Journal of Computational and Applied Mathematics 350 (2019) 130–138

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

# Novel symplectic integrators for the Klein–Gordon equation with space- and time-dependent mass

Philipp Bader [b,*], Sergio Blanes [a], Fernando Casas [b], Nikita Kopylov [a]

[a] Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Spain
[b] Universitat Jaume I, IMAC and Departament de Matemàtiques, 12071 Castellón, Spain

ARTICLE INFO

*Article history:*
Received 6 June 2018
Received in revised form 30 August 2018

*MSC:*
65L07
65L05
65Z05

*Keywords:*
Klein–Gordon equation
Time-dependent mass
Second-order linear differential equations
Symplectic integrators
Magnus expansion

ABSTRACT

We consider the numerical time-integration of the non-stationary Klein–Gordon equation with position- and time-dependent mass. A novel class of time-averaged symplectic splitting methods involving double commutators is analyzed and 4th- and 6th-order integrators are obtained. In contrast with standard splitting methods (that contain negative coefficients if the order is higher than two), additional commutators are incorporated into the schemes considered here. As a result, we can circumvent this order barrier and construct high order integrators with positive coefficients and a much reduced number of stages, thus improving considerably their efficiency. The performance of the new schemes is tested on several examples.

## 1. Introduction

Within the framework of relativistic quantum mechanics, the non-stationary Klein–Gordon (KG) equation in one of its simplest versions, written in units so that $\hbar = c = 1$, reads

$$\left(\frac{\partial^2}{\partial t^2} - \hat{N}(t)\right)\psi(x,t) = 0, \qquad \text{with} \qquad \hat{N}(t) = \Delta - m(x,t)^2, \tag{1}$$

and no external electromagnetic field is considered, so that the kinetic energy is represented by the Laplacian operator $\Delta$ [1–3]. Since the stationary position-dependence of the mass term $m(x)^2$ violates the Lorentz covariance of the interaction, it is highly desirable to introduce a time-dependence in this term [3]. In general, Eq. (1) has no closed form solutions and thus one has to rely on numerical algorithms for their computation. In this respect, its particular structure allows us to construct specially oriented numerical integrators which turn out to be more efficient than other general purpose integration schemes. In what follows we consider periodic boundary conditions. These can also be applied even when $\psi(x,t)$ and its derivatives attenuate far away from the region of interest as happens, for instance, in scattering [4,5] or when $\hat{N}(t)$ is a linear operator that traps the solution in a bounded region. In this way, accurate solutions can be obtained using high-order numerical methods as the schemes will not suffer from order reduction [6–9].

After spatial discretization, the operator $\hat{N}$ in (1) is approximated by a $n \times n$ matrix

$$N(t) = T + V(t), \tag{2}$$

[*] Corresponding author.
*E-mail addresses:* bader@uji.es (P. Bader), serblaza@imm.upv.es (S. Blanes), fernando.casas@uji.es (F. Casas), nikop1@upvnet.upv.es (N. Kopylov).

https://doi.org/10.1016/j.cam.2018.10.011
0377-0427/© 2018 Elsevier B.V. All rights reserved.

where $T$ corresponds to the (discretized) Laplacian $\Delta$ and $V(t)$ a diagonal matrix associated to $-m(x,t)^2$. Therefore, in discrete form, (1) is expressed either as the 2nd-order linear ODE

$$y''(t) = N(t)y(t), \qquad y(t_0) = y_0, \qquad y'(t_0) = y'_0, \quad y \in \mathbb{R}^n, \ t \geq 0 \tag{3}$$

or as the equivalent 1st-order system

$$z'(t) = A(t)z(t), \quad A(t) = \begin{pmatrix} 0 & I \\ N(t) & 0 \end{pmatrix}, \qquad z = \begin{pmatrix} y \\ y' \end{pmatrix}, \quad z \in \mathbb{R}^{2n}. \tag{4}$$

The matrix $N(t)$ has some remarkable properties which should be taken into consideration when designing numerical schemes for the integration of (4), namely:

1. Using a fine spatial grid results in a matrix of large dimension and thus numerical schemes involving only matrix–vector multiplications are preferred to methods that use matrix–matrix products due to the exceedingly large computational cost of the latter. In addition, as $V(t)$ is a diagonal matrix, the cost of the algorithms will be measured by the number of actions of $T$ on a vector.
2. In general, $N(t)$ is a symmetric negative definite matrix and the fundamental matrix solution of (4) is an oscillatory symplectic transformation. For this reason, symplectic integrators showing a good efficiency when dealing with oscillatory solutions over a wide range of frequencies are highly desirable.

A variety of methods can in principle be applied for the numerical integration of Eq. (4): classical Runge–Kutta (RK), Runge–Kutta–Nyström (RKN), multistep, extrapolation methods, etc., but they are not, in general, symplectic [10–13]. Moreover, they show a low performance when solving oscillatory problems. On the other hand, exponential integrators do show high accuracy when the solution is highly oscillatory [11,12,14] but their computational cost is sometimes exceedingly large.

The methods we propose here to solve (1) (or more specifically its discretized version (4)) are based on the key observation that the matrix $A(t)$ can be separated as

$$A(t) = B(t) + C \qquad \text{with} \qquad B(t) = \begin{pmatrix} 0 & 0 \\ N(t) & 0 \end{pmatrix}, \qquad C = \begin{pmatrix} 0 & I \\ 0 & 0 \end{pmatrix}, \tag{5}$$

where only the part $B(t)$ depends on time. They belong to the category of symplectic splitting schemes, but with the peculiarity that they include at no extra cost some time-derivatives of $V(t)$ in addition to the maps corresponding to each part $C$ and $B$ evaluated at certain nodes of a quadrature rule. In this way, it is possible to construct high order methods with a much reduced number of intermediate stages and therefore improve their overall efficiency. Furthermore, the inclusion of nested commutators into the algorithms allows one to use positive fractional time steps, something that cannot be done with classical splitting methods of higher order $p > 2$ [15]. This leads to an additional reduction of the error terms.

