



# Exponential Integrators for Second-Order in Time Partial Differential Equations

Duy Phan<sup>1</sup> · Alexander Ostermann<sup>1</sup>

Received: 21 December 2021 / Revised: 10 August 2022 / Accepted: 28 August 2022 /  
Published online: 12 October 2022  
© The Author(s) 2022

## Abstract

Two types of second-order in time partial differential equations, namely semilinear wave equations and semilinear beam equations are considered. To solve these equations with exponential integrators, we present an approach to compute efficiently the action of the matrix exponential as well as those of related matrix functions. Various numerical simulations are presented that illustrate this approach.

**Keywords** Semilinear wave equations · Semilinear beam equations · Exponential integrators · Computation of matrix functions

## 1 Introduction

We consider semilinear damped wave equations with damping term, structural (visco-elastic) damping term, and mass term

$$\partial_{tt}^2 u - \alpha \Delta u - \beta \Delta(\partial_t u) + \gamma \partial_t u + \delta u = g(u) + h(\partial_t u), \quad (x, t) \in \Omega \times (0, T), \quad (1a)$$

$$u|_{\partial\Omega} = 0, \quad t \in (0, +\infty), \quad (1b)$$

$$u(0, x) = p(x), \quad \partial_t u(0, x) = q(x), \quad x \in \Omega \quad (1c)$$

on a bounded and open domain  $\Omega \subset \mathbb{R}^d$  with smooth compact boundary  $\partial\Omega$ . The term  $\Delta(\partial_t u)$  is the structural (visco-elastic) damping while the term  $\partial_t u$  is the damping term. We assume that  $\beta, \gamma$ , and  $\delta$  are three non-negative coefficients. Moreover, the coefficient  $\alpha$  must be positive. The initial data  $p$  and  $q$  are chosen from the usual energy space  $(p, q) \in H_0^1(\Omega) \times L^2(\Omega)$ . Concerning the nonlinear term, we recall some particular equations from the literature:

---

✉ Duy Phan  
duy.phan-duc@uibk.ac.at

Alexander Ostermann  
alexander.ostermann@uibk.ac.at

<sup>1</sup> Institut für Mathematik, Universität Innsbruck, Technikerstraße 13, 6020 Innsbruck, Austria

(i) the perturbed sine-Gordon equation (see [4, 13, 27])

$$\partial_{tt}^2 u - \alpha \Delta u - \beta \Delta(\partial_t u) + \gamma \partial_t u = \sin u; \tag{2}$$

(ii) the perturbed wave equation of quantum mechanics (see [4, 13, 27])

$$\partial_{tt}^2 u - \alpha \Delta u - \beta \Delta(\partial_t u) = -|u|^q u - |\partial_t u|^p (\partial_t u), \quad p, q \geq 0. \tag{3}$$

Another type of second-order in time PDE is the Euler–Bernoulli beam equation with Kelvin–Voigt damping

$$\partial_{tt}^2 u + \partial_{xx}^2 (\alpha \partial_{xx}^2 u + \beta \partial_{xx}^2 (\partial_t u)) + \gamma \partial_t u + \delta u = g(u), \quad (x, t) \in (0, L) \times (0, T), \tag{4a}$$

$$u(0, x) = p(x), \quad \partial_t u(0, x) = q(x), \quad x \in (0, L), \tag{4b}$$

where  $u(t, x)$  denotes the deflection of the beam of its rigid body motion at time  $t$  and position  $x$ . For given parameters  $\alpha > 0$  and  $\beta \geq 0$ , the moment function is

$$m(t, x) = \alpha \partial_{xx}^2 u(t, x) + \beta \partial_{xx}^2 (\partial_t u)(t, x).$$

The first derivative of the moment  $m(t, x)$  with respect to the variable  $x$  represents the shear force. The following boundary conditions will be considered, where  $\xi \in \{0, L\}$ :

- (a) Hinged end:  $u(t, \xi) = 0, \quad m(t, \xi) = 0.$
- (b) Clamped end:  $u(t, \xi) = 0, \quad \partial_x u(t, \xi) = 0.$
- (c) Free end:  $m(t, \xi) = 0, \quad \partial_x m(t, \xi) = 0.$
- (d) Sliding end:  $\partial_x u(t, \xi) = 0, \quad m(t, \xi) = 0.$

Depending on the set up of the beam model, various combinations of boundary conditions are of interest, for example: hinged-hinged boundary conditions

$$u(t, 0) = 0, \quad m(t, 0) = 0, \quad u(t, L) = 0, \quad m(t, L) = 0.$$

Concerning semilinear beam equations, in [1, 10, 11], a nonlinear term  $g(u) = -lu^3, \quad l > 0$  was used when the authors considered a railway track model.

Both problems (1) and (4) can be rewritten as abstract ordinary differential equations in a product space  $X = H \times L^2(\Omega)$  by denoting a new variable  $y(t, x) = (u(t, x), w(t, x))'$  as follows

$$\dot{y}(t) = \mathcal{A}y(t) + \mathcal{F}(y(t)),$$

where

$$\mathcal{A} = \begin{bmatrix} 0 & I \\ -\alpha(-\Delta) - \delta I & -\beta(-\Delta) - \gamma I \end{bmatrix} \text{ for (1),}$$

$$\mathcal{A} = \begin{bmatrix} 0 & I \\ -\alpha \partial_{xxxx}^4 - \delta I & -\beta \partial_{xxxx}^4 - \gamma I \end{bmatrix} \text{ for (4),}$$

and

$$\mathcal{F}(u, w) = \begin{bmatrix} 0 \\ g(u) + h(w) \end{bmatrix}.$$

The space  $H$  and the domain of the operator  $\mathcal{A}$  will be chosen to be consistent with the boundary conditions. Here and henceforth, the transpose of a matrix  $E$  is denoted by  $E'$ .

These types of equations have been studied extensively in many fields of mathematics. For damped wave equations, see [4, 5, 13, 20, 27, 30]; for Euler–Bernoulli beam equations,

see [1, 2, 10, 11, 21, 23, 25, 26, 28, 29] and references therein. The time discretization of these equations, to the best of our knowledge, is usually carried out by standard integration schemes such as Runge–Kutta methods or multistep methods. In this article, we will consider exponential integrators to solve this class of PDEs. By spatial discretization of (1) or of (4), we get a semi-discretization of the equation in matrix form

$$\dot{y}(t) = Ay(t) + F(y(t)), \quad y(0) = y_0 = (p, q)', \tag{5}$$

where

$$A = \begin{bmatrix} 0 & I \\ -\alpha S - \delta I & -\beta S - \gamma I \end{bmatrix}. \tag{6}$$

and the square matrix  $S$  is the discretized version of the operator  $(-\Delta)$  or  $\partial_{xxxx}^4$ . The linear part of (5)

$$\dot{y}(t) = Ay(t), \quad y(0) = y_0 = (p, q)', \tag{7}$$

can be solved exactly, e.g.,

$$y(t) = e^{tA}y_0, \quad t > 0. \tag{8}$$

For the undamped wave equations (i.e.  $\beta = \gamma = 0$  in (6)), by using the matrix sine and matrix cosine functions, the explicit form of the matrix exponential  $e^{tA}$  is easily obtained (see [19, Section 3.2]). Based on this formula, Gautschi (in [12]) and Deuffhard (in [8]) developed a number of schemes to tackle semilinear second-order differential equations. Nevertheless, when damping terms appear in (6), a direct approach to compute the matrix exponential  $e^{tA}$  is more involved and not yet discussed in the literature. Therefore, in this paper, we firstly present an approach to exactly evaluate the matrix exponential of (6).

Let us briefly explain our procedure to compute the matrix exponential. We start by employing two linear transformations to represent the matrix  $A$  as  $A = \tilde{Q}PCP'\tilde{Q}'$  where the new matrix  $C$  is a block diagonal matrix, i.e.  $C = \text{diag}(G_1, \dots, G_n)$ . Each block  $G_i$  is a  $2 \times 2$  matrix. The exponential of such a matrix  $G_i$  will be computed explicitly. Regarding its eigenvalues, a suitable formula will be constructed. In this way the matrix exponential  $e^{tA}$  can be computed cheaply even for large values of  $t$ . We also discuss the cases  $\beta = \gamma = \delta = 0$  and  $\beta, \gamma \ll \alpha$  (see [21, Section 3] for typical physical parameters). In both cases, the matrix  $G_i$  has usually two conjugate complex eigenvalues. To reduce the computation cost, we avoid complex arithmetic. The exact matrix exponential will not only give us a huge advantage to solve the class of linear damped wave equations or linear beam equations but also be valuable in computing solutions of semilinear problems. The numerical schemes for the full equation (18) were constructed by incorporating the exact solution of (7) in an appropriate way. In the literature, these methods were investigated by many authors (see, e.g., [7, 9, 17–19, 22, 24, 31]). To employ these known exponential integrators, the core point is the computation of related matrix functions  $\varphi_k(tA)$ . As for the matrix exponential, we will use two linear transformations and compute the action of the matrix functions  $\varphi_k(tG_i)$ . Explicit formulas will be established in the same way as for computing the matrix exponential  $e^{tG_i}$ . Concerning the computation of matrix functions, we refer to the review work by Higham and Al-Mohy [16] as well as the monograph by Higham [15].

The outline of the paper is as follows. We start with the discussion of computing the matrix exponential  $e^{tA}$  in Sect. 2. Two linear transformations  $P$  and  $Q$  will be presented. The computations of the matrix exponential  $e^{tG_i}$  will be discussed for three different cases. In simulations, instead of computing the matrix exponential, we will rather compute its action

on a given vector. A detailed instruction will be presented in Remark 2.7. In Sect. 3 we recall some exponential integrators and discuss an approach to compute the action of the related matrix functions  $\varphi_k(tA)$ . The procedure will be summarized in Sect. 3.3. In Sect. 4, we will present some numerical examples of semilinear equations. The operators  $(-\Delta)$  and  $\partial_{xxx}^4$  will be discretized by finite differences. We will use exponential integrators for the time integration of these examples. Some comparisons with standard integrators will be presented in Sect. 4.3 to clarify the efficiency of our approach.

## 2 Exact Matrix Exponential

In this section, we propose an approach to compute efficiently the matrix exponential  $e^{tA}$  for a matrix  $A$  of the form (6). With this at hand, the solution of linear system (7) can be evaluated for an arbitrary time  $t > 0$  in a fast and reliable way.

### 2.1 Two Linear Transformations

The key idea is to transform  $A$  to a simple block-diagonal matrix for which the exponential can be computed cheaply.

**Lemma 2.1** *Assume that there exist an orthogonal matrix  $Q$  and a diagonal matrix  $D = \text{diag}\{\lambda_1, \dots, \lambda_n\}$  such that  $S = QDQ'$ , then the matrix  $A$  of form (6) can be transformed to the block form*

$$B = \begin{bmatrix} 0 & I \\ D_1 & D_2 \end{bmatrix}, \tag{9}$$

where  $D_1$  and  $D_2$  are two diagonal matrices.

