Exponential integrators for coupled self-adjoint non-autonomous partial differential systems

Enrique Ponsoda *, Sergio Blanes

Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, Building 8G, 46022 Valencia, Spain

1. Introduction

Let us consider the numerical integration of self-adjoint partial differential equations of the type

\[ (P(t)u_t(x,t))_t = Q(t)u_{xx}(x,t); \quad 0 \leq x \leq d, \quad t \geq 0, \]  

with initial and boundary conditions given by

\[
\begin{aligned}
u(0,t) &= u(d,t) = 0, \quad t \geq 0, \\
u(x,0) &= f(x), \quad 0 \leq x \leq d, \\
u_t(x,0) &= g(x), \quad 0 \leq x \leq d,
\end{aligned}
\]  

where \( u(x,t), f(x), g(x) \in \mathbb{R}^r \). We consider the case where:

(1) \( P(t), Q(t) \in \mathbb{R}^{r \times r} \) are symmetric positive definite matrices.
(II) \( -P(t) \) and \( Q(t) \) are both symmetric positive (or negative) semidefinite matrices.
(III) \( f(x) \) is three times differentiable and \( f^{(3)}(x) \) is piecewise continuous in \([0,d]\) with \( f(0) = f(d) = f^{(2)}(0) = f^{(2)}(d) = 0 \).
(IV) \( g(x) \) is twice differentiable with \( g^{(2)}(x) \) piecewise continuous in \([0,d]\) and \( g(0) = g(d) = 0 \).

* Corresponding author.
E-mail addresses: eponsoda@imm.upv.es (E. Ponsoda), serblaza@imm.upv.es (S. Blanes).

http://dx.doi.org/10.1016/j.amc.2014.05.050
0096-3003/© 2014 Elsevier Inc. All rights reserved.
The system (1) appears frequently in the study of microwave heating processes, where the variations of the dielectric properties of the material with temperature, density, moisture content and other parameters make the system non-autonomous, see [10,13] for more details. Systems of type (1) can also be found on models for the study of electromagnetic processing of homogeneous materials at high power densities or in the analysis of multi mode microwave applicators, see [8,17].

Under conditions (I)–(IV), the problem (1)(2) has, at most, a twice continuously differentiable solution [16, Section 2]. We look for a numerical solution, and to this purpose we first consider separation of variables. In a bounded domain \(D(d, T) = \{(x, t), 0 \leq x \leq d, 0 \leq t \leq T\}\), for a given \(T > 0\), the solution can be formally written as a convergent series

\[
u(x, t) = \sum_{n \geq 0} \left\{ Y_n(t) a_n + \tilde{Y}_n(t) b_n \right\} \sin \left(\frac{n \pi x}{d}\right),
\]

where \(a_n, b_n \in \mathbb{R}^r\) are given by

\[
a_n = \frac{2}{d} \int_0^d f(x) \sin \left(\frac{n \pi x}{d}\right) dx, \quad b_n = \frac{2}{d} \int_0^d g(x) \sin \left(\frac{n \pi x}{d}\right) dx.
\]

The matrices \(Y_n(t), \tilde{Y}_n(t) \in \mathbb{R}^{2 \times r}\) are given by

\[
Y_n(t) = [I, \ 0_{r \times r}] V_n(t), \quad \tilde{Y}_n(t) = [I, \ 0_{r \times r}] W_n(t),
\]

with \(V_n(t), W_n(t) \in \mathbb{R}^{2 \times r}\) verifying the initial value problems (IVPs)

\[
V_n'(t) = M(t, n) V_n(t), \quad V_n(0) = \begin{bmatrix} I_r, \ 0_{r \times r} \end{bmatrix} ;
\]

\[
W_n'(t) = M(t, n) W_n(t), \quad W_n(0) = \begin{bmatrix} 0_{r \times r}, \ P(0) \end{bmatrix} ;
\]

where

\[
M(t, n) = \begin{bmatrix} 0_{r \times r} & P^{-1}(t) \\ -\left(\frac{n \pi}{d}\right)^2 Q(t) & 0_{r \times r} \end{bmatrix} \in \mathbb{R}^{2 \times 2},
\]

see [16] for details. Here, \(0_{r \times r}, I_r \in \mathbb{R}^{r \times r}\) denotes the null and identity matrices, respectively. Note that, from our assumptions, \(P(t), Q(t)\) are non-singular matrices for all \(t \geq 0\).

Given a tolerance, it is possible to find \(n_0\) such that the truncated series for \(n \leq n_0\) has an error below than this tolerance. However, in general, the solution for the matrices \(Y_n(t), \tilde{Y}_n(t), n = 1, 2, \ldots, n_0\) cannot be obtained in a closed form and must be computed numerically (typically on a mesh \(0 < t_1 < t_2 < \ldots < t_L\) where \(L\) also depends on \(n_0\)), being this the most costly part for the algorithm.