In Section 2, we will discuss the underlying framework on which this and previous analysis [16–18] are based, considering first, for clarity, the autonomous case. In Section 3, we will derive new optimized methods within this framework and conclude with numerical experiments in Section 4, illustrating the regimes where the new schemes outperform the existing algorithms.

## 2. Class of methods

### 2.1. Autonomous case

As mentioned before, decomposition (5) is particularly suitable for the numerical time integration of Eq. (4) with splitting methods. If $N$ (or equivalently, $B$) in (5) is independent of time, they have the form

$$\begin{aligned} \Phi(\tau) &= e^{\tau a_m C}\, e^{\tau b_m B} \dots e^{\tau a_1 C}\, e^{\tau b_1 B} \\ &= \begin{pmatrix} I & \tau a_m I \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ \tau b_m N & I \end{pmatrix} \cdots \begin{pmatrix} I & \tau a_1 I \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ \tau b_1 N & I \end{pmatrix}, \end{aligned} \tag{6}$$

for a time step size $\tau$.

Here, the coefficients $a_i, b_i$ are chosen such that the method is of order $p$, i.e., $\Phi(\tau) = e^{\tau(B+C)} + \mathcal{O}(\tau^{p+1})$ [19]. But for the problem at hand we can do still better. A simple calculation shows that

$$[C,[C,B]] = \begin{pmatrix} 0 & -2N \\ 0 & 0 \end{pmatrix}, \qquad [B,[C,B]] = \begin{pmatrix} 0 & 0 \\ 2N^2 & 0 \end{pmatrix},$$

where $[C,B] = CB - BC$ stands for the (matrix) commutator. Notice, then, that $[C,[C,B]]$ and $[B,[C,B]]$ share the structure of the building blocks $C$ and $B$, respectively, so that we can introduce new parameters in (6) without increasing the complexity of the composition with the replacements

$$\begin{aligned} \exp(\tau a_k C) &\longrightarrow \exp(\tau a_k C + \tau^3 c_k [C,[C,B]]), \\ \exp(\tau b_k B) &\longrightarrow \exp(\tau b_k B + \tau^3 d_k [B,[C,B]]). \end{aligned} \tag{7}$$

In this way (6) becomes

$$\Phi(\tau) = e^{\tau a_m C + \tau^3 c_m [C,[C,B]]}\, e^{\tau b_m B + \tau^3 d_m [B,[C,B]]} \cdots e^{\tau a_1 C + \tau^3 c_1 [C,[C,B]]}\, e^{\tau b_1 B + \tau^3 d_1 [B,[C,B]]}, \tag{8}$$

and a higher order can in principle be achieved without increasing the number of stages. This procedure can be considered a generalization of the so-called modified potentials that have been used previously in the literature [20,21] for Schrödinger equations. The first replacement requires an additional matrix–vector product since, for a vector $v$, $e^{\tau a C} v$ only involves sums of diagonal matrices, whereas $\exp(\tau a C + \tau^3 d[C,[C,B]])v$ requires the computation of $Nv$. Similarly, the number of matrix–vector products for $e^{\tau b B} v$ is increased by one after including the nested commutator $[B,[C,B]]$. Composition (8) has the additional advantage, in comparison with (6), that it leads to methods of order six with all coefficients $a_i$, $b_i$ being positive.

### *2.2. General, time-dependent case*

The splitting scheme (6) can be easily adapted to the non-autonomous case by considering time as a new coordinate [16,17]. Then, it becomes

$$\Phi(\tau) = \begin{pmatrix} I & \tau a_m I \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ \tau b_m N_m & I \end{pmatrix} \cdots \begin{pmatrix} I & \tau a_1 I \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ \tau b_1 N_1 & I \end{pmatrix} \tag{9}$$

where now $N_j = N(t_n + c_j\tau)$, $c_j = \sum_{k=0}^{j-1} a_k$ and $a_0 = 0$.

Generalizing the composition (8) is not as straightforward, however, and requires a more detailed analysis. Essentially, the idea has two parts, first, approximate the solution of (4) up to a given degree in $\tau$ by means of the Magnus expansion and then reproduce this expression by an appropriately modified version of (8). Whereas a detailed treatment can be found in [16,17], here we only collect the most relevant steps in this approach.

For the sake of brevity, we fix the quadrature rule which is used to approximate the Magnus expansion. Since we consider only methods up to order six, a good choice is the 6th-order Gauss–Legendre (GL) quadrature rule (although other quadrature rules can also be used, see [22] and references therein),

$$c_1 = \frac{5-\sqrt{15}}{10}, \quad c_2 = \frac{1}{2}, \quad c_3 = \frac{5+\sqrt{15}}{10}. \tag{10}$$

Next, we form the linear combinations[1]

$$\alpha_1 = \tau A_2, \quad \alpha_2 = \frac{\sqrt{15}\tau}{3}(A_3 - A_1), \quad \alpha_3 = \frac{10\tau}{3}(A_3 - 2A_2 + A_1), \tag{11}$$

with $A_j = A(t_n + c_j\tau)$. It can readily be shown that $\alpha_1 = \mathcal{O}(\tau)$, $\alpha_2 = \mathcal{O}(\tau^2)$, $\alpha_3 = \mathcal{O}(\tau^3)$ and (see [23])

$$\Omega^{[6]} = \alpha_1 + \frac{1}{12}\alpha_3 - \frac{1}{12}[\alpha_1,\alpha_2] + \frac{1}{360}[\alpha_1,[\alpha_1,\alpha_3]] - \frac{1}{240}[\alpha_2,[\alpha_1,\alpha_2]] + \frac{1}{720}[\alpha_1,[\alpha_1,[\alpha_1,\alpha_2]]], \tag{12}$$

leads to an approximation of order six to the exact solution of (4), i.e.,

$$z(t_{n+1}) = \exp(\Omega^{[6]})z(t_n) + \mathcal{O}(\tau^7).$$

Notice that for this problem one has $[\alpha_2,\alpha_3] = 0$, and moreover

$$\alpha_1 = \gamma_1 + \beta_1, \qquad \alpha_2 = \beta_2, \qquad \alpha_3 = \beta_3,$$

with

$$\gamma_1 = \tau \begin{pmatrix} 0 & I \\ 0 & 0 \end{pmatrix}, \qquad \beta_1 = \tau \begin{pmatrix} 0 & 0 \\ T + W_1 & 0 \end{pmatrix}, \qquad \beta_i = \tau \begin{pmatrix} 0 & 0 \\ W_i & 0 \end{pmatrix}, \; i > 1$$

and

$$W_1 = V_2, \qquad W_2 = \frac{\sqrt{15}}{3}(V_3 - V_1), \qquad W_3 = \frac{10}{3}(V_3 - 2V_2 + V_1). \tag{13}$$