**Proof** By substituting  $S = QDQ'$  and  $QQ' = I$  into (6), we get that

$$\begin{aligned} A &= \begin{bmatrix} 0 & QQ' \\ -\alpha QDQ' - \delta QQ' & -\beta QDQ' - \gamma QQ' \end{bmatrix} \\ &= \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} 0 & I \\ -\alpha D - \delta I & -\beta D - \gamma I \end{bmatrix} \begin{bmatrix} Q' & 0 \\ 0 & Q' \end{bmatrix}. \end{aligned}$$

The proof is complete by identifying two diagonal matrices  $D_1 = -\alpha D - \delta I$  and  $D_2 = -\beta D - \gamma I$ . □

**Lemma 2.2** *Consider  $P \in \mathbb{R}^{2n \times 2n}$  the permutation matrix satisfying*

$$P_{i,2i-1} = 1, \quad P_{i+n,2i} = 1 \quad \text{for } 1 \leq i \leq n, \quad P_{k,l} = 0 \text{ else.} \tag{10}$$

*The matrix  $B$  given in (9) can be transformed under the permutation  $P$  to a block diagonal matrix  $C$ , i.e.  $B = PCP'$ , where*

$$C = \begin{bmatrix} G_1 & 0 & \dots & 0 \\ 0 & G_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & G_n \end{bmatrix} \quad \text{with } G_i = \begin{bmatrix} 0 & 1 \\ -\alpha\lambda_i - \delta & -\beta\lambda_i - \gamma \end{bmatrix}.$$

**Proof** Following the definitions of the matrices  $B$  and  $C$ , for  $1 \leq i \leq n$  we have

$$B_{i,n+i} = 1, \quad B_{n+i,i} = -\alpha\lambda_i - \delta, \quad B_{n+i,n+i} = -\beta\lambda_i - \gamma,$$

$$C_{2i-1,2i} = 1, \quad C_{2i,2i-1} = -\alpha\lambda_i - \delta, \quad C_{2i,2i} = -\beta\lambda_i - \gamma.$$

We will prove that  $B = PCP'$ . Indeed, for  $1 \leq i \leq n$ , we have

$$(PCP')_{n+i,n+i} = P_{n+i,2i}(CP')_{2i,n+i} = (CP')_{2i,n+i} = C_{2i,2i}P'_{2i,n+i}$$

$$= C_{2i,2i}P_{i+n,2i} = C_{2i,2i} = -\beta\lambda_i - \gamma = B_{n+i,n+i},$$

$$(PCP')_{i,n+i} = P_{i,2i-1}(CP')_{2i-1,n+i} = (CP')_{2i-1,n+i} = C_{2i-1,2i}P'_{2i,n+i}$$

$$= C_{2i-1,2i} = 1 = B_{i,n+i},$$

$$(PCP')_{n+i,i} = P_{n+i,2i}(CP')_{2i,i} = (CP')_{2i,i} = C_{2i,2i-1}P'_{2i-1,i}$$

$$= C_{2i,2i-1} = -\alpha\lambda_i - \delta = B_{n+i,i}.$$

We will not be concerned with the remaining elements of  $B$  and  $C$  since they are all zero. Thus, the proof is complete. □

**Example 2.3** For  $n = 2$  and  $n = 3$  the permutation matrices  $P$  have the following form

$$P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Next, we recall some important properties of matrix functions (see [15, Theorem 1.13] or [16, Theorem 2.3]).

**Theorem 2.4** Let  $A \in \mathbb{C}^{n \times n}$  and  $f$  be defined on the spectrum of  $A$ . Then

- (a)  $f(A)$  commutes with  $A$ ;
- (b)  $f(A') = f(A)'$ ;
- (c)  $f(XAX^{-1}) = Xf(A)X^{-1}$ .
- (d) The eigenvalues of  $f(A)$  are  $f(\lambda_i)$ , where  $\lambda_i$  are the eigenvalues of  $A$ .
- (e) If  $A = (A_{ij})$  is block triangular then  $F = f(A)$  is block triangular with the same block structure as  $A$ , and  $F_{ii} = f(A_{ii})$ .
- (f) If  $A = \text{diag}(A_{11}, A_{22}, \dots, A_{mm})$  is block diagonal then

$$f(A) = \text{diag}(f(A_{11}), f(A_{22}), \dots, f(A_{mm})).$$

A direct consequence of this theorem is the following result.

**Theorem 2.5** Assume that there exist an orthogonal matrix  $Q$  and a diagonal matrix  $D = \text{diag}\{\lambda_1, \dots, \lambda_n\}$  such that  $S = QDQ'$ . Then, for  $t > 0$ , the exponential of the matrix  $tA$  is computed as follows

$$e^{tA} = \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} P \begin{bmatrix} e^{tG_1} & 0 & \dots & 0 \\ 0 & e^{tG_2} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & e^{tG_n} \end{bmatrix} P' \begin{bmatrix} Q' & 0 \\ 0 & Q' \end{bmatrix}, \tag{11}$$

where  $P \in \mathbb{R}^{2n \times 2n}$  is defined by (10) and  $G_i = \begin{bmatrix} 0 & 1 \\ -\alpha\lambda_i - \delta & -\beta\lambda_i - \gamma \end{bmatrix}$ .

**Proof** The two Lemmas 2.1 and 2.2 imply that

$$A = \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} B \begin{bmatrix} Q' & 0 \\ 0 & Q' \end{bmatrix} = \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} P \begin{bmatrix} G_1 & 0 & \cdots & 0 \\ 0 & G_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G_n \end{bmatrix} P' \begin{bmatrix} Q' & 0 \\ 0 & Q' \end{bmatrix}.$$

Formula (11) is proved by using the properties (c) and (f) in Theorem 2.4. □

**Remark 2.6** We need to compute the matrix exponential of the small matrices  $tG_i$ . This will be presented in the next section. The exponential matrix  $e^{tG_i}$  can be computed explicitly by using formula (14), (15), or (16) depending on the sign of  $(\beta\lambda_i + \gamma)^2 - 4(\alpha\lambda_i + \delta)$ .

**Remark 2.7** In practical situations, to reduce the computational cost, we will compute the action of the matrix exponential to a vector instead of computing it explicitly. In (11),  $P$  and  $Q$  are two square matrices of orders  $2n$  and  $n$ , respectively. Since  $P$  is a permutation matrix, it can be stored, however, as a column matrix with  $n$  entries by indicating the positions of the non-zero elements, for example:

$$P_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 3 \\ 5 \\ 2 \\ 4 \\ 6 \end{bmatrix}, \quad P'_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 4 \\ 2 \\ 5 \\ 3 \\ 6 \end{bmatrix}$$

The block matrix  $[e^{tG_i}]$  can be stored as a  $2 \times 2n$  matrix. Given a compound vector  $v_0 = [a, b]'$ , where  $a$  and  $b$  are two column vectors with  $n$  entries, we start by evaluating a new vector  $v_1 = [Q'a, Q'b]'$ . Next, the action of the permutation matrix  $P'$  to the vector  $v_1$  is the reorder of its entries to get a new vector  $v_2$ . Then we compute the multiplication of the block exponential matrix with the vector  $v_3$  by cheaply multiplying each block  $2 \times 2$  matrix  $e^{tG_i}$  with two corresponding elements of  $v_2$ . Analogously applying  $P$  and  $Q$ , we get an exact valuation of the action of the matrix exponential to an arbitrary vector.

### 2.2 The Matrix Exponential $G_i$

From (11), instead of evaluating the matrix exponential of  $A \in \mathbb{R}^{2n \times 2n}$ , we need to compute the matrix exponential of each  $G_i \in \mathbb{R}^{2 \times 2}$ . In this section, we give some explicit formulas. For simplification, we omit the index  $i$ .

**Theorem 2.8** Assume that  $f$  is an analytic function. For a  $2 \times 2$  matrix  $G$ , the matrix function  $f(G)$  can be computed explicitly as

$$f(G) = \frac{f(z_1) - f(z_2)}{z_1 - z_2} G + \frac{z_1 f(z_2) - z_2 f(z_1)}{z_1 - z_2} I, \tag{12}$$

where  $z_1$  and  $z_2$  are the two distinct eigenvalues of the matrix  $G$ . In case the matrix  $G$  has a double eigenvalue  $z_1$ , we get

$$f(G) = f'(z_1)G + (f(z_1) - f'(z_1)z_1) I. \tag{13}$$

**Proof** Let  $p(z)$  be the characteristic polynomial of the matrix  $G$  and assume for a moment that the equation  $p(z) = 0$  has two distinct roots  $z_1$  and  $z_2$ . The Cayley–Hamilton theorem states that  $p(G) = 0$ .

The function  $f$  can be rewritten in the form  $f(z) = q(z)p(z) + r(z)$  where  $q(z)$  is some quotient and  $r(z)$  is a remainder polynomial with  $0 \leq \deg r(z) < 2$ . From  $p(G) = 0$ , we obtain

$$f(G) = r(G) = d_1G + d_0I.$$

To complete the proof, we determine the coefficients  $d_1$  and  $d_0$ . From  $f(z_1) = r(z_1)$  and  $f(z_2) = r(z_2)$ , we obtain that

$$d_1 = \frac{f(z_1) - f(z_2)}{z_1 - z_2}, \quad d_0 = \frac{z_1f(z_2) - z_2f(z_1)}{z_1 - z_2}.$$

In case of a double eigenvalue  $z_1$ , we use the conditions  $f(z_1) = r(z_1)$  and  $f'(z_1) = r'(z_1)$ . As a consequence, we obtain that  $d_1 = f'(z_1)$  and  $d_0 = f(z_1) - f'(z_1)z_1$ . □

We remark that similar formulas can be found in the work of Bernstein and So [3] or Cheng and Yau [6]. To reduce the computational cost, we try to avoid complex arithmetic.

**Lemma 2.9** *Assume that the matrix  $G$  is of the form*

$$G = \begin{bmatrix} 0 & 1 \\ -\alpha\lambda - \delta & -\beta\lambda - \gamma \end{bmatrix}$$

and denote  $m = -\frac{1}{2}(\beta\lambda + \gamma)$ .