Since the performance of standard explicit integrators deteriorates, in general, as the value of \(n\) grows, implicit methods are usually required to numerically solve the equations. However, in general, one needs to take \(L = \mathcal{O}(n_0^2)\), i.e. the mesh size has to be chosen inversely proportional to \(n_0^2\) and, in each interval, the method has to be applied \(n_0\) times (for \(n = 1, 2, \ldots, n_0\)). As a result, the matrices \(Q(t)\) and \(P^{-1}(t)\) need to be evaluated in a number of mesh points which grows as \(n_0^2\), and the method has to be applied \(\mathcal{O}(n_0^2)\) times.

On the other hand, most exponential integrators can deal efficiently with the numerical solution for relatively large values of \(n\). Usually, one can take \(L = \mathcal{O}(n_0)\) and on each mesh the exponentials have to be computed for each value of \(n \leq n_0\).

The contributions (under convergence conditions) of \(Y_n(t), \tilde{Y}_n(t)\) to the solution (3) decrease with \(n\), but the errors and computational cost of most numerical integrators increase with \(n\). For this reason, we consider a class of exponential integrators based on the Magnus series expansion which provides sufficiently accurate solutions as \(n\) grows and the exponentials can be computed using a simple recursive relation such that the computational cost of each term is irrespective of the value of \(n\). The numerical solutions obtained are also such that the series solution remains still convergent.

The paper is organized as follows. In Section 2, the convergence of the formal series solution (3) is established. In section 3, numerical methods based on Magnus expansion are proposed in order to solve the IVPs (6)–(8). Exploiting the structure of the matrices, we found that the computational cost of the proposed method is very advantageous with respect to standard numerical methods. The convergence of the series obtained by the numerical scheme is studied in Section 4. Section 5 deals with the presentation of numerical experiments in order to test the effectiveness of the proposed algorithm. Conclusions are presented in the last section.

Throughout this paper, \(\|\cdot\|_2\), denotes the usual Euclidean norm of a vector in \(\mathbb{R}^r\), and \(\|\cdot\|\), denotes the 2-norm of a square matrix in \(\mathbb{R}^{r \times r}\).

2. Convergence of the formal series solution

We first review the most relevant results on the convergence of the series solution (see [16] for more details). Under the assumptions (I)–(IV), existence solutions of (5)–(8) are guaranteed and, given \(T > 0\), there exists a constant \(\delta\) such that
\[\|Y_n(t)\| \leq \delta, \quad \|\bar{Y}_n(t)\| \leq \frac{\delta d}{n^2} \left(\|P^{-1}(0)\| \|Q(0)\|^{-1}\right)^{1/2},\]

for all \(n \geq 1\), and \(0 \leq t \leq T\), where \(\delta\) depends only of \(r\) and the data \(\|P(0)\|, \|P(t)\|, \|P^{-1}(0)\|, \|P^{-1}(t)\|, \|Q(0)\|, \|Q(t)\|, \|Q^{-1}(0)\|, \|Q^{-1}(t)\|\).

In addition, from [19, p. 71], the sine Fourier coefficients \(a_n, b_n\), satisfy

\[
\|a_n\|_2 \leq \frac{L_1}{n^2}, \quad \|b_n\|_2 \leq \frac{L_2}{n^2}, \quad n \geq 1,
\]

where

\[
L_1 = \frac{2d^2}{\pi^2} \int_0^d \|f^{(3)}\|_2(x) \, dx, \quad L_2 = \frac{2d^2}{\pi^2} \int_0^d \|g^{(2)}\|_2(x) \, dx,
\]

thus

\[
\sum_{n \geq 1} \|Y_n(t)\| \|a_n\|_2 < +\infty, \quad \sum_{n \geq 1} \|\bar{Y}_n(t)\| \|b_n\|_2 < +\infty,
\]

see [9, pp. 38–41], and the series (3) is uniformly convergent in \(D(d, T)\) and defines a continuous function that is a rigorous solution of problem (1), (2), see Theorem 4.1 of [16]. It is also proven that

\[
\left|\sum_{n \geq n_0} \left\{Y_n(t) a_n + \bar{Y}_n(t) b_n\right\} \sin\left(\frac{n\pi x}{d}\right)\right| < C \frac{1}{n_0^2},
\]

where \(C\) is a constant that depends on \(L_1, L_2, \delta, P^{-1}(0), Q(0)\), but does not depend on \(n_0\).

Our goal is to apply time-averaging methods based on the Magnus series expansion as exponential integrators for the numerical integration of IVPs (6),(7). Geometric integrators have shown in many cases a high performance for the numerical integration of IVPs, not only for a high accuracy, also because they preserve some of the qualitative properties of the exact solution, see [6]. As we will see, an appropriate choice of exponential integrators allows to obtain accurate and stable solutions at a low computational cost in comparison to standard numerical integrators.

### 3. Numerical approximation for the IVPs

#### 3.1. The autonomous case

Let us first consider the case in which \(Q\) and \(P^{-1}\) are positive definite constant matrices. In this case, the equations for \(Y_n(t)\) and \(\bar{Y}_n(t)\) have exact solution in a closed form. At this point, it is interesting to present the following results.