As in (11), $\beta_i = \mathcal{O}(\tau^i)$ and $V_j = V(t_n + c_j\tau)$, so that $W_j$ are diagonal matrices. With this notation, the following algebraic relations hold:

$$[\beta_i,\beta_j] = 0, \qquad [\gamma_1,[\gamma_1,[\gamma_1,\beta_i]]] = [\beta_i,[\beta_j,[\beta_k,\gamma_1]]] = 0 \tag{14}$$

[1] Let $\mathcal{L}_k(t)$ be the Lagrange interpolation polynomial of $A(t)$ at the nodes $A_i = A(t_n + c_i\tau)$, $i = 1,\ldots,k$, then $\alpha_{i+1} = \tau^{i+1}\frac{1}{i!}\frac{d^i}{dt^i}\mathcal{L}_k(t)\big|_{t=t_n+\tau/2}$, i.e., the $i$th time derivative of the interpolation polynomial at the midpoint of the current interval.

for any value of $i, j, k$, so that the Magnus exponent (12) reads

$$\begin{aligned}\Omega^{[6]} = {}& \gamma_1 + \beta_1 + \frac{1}{12}\beta_3 + \frac{1}{12}[\beta_2, \gamma_1] + \frac{1}{360}\Big(-[\gamma_1, [\beta_3, \gamma_1]] + [\beta_1, [\gamma_1, \beta_3]]\Big) \\ & - \frac{1}{240}[\beta_2, [\gamma_1, \beta_2]] + \frac{1}{720}\Big([\gamma_1, [\beta_1, [\gamma_1, \beta_2]]] - [\beta_1, [\gamma_1, [\beta_2, \gamma_1]]]\Big).\end{aligned} \tag{15}$$

Moreover, it is easy to check that

$$\begin{aligned} \beta_j = \begin{pmatrix} 0 & 0 \\ D_1 & 0 \end{pmatrix}, \qquad & [\gamma_1, \beta_j] = \begin{pmatrix} D_1 & 0 \\ 0 & -D_1 \end{pmatrix}, \qquad j \geq 1, \\ [\gamma_1, [\beta_i, \gamma_1]] = \begin{pmatrix} 0 & D_2 \\ 0 & 0 \end{pmatrix}, \qquad & [\beta_i, [\gamma_1, \beta_j]] = \begin{pmatrix} 0 & 0 \\ D_3 & 0 \end{pmatrix}, \qquad i, j \geq 1, \end{aligned} \tag{16}$$

where the $D_j$ are placeholders for diagonal matrices which are functions of the potential $V$. Therefore any linear combination of all these terms not containing $T$ results in a $2 \times 2$ block matrix where the blocks are again diagonal matrices, i.e.,

$$\begin{aligned} & \exp\Big(x_1\gamma_1 + x_2\beta_2 + x_3\beta_3 + x_4[\gamma_1, \beta_2] + x_5[\gamma_1, \beta_3] + x_6[\beta_2, [\gamma_1, \beta_2]] + x_7[\gamma_1, [\beta_2, \gamma_1]] + x_8[\gamma_1, [\beta_3, \gamma_1]]\Big) \\ & = \exp\begin{pmatrix} D_4 & D_5 \\ D_6 & -D_4 \end{pmatrix}. \end{aligned} \tag{17}$$

This exponential contributes with eight independent parameters to the scheme, and this helps a good deal to solve the order conditions. In addition, its computational cost can be neglected[2] since the blocks $D_j$ are diagonal matrices.

With respect to the autonomous part, when a composition of type (8) is used, we are dealing with the following exponents:

$$x_1\gamma_1 + x_2[\gamma_1, [\beta_1, \gamma_1]] = \begin{pmatrix} 0 & 2x_2T + D_7 \\ 0 & 0 \end{pmatrix}, \qquad x_3\beta_1 + x_4[\beta_1, [\gamma_1, \beta_1]] = \begin{pmatrix} 0 & 0 \\ (y_1T + D_8)(y_2T + D_9) + D_{10} & 0 \end{pmatrix}.$$

These nilpotent matrices (with index 2) are trivially exponentiated and their actions on a vector require one and two products, respectively. With the same reasoning, additional commutators can be added for the non-autonomous part at no extra cost:

$$\exp\Big(x_1\gamma_1 + x_2[\gamma_1, [\beta_1, \gamma_1]] + x_3[\gamma_1, [\beta_2, \gamma_1]] + x_4[\gamma_1, [\beta_3, \gamma_1]]\Big) = \exp\begin{pmatrix} 0 & 2x_2T + D_{11} \\ 0 & 0 \end{pmatrix}, \tag{18}$$

and

$$\exp\Big(x_1\beta_1 + x_2\beta_2 + x_3\beta_3 + x_4[\beta_1, [\gamma_1, \beta_1]] + x_5[\beta_2, [\gamma_1, \beta_2]]\Big) = \exp\begin{pmatrix} 0 & 0 \\ (y_1T + D_{12})(y_2T + D_{13}) + D_{14} & 0 \end{pmatrix}, \tag{19}$$

respectively.