(i) *If  $(\beta\lambda + \gamma)^2 > 4(\alpha\lambda + \delta)$ , denoting  $n = \frac{1}{2}\sqrt{(\beta\lambda + \gamma)^2 - 4(\alpha\lambda + \delta)}$ , the exponential matrix  $e^{tG}$  can be computed explicitly as follows*

$$e^{tG} = \frac{e^{t(m+n)} - e^{t(m-n)}}{2n} \begin{bmatrix} -m - n & 1 \\ n^2 - m^2 & m - n \end{bmatrix} + e^{t(m+n)}I. \tag{14}$$

(ii) *If  $(\beta\lambda + \gamma)^2 = 4(\alpha\lambda + \delta)$ , we obtain that*

$$e^{tG} = e^{tm} \begin{bmatrix} 1 - tm & t \\ -tm^2 & tm + 1 \end{bmatrix}. \tag{15}$$

(iii) *If  $(\beta\lambda + \gamma)^2 < 4(\alpha\lambda + \delta)$ , denoting  $n = \frac{1}{2}\sqrt{4(\alpha\lambda + \delta) - (\beta\lambda + \gamma)^2}$ , we get that*

$$e^{tG} = \frac{e^{tm} \sin(tn)}{n} \begin{bmatrix} -m & 1 \\ -n^2 - m^2 & m \end{bmatrix} + e^{tm} \cos(tn)I. \tag{16}$$

**Proof** Let  $z_1$  and  $z_2$  be the two eigenvalues of the matrix  $tG$ . Thus  $z_1$  and  $z_2$  satisfy the characteristic equation  $z^2 + (\beta\lambda + \gamma)tz + (\alpha\lambda + \delta)t^2 = 0$ . By using formula (12), we obtain that

$$e^{tG} = \frac{e^{z_1} - e^{z_2}}{z_1 - z_2}tG + \frac{z_1e^{z_2} - z_2e^{z_1}}{z_1 - z_2}I. \tag{17}$$

The discriminant of the characteristic equation is

$$D = ((\beta\lambda + \gamma)^2 - 4(\alpha\lambda + \delta))t^2.$$

We consider three cases:

(i) If  $D > 0$  or  $(\beta\lambda + \gamma)^2 > 4(\alpha\lambda + \delta)$ , the two real roots of the characteristic equation are  $z_1 = tm + tn$  and  $z_2 = tm - tn$ , where  $m = -\frac{1}{2}(\beta\lambda + \gamma)$  and  $n = \frac{1}{2}\sqrt{(\beta\lambda + \gamma)^2 - 4(\alpha\lambda + \delta)}$ . From the definitions of the parameters  $m$  and  $n$ , we get that  $-\alpha\lambda - \delta = n^2 - m^2$  and  $-\beta\lambda - \gamma = 2m$ . We will simplify the two coefficients in formula (17):

$$\begin{aligned} \frac{e^{z_1} - e^{z_2}}{z_1 - z_2} &= \frac{e^{t(m+n)} - e^{t(m-n)}}{2tn}, \\ \frac{z_1 e^{z_2} - z_2 e^{z_1}}{z_1 - z_2} &= \frac{(m+n)e^{t(m-n)} - (m-n)e^{t(m+n)}}{2n} \\ &= e^{t(m+n)} - (m+n) \frac{e^{t(m+n)} - e^{t(m-n)}}{2n}. \end{aligned}$$

By substituting the two simplified coefficients into (17), we get that

$$\begin{aligned} e^{tG} &= \frac{e^{t(m+n)} - e^{t(m-n)}}{2n} (G - (m+n)I) + e^{t(m+n)} I \\ &= \frac{e^{t(m+n)} - e^{t(m-n)}}{2n} \left( \begin{bmatrix} 0 & 1 \\ n^2 - m^2 & 2m \end{bmatrix} - \begin{bmatrix} m+n & 0 \\ 0 & m+n \end{bmatrix} \right) + e^{t(m+n)} I \\ &= \frac{e^{t(m+n)} - e^{t(m-n)}}{2n} \begin{bmatrix} -m-n & 1 \\ n^2 - m^2 & m-n \end{bmatrix} + e^{t(m+n)} I. \end{aligned}$$

(ii) If  $D = 0$  or  $(\beta\lambda + \gamma)^2 = 4(\alpha\lambda + \delta)$ , the characteristic equation has only one root  $z_1 = tm$ , where  $m = -\frac{1}{2}(\beta\lambda + \gamma)$ . In this case, we have

$$\begin{aligned} e^{tG} &= e^{z_1}(tG) + e^{z_1}(1 - z_1)I \\ &= e^{tm} t \begin{bmatrix} 0 & 1 \\ -m^2 & 2m \end{bmatrix} + e^{tm}(1 - tm) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = e^{tm} \begin{bmatrix} 1 - tm & t \\ -tm^2 & tm + 1 \end{bmatrix}. \end{aligned}$$

(iii) If  $D < 0$  or  $(\beta\lambda + \gamma)^2 < 4(\alpha\lambda + \delta)$ , the characteristic equation has two conjugate complex roots  $z_1 = tm + itn$  and  $z_2 = tm - itn$ , where

$$m = -\frac{1}{2}(\beta\lambda + \gamma), \quad n = \frac{1}{2}\sqrt{4(\alpha\lambda + \delta) - (\beta\lambda + \gamma)^2}.$$

Since  $z_2 = \bar{z}_1$ , we infer that  $e^{z_2} = \bar{e}^{z_1}$  and  $z_2 e^{z_1} = \bar{z}_1 \bar{e}^{z_2} = \bar{z}_1 e^{z_2}$ . We analogously simplify the two coefficients in (17) as follows

$$\begin{aligned} \frac{e^{z_1} - e^{z_2}}{z_1 - z_2} &= \frac{e^{z_1} - \bar{e}^{z_1}}{z_1 - \bar{z}_1} = \frac{2\text{Im}(e^{z_1})}{2\text{Im}(z_1)} = \frac{e^{tm} \sin(tn)}{tn}, \\ \frac{z_1 e^{z_2} - z_2 e^{z_1}}{z_1 - z_2} &= \frac{\text{Im}(z_1 \bar{e}^{z_1})}{\text{Im}(z_1)} = e^{tm} \cos(tn) - m \frac{e^{tm} \sin(tn)}{n}. \end{aligned}$$

By substituting the two simplified coefficients into (17) and noting that  $-\alpha\lambda - \delta = -n^2 - m^2$  and  $-\beta\lambda - \gamma = 2m$ , we get that

$$\begin{aligned} e^{tG} &= \frac{e^{tm} \sin(tn)}{n} (G - mI) + e^{tm} \cos(tn) I \\ &= \frac{e^{tm} \sin(tn)}{n} \begin{bmatrix} -m & 1 \\ -n^2 - m^2 & m \end{bmatrix} + e^{tm} \cos(tn) I. \end{aligned}$$

This concludes the proof. □

**Remark 2.10** Formula (16) is useful in computations. For example, for beam equations with typical physical parameters proposed by Ito and Morris in [21, Section 3], the matrix  $G$  has usually two complex conjugate eigenvalues.

### 3 Exponential Integrators

#### 3.1 Exponential Integrators for Semilinear Problems

We consider semilinear differential equations of the form

$$\dot{y}(t) = Ay(t) + F(y(t)). \tag{18}$$

The solution of this equation at time  $t_{n+1} = t_n + \tau_n$ ,  $t_0 = 0, n \in \mathbb{N}$  is given by the variation-of-constants formula

$$y(t_{n+1}) = e^{\tau_n A} y(t_n) + \int_0^{\tau_n} e^{(t_n - \tau)A} F(y(t_n + \tau)) d\tau.$$

For the numerical solution of (18), we recall a general class of one-step exponential integrators from [17–19]

$$y_{n+1} = e^{\tau_n A} y_n + \tau_n \sum_{i=1}^s b_i(\tau_n A) F_{ni},$$

$$Y_{ni} = e^{c_j \tau_n A} y_n + \tau_n \sum_{j=1}^{i-1} a_{ij}(\tau_n A) F_{nj},$$

$$F_{nj} = F(Y_{nj}).$$

The coefficients are as usually collected in a Butcher tableau

$c_1$				
$c_2$	$a_{21}(\tau_n A)$			
$\vdots$	$\vdots$	$\ddots$		
$c_s$	$a_{s1}(\tau_n A)$	$\dots$	$a_{s,s-1}(\tau_n A)$	
	$b_1(\tau_n A)$	$\dots$	$b_{s-1}(\tau_n A)$	$b_s(\tau_n A)$

The method coefficients  $a_{ij}$  and  $b_i$  are constructed from a family of functions  $\varphi_k$  evaluated at the matrix  $(\tau_n A)$ . We next recall this family  $\varphi_k$ , which was introduced before in [17–19].

**Corollary 3.1** Consider the entire functions

$$\varphi_k(z) = \int_0^1 e^{(1-\theta)z} \frac{\theta^{k-1}}{(k-1)!} d\theta, \quad k \geq 1, \quad \varphi_0(z) = e^z.$$

These functions satisfy the following properties:

- (i)  $\varphi_k(0) = \frac{1}{k!}$ ;
- (ii) they satisfy the recurrence relation

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi(0)}{z};$$

(iii) the Taylor expansion of the function  $\varphi_k$  is

$$\varphi_k(z) = \sum_{n=0}^{\infty} \frac{z^n}{(n+k)!}.$$

To simplify notation, we denote

$$\varphi_{k,j} = \varphi_k(c_j \tau_n A), \quad \varphi_k = \varphi_k(\tau_n A).$$

Next, we recall five exponential integrators that will be used in our numerical examples.

**Example 3.2** For  $s = 1$ , the exponential Euler method has the form

$$y_{n+1} = e^{\tau_n A} y_n + \tau_n \varphi_1(\tau_n A) F(y_n). \tag{19}$$

$$(EI-E1) \quad \begin{array}{c|c} 0 & \\ \hline c_2 & \varphi_1 \end{array}$$

We denote this method by EI-E1.

**Example 3.3** For  $s = 2$ , we recall a second-order method proposed by Strehmel and Weiner in [31, Section 4.5.3]:

$$(EI-SW21) \quad \begin{array}{c|cc} 0 & & \\ \hline c_2 & c_2 \varphi_{1,2} & \\ \hline & \varphi_1 - \frac{1}{c_2} \varphi_2 & \frac{1}{c_2} \varphi_2 \end{array}$$

A simplified version, where only  $\varphi_1$  is used, is also proposed by Strehmel and Weiner

$$(EI-SW22) \quad \begin{array}{c|cc} 0 & & \\ \hline c_2 & c_2 \varphi_{1,2} & \\ \hline & \left(1 - \frac{1}{2c_2}\right) \varphi_1 & \frac{1}{2c_2} \varphi_1 \end{array}$$

**Example 3.4** For  $s = 4$ , we recall two schemes. The first one, proposed by Krogstad [22], is given by

$$(EI-K4) \quad \begin{array}{c|ccc} 0 & & & \\ \hline \frac{1}{2} & \frac{1}{2} \varphi_{1,2} & & \\ \frac{1}{2} & \frac{1}{2} \varphi_{1,3} - \varphi_{2,3} & \varphi_{2,3} & \\ \hline 1 & \varphi_{1,4} - 2\varphi_{2,4} & 0 & 2\varphi_{2,4} \\ \hline & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 2\varphi_2 - 4\varphi_3 & 2\varphi_2 - 4\varphi_3 \quad -\varphi_2 + 4\varphi_3 \end{array}$$

The second method is suggested by Strehmel and Weiner (see [31, Example 4.5.5])

$$(EI-SW4) \quad \begin{array}{c|ccc} 0 & & & \\ \hline \frac{1}{2} & \frac{1}{2} \varphi_{1,2} & & \\ \frac{1}{2} & \frac{1}{2} \varphi_{1,3} - \frac{1}{2} \varphi_{2,3} & \frac{1}{2} \varphi_{2,3} & \\ \hline 1 & \varphi_{1,4} - 2\varphi_{2,4} & -2\varphi_{2,4} & 4\varphi_{2,4} \\ \hline & \varphi_1 - 3\varphi_2 + 4\varphi_3 & 0 & 4\varphi_2 - 8\varphi_3 \quad -\varphi_2 + 4\varphi_3 \end{array}$$

### 3.2 Computing Matrix Functions of $tA$

To apply these exponential integrators to semilinear problems, we next introduce an approach to explicitly compute the matrix functions  $\varphi_k(tA)$ . We first present an analogous version of Theorem 2.5.

