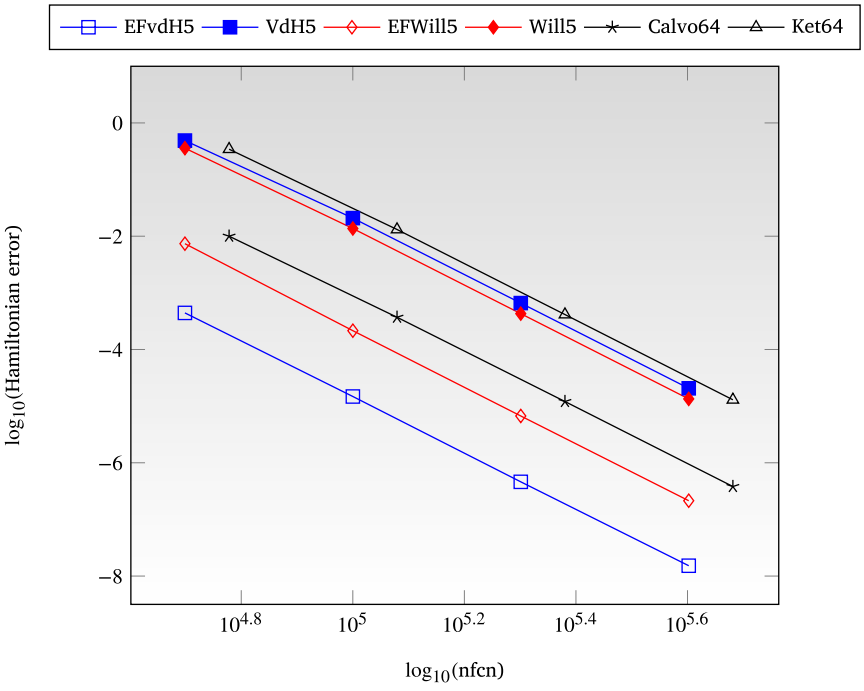

Fig. 9. Efficiency of fourth-order methods for the Fermi-Pasta-Ulam problem.

Appendix A

In this section we show the Taylor series of the fourth-order five-stage LSEFRK method of van der Houwen type constructed in Section 3.1.

$$
\begin{aligned}
b_1(v) = {} & 0.052424337166216375236 \\
b_2(v) = {} & 0.370580843482625928297 + 0.000731029688264186137240\, v^2 + \\
& 1.08855759513526779403\, 10^{-7}\, v^4 - 1.337637738358268045 99\, 10^{-8}\, v^6 + \\
& 1.17699884850564259905\, 10^{-11}\, v^8 + \mathcal{O}(v^{10}) \\
b_3(v) = {} & -0.288332305928004570489 - 0.000051254706605866272 35\, v^2 - \\
& 0.000022695258607845917446\, v^4 - 9.2493258381505681\, 10^{-8}\, v^6 - \\
& 2.330909089677 54\, 10^{-11}\, v^8 + \mathcal{O}(v^{10}) \\
b_4(v) = {} & 0.51540838214412296 - 0.00165282871657485\, v^2 + \\
& 0.00001946928495167\, v^4 + 1.11434498995\, 10^{-7}\, v^6 - \\
& 7.0731376\, 10^{-12}\, v^8 + \mathcal{O}(v^{10}) \\
b_5(v) = {} & 0.349918743135039307918 + 0.000973053734916531044\, v^2 + \\
& 3.11711789666624182\, 10^{-6}\, v^4 - 5.56486323020453\, 10^{-9}\, v^6 + \\
& 1.8612239983284\, 10^{-11}\, v^8 + \mathcal{O}(v^{10})
\end{aligned}
\tag{18}
$$

$$
\begin{aligned}
\gamma_2(v) = {} & 1 + 0.024712816747569319970 5\, v^2 + \\
& 0.000101787218599823777035\, v^4 + 1.676965920334816283 10\, 10^{-7}\, v^6 + \\
& 1.480091124326901638 44\, 10^{-10}\, v^8 + \mathcal{O}(v^{10}) \\
\gamma_3(v) = {} & 1 + 0.063419513515272890056 4\, v^2 + \\
& 0.000140894722316347228218\, v^4 + 2.833308166369463471 18\, 10^{-6}\, v^6 - \\
& 1.969332683856905775 50\, 10^{-8}\, v^8 + \mathcal{O}(v^{10}) \\
\gamma_4(v) = {} & 1 - 0.010426186233889661302\, v^2 + \\
& 0.000213221375449530425 4\, v^4 - 3.840741633273026 99\, 10^{-6}\, v^6 + \\
& 6.539930091339183 63\, 10^{-8}\, v^8 + \mathcal{O}(v^{10}) \\
\gamma_5(v) = {} & 1 + 0.041442598205409945317\, v^2 + \\
& 0.000514695065799405947\, v^4 + 6.8184485186555137\, 10^{-6}\, v^6 + \\
& 9.194516794358457\, 10^{-8}\, v^8 + \mathcal{O}(v^{10})
\end{aligned}
$$