Let \(N, M \in \mathbb{R}^t\) symmetric positive definite matrices, then

\[
\exp\left(\begin{bmatrix} t & 0 & M \\ -N & 0 & \end{bmatrix}\right) = \begin{bmatrix} \cos(t\sqrt{NM}) & M(\sqrt{NM})^{-1} \sin(t\sqrt{NM}) \\ -N(\sqrt{NM})^{-1} \sin(t\sqrt{NM}) & \cos(t\sqrt{NM}) \end{bmatrix}.
\]

Given a matrix \(R \in \mathbb{R}^{n \times n}\), if we consider the property

\[
\exp((m + 1)R) = \exp(mR) \exp(R),
\]

it is obvious, by replacing \(t\) by \(m + 1\) and taking \(S\) either \(\sqrt{NM}\) or \(\sqrt{NM}\) in (10), that

\[
\cos((m + 1)S) = \cos(mS) \cos(S) - \sin(mS) \sin(S),
\]

\[
\sin((m + 1)S) = \sin(mS) \cos(S) + \cos(mS) \sin(S).
\]

If we take \(M = P^{-1}, N = n^2Q\), which are positive definite matrices, and taking into account that

\[
\Phi_0(T, 0) = \exp\left(\begin{bmatrix} 0 & P^{-1} \\ -n^2Q & 0 \end{bmatrix}\right),
\]

we find that

\[
\Phi_n(T, 0) = \begin{bmatrix}
\cos(nT\sqrt{P^{-1}Q}) & \frac{1}{n}P^{-1}(\sqrt{QP^{-1}})^{-1} \sin(nT\sqrt{QP^{-1}})
\\
-nQ(\sqrt{P^{-1}Q})^{-1} \sin(nT\sqrt{P^{-1}Q}) & \cos(nT\sqrt{QP^{-1}})
\end{bmatrix}.
\]

To compute \(\Phi_1(T, 0)\) requires to evaluate
\[
\cos \left( T \sqrt{P^{-1}Q} \right), \quad \sin \left( T \sqrt{QP^{-1}} \right), \quad \sin \left( T \sqrt{Q} \right). 
\]

Then, the computation of \( \Phi_n(T, 0) \), \( n > 1 \) can be carried out just by taking into account the recursive relations (11) and (12), which only involve few matrix multiplications.

It is very important to notice that the computational cost to evaluate \( \Phi_n \) once \( \Phi_{n-1} \) has already been obtained, is independent of \( n \), and we are interested in looking for numerical methods with this property.

### 3.2. The non-autonomous case

Suppose we are interested in approximating the solution of (3) on a time mesh, i.e., \( u(x, t) \) for \( t_i = t_0 + lh, \ l = 1, 2, \ldots, L \) and \( h = T/L \). This requires to compute numerical approximations to

\[
Y_n(t_i), \quad \bar{Y}_n(t_i), \quad n = 1, 2, \ldots, n_0, \quad l = 1, 2, \ldots, L.
\]

This can be achieved by solving numerically the following matrix equations

\[
V_n(t) = M(t, n) V_n(t), \quad V_{n,0} = V_n(t_{i-1}), \quad t \in [t_i, t_{i-1}]
\]

\[
W_n(t) = M(t, n) W_n(t), \quad W_{n,0} = \bar{W}_n(t_{i-1})
\]

\( l = 1, 2, \ldots, L, \quad n = 1, 2, \ldots, n_0 \), and where \( V_n(t_{i-1}), \ W_n(t_{i-1}) \) are the approximate solutions from the previous step.

To solve numerically this problem using standard methods, requires to apply the integrator repeatedly for each value of \( l \) and \( n \). In general, the computational cost increases considerably with \( n \), and what is even worst, the error also grows considerably with \( n \) and can make the numerical series expansion divergent. We will propose an algorithm which circumvent these two main troubles.

Let us now consider the IVPs (6) and (7), where \( M(t, n) \) is given by (8). If \( \Phi_n(t, t_0) \) denotes the fundamental matrix solution of the homogeneous matrix equation

\[
\Phi_n(t_0, t_0) = M(t, n) \Phi_n(t_0, t_0), \quad \Phi_n(t, t) = I \in \mathbb{R}^{2x2},
\]

then, the solution of the homogeneous Eqs. (6),(7) can be written in the form

\[
V_n(t) = \Phi_n(t_0, t) V_n(0), \quad W_n(t) = \Phi_n(t_0, t) W_n(0).
\]

Note that the dependence on \( n \) is given by \( \Phi_n \) because the initial conditions, \( V_n(0) \) and \( W_n(0) \), remain the same for all \( n \geq 1 \).