**Theorem 3.5** *Assume that there exist an orthogonal matrix  $Q$  and a diagonal matrix  $D = \text{diag}\{\lambda_1, \dots, \lambda_n\}$  such that  $S = QDQ'$ . Then, for  $t > 0$  and  $k \geq 1$ , the functions  $\varphi_k(tA)$  are computed as follows*

$$\varphi_k(tA) = \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} P \begin{bmatrix} \varphi_k(tG_1) & 0 & \dots & 0 \\ 0 & \varphi_k(tG_2) & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \varphi_k(tG_n) \end{bmatrix} P' \begin{bmatrix} Q' & 0 \\ 0 & Q' \end{bmatrix}, \tag{20}$$

where  $P \in \mathbb{R}^{2n \times 2n}$  is given in (10) and  $G_i = \begin{bmatrix} 0 & 1 \\ -\alpha\lambda_i - \delta & -\beta\lambda_i - \gamma \end{bmatrix}$ .

The matrix functions  $\varphi_k(tG_i)$  are computed explicitly. The actual formula depends on the sign of  $(\beta\lambda_i + \gamma)^2 - 4(\alpha\lambda_i + \delta)$ . Next, we will present two lemmas concerning these functions.

**Lemma 3.6** *Assume that the matrix  $G$  is of the form*

$$G = \begin{bmatrix} 0 & 1 \\ -\alpha\lambda - \delta & -\beta\lambda - \gamma \end{bmatrix}$$

and denote  $m = -\frac{1}{2}(\beta\lambda + \gamma)$ .

(i) *If  $(\beta\lambda + \gamma)^2 > 4(\alpha\lambda + \delta)$ , denoting  $n = \frac{1}{2}\sqrt{(\beta\lambda + \gamma)^2 - 4(\alpha\lambda + \delta)}$ , the matrix functions  $\varphi_k(tG)$  can be computed explicitly as follows*

$$\varphi_k(tG) = \frac{\varphi_k^+ - \varphi_k^-}{2n} \begin{bmatrix} -m - n & 1 \\ n^2 - m^2 & m - n \end{bmatrix} + \varphi_k^+ I, \tag{21}$$

where  $\varphi_k^+ = \varphi_k(t(m + n))$  and  $\varphi_k^- = \varphi_k(t(m - n))$ .

(ii) *If  $(\beta\lambda + \gamma)^2 = 4(\alpha\lambda + \delta)$ , we obtain that*

$$\varphi_k(tG) = \varphi'_k(tm) \begin{bmatrix} -tm & t \\ -tm^2 & tm \end{bmatrix} + \varphi_k(tm)I, \tag{22}$$

where the derivative  $\varphi'_k(z)$  can be computed recursively

$$\begin{aligned} \varphi_0(z) &= \varphi'_0(z) = e^z, \\ \varphi'_{k+1}(z) &= \frac{\varphi'_k(z) - \varphi_{k+1}(z)}{z}, \quad \varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}. \end{aligned}$$

**Proof** By applying Theorem 2.8, the proof follows the lines of the first two points in the proof of Lemma 2.9. □

The last lemma concentrates on the case of two complex eigenvalues of  $G_i$ . Again the idea is to compute the matrix functions without explicitly using complex numbers. It is inspired by formula (16) above.

**Lemma 3.7** *In the case  $(\beta\lambda + \gamma)^2 < 4(\alpha\lambda + \delta)$ , the matrix  $G$  has two conjugate complex eigenvalues  $z_1$  and  $z_2$  with  $z_1 = tm + itn$ , where*

$$m = -\frac{1}{2}(\beta\lambda + \gamma), \quad n = \frac{1}{2}\sqrt{4(\alpha\lambda + \delta) - (\beta\lambda + \gamma)^2}.$$

The matrix  $\varphi_k(tG)$  can be explicitly computed as follows

$$\varphi_k(tG) = \frac{i_k}{n} \begin{bmatrix} -m & 1 \\ -n^2 - m^2 & m \end{bmatrix} + r_k I. \tag{23}$$

Here, the two coefficients  $i_k$  and  $r_k$  depend on  $(t, m, n)$  and can be computed recursively as follows

$$i_0 = e^{tm} \sin(tn), \quad r_0 = e^{tm} \cos(tn), \tag{24a}$$

$$i_k = \frac{1}{t(m^2 + n^2)} \left( mi_{k-1} - n \left( r_{k-1} - \frac{1}{(k-1)!} \right) \right), \tag{24b}$$

$$r_k = \frac{1}{t(m^2 + n^2)} \left( ni_{k-1} + m \left( r_{k-1} - \frac{1}{(k-1)!} \right) \right). \tag{24c}$$

**Proof** By using formula (12), we obtain that

$$\varphi_k(tG) = \frac{\varphi_k(z_1) - \varphi_k(z_2)}{z_1 - z_2} tG + \frac{z_1\varphi_k(z_2) - z_2\varphi_k(z_1)}{z_1 - z_2} I.$$

First, we note that  $\varphi_k(z_2) = \overline{\varphi_k(z_1)}$  because  $\varphi_k$  has real coefficients. Thus we can simplify as follows

$$\frac{\varphi_k(z_1) - \varphi_k(z_2)}{z_1 - z_2} = \frac{\text{Im}(\varphi_k(z_1))}{\text{Im}z_1}, \quad \frac{z_1\varphi_k(z_2) - z_2\varphi_k(z_1)}{z_1 - z_2} = \frac{\text{Im}(z_1\overline{\varphi_k(z_1)})}{\text{Im}z_1}.$$

Next, we rewrite the recursion as follows

$$\varphi_{k+1}(z_1) = \frac{\varphi_k(z_1) - \varphi_k(0)}{z_1} = \frac{(\varphi_k(z_1) - \varphi_k(0))\overline{z_1}}{|z_1|^2}.$$

To simplify notation, we denote  $i_k = \text{Im}(\varphi_k(z_1))$  and  $r_k = \text{Re}(\varphi_k(z_1))$ . Thus we obtain that

$$\begin{aligned} i_{k+1} &= \text{Im}(\varphi_{k+1}(z_1)) = \frac{1}{|z_1|^2} \text{Im}\left((\varphi_k(z_1) - \varphi_k(0))\overline{z_1}\right) \\ &= \frac{1}{t^2(m^2 + n^2)} \left( \text{Im}(\varphi_k(z_1) - \varphi_k(0))\text{Re}(\overline{z_1}) + \text{Re}(\varphi_k(z_1) - \varphi_k(0))\text{Im}(\overline{z_1}) \right) \\ &= \frac{1}{t(m^2 + n^2)} \left( mi_k - n \left( r_k - \frac{1}{k!} \right) \right), \\ r_{k+1} &= \text{Re}(\varphi_{k+1}(z_1)) = \frac{1}{|z_1|^2} \text{Re}\left((\varphi_k(z_1) - \varphi_k(0))\overline{z_1}\right) \\ &= \frac{1}{t^2(m^2 + n^2)} \left( \text{Re}(\varphi_k(z_1) - \varphi_k(0))\text{Re}(\overline{z_1}) - \text{Im}(\varphi_k(z_1) - \varphi_k(0))\text{Im}(\overline{z_1}) \right) \\ &= \frac{1}{t(m^2 + n^2)} \left( ni_k + m \left( r_k - \frac{1}{k!} \right) \right). \end{aligned}$$

Besides, we also get that  $\text{Im}(z_1\overline{\varphi_k(z_1)}) = tnr_k - tmi_k$ . This finally yields that

$$\varphi_k(tG) = \frac{i_k}{n} G + \frac{nr_k - mi_k}{n} I = \frac{i_k}{n} (G - mI) + r_k I$$

$$= \frac{i_k}{n} \begin{bmatrix} -m & 1 \\ -n^2 - m^2 & m \end{bmatrix} + r_k I,$$

which completes the proof. □

### 3.3 Summary of the Integration Procedure

The above described procedure can be summarized in two main parts. The **Preparation** part, which is done once at the beginning, consists of three steps:

- P1: Discretize the operator  $(-\Delta)$  or  $\partial_{xxxx}^4$  as a square symmetric matrix  $S$  (e.g., by finite differences, see also Sect. 4).
- P2: Find an orthogonal matrix  $Q$  and a diagonal matrix  $D = \text{diag}\{\lambda_1, \dots, \lambda_n\}$  such that  $S = QDQ'$ . The matrix  $D$  is stored as a vector.
- P3: Create a column vector which stores the positions of all non-zero entries of the permutation matrix  $P$  by using formula (10).

The **Main** part is used to compute the action of a matrix functions. This is required in the time stepping. Computing this action consists of two steps:

- M1: Compute the matrix functions  $\varphi_k(tG_i)$  by using formulas (14), (15), or (16) for  $\varphi_0(tG_i) = e^{tG_i}$ ; formulas (21), (22), or (23) for  $\varphi_k(tG_i)$  with  $k \geq 1$ .
- M2: Compute the action of the matrix functions  $\varphi_k(tA)$  using formula (11) for  $k = 0$  and formula (20) for  $k \geq 1$ .