## 3. Derivation of new schemes

The methods we construct in this section for the numerical integration of Eq. (4) are compositions (8), where in some (or all) the exponents we include additional terms of the form (17)–(19), which allows one to introduce additional parameters without increasing significantly the computational cost. In this way, our ansatz is

$$\begin{aligned} \Phi_m^{[p]}(\tau) = {}& \prod_{i=1}^{m+1} \exp\left(b_i\beta_1 + d_i[\beta_1, [\gamma_1, \beta_1]] + x_i\beta_2 + y_i\beta_3\right) \\ & \times \exp\left(a_i\gamma_1 + c_i[\gamma_1, [\beta_1, \gamma_1]] + z_i[\gamma_1, [\beta_3, \gamma_1]] + u_i\beta_2 + v_i\beta_3 + w_i[\gamma_1, \beta_2] + r_i[\gamma_1, \beta_3]\right) \\ = {}& e^{\Omega^{[p]}(\tau)} + \mathcal{O}(\tau^{p+1}). \end{aligned} \tag{20}$$

A common procedure to obtain these parameters is to apply sequentially the Baker–Campbell–Hausdorff formula to express the whole composition as the exponential of a power series in $\tau$, $P_m(\tau)$, whose terms are linear combinations of $\gamma_1$, $\beta_j$ and their nested commutators, and then impose that $P_m(\tau) = \Omega^{[p]}$ up to the order $p$ considered (in practice, $p = 4, 6$). This requirement imposes a set of polynomial equations in the coefficients, the so-called order conditions.

Since we are considering a Gauss–Legendre quadrature rule, and this is symmetric with respect to the midpoint, it is clear that $\tau W_i$, $i = 1, 2, 3$ only contain odd powers of $\tau$ in its Taylor expansion around the midpoint $t_n + \tau/2$. In consequence, if we write explicitly the dependence on $\tau$, one has

$$\gamma_1(-\tau) = -\gamma_1(\tau), \qquad \beta_i(-\tau) = (-1)^i\beta_i(\tau)$$

[2] The exponential formula for $2 \times 2$ matrices can be used here.

**Table 1**
$M$ steps of algorithm $\Sigma_{3c}^{[4]}(\tau)$. Each step requires three products ($Tq_1$, $Tp_1$ and $Tq_2$).

```
do $k = 0, M-1$
  $V_i = V(t_k + c_i\tau),\ i = 1, 2, 3$
  $W_1 = V_2, \quad W_2 = \frac{\sqrt{15}}{3}(V_3 - V_1), \quad W_3 = \frac{10}{3}(V_1 - 2V_2 + V_3)$
  $q_0 = z_1, \quad p_0 = z_2$
      $q_1 = q_0 + \tau\,\frac{1}{6}p_0$
      $p_1 = p_0 + \tau(\frac{1}{2}\mathbf{T} + (\frac{1}{2}W_1 - \frac{1}{8}W_2 + \frac{1}{24}W_3))q_1$
      $q_2 = q_1 + \tau\,\frac{2}{3}p_1 + \tau^3(\frac{1}{36}\mathbf{T} - \frac{7}{2160}W_3)p_1$
      $p_2 = p_1 + \tau(\frac{1}{2}\mathbf{T} + (\frac{1}{2}W_1 + \frac{1}{8}W_2 + \frac{1}{24}W_3))q_2$
      $q_3 = q_2 + \tau\,\frac{1}{6}p_2$
  $z_1 = q_3, \quad z_2 = p_2$
enddo
```

The matrix $T$ that carries the cost of the algorithm is highlighted in bold.

and therefore $\sigma(-\tau) = (-1)^{i+1}\sigma(\tau)$, where

$$\sigma(\tau) \equiv [\gamma_1(\tau), \beta_i(\tau)] = \mathcal{O}(\tau^{i+1}).$$

In practice, we only consider symmetric compositions and therefore $P_m(\tau)$ only contains odd powers of $\tau$,

$$\begin{aligned} P_m(\tau) =& f_{1,1}\gamma_1 + f_{1,2}\beta_1 + f_{3,1}[\gamma_1, [\gamma_1, \beta_1]] + f_{3,2}[\beta_1, [\gamma_1, \beta_1]] + f_{3,3}\beta_3 \\ & + f_{3,4}[\beta_2, \gamma_1] + f_{5,1}[\gamma_1, [\gamma_1, [\beta_1, [\gamma_1, \beta_1]]]] + f_{5,2}[\beta_1, [\gamma_1, [\beta_1, [\gamma_1, \beta_1]]]] \\ & + f_{5,3}[\gamma_1, [\beta_3, \gamma_1]] + f_{5,4}[\beta_1, [\gamma_1, \beta_3]] + f_{5,5}[\beta_2, [\gamma_1, \beta_2]] + f_{5,6}[\gamma_1, [\beta_1, [\gamma_1, \beta_2]]] \\ & + f_{5,7}[\beta_1, [\gamma_1, [\beta_2, \gamma_1]]] + \mathcal{O}(\tau^7), \end{aligned} \tag{21}$$

where $f_{ij}$ are polynomials in the coefficients of the method. Notice that the elements in $P_m(\tau)$ that involve commutators with $\gamma_1, \beta_1$ like $[\beta_1, [\gamma_1, \beta_1]]$, do not appear in the Magnus expansion and so, their coefficients up to the desired order must be canceled to reach the proper order. Furthermore, an efficient scheme has to minimize the coefficients $f_{j,k}$ multiplying commutators (and hence errors) at higher orders.

The construction procedure has indeed two parts. First, an efficient propagator is built for the autonomous problem. In other words, we first approximate $e^{\gamma_1+\beta_1}$. Second, new terms involving $\beta_2$, $\beta_3$ and different commutators are incorporated into the composition to solve the remaining order conditions for the non-autonomous problem. In doing so, it might be the case that some extra exponentials have to be included to solve all the order conditions arising from the non autonomous part.