If we denote by \( V_{in}, I = 0, 1, \ldots, L, \) the values of \( V_n(t) \), at a mesh \( t_0 = 0, \ t_1 = h, \ldots, t_l = Lh, \) with \( h = T/L \), and \( \Phi_{nl} = \Phi_n(t_{i} + h, t_0) \), then

\[
V_{n+1, l} = \Phi_{nl} V_{in}, \quad l = 0, 1, \ldots, L - 1.
\]

At this point it is convenient to analyze the structure of the fundamental matrix solution. It is easy to prove that, for this problem, the matrix, \( \Phi_{nl} \), takes the form

\[
\Phi_{nl} = \begin{bmatrix}
\frac{1}{n} \tau_{nl} & \frac{1}{n} \zeta_{nl} \\
\frac{1}{n} \eta_{nl} & \frac{1}{n} \eta_{nl}
\end{bmatrix},
\]

where \( \tau_{nl}, \zeta_{nl}, \eta_{nl} \) are \( \mathbb{R}^{2x2} \)-valued functions bounded by a constant which does not depend of \( n \), and \( \Phi_{nl} = I + O(h) \) in the limit \( h \to 0 \) (or \( L \to \infty \)).

Then, for the integration over a finite time interval, \( t \in [0, T] \), with \( h = T/L \), we have that

\[
\Phi_n = \prod_{l=0}^{L-1} \Phi_{nl} = \begin{bmatrix}
A & \frac{1}{n} B \\
B & D
\end{bmatrix},
\]

where \( A, B, C, D \) are \( \mathbb{R}^{2x2} \)-valued functions which can also be bounded by functions depending on \( T, \ P(t), \ Q(t) \), but not on \( n \). As a result, standard explicit methods like explicit Runge–Kutta methods are not appropriate for solving this problem because for a fixed value of \( h \), the numerical solution grows polynomially with \( n \) and the series solution will not converge so, one has to use implicit methods.

The matrices \( \Phi_{nl} \) for \( l = 0, 1, \ldots, L - 1, \ n = 1, 2, \ldots, n_0 \), have to be numerically approximated, and this can lead to exceedingly costly algorithms. If we take the same time step, \( h \), for all values of \( n \), the matrices \( P(t), \ Q(t) \) need to be computed only once on each interval (if the same quadrature rule is used for all values of \( n \), i.e., the same values \( P(t_i + c_i h), \ Q(t_i + c_i h) \), can be used for all \( n \). However, as we will see, the performance of most standard methods rapidly deteriorate with \( n \). In addition, implicit methods usually need to compute the inverse of a matrix (the problem is linear) and the algorithm has to repeat this computation \( L \cdot n_0 \) times.

In the following, we show that some exponential methods have many advantages for the numerical integration of this problem. The exponential methods we consider in this work are explicit methods, but closely related to implicit methods for linear problems [6]. They show a better performance for large values of \( n \) (they converge to the exact solution in the limit
when \( P(t), Q(t) \) are constant) and there is a recursive algorithm which allows to compute all matrices \( \Phi_n, n = 2, 3, \ldots, n_0 \) from \( \Phi_{1,t} \).

### 3.3. Second order exponential integrator

Exponential integrators for non-autonomous linear problems have shown to be superior, both qualitatively and quantitatively, to standard methods for solving many linear IVPs with oscillatory solutions or for stiff problems. The effectiveness of these methods is far superior to traditional methods, like Runge–Kutta methods, see [4–7,11,15].

We first consider in detail the following second order approximation (in the time step \( h \)) to the fundamental matrix \( \Phi_n \)

\[
\exp \left( \int_{t_i}^{t_i+h} M(t, n) \right) = \Phi_n(t + h, t) + O(h^3),
\]

that corresponds to the first order approximation for most exponential methods like e.g. the Magnus, Fer or Wilcox expansions (see [2] and the references therein for details). Here, it suffices to approximate the integral by a second order symmetric rule. From the computational point of view, we found it useful to consider the trapezoidal rule

\[
\Upsilon_{n,t} = \exp \left( \frac{h}{2} (M(t_i + h, n) + M(t_i, n)) \right) = \Phi_{n,t} + O(h^3).
\]

Notice that given \( s > 0 \) and \( b_i > 0, 0 \leq c_i \leq 1, i = 1, \ldots, s \) we have that

\[
\sum_{i=1}^{s} b_i M(t + c_i h, n) = \begin{bmatrix}
0 & \hat{P}^{-1}(t) \\
-(\frac{\hat{Q}}{2})^T \hat{Q}(t) & 0
\end{bmatrix},
\]

where

\[
\hat{P}(t) = \left( \sum_{i=1}^{s} b_i \hat{P}^{-1}(t + c_i h) \right)^{-1}, \quad \hat{Q}(t) = \left( \sum_{i=1}^{s} b_i \hat{Q}(t + c_i h) \right).
\]

so \( \hat{P}, \hat{Q} \) are symmetric and positive definite matrices which do not depend on \( n \). If one relaxes the condition \( b_i > 0 \) to \( \sum_{i=1}^{s} b_i > 0 \) then, from the smoothness of the matrices \( P(t), Q(t) \) we can find a value, \( h^* \) such that if \( h < h^* \) it is still guaranteed that \( P(t), Q(t) \) are symmetric and positive definite matrices. Notice that in these cases we can compute \( \Upsilon_{n,t} \) from \( \Upsilon_{1,t} \) using the recursive schemes (11) and (12).