## 4 Numerical Examples

### 4.1 Semilinear Wave Equations

We consider a 1D semilinear wave equation on  $\Omega = (0, \ell)$

$$\partial_{tt}^2 u - \alpha \partial_{xx}^2 u - \beta \partial_{xxt}^3 u + \delta u + \gamma \partial_t u = g(u) + h(\partial_t u), \quad 0 < x < \ell, \quad t \in (0, T], \tag{25a}$$

$$u(t, 0) = 0, \quad u(t, \ell) = 0, \tag{25b}$$

$$u(0, x) = p(x), \quad u_t(0, x) = q(x). \tag{25c}$$

We consider the product space  $X = H_0^1(\Omega) \times L^2(\Omega)$  and rewrite (25) in abstract form

$$\dot{y}(t) = \mathcal{A}y(t) + \mathcal{F}(y(t)), \quad y(0) = (p, q)',$$

where  $\mathcal{A} : D(\mathcal{A}) \rightarrow X$  is the operator defined by

$$\begin{aligned} \mathcal{A} \begin{bmatrix} u \\ w \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -\alpha(-\partial_{xx}^2) - \delta I & -\beta(-\partial_{xx}^2) - \gamma I \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} \\ &= \begin{bmatrix} w \\ -\alpha(-\partial_{xx}^2 u) - \beta(-\partial_{xx}^2 w) - \delta u - \gamma w \end{bmatrix}, \\ D(\mathcal{A}) &= (H^2(\Omega) \cap H_0^1(\Omega))^2, \\ \mathcal{F} \begin{pmatrix} u \\ w \end{pmatrix} &= \begin{bmatrix} 0 \\ g(u) + h(w) \end{bmatrix}. \end{aligned}$$

Define the closed self-adjoint positive operator  $\mathcal{A}_0$  on  $L^2(0, \ell)$  by

$$\begin{aligned} \mathcal{A}_0\phi &= -\partial_{xx}^2\phi, \\ D(\mathcal{A}_0) &= \{\phi \in H^2(0, \ell) \mid \phi(0) = \phi(1) = 0\}. \end{aligned}$$

We use symmetric finite differences to discretize the operator  $\mathcal{A}_0$ . For this, the space interval  $(0, \ell)$  is divided equidistantly by the nodes  $x_i = i \Delta x$ ,  $i \in \{0, \dots, N + 1\}$ , where  $N$  is a given integer and  $\Delta x = \frac{\ell}{N+1}$ . Then, the discrete operator  $\mathcal{A}_0$  is given by the matrix  $S_w \in \mathbb{R}^{N \times N}$  defined by

$$S_w = \frac{1}{\Delta x^2} \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix}. \tag{26}$$

In the four examples below, we consider the space interval  $\Omega = (0, 1)$  with  $N = 200$ .

**Example 4.1** Consider Eq. (25) with  $\alpha = \pi^2$ ,  $\beta = 10^{-2}$ ,  $\delta = 0$ ,  $\gamma = 10^{-2}$ . The nonlinear source term is  $g(u) = \sin u$ . This is a perturbed sine-Gordon equation of the form (2). The initial conditions are  $p(x) = 5 \sin(2\pi x)$  and  $q(x) = 0$ . We use four different schemes, namely EI-E1, EI-SW21 (with  $c_2 = 0.75$ ), EI-SW4, and EI-K4 to compute the solution at time  $T = 6$  with  $M \in \{5, 10, 20, \dots, 5 \cdot 2^{11}\}$  time steps. The reference solution  $y_{ref} = (u_{ref}, (\partial_t u)_{ref})'$  plotted in Fig. 1a, b is computed by using EI-SW4 with  $M = 200000$  time steps. The discrete  $\ell_2$  error between the approximate solution obtained with the mentioned integrators at the final time  $y(T) = (u(T), \partial_t u(T))'$  and the reference solution  $y_{ref}$  is computed by the formula

$$\|y(T) - y_{ref}\|_{\ell_2}^2 = \Delta x \sum_{i=1}^N |y_i(T) - y_{ref,i}|^2. \tag{27}$$

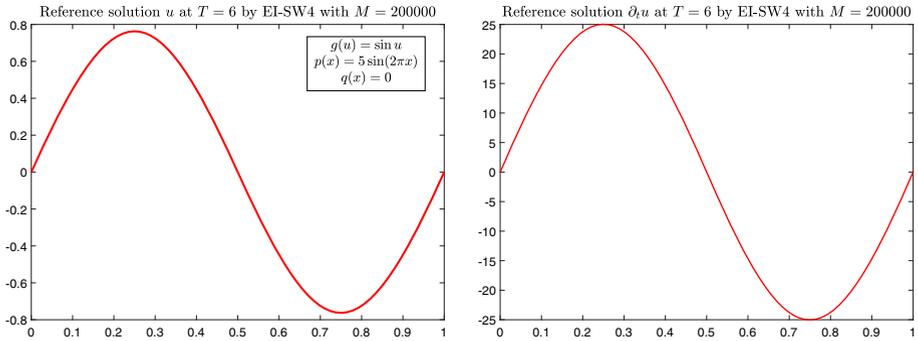
These errors are plotted in Fig. 1c. The expected convergence rate is observed for each scheme. Even when we use a rather coarse time mesh with  $M = 5$  and  $\Delta t = 1.2$ , the error is quite small (approximate  $10^{-1}$ ).

**Example 4.2** Consider Eq. (25) with  $\alpha = 100$ ,  $\beta = 10^{-2}$ ,  $\delta = 0$ ,  $\gamma = 10^{-3}$ . The nonlinear source term is  $g(u) = u|u|$ . The initial conditions are

$$p(x) = \begin{cases} 2x & \text{if } x \leq \frac{1}{2}, \\ -2x + 2 & \text{if } x > \frac{1}{2}, \end{cases} \quad q(x) = \pi^2 \sin(\pi x).$$

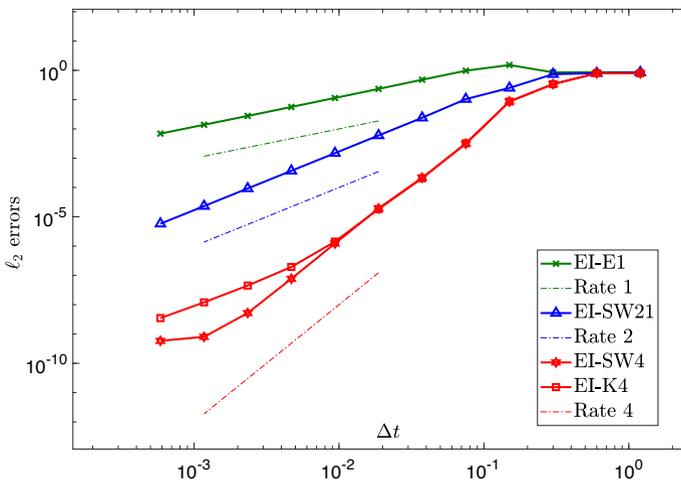
We use five different schemes, namely EI-E1, EI-SW21, EI-SW22 (both schemes with  $c_2 = 0.2$ ), EI-SW4, and EI-K4 to compute the solution at time  $T = 15$  with  $M \in \{20, 40, 80, \dots, 20 \cdot 2^{12}\}$  time steps. The reference solution plotted in Fig. 2a, b is computed by using EI-K4 with  $M = 200000$ . The errors are plotted in Fig. 2c.

**Example 4.3** Consider Eq. (25) with  $\alpha = 15$ ,  $\beta = 10^{-3}$ ,  $\delta = 1$ ,  $\gamma = 10^{-6}$ . The nonlinear source term is  $g(u) = u^3$ . The initial conditions are  $p(x) = 10 \sin(3\pi x)$ ,  $q(x) = -10 \cos(3\pi x)$ . We use all five schemes, namely EI-E1, EI-SW22 (with  $c_2 = 0.9$ ), EI-K4,



(a) Reference solution  $u$  at  $T = 6$ .

(b) Reference solution  $\partial_t u$  at  $T = 6$ .



(c) Convergence rates.

**Fig. 1** Example 4.1

and EI-SW4 to compute the solution at time  $T = 30$  with  $M \in \{20, 40, 80, \dots, 20 \cdot 2^{11}\}$  time steps. We plot the reference solution computed with EI-SW4 and  $M = 300000$  in Fig. 3a, b. Again the expected convergence rates are observed for each scheme and plotted in Fig. 3c. An order reduction occurs for EI-K4 (reduction to order 2) while EI-SW4 still preserves its convergence rate.

**Example 4.4** This example concerns discontinuous initial conditions

$$p(x) = \begin{cases} -1 & \text{if } x \leq \frac{1}{2}, \\ 5 & \text{if } x > \frac{1}{2}, \end{cases} \quad q(x) = 0.$$

The other parameters are  $\alpha = 5$ ,  $\beta = 10^{-3}$ ,  $\delta = 1$ ,  $\gamma = 10^{-4}$ . The nonlinear term is  $g(u) = |u|$ . The approximate solutions at  $T = 3$  are computed by using five exponential integrators, namely EI-E1, EI-SW21, EI-SW22 (both schemes with  $c_2 = 0.5$ ), EI-SW4,

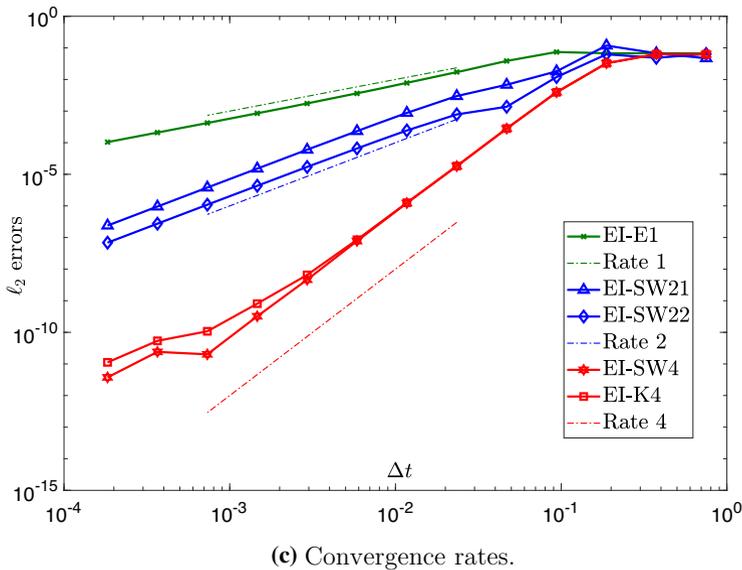
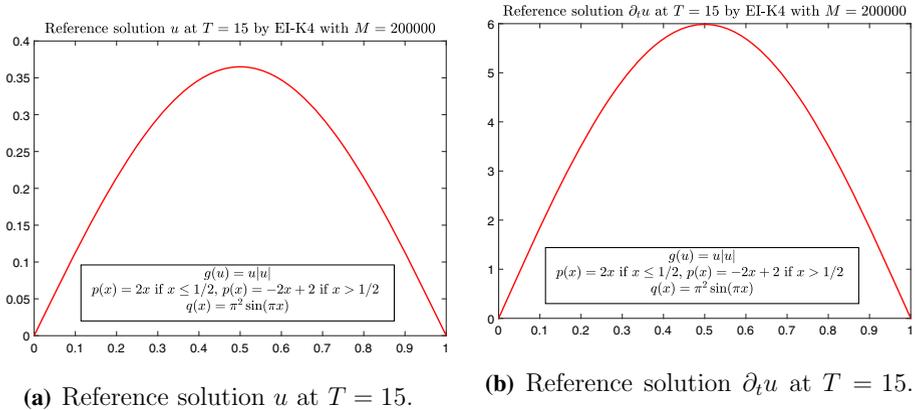


Fig. 2 Example 4.2

and EI-K4 with  $M \in \{20, 40, \dots, 20 \cdot 2^{13}\}$  time steps. The reference solution computed with  $M = 300000$  time steps by using EI-SW4 is plotted in Fig. 4a, b. The errors are plotted in Fig. 4c. We observe an order reduction to order 2 for the two fourth-order exponential integrators EI-SW4 and EI-K4 while the other integrators preserve their convergence rate.

