ORIGINAL PAPER

High-order exponential integration for seismic wave modeling

Fernando V. Ravelo[1] · **Martin Schreiber[2,3]** · **Pedro S. Peixoto[1]**

Abstract

Seismic imaging is a major challenge in geophysics with broad applications. It involves solving wave propagation equations with absorbing boundary conditions (ABC) multiple times. This drives the need for accurate and efficient numerical methods. This study examines a collection of exponential integration methods, known for their good numerical properties on wave representation, to investigate their efficacy in solving the wave equation with ABC. The purpose of this research is to assess the performance of these methods. We compare a recently proposed Exponential Integration based on Faber polynomials with well-established Krylov exponential methods alongside a high-order Runge-Kutta scheme and low-order classical methods. Through our analysis, we found that the exponential integrator based on the Krylov subspace exhibits the best convergence results among the high-order methods. We also discovered that high-order methods can achieve computational efficiency similar to low-order methods while allowing for considerably larger time steps. Most importantly, the possibility of undertaking large time steps could be used for important memory savings in full waveform inversion imaging problems.

1 Introduction

The resolution of wave propagation equations is a widely researched topic due to its broad range of applications in various fields. One particularly prominent application is seismic imaging, where material parameters of underground regions are estimated based on seismic data. This technique is extensively utilized in the industry for the exploration and extraction of fossil fuels [27].

The numerical approximation of propagating wave equations is a critical stage in this procedure. Consequently, the complexity of the problem impels the development of novel

✉ Fernando V. Ravelo
  fernanvr@ime.usp.br

  Martin Schreiber
  martin.schreiber@univ-grenoble-alpes.fr

  Pedro S. Peixoto
  ppeixoto@usp.br

1   Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, Brazil

2   Université Grenoble Alpes/ Laboratoire Jean Kuntzmann/ Inria, Grenoble, France

3   Technical University of Munich, Munich, Germany

techniques competitive to the efficiency and accuracy of existing schemes [5, 33, 35].

The propagation of elastic waves can be described as a linear hyperbolic system of PDEs. Nonetheless, the addition of absorbing boundary conditions to replicate an infinite domain modifies the eigenvalues, and they are no longer purely imaginary. In this context, low-order classical explicit schemes such as the Leap-Frog [49] approximation, fourth-order Runge-Kutta [58], and similar methods [29, 36] have proven effective. Nevertheless, despite their computational speed, these algorithms require very small time steps to approximate the solution accurately. Consequently, this leads to high memory requirements, which can be a significant challenge in solving inverse problems, which is another crucial step in seismic imaging.

In recent decades, a class of numerical algorithms known as exponential integrators have emerged and demonstrated successful applications in various fields. These algorithms have been effectively utilized in areas such as photonics [46], the development of numerical methods for weather prediction [45], and the modeling of diverse physical phenomena [38], often surpassing the performance of classical schemes. Another example of successful applications of exponential integrators is provided by Brachet et al. [11], where classical explicit and implicit schemes were compared with exponential integrators, revealing that exponential integra-

tors exhibit superior dissipation and dispersion properties. In Cohen and Dujardin [17], exponential integrators were compared with explicit and implicit schemes for solving the non-linear Schrödinger equation, demonstrating accuracy comparable to the best performance of classical methods and surpassing other schemes such as the Crank-Nicholson method. In Iyiola and Wade [28], an exponential integrator was compared with an Implicit-Explicit (IMEX) scheme and a second-order backward difference scheme for solving non-linear space-fractional equations, concluding that it offers a significantly larger stability region. In general, this class of methods enables the use of larger time steps, with the potential for parallel implementation to enhance efficiency. Exponential integrators are typically employed to preserve favorable dispersion properties while allowing for larger time steps [51].