Summarizing, a symmetric second order exponential integrator is given by

\[
V_{i+1,n} = \exp \left( \frac{h}{2} (M(t_i + h, n) + M(t_i, n)) \right) V_{i,n};
\]

\[
W_{i+1,n} = \exp \left( \frac{h}{2} (M(t_i + h, n) + M(t_i, n)) \right) W_{i,n};
\]

where

\[
V_{0,n} = I_r; \quad W_{0,n} = \begin{bmatrix} I_r \\ P(0) \end{bmatrix}, \quad I_r = 0, 1, \ldots, L-1, \quad 1h = T.
\]

The same arguments applies to \( W_{i+1,n} = \exp \left( \frac{h}{2} (M(t_i + h, n) + M(t_i, n)) \right) W_{i,n}; \)

\[
W_{0,n} = \begin{bmatrix} 0 \\ P(0) \end{bmatrix}, \quad I_r = 0, 1, \ldots, L-1, \quad 1h = T.
\]

### 3.4. High order exponential integrators. Commutator-free methods

Magnus integrators up to very high orders can be found in the literature (see [2,3] and references therein). These methods, as well as most exponential integrators for linear problems, involve the computation of the exponential of a matrix that contains commutators. While this is not a serious problem for one single exponential, it is not a simple task to build a recursive algorithms to compute the matrix solution for \( n > 1 \) from the computation for \( n = 1 \).

For example,

\[
\exp \left( \Omega^{(4)}_{n,t} \right) = \exp \left( \frac{h}{2} (M_1 + M_2) - \frac{h^2}{12} (M_1, M_2) \right)
\]

with \( M_1 = M(t_i + c_1 h), \) and \( c_1 = \frac{1}{2} - \frac{\sqrt{7}}{4}, \quad c_2 = \frac{1}{2} + \frac{\sqrt{7}}{4}, \) correspond to a fourth-order Magnus integrator. Observe that

\[
\Omega^{(4)}_{n,t} = \begin{bmatrix}
-h^2 \frac{\sqrt{7}}{12} \frac{w^2}{\pi^2} [P_1^{-1}, Q_2] & \frac{h}{2} (P_1 + P_2^{-1}) \\
-(\frac{w}{\pi})^2 \frac{h}{2} (Q_1 + Q_2) & h^2 \frac{\sqrt{7}}{12} \frac{w^2}{\pi^2} [P_2^{-1}, Q_1]
\end{bmatrix},
\]
which makes very involved to write $\exp(\Omega_{i-1}^{[k]})$ in terms of $\exp(\Omega_{i-1}^{[k]})$, and to find a recursive algorithm is not an obvious task. Here, $[\cdot, \cdot]$ denotes the commutator, i.e. $[S, S] = SS - SS$.

For such reason, we consider more appropriate to use commutator-free Magnus integrators. A simple two exponential fourth-order commutator-free method for the interval $t \in [t_i, t_{i+1}]$ is given by [5]

$$Y^{[4]} = \exp\left(\frac{h}{2}(\beta M_1 + \alpha M_2)\right) \exp\left(\frac{h}{2}(\alpha M_1 + \beta M_2)\right),$$

with $\alpha = \frac{1}{2} + \frac{\sqrt{3}}{4}$, $\beta = \frac{1}{2} - \frac{\sqrt{3}}{4}$, with $M_1$, $M_2$ evaluated on the same Gaussian quadrature rule.

Notice that each exponent has the form (14) and a similar recursion as for the second order method can be used to compute the exponentials for $n > 1$. Notice that $\beta < 0$, but $\frac{1}{2}(\alpha + \beta) = \frac{1}{2} > 0$. However, all existing commutator-free methods of order six or higher [1,5] are such that at least one of the exponents has the form (14) with $\sum_i b_i < 0$, and this can cause serious troubles on the existence and positivity of the matrices.

To illustrate the comparison between the second and fourth-order commutator-free Magnus integrator is seems more appropriate to take an equidistant time mesh and to use the following fourth-order commutator-free method

$$Y^{[4]} = \exp\left(\frac{h}{2}\left(-\frac{1}{6}M_1 + \frac{2}{3}M_2 + \frac{1}{2}M_3\right)\right) \exp\left(\frac{h}{2}\left(\frac{1}{2}M_1 + \frac{2}{3}M_2 - \frac{1}{6}M_3\right)\right),$$

where now $M_1 = M(t_i)$, $M_2 = M(t_i + h/2)$, $M_3 = M(t_i + h)$. This scheme should be compared with two consecutive steps with time step $\frac{h}{2}$ of the second order method, i.e.

$$Y^{[2]} = \exp\left(\frac{h}{2}\left(\frac{1}{2}M_2 + \frac{1}{2}M_3\right)\right) \exp\left(\frac{h}{2}\left(\frac{1}{2}M_1 + \frac{1}{2}M_2\right)\right).$$

Both methods require the same evaluations of $M_t$, and the same number of exponentials. Unfortunately, the analysis carried in Section 3.2 does not apply to higher order Magnus integrators.\(^1\) The error with $n$ will not follow the same rule. The convergence of the Magnus series expansion is not necessarily guaranteed as the value of $n$ grows, and a higher order method can provide worse results. This will be illustrated in the numerical examples.