**Example 4.5** The last example concerns two nonlinear terms, namely  $g(u) = -u|u|^3$ ,  $h(w) = -w|w|$ . The other parameters are  $\alpha = 50$ ,  $\beta = 10^{-6}$ ,  $\delta = 10$ ,  $\gamma = 10^{-3}$ . The initial conditions are  $p(x) = 20 \sin(4\pi x)$  and  $q(x) = -25 \cos(3\pi x)$ . The approximate solution at  $T = 1$  are computed by using five exponential integrators, namely EI-E1, EI-SW21, EI-SW22 (both schemes with  $c_2 = 0.85$ ), EI-SW4, and EI-K4 with  $M \in \{160, 320, \dots, 160 \cdot 2^{10}\}$  time steps. The reference solution is computed by EI-SW4

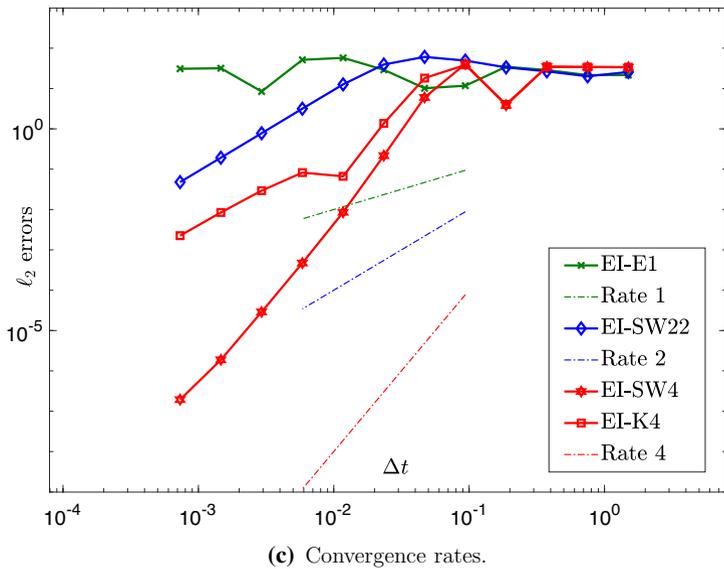
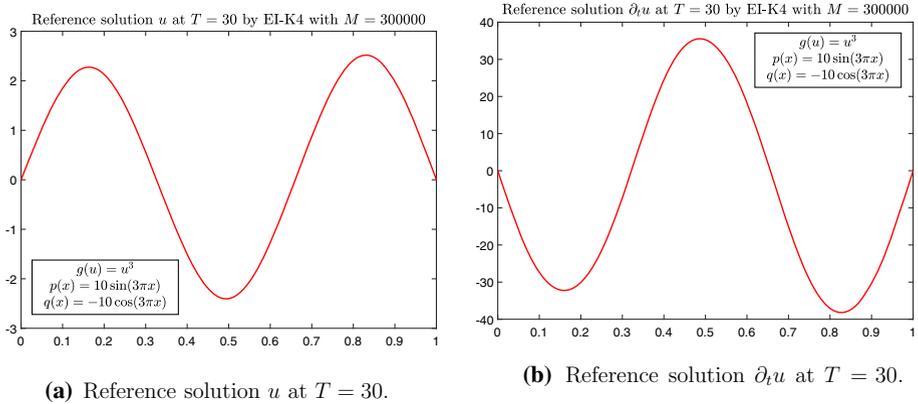


Fig. 3 Example 4.3

with  $M = 800000$  and plotted in Fig. 5a, b. The convergence rates are plotted in Fig. 5c. The two fourth-order exponential integrators EI-SW4 and EI-K4 show order reductions. While EI-SW4 works still well with an order reduction to order 3; EI-K4 on the other hand works badly.

### 4.2 Railway Track Model

Assume that a track beam is made of Kelvin-Voigt material. The resulting railway track model is a semilinear PDE on  $\Omega = (0, L)$ :

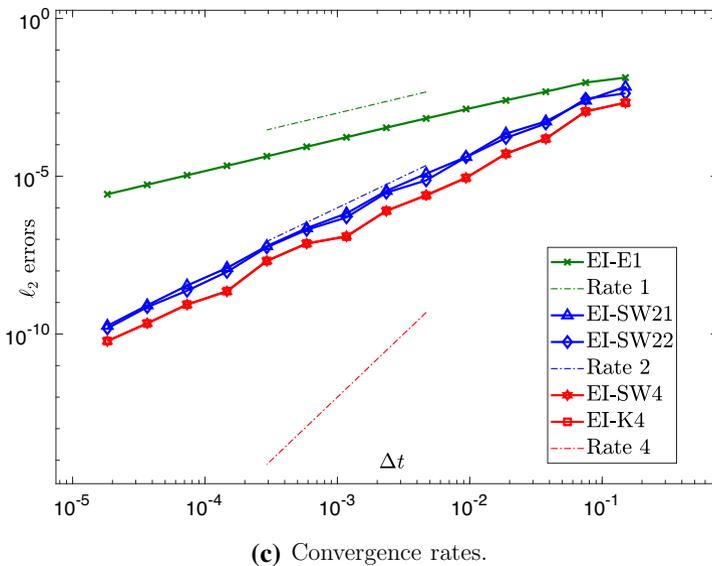
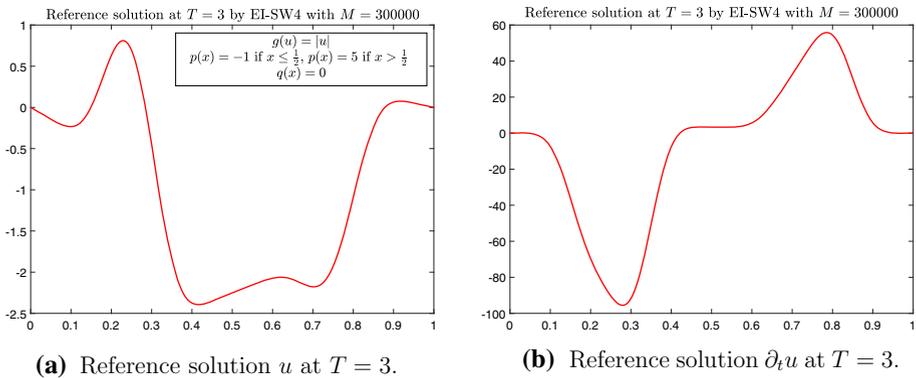


Fig. 4 Example 4.4

$$\partial_{tt}^2 u + \partial_{xx}^2 (\alpha \partial_{xx}^2 u + \beta \partial_{xx}^2 (\partial_t u)) + \gamma \partial_t u + \delta u = -lu^3, \quad x \in \Omega, t \in (0, T], \quad (28a)$$

$$u(t, 0) = u(t, L) = 0, \quad (28b)$$

$$\alpha \partial_{xx}^2 u(t, 0) + \beta \partial_{xx}^2 (\partial_t u)(t, 0) = \alpha \partial_{xx}^2 u(t, L) + \beta \partial_{xx}^2 (\partial_t u)(t, L) = 0, \quad (28c)$$

$$u(0, x) = p(x), \quad \partial_t u(0, x) = q(x). \quad (28d)$$

Denote the closed self-adjoint positive operator  $\mathcal{A}_0$  on  $L^2(0, L)$  as

$$\begin{aligned} \mathcal{A}_0 \phi &:= \partial_{xxxx}^4 \phi, \\ D(\mathcal{A}_0) &:= \{ \phi \in H^4(\Omega) \mid \phi(0) = \phi(L) = 0, \phi''(0) = \phi''(L) = 0 \}. \end{aligned}$$

Concerning the analysis of the linear operators, we refer to the literature [2, 26]. We use finite differences to discretize the operator  $\mathcal{A}_0$  with an equidistant space mesh  $x_i = i\Delta x$ ,  $i \in \{0, \dots, N + 1\}$ , where  $N$  is a given integer and  $\Delta x = \frac{L}{N+1}$ . Then the discrete operator  $\mathcal{A}_0$

is given by the matrix  $S_b \in \mathbb{R}^{N \times N}$ , defined as below

$$S_b = \frac{1}{\Delta x^4} \begin{bmatrix} 5 & -4 & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ -4 & 6 & -4 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & \dots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & \dots & \ddots & 1 & -4 & 6 & -4 & 1 \\ 0 & \dots & \dots & \dots & \dots & \dots & \ddots & 0 & 1 & -4 & 6 & -4 \\ 0 & \dots & 0 & 1 & -4 & 5 \end{bmatrix} \quad (29)$$

**Example 4.6** Consider Eq. (28) with  $\alpha = 15$ ,  $\beta = 3 \cdot 10^{-6}$ ,  $\delta = 10$ ,  $\gamma = 3 \cdot 10^{-4}$ . The nonlinear term is  $g(u) = -5u^3$ . The initial conditions are

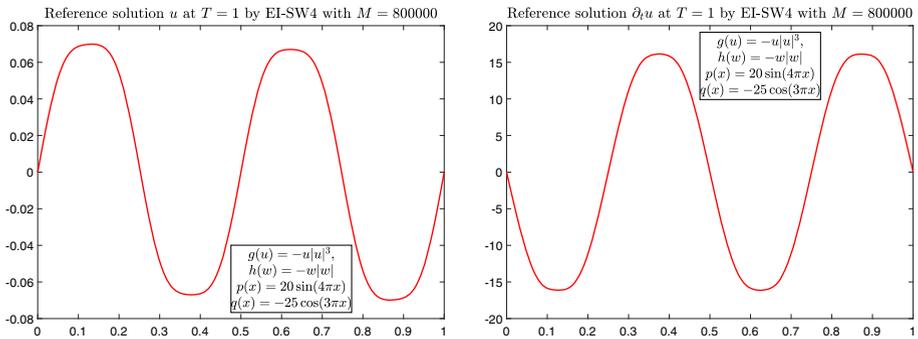
$$p(x) = 5e^{-100\left(x-\frac{2}{3}\right)^2}, \quad q(x) = 0.$$

For our numerical solution, the space interval  $\Omega = (0, 1)$  is divided into 300 equidistant subintervals. We compute approximate solutions at  $T = 5$  with four exponential integrators EI-E1, EI-SW22 ( $c_2 = 0.9$ ), EI-SW4, and EI-K4 with  $M \in \{160, 320, \dots, 160 \cdot 2^{10}\}$  time steps. We compare these numerical results with the reference solution evaluated by EI-K4 with  $M = 600000$  time steps. The reference solution is plotted in Fig. 6a, b. Notice that the magnitude of the velocity  $\partial_t u$  is extremely big. The errors are plotted in Fig. 6c. The four exponential integrators preserve their convergence rate. Since the matrix  $S_b$  is stiffer than the one  $S_w$ , the computation for beam equations is more expensive than for wave equations. However, we note that solving beam equations with exponential integrators is a good option. Some comparisons in the next section will elucidate this point.