*Order 4.* In particular, although the composition

$$\Phi_{3,1}^{[4]}(\tau) = \exp\left(\frac{1}{2}\beta_1 + \frac{1}{48}[\beta_1, [\gamma_1, \beta_1]]\right) \exp\left(\gamma_1 + \frac{1}{12}[\gamma_1, [\beta_1, \gamma_1]]\right) \exp\left(\frac{1}{2}\beta_1 + \frac{1}{48}[\beta_1, [\gamma_1, \beta_1]]\right)$$

satisfy all the order conditions up to order 4 for the autonomous case, there are so few terms here that the inclusion of additional commutators of the form (17)–(19) is just enough to solve the remaining order conditions for the non autonomous problem, but in order to have free parameters to minimize the error coefficients, more exponentials have to be included. At the same computational cost, a more efficient method for this problem with 5 exponentials is

$$\begin{aligned} \Sigma_{3c}^{[4]}(\tau) \equiv & \exp\left(\frac{1}{6}\gamma_1\right) \exp\left(\frac{1}{2}\beta_1 + \frac{1}{8}\beta_2 + \frac{1}{24}\beta_3\right) \exp\left(\frac{2}{3}\gamma_1 + \left[\gamma_1, \left[\gamma_1, -\frac{1}{72}\beta_1 + \frac{7}{4320}\beta_3\right]\right]\right) \\ & \times \exp\left(\frac{1}{2}\beta_1 - \frac{1}{8}\beta_2 + \frac{1}{24}\beta_3\right) \exp\left(\frac{1}{6}\gamma_1\right), \end{aligned} \tag{22}$$

and it satisfies, in addition to the 4th-order conditions $f_{1,1} = f_{1,2} = 1, f_{3,1} = f_{3,2} = 0, f_{3,3} = f_{3,4} = \frac{1}{12}$, also the following at order 5:

$$f_{5,3} = -\frac{1}{360}. \tag{23}$$

If we denote $z = (z_1, z_2)$, the algorithm (22) to integrate for $t \in [t_0, t_f]$ with in $M$ steps of length $\tau = (t_f - t_0)/M$ is stated in Table 1.

*Order 6.* Now the full set of 13 order conditions arising from $\Phi_m^{[6]}(\tau) = \exp(P_m(\tau)) = \exp(\Omega^{[6]}) + \mathcal{O}(\tau^7)$ are required to achieve order six for a symmetric composition. We have carried out an exhaustive analysis of the composition (20) for several

**Table 2**
$M$ steps of algorithm $\Sigma_{5c}^{[6]}$ with $W_i$ defined as in Table 1. Each step requires 5 products $Tq_1, Tq_2, Tp_4, Tq_3, Tq_3$.

```
do $k = 0, M-1$
    //Calculate exponentials:
    $D_1 = \tau^2 y_1 W_2, D_2 = \tau^2 y_4 W_2,$
    $E_i(u) = \exp\begin{pmatrix} D_i & uI \\ 0 & -D_i \end{pmatrix} = \begin{pmatrix} \exp(D_i) & uD_i^{-1}\sinh D_i \\ 0 & \exp(D_i)^{-1} \end{pmatrix}$, //where $D_i = \mathcal{O}(\tau^3)$ are diagonal matrices.
    $q_0 = z_1, \quad p_0 = z_2$
        $\begin{pmatrix} q_1 \\ p_1 \end{pmatrix} = E_1(\tau x_1) \begin{pmatrix} q_0 \\ p_0 \end{pmatrix}$
        $p_2 = p_1 + \tau(x_2 \mathbf{T} + (x_2 W_1 - y_2 W_2 + y_3 W_3)) q_1$
        $\begin{pmatrix} q_2 \\ p_3 \end{pmatrix} = E_2(\tau x_3) \begin{pmatrix} q_1 \\ p_2 \end{pmatrix}$
        $p_4 = p_3 + \tau(x_4 \mathbf{T} + (x_4 W_1 - y_5 W_2 + y_6 W_3)) q_2$
        $q_3 = q_2 + (\tau x_5 I + \tau^3 (2x_6 \mathbf{T} + 2x_6 W_1 + 2y_7 W_3)) p_4$
        $p_5 = p_4 + \tau(x_4 \mathbf{T} + (x_4 W_1 + y_5 W_2 + y_6 W_3)) q_3$
        $\begin{pmatrix} q_4 \\ p_6 \end{pmatrix} = E_2(\tau x_3) \begin{pmatrix} q_3 \\ p_5 \end{pmatrix}$
        $p_7 = p_6 + \tau(x_2 \mathbf{T} + (x_2 W_1 + y_2 W_2 + y_3 W_3)) q_3$
        $\begin{pmatrix} q_5 \\ p_8 \end{pmatrix} = E_1(\tau x_1) \begin{pmatrix} q_4 \\ p_7 \end{pmatrix}$
    $z_1 = q_5, \quad z_2 = p_8.$
enddo
```

The matrix $T$ that carries the cost of the algorithm is highlighted in bold.

values of $m$ and found that the specific choice

$$\begin{aligned} \Sigma_{5c}^{[6]}(\tau) = & \exp\left(x_1\gamma_1 + y_1[\gamma_1, \beta_2]\right) \exp\left(x_2\beta_1 + y_2\beta_2 + y_3\beta_3\right) \\ & \times \ \exp\left(x_3\gamma_1 + y_4[\gamma_1, \beta_2]\right) \exp\left(x_4\beta_1 + y_5\beta_2 + y_6\beta_3\right) \\ & \times \ \exp\left(x_5\gamma_1 + x_6\,[\gamma_1, [\beta_1, \gamma_1]] + y_7\,[\gamma_1, [\beta_3, \gamma_1]]\right) \\ & \times \ \exp\left(x_4\beta_1 - y_5\beta_2 + y_6\beta_3\right) \exp\left(x_3\gamma_1 + y_4[\gamma_1, \beta_2]\right) \\ & \times \ \exp\left(x_2\beta_1 - y_2\beta_2 + y_3\beta_3\right) \exp\left(x_1\gamma_1 + y_1[\gamma_1, \beta_2]\right), \end{aligned} \tag{24}$$