Exponential integrators can be categorized into two types: one primarily concerned with approximating the exponential (or related φ-functions) of a large matrix resulting from the spatial discretization of the linear term of a system of PDE, and the other focused on different schemes to approximate the non-linear term [25, 43]. In the context of wave propagation equations with absorbing boundary conditions, these equations are primarily governed by the linear term, and a source function replaces the non-linear term with a well-defined analytic representation. This leads to a transformation of the problem, as demonstrated by Al-Mohy and Higham [3], which is a generalization of the work of Sidje [52], where the problem transforms into calculating the exponential of a slightly enlarged matrix.

The approximation of a matrix exponential has received significant attention [1, 6, 42]. Numerous exponential integrators have been developed to address this matrix function calculation [3, 25, 39]. One notable exponential integrator is based on the Krylov subspace, with several schemes utilizing this approach and demonstrating good performance [22, 44, 52]. Another method relies on rational approximations [2], which are generally implicit and less suitable for large operators. However, they can be combined with the Krylov method to reduce matrix dimensions [3]. Another approach utilizes Chebyshev polynomials, an explicit method that can be formulated as a three-term recurrence relation [9, 30]. Additionally, there are other methodologies, such as Leja points interpolation [10, 19], optimized Taylor approximations [8], and contour integrals [50], among others.

When applied to solve hyperbolic systems, such as the wave equations in heterogeneous media, their performance is poorly understood. To the best of our knowledge, only a limited number of literature publications have focused on methods of practical relevance for this specific problem [30, 31, 47, 55, 56, 61].

In Zhang et al. [61], an implicit exponential integrator method is developed, and a comparison with other methods is presented, demonstrating superior results in terms of accuracy and dispersion. However, a notable drawback of the implicit method is its high computational cost for each time step, making it primarily suitable for very stiff problems.

Kole [30] proposes an explicit exponential integrator based on Chebyshev polynomial approximations, which achieves high solution accuracy and permits large time steps. Nevertheless, the applicability of Chebyshev polynomials for approximating the solution is limited to cases where the system matrix is symmetric or antisymmetric, preventing the modeling of absorbing boundary conditions. As a result, its usage in seismic applications is constrained.

Chebyshev expansions have been proposed [31, 55, 56] to approximate the matrix exponential, with the use of absorbing boundary conditions. Nonetheless, the numerical results are primarily validated using simplistic ABCs, and there is a lack of proof demonstrating convergence for these boundary conditions.

In a previous work [47], we explored a generalization of the exponential integrator using Faber polynomials, a variant of Chebyshev polynomials. This approach enabled us to solve the wave equations with absorbing boundary conditions. We found that employing higher approximation degrees in the Faber polynomial-based method allows for increased time step sizes without incurring additional computational costs. Furthermore, the augmented time step approximations exhibit favorable accuracy and dispersion properties.

A notable gap in the existing literature is the absence of experiments comparing high-order methods with classical low-order schemes for solving the wave equations with absorbing boundary conditions. Our work fills in this gap by comparing exponential integrators based on Faber polynomials, Krylov subspace projection, and High-order Runge-Kutta with various classical methods. Specifically, we consider classical low-order methods such as Leap-frog, fourth-order and four-stage Runge-Kutta (RK4-4), second-order and three-stage Runge-Kutta (RK3-2), and seventh-order and nine-stage Runge-Kutta (RK9-7). Detailed descriptions of these methods can be found in Section 3. The comparison between these algorithms focuses on several key characteristics, including numerical dispersion, dissipation, convergence, and computational cost, which are thoroughly discussed in Sections 4 and 5. By investigating these aspects, we aim to comprehensively evaluate the different methods and their suitability for solving wave equations with absorbing boundary conditions. Finally, in Section 6, we summarize the main findings and draw conclusive remarks based on our research.

2 The wave equation

The execution of finite difference methods when solving a system of partial differential equations depends on the contin-

uum formulation and the approximation of the spatial derivatives [57]. These factors directly impact the discrete operator used in the computations. This section lays the groundwork for the entire analysis presented in the manuscript. We discuss the fundamental elements defining the discrete spatial operator present in seismic imaging applications. These elements include formulating wave propagation equations with absorbing boundary conditions (ABC), spatial discretization using derivative approximations, and free surface treatment.