### 4.3 Comparisons with Standard Integrators

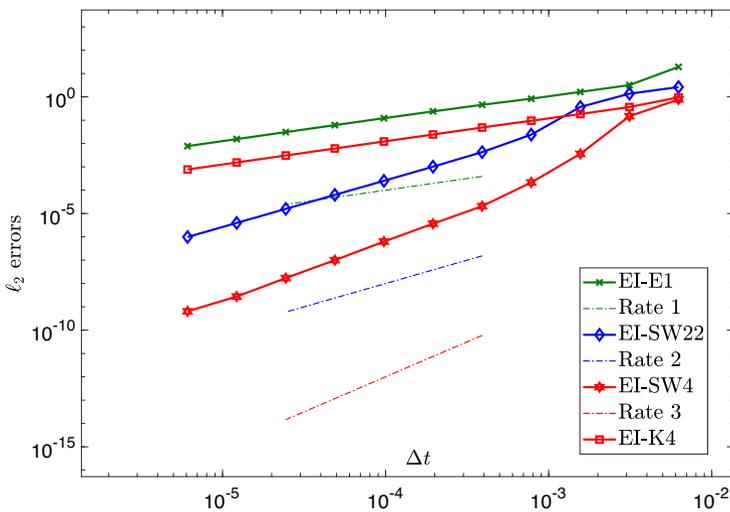
Some comparisons between our approach and standard integrators will be presented to clarify the efficiency. In particular, we consider the explicit Runge-Kutta method ode45 from MATLAB. Note that ode45 needs sufficiently small time steps to guarantee stability. The CFL condition depends on the set up of our model. For example, the larger the parameter  $\alpha$  we choose in the example, the smaller the time step  $\Delta t$  has to be chosen. Besides, the maximum step size depends on the type of equation, i.e. the beam equation is stiffer than the wave equation. In particular, the relation between  $\Delta t$  and  $\Delta x$  is of the form:  $\Delta t \sim (\Delta x)^2$  for beam equations, and  $\Delta t \sim \Delta x$  for wave equations.

Stiff problems are often solved with implicit schemes. Therefore, we will make another comparison with a class of implicit Runge-Kutta methods, namely the Radau IIA methods (see [14, Section IV-5]). Though these methods do not require any CFL condition to guarantee



(a) Reference solution  $u$  at  $T = 1$ .

(b) Reference solution  $\partial_t u$  at  $T = 1$ .



(c) Convergence rates.

Fig. 5 Example 4.5

stability, their computational cost is high since they require the solution of linear systems with large matrices. Below, we illustrate by some examples that both explicit and implicit Runge-Kutta methods are more expensive than our exponential integrators in the present context.

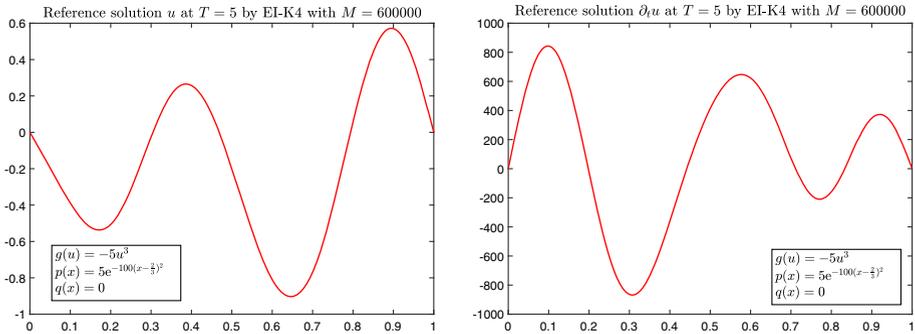
**Example 4.7** Exact matrix exponential versus ode45 and Radau scheme for a linear example.

Consider a linearized version of Eq. (25) with  $\alpha = 100$ ,  $\beta = 10^{-2}$ ,  $\delta = 10^{-2}$ ,  $\gamma = 10^{-6}$ . The initial conditions are  $p(x) = 5 \sin(2\pi x)$  and  $q(x) = 0$ . We consider the space interval  $\Omega = (0, 1)$  with  $N = 200$  (number of grid points).

We compute the solution at time  $T = 10$  by using different methods. For any exponential integrator, the solution is obtained immediately by the formula

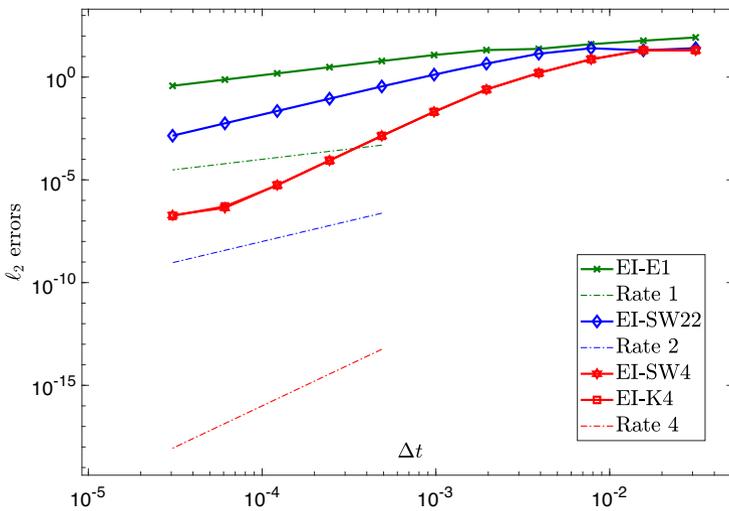
$$y(T) = \begin{bmatrix} u(T) \\ w(T) \end{bmatrix} = e^{TA} \begin{bmatrix} p \\ q \end{bmatrix}, \quad A = \begin{bmatrix} 0 & I \\ -\alpha S_w - \beta I & -\beta S_w - \gamma I \end{bmatrix},$$

where  $S_w$  was defined in (26). The computational time for using this approach is 0.036s.



(a) Reference solution  $u$  at  $T = 5$ .

(b) Reference solution  $\partial_t u$  at  $T = 5$ .



(c) Convergence rates.

Fig. 6 Example 4.6

For comparisons, we use an explicit Runge-Kutta method, namely `ode45` from MATLAB to obtain the solution at final time  $T = 10$  with various tolerances. Due to the stability reason, `ode45` needs a huge number of time steps. The minimum number of time steps  $M$  which `ode45` needs to achieve the corresponding accuracy are presented in the second column of Table 1. The relating computational time is reported in the third column. As implicit method, we use the Radau IIA scheme. The minimum number of time steps which `Radau`<sup>1</sup> needs to obtain the corresponding accuracy is shown in the fourth column of Table 1. The relating computational time is reported in the fifth column. The implicit scheme is more efficient than the explicit one for solving this linear equation. Obviously to tackle the linear case, both `ode45` and `Radau` are expensive choices.

**Example 4.8** EI-K4 versus `ode45` and `Radau` for a semilinear wave equation. Consider the Eq. (28) with  $\alpha = 100$ ,  $\beta = \gamma = 10^{-3}$ ,  $\delta = 10$ . The nonlinear source term is  $g(u) = u^2$ .

<sup>1</sup> <http://www.unige.ch/~hairer/prog/MatlabStiff.7z>.

**Table 1** Number of time steps  $M$  that ode45 and Radau need to compute the solution at  $T = 10$  and the corresponding computational time

Tolerance	ode45		Radau	
	$M$	Time (s)	$M$	Time (s)
$10^{-3}$	55621	18.17	32	1.33
$10^{-6}$	55865	17.63	58	2.09
$10^{-10}$	888970	26.76	99	4.03
$10^{-12}$	224733	68.30	284	5.25

On the other hand, an exponential integrator needs just 0.036 s to get the exact solution

**Table 2** Comparison among EI-K4, ode45, and Radau for a semilinear wave equation

Tolerance	EI-K4		ode45		Radau	
	$M$	Time (s)	$M$	Time (s)	$M$	Time (s)
$10^{-4}$	20	0.36	123149	41.05	1170	213.01
$10^{-6}$	640	6.56	128693	42.75	1734	319.57
$10^{-8}$	2560	14.26	167861	54.85	2128	273.88
$10^{-10}$	20480	99.65	378641	119.29	3917	827.20

The number of time steps for the exponential integrator is empirically chosen to reach the prescribed accuracy

The initial conditions are

$$p(x) = \begin{cases} 2x & \text{if } x \leq \frac{1}{2}, \\ -2x + 2 & \text{if } x > \frac{1}{2}, \end{cases} \quad q(x) = \pi^2 \sin(\pi x).$$

We compute the solution at the final time  $T = 15$  by EI-K4, ode45, and Radau. To solve semilinear equations, three integrators need a sufficient number of substeps to attain the solution at final time. Corresponding to the desired accuracy, the number of steps as well as the required computational time of two schemes are reported in the Table 2. For all accuracies, EI-K4 is more efficient than the two other schemes. Even though Radau needs less number of time steps to achieve the desired accuracy, the computational cost is really high since it needs to solve a linear system involving a large matrix at each time step.

**Example 4.9** EI-K4 versus ode45 and Radau for a semilinear beam equation. We repeat Example 4.6 with a smaller matrix  $S_b$ , i.e., the space interval  $\Omega = (0, 1)$  is divided into 200 equidistant subintervals. The solution at time  $T = 1$  is computed with EI-K4, ode45, and Radau. The number of time steps and the corresponding computational time are presented in Table 3 for some desired tolerances. As we mentioned in the beginning of this section, this problem is stiffer than the wave equation. To tackle this challenging situation, ode45 needs more than 1 millions time steps to achieve the solution even for a low tolerance like  $10^{-2}$ . On the other hand, the implicit Radau scheme requires less number of time steps to obtain a desired accuracy. However, its computational cost is very high.

**Example 4.10** Adding all damping terms into the nonlinear part. Another common approach for solving (25) consists in merging the damping terms with the nonlinear terms and then

**Table 3** Comparison among EI-K4, ode45, and Radau for a semilinear beam equation

Tolerance	EI-K4		ode45		Radau	
	M	Time (s)	M	Time (s)	M	Time (s)
10 <sup>-2</sup>	320	3.56	1067777	420.77	1663	369.91
10 <sup>-5</sup>	2560	26.67	1068325	432.29	3043	699.78
10 <sup>-8</sup>	20480	215.86	1073297	438.69	5234	1264.90

The number of time steps for the exponential integrator is empirically chosen to reach the prescribed accuracy

employing various exponential integrators to solve the resulting semilinear equation. In this case, we have to solve the system

$$\dot{y}(t) = \tilde{A}y(t) + \tilde{F}(y(t)),$$

where

$$\tilde{A} = \begin{bmatrix} 0 & I \\ -\alpha S - \delta I & 0 \end{bmatrix}, \quad \tilde{F}(y(t)) = \tilde{F} \begin{pmatrix} u \\ w \end{pmatrix} = \begin{bmatrix} 0 \\ g(u) + h(w) - \beta Sw - \gamma w \end{bmatrix}.$$

The matrix  $S$  is either  $S = S_w$  as in (26) or  $S = S_b$  as in (29). The exponential function of  $\tilde{A}$  can be computed by the following formula

$$e^{t\tilde{A}} = \begin{bmatrix} \cos(t\Omega) & \Omega^{-1} \sin(t\Omega) \\ -\Omega \sin(t\Omega) & \cos(t\Omega) \end{bmatrix}, \quad \Omega = \sqrt{\alpha S + \delta I}, \tag{30}$$

An explicit form of  $\varphi_k(t\tilde{A})$  was also presented in the literature (for example: see [32]). We illustrate by two examples below the claim that this approach is more expensive and it also lacks accuracy.