As a result, the choice of the method to be used will depend on the accuracy desired, i.e. the size of the time step considered, as well as the number of terms to be considered in the series. For large values of $h$ or large values of $n$ the second order method performs better but for medium or relatively small values of $h$ and $n$ fourth-order methods are clearly superior, and this will depend on each problem.

Obviously, in spite of the recurrence relations, the performance of the algorithms will be improved by using efficient algorithms to compute the matrix exponential for $n = 1$ or its action on a vector (see for example the appendix B of [6], or [7,14,18]).

### 4. On the convergence of the numerical scheme

For the sake of clarity in the presentation, let us consider the matrix problem

$$Z'(t) = \tilde{M}(t, n)Z(t),$$

where $\tilde{M}(t, n)$ is given by

$$\tilde{M}(t, n) = \begin{bmatrix} 0 & p^{-1}(t) \\ -\left(\frac{\pi}{\sqrt{2}}\right)^2 q(t) & 0 \end{bmatrix}.$$ 

Here, we consider that $p(t)$ and $q(t)$ are scalar functions verifying the conditions (I)–(II) in the scalar case. Applying the numerical algorithm (13) we have

$$Z_{1,1,1} = \exp\left(\frac{h}{2}\left(\tilde{M}(t_i + h, n) + \tilde{M}(t_i, n)\right)\right) Z_{t_i}.$$

Note that

$$\frac{h}{2}\left(\tilde{M}(t_i + h, n) + \tilde{M}(t_i, n)\right) = \begin{bmatrix} 0 & \alpha(l, h) \\ -n^2 \beta(l, h) & 0 \end{bmatrix},$$

where

$$\alpha(l, h) = \frac{h}{2} (p^{-1}(t_i + h) + p^{-1}(t_i)), \quad \beta(l, h) = \frac{h \pi^2}{2d^2} (q(t_i + h) + q(t_i)).$$

\(^1\) The two exponential commutator-free methods have negative coefficients in both exponents, and $\Omega_{i-1}^{[k]}$ in (17) has a different matrix structure.
Thus, following the same arguments shown in the Section 3.3

\[
\exp \begin{bmatrix} 0 & \gamma(l, h) \\ -n^2 \beta(l, h) & 0 \end{bmatrix} = \begin{bmatrix} \cos(n \gamma(l, h)) & \frac{1}{n} \sin(n \gamma(l, h)) \\ -n \rho(l, h) \sin(n \gamma(l, h)) & \cos(n \gamma(l, h)) \end{bmatrix},
\]

where

\[
\gamma(l, h) = \sqrt{\alpha(l, h) \beta(l, h)}, \quad \rho(l, h) = \frac{\beta(l, h)}{\alpha(l, h)}.
\]

In order to achieve the step \( L \), the product of \( L \) matrices of this type must be performed. But note that the result is a matrix with the same dependence in \( n \)

\[
\begin{bmatrix}
\mu_1(n, h) & \frac{1}{n} \mu_2(n, h) \\
-\mu_3(n, h) & \mu_4(n, h)
\end{bmatrix},
\]

(19)

where \( \mu_i(n, h), \ 1 \leq i \leq 4, \) are a span of bounded trigonometric functions.

Multiplying for the appropriate starting condition and taking into account (5)–(8), it follows that

\[
y_{nL} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix}
\mu_1(n, h) & \frac{1}{n} \mu_2(n, h) \\
-\mu_3(n, h) & \mu_4(n, h)
\end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \mu_1(n, h),
\]

\[
\bar{y}_{nL} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix}
\mu_1(n, h) & \frac{1}{n} \mu_2(n, h) \\
-\mu_3(n, h) & \mu_4(n, h)
\end{bmatrix} \begin{bmatrix} 0 \\ p(0) \end{bmatrix} = \frac{1}{n} \mu_2(n, h) p(0).
\]

From (9) we have

\[
y_{nL} a_n + \bar{y}_{nL} b_n \to \frac{1}{n^4},
\]

and the series

\[
\sum_{n \geq 1} \{y_{nL} a_n + \bar{y}_{nL} b_n\},
\]

is uniformly convergent.

In this way, the advantages of exponential methods versus standard multi-step numerical methods for this problem are clear. The dependence on \( n \) after \( L \) steps given by (19) is essential in order to ensuring the convergence of the series. For instance, if an explicit Runge–Kutta method is applied, we have

\[
Z_{L,n} = \text{P}(\tilde{M}(t, n)) Z_{0,n},
\]

where \( \text{P}(\tilde{M}(t, n)) \) is a polynomial in \( n \) of degree \( L + 1. \) Also, for implicit Runge–Kutta methods, the relationship between the last step and the starting condition has the form

\[
Z_{L,n} = n F(\tilde{M}_n) Z_{0,n},
\]

such that, if the series is truncated at \( n_0, \) requires to take \( h \ll 1/n_0 \) in order to ensure that \( F(n, h) \) provides a close approximation of the solution (see [12] and references therein).