$$
\begin{aligned}
\alpha_2(v) = {} & 0.169894428337594930766 + 0.000535821266312348700117\, v^2 - \\
& 8.10285710452651775915\, 10^{-7}\, v^4 - 3.4653684938058310759 0\, 10^{-9}\, v^6 - \\
& 4.10314592947150079330\, 10^{-12}\, v^8 + \mathcal{O}(v^{10}) \\
\alpha_3(v) = {} & 0.190407988654597441969 - 0.002952680056810281445\, v^2 + \\
& 0.000034672881165197863 79\, v^4 - 3.668133012530613 40\, 10^{-7}\, v^6 + \\
& 4.47216406773553604\, 10^{-9}\, v^8 + \mathcal{O}(v^{10}) \\
\alpha_4(v) = {} & 0.443475608212419978750 - 0.005413840988819886359\, v^2 + \\
& 0.000126317710032767670 7\, v^4 - 1.98671160935421986\, 10^{-6}\, v^6 + \\
& 3.5691414758717143\, 10^{-8}\, v^8 + \mathcal{O}(v^{10}) \\
\alpha_5(v) = {} & 0.19725027235042413 + 0.0014082307211743\, v^2 + \\
& 0.000025526267491 20\, v^4 + 2.9800532375\, 10^{-7}\, v^6 + \\
& 4.487915381\, 10^{-9}\, v^8 + \mathcal{O}(v^{10})
\end{aligned}
$$

Appendix B

In this section we show the Taylor series of the fourth-order five-stage LSEFRK method of Williamson type constructed in section 3.2.

$$
\begin{aligned}
\alpha_2(v) = {} & -0.606619442246971406 94 + 0.016744454826839299 05\, v^2 - \\
& 0.002088223188892940 97\, v^4 + 0.000026767889826936 82\, v^6 -
\end{aligned}
$$

$$
\begin{aligned}
&0.00007745662355005192\,\nu^8 + O(\nu^{10})\\
\alpha_3(\nu) &= -2.97023307150389229342 - 0.57451342416039459809\,\nu^2 -\\
&0.15015422219003597823\,\nu^4 - 0.03127897243094950391\,\nu^6 -\\
&0.00954582722350558500\,\nu^8 + O(\nu^{10})\\
\alpha_4(\nu) &= -0.66869682611209047736 + 0.96035559033884651690\,\nu^2 -\\
&0.45506511029045194521\,\nu^4 + 0.13117368674210919327\,\nu^6 -\\
&0.01714834477789480876\,\nu^8 + O(\nu^{10})\\
\alpha_5(\nu) &= -0.89877145606357789537\\
\beta_1(\nu) &= 0.26958221718999705470 + 0.0032652953676562233569\,\nu^2 +\\
&0.00001186519713634080229 3\,\nu^4 + 2.05309076548493047683\,10^{-8}\nu^6 +\\
&2.07232628193933702888\,10^{-11}\nu^8 + O(\nu^{10})\\
\beta_2(\nu) &= 0.92031885308133836441 + 0.03250011465200184698\,\nu^2 +\\
&0.00434376781032999685 50\,\nu^4 + 0.00029923689672164461202\,\nu^6 +\\
&0.00020561539581064686640\,\nu^8 + O(\nu^{10})\\
\beta_3(\nu) &= 0.40552418761878786893 - 0.26689708990948891042\,\nu^2 +\\
&0.10191260035211000112\,\nu^4 - 0.02890972555589863693 2\,\nu^6 +\\
&0.00300849469539424409 84\,\nu^8 + O(\nu^{10})\\
\beta_4(\nu) &= 0.39248043180810899723 + 0.04225025125645926993\,\nu^2 +\\
&0.00763178981686994849 35\,\nu^4 + 0.00031978903679079613 51\,\nu^6 +\\
&0.00029580831519017824291\,\nu^8 + O(\nu^{10})\\
\beta_5(\nu) &= 0.14028607469781692707 - 0.00006564973996063854 3330\,\nu^2 +\\
&0.00016953884247572404 87\,\nu^4 - 1.61816012074038833941\,10^{-6}\,\nu^6 +\\
&7.80192049938729717733\,10^{-6}\,\nu^8 + O(\nu^{10})\\
\gamma_1(\nu) &= 1 + 0.03633728591253737174\,\nu^2 + 0.0002200663912482478222 5\,\nu^4 +\\
&5.33107691901859543190\,10^{-7}\,\nu^6 + 6.91845950814669795651\,10^{-10}\,\nu^8 + O(\nu^{10})\\
\gamma_2(\nu) &= 1 - 0.08496838488952350417\,\nu^2 - 0.00226769849124231098 54\,\nu^4 -\\
&0.00109929245546156227 13\,\nu^6 - 0.00005413996885408564 91\,\nu^8 + O(\nu^{10})\\
\gamma_3(\nu) &= 1 + 0.02776700040614306880\,\nu^2 - 0.00339843040334329475 47\,\nu^4 +\\
&0.00181781369798895982 50\,\nu^6 - 0.00021144857465788957 06\,\nu^8 + O(\nu^{10})\\
\gamma_4(\nu) &= 1 + 0.07586695314554232686\,\nu^2 + 0.01673075301809917810\,\nu^4 +\\
&0.00025868056960961401938\,\nu^6 + 0.00074028981069076793270\,\nu^8 + O(\nu^{10})\\
\gamma_5(\nu) &= 1 - 0.05500285457469926324\,\nu^2 - 0.00223862067439588375 70\,\nu^4 +\\
&0.00061228129856346006635\,\nu^6 - 0.00024065831875901486620\,\nu^8 + O(\nu^{10})
\end{aligned}
\tag{19}
$$