Consider a wave Eq. (25) with  $\alpha = 1$ ,  $\beta = 10^{-2}$ ,  $\delta = 1$ ,  $\gamma = 10^{-1}$ . The nonlinear term is  $g(u) = -5u^3$ . The initial conditions are  $p(x) = 5 \sin(5\pi x)$  and  $q(x) = 5 \cos(10\pi x)$ . We compute the solution at  $T = 1$  by using EI-SW21 ( $c_2 = \frac{1}{3}$ ) with  $M \in \{10, 20, \dots, 10 \cdot 2^{10}\}$  time steps. We compare these numerical results with the reference solution evaluated by EI-K4 with  $M = 100000$  time steps. The convergence rates of two approaches are plotted in Fig. 7a. When we add all damping terms into the nonlinear part, the corresponding approximation is worse than the one given by our approach. Note that the errors are reduced approximately 100 times with our approach. Moreover, with a bigger  $\Delta t$ , we obtain the solution with an acceptable accuracy while the traditional approach needs much smaller time steps to achieve stability.

We next repeat Example 4.6. The solution at final time step  $T = 1$  is computed by using EI-SW21 ( $c_2 = 0.2$ ) with  $M \in \{320, \dots, 320 \cdot 2^7\}$  time steps. These numerical solutions are compared with the reference solution obtained by using EI-SW4 with  $M = 100000$  time steps. The errors are plotted in Fig. 7b. While our approach works and preserves the convergence rate of the exponential integrator EI-SW21, the traditional approach fails even with a small  $\Delta t$ . Since the matrix  $S_b$  is stiffer than the matrix  $S_w$ , it leads to the stability problem when the structural damping term  $\partial_{xxxxt}u$  is added into the nonlinear part.

In conclusion, the two examples clearly demonstrate the importance of using the matrix exponential of the linearization.

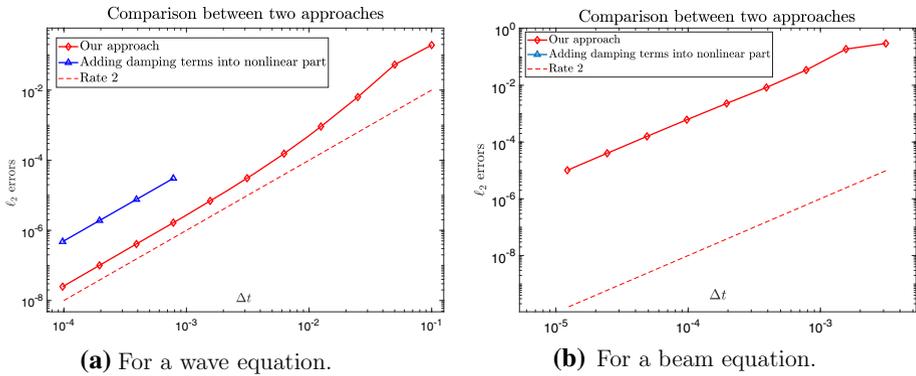


Fig. 7 The importance of using the matrix exponential of the linearization

### 5 Conclusion

We presented an approach to cheaply compute the action of the matrix exponential  $e^{tA}$  as well as the action of the matrix functions  $\varphi_k(tA)$  on a given vector by employing two linear transformations. Thus, the solution of certain linear differential equations can be computed in a fast and efficient way. By applying the exponential integrators in the literature, we can solve semilinear wave and semilinear beam equations.

Note that the described procedure can be extended to the case

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}, \text{ where } [A_i, A_j] = A_i A_j - A_j A_i = 0 \text{ for } 1 \leq i, j \leq 4, i \neq j.$$

Indeed, under the above assumption, the four matrices  $A_i$  share the same eigenvalues and eigenvectors. Thus, there exist a matrix  $Q$  and four corresponding diagonal matrices  $D_i$  such that  $A_i = Q D_i Q'$  for  $1 \leq i \leq 4$ . This implies that

$$A = \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} D_1 & D_2 \\ D_3 & D_4 \end{bmatrix} \begin{bmatrix} Q' & 0 \\ 0 & Q' \end{bmatrix}$$

Thus, the scheme can be analogously applied by evaluating the exponential of each  $2 \times 2$  block matrix  $\begin{bmatrix} \lambda_{1,i} & \lambda_{2,i} \\ \lambda_{3,i} & \lambda_{4,i} \end{bmatrix}$  where  $\lambda_{k,i}$  is an entry of the diagonal matrix  $D_k$  with  $1 \leq k \leq 4$ .

**Acknowledgements** We would like to thank the anonymous referees for the fruitful discussions which lead to improvements in the current version.

**Funding** Open access funding provided by University of Innsbruck and Medical University of Innsbruck. The authors have not disclosed any funding.

**Data Availability** Enquiries about data availability should be directed to the authors.

### Declarations

**Conflict of interest** The authors have not disclosed any competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give

appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Ansari, M., Esmailzadeh, E., Younesian, D.: Frequency analysis of finite beams on nonlinear Kelvin–Voigt foundation under moving loads. *J. Sound Vib.* **330**, 1455–1471 (2011)
2. Banks, H.T., Ito, K.: Approximation in LQR problems for infinite dimensional systems with unbounded input operators. *J. Math. Syst. Estim. Control* **7**, 1–34 (1997)
3. Bernstein, D.S., So, W.: Some explicit formulas for the matrix exponential. *IEEE Trans. Autom. Control* **38**(8), 1228–1232 (1993)
4. Carvalho, A., Cholewa, J., Dlotko, T.: Strongly damped wave problems: bootstrapping and regularity of solutions. *J. Differ. Equ.* **244**, 2310–2333 (2008)
5. Chen, W., Fino, A.Z.: Blow-up of solutions to semilinear strongly damped wave equations with different nonlinear terms in an exterior domain. *Math. Methods Appl. Sci.* **44**, 6787–6807 (2021)
6. Cheng, H.-W., Yau, S.S.-T.: More explicit formulas for the matrix exponential. *Linear Algebra Appl.* **262**, 131–163 (1997)
7. Cox, S., Matthews, P.: Exponential time differencing for stiff systems. *J. Comput. Phys.* **176**, 430–455 (2002)
8. Deuffhard, P.: A study of extrapolation methods based on multistep schemes without parasitic solutions. *Z. Angew. Math. Phys.* **30**, 177–189 (1979)
9. Do, V.N.V., Ong, T.H., Thai, C.H.: Dynamic responses of Euler–Bernoulli beam subjected to moving vehicles using isogeometric approach. *Appl. Math. Model.* **51**, 405 (2017)
10. Edalatzaeh, M.S., Morris, K.A.: Optimal actuator design for semilinear systems. *SIAM J. Control Optim.* **57**, 2992–3020 (2019)
11. Edalatzaeh, M.S., Morris, K.A.: Stability and well-posedness of a nonlinear railway track model. *IEEE Control Syst. Lett.* **3**, 162–167 (2019)
12. Gautschi, W.: Numerical integration of ordinary differential equations based on trigonometric polynomials. *Numer. Math.* **3**, 381–397 (1961)
13. Ghidaglia, J.M., Marzocchi, A.: Longtime behaviour of strongly damped wave equations, global attractors and their dimension. *SIAM J. Math. Anal.* **22**, 879–895 (1991)
14. Hairer, E., Wanner, G.: *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, Berlin (1996)
15. Higham, N.J.: *Functions of Matrices. Soc. Ind. Appl. Math.* (2008). <https://doi.org/10.1137/1.9780898717778>
16. Higham, N.J., Al-Mohy, A.H.: Computing matrix functions. *Acta Numer.* **19**, 159–208 (2010)
17. Hochbruck, M., Ostermann, A.: Explicit exponential Runge–Kutta methods for semilinear parabolic problems. *SIAM J. Numer. Anal.* **43**, 1069–1090 (2005)
18. Hochbruck, M., Ostermann, A.: Exponential Runge–Kutta methods for parabolic problems. *Appl. Numer. Math.* **53**, 323–339 (2005)
19. Hochbruck, M., Ostermann, A.: Exponential integrators. *Acta Numer.* **19**, 209–286 (2010)
20. Ikehata, R., Todorova, G., Yordanov, B.: Wave equations with strong damping in Hilbert spaces. *J. Differ. Equ.* **254**, 3352–3368 (2013)
21. Ito, K., Morris, K.: An approximation theory of solutions to operator Riccati equations for  $H^\infty$  control. *SIAM J. Control Optim.* **36**, 82–99 (1998)
22. Krogstad, S.: Generalized integrating factor methods for stiff PDEs. *J. Comput. Phys.* **203**, 72–88 (2005)
23. Liu, K., Liu, Z.: Exponential decay of energy of the Euler–Bernoulli beam with locally distributed Kelvin–Voigt damping. *SIAM J. Control Optim.* **36**, 1086–1098 (1998)
24. Luan, V.T., Ostermann, A.: Explicit exponential Runge–Kutta methods of high order for parabolic problems. *J. Comput. Appl. Math.* **256**, 168 (2014)
25. Mattsson, K., Stiernström, V.: High-fidelity numerical simulation of the dynamic beam equation. *J. Comput. Phys.* **286**, 194–213 (2015)
26. Morris, K.A.: *Controller Design for Distributed Parameter Systems*. Springer (2020)

27. Pata, V., Zelik, S.: Smooth attractors for strongly damped wave equations. *Nonlinearity* **19**, 1495–1506 (2006)
28. Paunonen, L., Phan, D.: Reduced order controller design for robust output regulation. *IEEE Trans. Autom. Control* **65**, 2480–2493 (2020)
29. Phan, D., Paunonen, L.: Finite-dimensional controllers for robust regulation of boundary control systems. *Math. Control Relat. Fields* **11**, 95–117 (2021)
30. Ponce, G.: Global existence of small solutions to a class of nonlinear evolution equations. *Nonlinear Anal. Theory Methods Appl.* **9**, 399–418 (1985)
31. Strehmel, K., Weiner, R.: *Linear-implizite Runge–Kutta–Methoden und ihre Anwendung*. Vieweg+Teubner Verlag (1992)
32. Wang, B., Wu, X.: Global error bounds of one-stage extended RKN integrators for semilinear wave equations. *Numer. Algorithms* **81**(4), 1203–1218 (2018)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.