5. Testing the numerical algorithm

For simplicity, let us suppose that \( u(x, 0) = u_t(x, 0), \) and then, \( f(x) = g(x), \ 0 \leq x \leq d. \) In this way, the Fourier coefficients \( a_n \) and \( b_n \) given by (4) are equal and, from (5), we can write (3) as

\[
u(x, t) = \int_0^t \sum_{n \geq 1} \sin \left( \frac{n \pi x}{d} \right) \Psi_n(t) a_n,
\]

where

\[
\Psi_n(t) = \langle V_n(t) + W_n(t) \rangle \in \mathbb{R}^{2 \pi}, \quad n \geq 1.
\]

Thus, from (15)–(16), a second order exponential integrator for \( \Psi_n(t) \) is given by

\[
\Psi_{l+1,n} = \exp \left\{ \begin{array}{l}
\frac{1}{2} \left( M(t_l) + h, n \right) + M(t_l, n) \\
\end{array} \right\} \Psi_{l,n}, \quad l = 0, 1, \ldots, L - 1, \quad Lh = T,
\]

(20)
where
\[ \Psi_{1n} = V_{1n} + W_{1n}, \quad \Psi_{0n} = V_{0n} + W_{0n} = \left[ I_r \right] P(0), \]
and \( M(t; n) \) are the values of \( M(t; h) \) given by (8), evaluated in the mesh points \( t_l = lh \).

As an illustrative example, let us consider the scalar problem with data
\[ p(t) = 3 - \arctan(t), \quad q(t) = 1 + \arctan(t), \]
that verifies the required conditions
\[ p(t) > 0, \quad q(t) > 0, \quad -p'(t) = q'(t) = \frac{1}{1+t^2} > 0, \quad t \geq 0. \]

Using (20) and the initial condition (21), with \( p(0) \) given by (22), we can write
\[ W_{1l} = \sum_{k=0}^{n-1} \exp \left( \frac{h}{2} (M_{l,k,n} + M_{l,k-1,n}) \right) \left[ \begin{array}{c} 1 \\ \frac{1}{3} \end{array} \right]. \]

Let us take \( T = 1 \) and \( h = 0.01 \), i.e., the interval \([0, 1]\), is divided into one hundred knots such that
\[ t_l = lh, \quad t_{l+1} = t_l + h = (l+1)h, \quad t_{100} = 100h = 1. \]

From (8) and the coefficients \( p(t), q(t) \) given by (22), we have
\[ M_{1n} = \begin{bmatrix} 0 & -\frac{1}{3} \arctan(lh) \\ \frac{1}{3} \arctan(lh) & 0 \end{bmatrix}. \]

Thus, if we denote by
\[ \psi_{l,n}(x) = \left[ \begin{array}{l} 0 \\ \sum_{k=1}^{n} \sin \left( \frac{knx}{d} \right) \Psi_{lk} \end{array} \right], \]
where \( \Psi_{lk} \) is given by (23), then the approximate solution of (1) at \( t = 1 \), is given by
\[ u(x, 1) = \sum_{n \geq 1} \psi_{100,n}(x) a_n, \]
where \( a_n = b_n \), is given by (4).

In order to show the performance of the exponential methods considered in this work as \( n \) increases, we consider separately both components of \( \Psi_{1n} = (y_1, y_2) \) in (21). In Fig. 1, we show the error at final time, \( T = 1 \), and constant time step

**Fig. 1.** Error at \( T = 1 \) to approximate \( \Psi_{1n} = (y_1(T), y_2(T)) \) for \( n = 1, 2, ..., 100 \) using the following methods: the implicit second order trapezoidal Runge–Kutta method (IRK2), the second order Magnus integrator (MAG2), both with a time step \( h = 1/100 \), and the commutator-free Magnus integrator (18) with a time step twice as large, \( h = 1/50 \) (CF4). Here, \( y_1, y_2 \) denote the exact solutions obtained numerically to high accuracy while \( y_{1,ap}, y_{2,ap} \) denote the approximated solutions obtained by the numerical methods.
h = 1/100, for both components when considering the second order Magnus integrator (21) or the implicit second order trapezoidal Runge–Kutta method, and we take h = 1/50 for the fourth-order commutator-free method (18), so that the computational cost is the same as for the second order Magnus integrator. The exact solution is computed numerically with a sufficiently high accuracy. The computational cost is similar for all schemes in this simple scalar test problem. For higher dimensional problems, the computational cost of the implicit RK methods would increase with n because it would require a larger number of iteration to converge, but this is not the case of the exponential methods due to the recurrence relation, as already mentioned. The superiority of the exponential method as the value of n grows is then clear.

In order to check the convergence of the series (24) for l = 100, the Fig. 2 shows the results for ψ_{100,n}, choosing d = 1 and x = 0.3, x = 0.6, x = 0.9, respectively. Note that, taking just over ten terms, the series offers stability in the results for all cases.

In order to verify the effectiveness and simplicity in the implementation of the proposed numerical method, we repeat the previous computations for the case d = 2 at T = 1. Fig. 3 corresponds to ψ_{100,n} for values of x at 0.5, 1 and 1.5. Note that the number of terms in the series necessary to obtain convergence increase with the value of x. However, for larger values of T, the series stabilizes quickly.