$$
\begin{aligned}
c_1 &= 0\\
c_2 &= 0.26958221718999705470\\
c_3 &= 0.63161776092576151790\\
c_4 &= 0.56331453434548609609\\
c_5 &= 1
\end{aligned}
$$

References

[1] D.G. Bettis, Runge-Kutta algorithms for oscillatory problems, J. Appl. Math. Phys. 30 (1979) 699–704.
[2] M. Calvo, J.M. Franco, L. Rández, Minimum storage Runge-Kutta schemes for computational acoustics, Comput. Math. Appl. 45 (2003) 535–545.
[3] M. Calvo, J.M. Franco, L. Rández, A new minimum storage Runge-Kutta scheme for computational acoustics, J. Comput. Phys. 201 (2004) 1–12.
[4] W. Gautschi, Numerical integration of ordinary differential equations based on trigonometric polynomials, Numer. Math. 3 (1961) 381–397.
[5] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, Springer Series in Computational Mathematics, Springer, Berlin, Heidelberg, 1993.
[6] P.J. van der Houwen, Construction of Integration Formulas for Initial Value Problems, North-Holland, Amsterdam, 1977.
[7] L.Gr. Ixaru, G. Vanden Berghe, Exponential Fitting, Kluwer Academic Publishers, 2004.
[8] C.A. Kennedy, M.H. Carpenter, R.M. Lewis, Low storage, explicit Runge-Kutta schemes for computational acoustics, Appl. Numer. Math. 35 (2000) 177–219.
[9] D.I. Ketcheson, Runge-Kutta methods with minimum-storage implementations, J. Comput. Phys. 229 (2010) 1763–1773.
[10] J.D. Lambert, Numerical Methods for Ordinary Differential Systems: The Initial Value Problem, John Wiley & Sons, 1991.
[11] J. Oliver, A curiosity of low-order explicit Runge-Kutta methods, Math. Comput. 29 (1975) 1032–1036.
[12] A. Paris, L. Rández, New embedded explicit pairs of exponentially fitted Runge-Kutta methods, J. Comput. Appl. Math. 234 (2010) 767–776.
[13] B. Paternoster, Runge-Kutta(–Nyström) methods for ODEs with periodic solutions based on trigonometric polynomials, Appl. Numer. Math. 28 (1998) 401–412.
[14] T.E. Simos, An exponentially-fitted Runge-Kutta method for the numerical integration of initial-value problems with periodic or oscillating solutions, Comput. Phys. Commun. 115 (1998) 1–8.
[15] T.E. Simos, Ch. Tsitouras, Evolutionary derivation of Runge-Kutta pairs for addressing inhomogeneous linear problems, Numer. Algorithms 87 (2) (2020) 1–15.
[16] G. Vanden Berghe, H. De Meyer, M. Van Daele, T. Van Hecke, Exponentially-fitted explicit Runge-Kutta methods, Comput. Phys. Commun. 123 (1999) 7–15.
[17] J.H. Williamson, Low storage Runge-Kutta schemes, J. Comput. Phys. 35 (1980) 48–56.