where

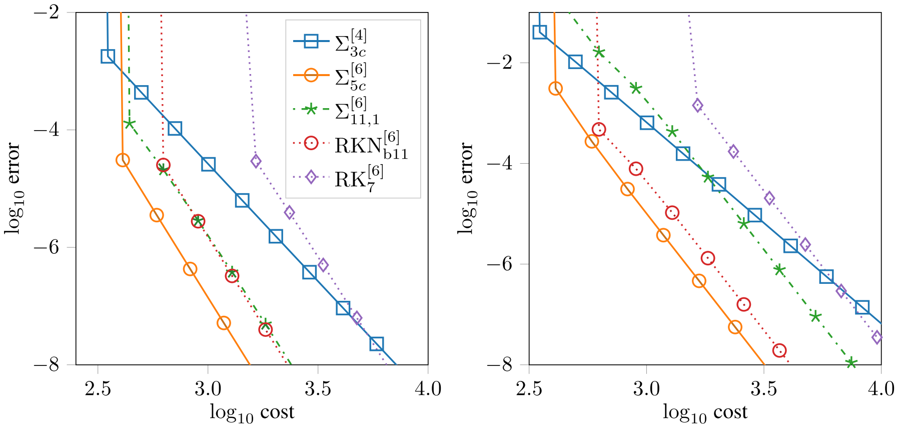
$$\begin{array}{lll} x_1 = 0.08910076599011520575 & x_2 = 0.24004250742649120555 & x_3 = 0.28694996084207488677 \\ x_4 = 0.25995749257350879444 & x_5 = 0.24789854633561981494 & x_6 = 0.00285551027560918571 \\ y_1 = -0.00097618964290807330 & y_2 = 0.06618969871667327349 & y_3 = 0.03862265557473451707 \\ y_4 = -0.00501240016226056089 & y_5 = 0.06842138031733469147 & y_6 = 0.00304401109193214959 \\ y_7 = 0.00031774532164766212, & & \end{array}$$

leads to particularly small error terms at higher orders. This method, together with $\Sigma_{3c}^{[4]}$ given in (22), will be used in the numerical experiments reported in the next section. For the convenience of the reader, we include the implementation of the corresponding algorithm in Table 2, with $W_i$ defined as before in Table 1. Notice that since the appearing diagonal matrices $D_i$ are of size $\mathcal{O}(\tau^3)$, the computations of $\exp(D_i)$ and $\sinh(D_i)$ in $E_i$, cf. Table 2, are relatively cheap to perform.

## 4. Numerical examples

In this section, we compare the methods $\Sigma_{3c}^{[4]}$ and $\Sigma_{5c}^{[6]}$ constructed specifically for the numerical integration of the Klein–Gordon equation in the previous section with several well known reference schemes from the literature. In this respect, the following 6th-order methods are considered:

1. $\mathrm{RK}_7^{[6]}$: The 7-stage 6th-order RK method with coefficients originally obtained in [24, p. 203–205], and reproduced in [22, p. 201], etc.
2. $\mathrm{RKN}_{b11}^{[6]}$: The 11-stage 6th-order symplectic splitting RKN method of type (9) from [25]. The scheme uses four additional stages to add four free parameters for optimization.



(a) $\mu = 1/5$

(b) $\mu = 5$

**Fig. 1.** Efficiency plots for Klein–Gordon Eq. (27).

3. $\Sigma_{11,1}^{[6]}$: This method belongs to the family of schemes (20) but contains no commutators. It is the 11-stage 6th-order symplectic splitting method for a general linear wave equation from [18] that corresponds to an improvement on the more general scheme given in [17] and referred as $\mathrm{SM}_{11}^{[6]}$. This scheme has six additional stages for optimization in the autonomous problem and is given by the composition

$$\Psi_{11}^{[6]} = \begin{pmatrix} I & \tau a_{12} I \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ \tau C_{11} & I \end{pmatrix} \begin{pmatrix} I & \tau a_{11} I \\ 0 & I \end{pmatrix} \cdots \begin{pmatrix} I & 0 \\ \tau C_{1} & I \end{pmatrix} \begin{pmatrix} I & \tau a_{1} I \\ 0 & I \end{pmatrix}, \tag{25}$$

where $C_i = (b_{i,1}N_1 + b_{i,2}N_2 + b_{i,3}N_3)$ and $N_i = N(t_n + c_i\tau)$ with $c_i$ being the nodes of the 6th-order GL quadrature rule (10). The coefficients $a_i$, $b_{i,j}$ are given by (more significant digits can found in [18]):

$$a_1 = 0.0464874547908631 \quad a_2 = -0.0606916711656429 \quad a_3 = 0.2184665264634068$$
$$a_4 = 0.1680535794830927 \quad a_5 = 0.3143923641703534 \quad a_6 = -0.1867082537420731$$

$$(b_{i,j}) = \begin{pmatrix} 0.152309756970167 & 0.078927889445323 & -0.046907162912825 \\ 0.006406269275594 & -0.091413523927685 & 0.043950351354379 \\ 0.086778862327312 & 0.051027214890409 & -0.004050397550970 \\ 0.066634120201024 & 0.148499347182669 & -0.011368920251338 \\ -0.020231991304321 & 0.030206484536889 & -0.021734660147529 \\ 0.025991549816284 & 0.009949620189233 & 0.025991549816284 \end{pmatrix} \tag{26}$$

and $a_{13-i} = a_i, \quad i = 1, \ldots, 6, \quad b_{6+i,j} = b_{6-i,4-j}, \quad i = 1, \ldots, 5, \;\; j = 1, 2, 3.$