Fig. 4 shows the results obtained for d = 2 and T = 10 with the same h = 0.01, i.e., taking one thousand knots in the time domain [0, 10]. We observe that it suffices to take less than five terms in the series to stabilize the results.

Finally, Fig. 5 shows the evolution in the variable x of ψ_{l,n}(x) for T = 1 i.e., taking l = 100. Here, the number of terms in the series is taken for n = 100 and d = 1.2.

Fig. 2. ψ_{100,n}(x) at x = 0.3, 0.6 and 0.9, with d = 1. Here, n is the number of terms taken in the series (24).

Fig. 3. ψ_{100,n}(x) at x = 0.5, 1 and 1.5, with d = 2. Here, n is the number of terms taken in the series (24).
6. Conclusions

We have considered the numerical integration of coupled self-adjoint non-autonomous partial differential systems. Under convergence conditions the solution can be written as a series expansion where each term (oscillatory time-dependent matrices) must be computed numerically. Accurate results require, in general, to increase the number of terms in the series. While these new terms provide small contributions to the final solution, their numerical solutions are the most challenging tasks due to the numerical error which can make the numerical series to diverge as well as they are the most computationally costly terms.

In this work we analyze second order of Magnus integrators whose numerical error grows with $n$ (the number of terms considered in the truncated series) at a rate that still allows us to guarantee convergence of the numerical series. In addition, the integrator can be implemented with a recursive algorithm such that the computational cost of the method grows only linearly with the number of terms of the series.

We present a time-symmetric second order exponential integrator which shows a high performance for these problems. This method leads to expression (23),(24), that it is easy to implement at a relatively low computational cost. Here, it is important to recall that, as seen in Section 3, once evaluated for $n = 1$, the proposed method admits a recursive scheme to find the solutions for $n > 1$ and it is proved that, from the structure of the numerical solution, at successive iterations the error grows similarly as does the exact solution and hence the convergence of the series is unaffected. This does not happen when other standard numerical methods are used. The numerical solutions provided verify that the difference with the theoretical exact solution differs in a quantity less than $h^3$, where $h$ is the measure of meshing which divides the time domain. To improve the accuracy we can make a finest mesh. Alternatively, higher order Magnus integrators can be used. Commutator-free Magnus integrators can be used with a similar recursive algorithm and can provide highly accurate results, but they show a faster error growth with $n$, and some caution must be taken if these methods are used. The numerical experiments from the previous section confirm the performance of the proposed algorithm, showing excellent results at a relatively low computational cost.

Fig. 4. $\psi_{1000,n}(x)$ at $x = 0.5, 1$ and $1.5$, with $d = 2$. Here, $n$ is the number of terms taken in the series (24).

Fig. 5. Left. $\psi_{100,100}(x)$, i.e, the approximation of the function $\psi(x)$ at $T = 1$, taking one hundred terms in the series (24) and evaluated also in one hundred points of the domain $[0, 1]$. Right. The same for $x \in [0, 2]$.

6. Conclusions

We have considered the numerical integration of coupled self-adjoint non-autonomous partial differential systems. Under convergence conditions the solution can be written as a series expansion where each term (oscillatory time-dependent matrices) must be computed numerically. Accurate results require, in general, to increase the number of terms in the series. While these new terms provide small contributions to the final solution, their numerical solutions are the most challenging tasks due to the numerical error which can make the numerical series to diverge as well as they are the most computationally costly terms.

In this work we analyze second order of Magnus integrators whose numerical error grows with $n$ (the number of terms considered in the truncated series) at a rate that still allows us to guarantee convergence of the numerical series. In addition, the integrator can be implemented with a recursive algorithm such that the computational cost of the method grows only linearly with the number of terms of the series.

We present a time-symmetric second order exponential integrator which shows a high performance for these problems. This method leads to expression (23),(24), that it is easy to implement at a relatively low computational cost. Here, it is important to recall that, as seen in Section 3, once evaluated for $n = 1$, the proposed method admits a recursive scheme to find the solutions for $n > 1$ and it is proved that, from the structure of the numerical solution, at successive iterations the error grows similarly as does the exact solution and hence the convergence of the series is unaffected. This does not happen when other standard numerical methods are used. The numerical solutions provided verify that the difference with the theoretical exact solution differs in a quantity less than $h^3$, where $h$ is the measure of meshing which divides the time domain. To improve the accuracy we can make a finest mesh. Alternatively, higher order Magnus integrators can be used. Commutator-free Magnus integrators can be used with a similar recursive algorithm and can provide highly accurate results, but they show a faster error growth with $n$, and some caution must be taken if these methods are used. The numerical experiments from the previous section confirm the performance of the proposed algorithm, showing excellent results at a relatively low computational cost.
Acknowledgements

The work of S. Blanes has been partially supported by Ministerio de Ciencia e Innovación (Spain) under project MTM2010-18246-C03. The work of E. Ponsoda has also been partially supported by the Universitat Politècnica de València under project 2087.

References