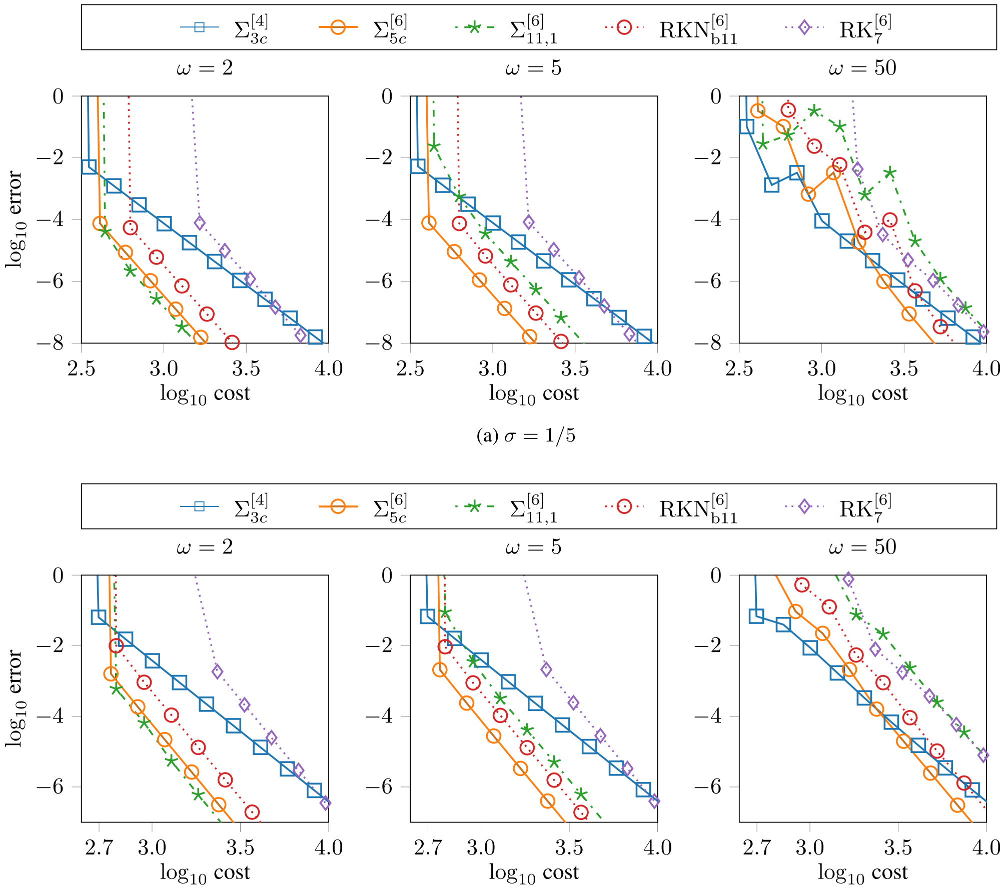
For our experiments, we assume that the solution is confined in a region $[x_0, x_N]$, and periodic boundary conditions are imposed. We divide the spatial region into $N = 128$ intervals of length $\Delta x = (x_N - x_0)/N$ and, after spatial discretization, we obtain an equation similar to (3) that we write as the first order system (4) where $z = (z_1, z_2)^T$ and $z_{1,i}(t) \approx u(x_i, t)$, $z_{2,i}(t) \approx u_t(x_i, t)$. We take as reference solution one calculated with a sufficiently small stepsize, and then we carry out the numerical integration with each method using different time steps and measure the $\ell_2$ error at the final time. This error is plotted in double-logarithmic scale versus the computational cost, estimated as the number of matrix–vector products.

**Example 1.** We first consider the Klein–Gordon equation with time-dependent mass [1]:

$$\begin{aligned} &\partial_t^2 u = \partial_x^2 u - \frac{\mu^2}{(1+t)^2} u, && x \in [-10,\ 10]; \\ &u(x,0) = e^{-\frac{1}{2}(x-3)^2} + e^{-\frac{1}{2}(x+2)^2}, && u'(x,0) = 0; \\ &u(-10,t) = u(10,t), \end{aligned} \tag{27}$$

which we integrate over the time interval $t \in [0, 10\pi]$.

As it can be seen from Fig. 1, $\Sigma_{5c}^{[6]}$ method shows notable improvement in efficiency, both when the initial mass $\mu$ is small, i.e., when the time dependence of the problem contributes less, and when $\mu$ is relatively large. This also applies to



(a) $\sigma = 1/5$

(b) $\sigma = 1$

**Fig. 2.** Efficiency plots for the wave Eq. (28).

the 4th-order $\Sigma_{3c}^{[4]}$ method: it compares favorably with the 6th-order Runge–Kutta method $\mathrm{RK}_7^{[6]}$, both in efficiency and especially in stability.

**Example 2.** As a second illustration we take a wave equation with time-dependent potential of amplitude $\sigma$ and frequency $\omega$, namely

$$\begin{aligned} &\partial_t^2 u = \partial_x^2 u - \sigma\left(1 + \frac{1}{5}\cos\omega t\right) x^2 u, && x \in [-10,\ 10]; \\ &u(x,0) = e^{-\frac{1}{2}(x-3)^2} + e^{-\frac{1}{2}(x+2)^2}, && u'(x,0) = 0; \\ &u(-10,t) = u(10,t), && \end{aligned} \tag{28}$$

and time interval used is again $[0,\ 10\pi]$. The results obtained by the different methods are summarized in Fig. 2. Notice that the results do not change qualitatively if a single Gaussian is used as initial condition.

This example differs from the previous one. When $\sigma$ and $\omega$ are small, the oscillations are low, and the equation is very close to the harmonic oscillator. This type of problems is best solved by $\Sigma_{11,1}^{[6]}$ which is designed to solve the autonomous harmonic oscillator with high precision and uses 6 free parameters for optimization, resulting in a cost of 11 multiplications per step. On the other hand, the new $\Sigma_{5c}^{[6]}$ costs only 5 such products and uses the minimal number of parameters to reach order 6, which leaves room for further development of this family of schemes. Due to included commutators, this method addresses better the case with higher oscillations and, hence, more significant contributions of the time derivatives of the potential. Similarly to the Klein–Gordon example above, $\Sigma_{3c}^{[4]}$ shows marginally improved performance and stability in some examples, when fewer time steps are used.

## 5. Conclusions

We have considered numerical integrators for the non-stationary Klein–Gordon equation with position- and time-dependent mass. The algebraic structure of this particular problem allowed us to build new families of symplectic schemes that only involve few matrix–vector products per step compared with standard methods. The improved performance of the new methods compared to symplectic integrators from the literature comes from the fact that a reduced number of order conditions has to be satisfied by the coefficients of the schemes and some elements of the algebra can be incorporated into the scheme without increasing the computational cost in terms of matrix–vector multiplications. Furthermore, these new terms allow to solve the order conditions while overcoming the order barrier on the negative coefficients. Consequently, schemes have a reduced number of stages but allow to solve the order conditions with small positive coefficients that provide highly accurate results at a low computational cost. These 4th- and 6th-order schemes are built without using additional stages to improve their performance on the autonomous case and already have shown their superiority versus other highly optimized schemes from the literature for moderately oscillatory problems. It is worth noting that such a setup opens up the possibility to build methods with extra coefficients for optimization to further increase their efficiency.

For the numerical integrations of problems with highly oscillatory solutions, exponential integrators typically exhibit a good performance. For such problems, it remains to analyze when one should turn to exponential integrators and this could depend either on the initial conditions as well as on the size of the spatial mesh or the external interactions. Since this analysis becomes very problem dependent and specific, it has not been considered in this work.

## Acknowledgments

This work has been funded by Ministerio de Economía, Industria y Competitividad (Spain) through project MTM2016-77660-P (AEI/FEDER, UE). Kopylov has also been partly supported by grant GRISOLIA/2015/A/137 from the Generalitat Valenciana.

## References

[1] C. Böhme, M. Reissig, A scale-invariant klein-gordon model with time-dependent potential, Ann. Univ. Ferrara Sez. VII Sci. Mat. 58 (2012) 229–250.
[2] W.N. Nascimento, Klein-Gordon Models with Non-Effective Time-Dependent Potential (Ph.D. thesis), Univ. Federal de São Carlos, 2016.
[3] M. Znojil, Non-Hermitian interaction representation and its use in relativistic quantum mechanics, Ann. Physics 385 (2017) 162–179.
[4] V. Petkov, Scattering Theory for Hyperbolic Operators, Elsevier Science Publishers, North Holland, 1989.
[5] G.F. Roach, Wave scattering by time-dependent potentials, Chaos Solitons Fractals 12 (2001) 2833–2847.
[6] I. Alonso-Mallo, B. Cano, N. Reguera, Avoiding order reduction when integrating linear initial boundary value problems with Lawson methods, IMA J. Num. Anal. 37 (4) (2017) 2091–2119.
[7] L. Einkemmer, A. Ostermann, Overcoming order reduction in diffusion-reaction splitting. Part 1: Dirichlet boundary conditions, SIAM J. Sci. Comput. 37 (2015) A1577–A1592.
[8] W. Hundsdorfer, J. Verwer, Numerical solution of time-dependent advection-diffusion-reaction equations, in: Springer Series in Computational Mathematics, vol. 33, Springer-Verlag, Berlin, 2003.
[9] J.M. Sanz-Serna, J.G. Verwer, W.H. Hundsdorfer, Convergence and order reduction of Runge–Kutta schemes applied to evolutionary problems in partial differential equations, Numer. Math. 50 (1986) 405–418.
[10] S. Blanes, F. Casas, A Concise Introduction to Geometric Numerical Integration, CRC Press, Boca Raton, 2016.
[11] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration, Structure-Preserving Algorithms for Ordinary Differential Equations, second ed., Springer Series in Computational Mathematics, vol. 31, 2006, pp. Springer–Verlag.
[12] B. Leimkuhler, S. Reich, Simulation Hamiltonian Dynamics, Cambridge University Press, Cambridge, 2005.
[13] J.M. Sanz-Serna, M.P. Calvo, Numerical Hamiltonian Problems, Chapman and Hall, London, 1994.
[14] M. Hochbruck, A. Ostermann, Exponential integrators, Acta Numer. 19 (2010) 209–286.
[15] Q. Shen, Solving linear partial differential equations by exponential splitting, IMA J. Numer. Anal. 9 (1989) 199–212.
[16] S. Blanes, F. Casas, A. Murua, Splitting methods for non-autonomous linear systems, Int. J. Comput. Math. 84 (2007) 713–727.
[17] S. Blanes, F. Casas, A. Murua, Symplectic time-average propagators for the Schrödinger equation with a time-dependent Hamiltonian, J. Chem. Phys. 146 (2017) 114109.
[18] P. Bader, S. Blanes, F. Casas, N. Kopylov, E. Ponsoda, Symplectic integrators for second-order linear non-autonomous equations, J. Comput. Appl. Math. 330 (2018) 909–919.
[19] R.I. McLachlan, G.R.W. Quispel, Splitting methods, Acta Numer. 11 (2002) 341–434.
[20] S.A. Chin, Symplectic integrators from composite operator factorizations, Phys. Lett. A 226 (1997) 344–348.
[21] I.P. Omelyan, I.M. Mryglod, R. Folk, On the construction of high-order force gradient algorithms for integration of motion in classical and quantum systems, Phys. Rev. E 66 (2002) 026701.
[22] S. Blanes, F. Casas, J.A. Oteo, J. Ros, The magnus expansion and some of its applications, Phys. Rep. 470 (2009) 151–238.
[23] H. Munthe-Kaas, B. Owren, Computations in a free lie algebra, Philos. Trans. R. Soc. Lond. Ser. A 357 (1999) 957–981.
[24] J.C. Butcher, The Numerical Analysis of Ordinary Differential Equations, John Wiley, 1987.
[25] S. Blanes, P.C. Moan, Practical symplectic partitioned Runge–Kutta and Runge–Kutta-Nyström methods, J. Comput. Appl. Math. 142 (2002) 313–330.