We employ Perfectly Matching Layers (PML) as the absorbing boundary condition [7] to simulate an infinite domain. Despite the significant computational cost associated with PML absorbing boundaries, they remain widely used in numerous numerical studies within the field of seismic imaging [16, 29, 54]. For computational efficiency, we implement the PML for the two-dimensional acoustic wave propagation equations. While we can extend our analysis to propagating waves in three dimensions, the complexity of the equations substantially increases, resulting in a significant rise in computational requirements. Thus, for our purposes, we define the system of equations within a rectangular domain $\Omega = [0, a] \times [0, -b]$ for $t > 0$, as follows:

all $y \in [0, \delta]$), and determine the solution approximation at the upper boundary based on the chosen spatial discretization

2.1 Spatial discretization

Several finite difference discretization schemes have been proposed for the wave propagation equations [29, 40, 41, 48, 62]. While determining the optimal approach remains an open problem, staggered grids have gained significant popularity for these equations, as noted in Moczo et al. [41]. Staggered grids have good numerical stability properties and usually allow better wave representation for high wave numbers. Additionally, in the study of Moczo et al. [41], the effectiveness of second and fourth order staggered grid spatial discretizations was compared for solving the wave equation, with the fourth-order discretization demonstrating superior accuracy and stability over the second-order counterpart. As our main interest is the time integration methods, to minimize spatial numerical errors we adopted an eighth-order staggered grid spatial discretization.

The spatial discretization consists of a uniform staggered grid ($\Delta x = \Delta y$) of 8th-order. The positions of the discrete points are depicted in Fig. 1.

$$\frac{\partial}{\partial t} \begin{pmatrix} u \\ v \\ w_x \\ w_y \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\beta_x \beta_y + c^2 \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) & -(\beta_x + \beta_y) & c^2 \frac{\partial}{\partial x} & c^2 \frac{\partial}{\partial y} \\ (\beta_y - \beta_x)\frac{\partial}{\partial x} & 0 & -\beta_x & 0 \\ (\beta_x - \beta_y)\frac{\partial}{\partial y} & 0 & 0 & -\beta_y \end{pmatrix} \begin{pmatrix} u \\ v \\ w_x \\ w_y \end{pmatrix} + \begin{pmatrix} 0 \\ f \\ 0 \\ 0 \end{pmatrix}, \tag{1}$$

where, $u = u(t, x, y)$ is the displacement, $c = c(x, y)$ is the given velocity distribution in the medium, $v = v(t, x, y)$ is the wave velocity, and $f = f(x, y, t)$ is the source term. The w-functions, $(w_x, w_y) = (w_x(t, x, y), w_y(t, x, y))$, are the auxiliary variables of the PML approach and the β-functions are known and control the damping factor in the absorbing layer.

$$\beta_z(z) = \begin{cases} 0, & \text{if } d(z, \partial\Omega) > \delta \\ \beta_0 \left(1 - \frac{d(z, \Omega)}{\delta} \right)^2, & \text{if } d(z, \partial\Omega) \leq \delta \end{cases}, \quad z \in \{x, y\}$$

where $d(z, \partial\Omega)$ is the distance from z to the boundary of Ω, δ is the thickness of the PML domain, and β_0 is the magnitude of the absorption factor. Thus, the domain Ω comprises a physical domain, where the wave propagates normally, and an outer layer of thickness δ (the domain of the PML), where the waves dampen.

Due to the attenuation of displacement within the PML domain, we opt for a Dirichlet boundary condition (null displacement) along three sides of the rectangular domain Ω. However, this boundary condition does not apply to the top side, as a free-surface boundary condition is more suitable for seismic-imaging simulations. Therefore, on the upper side of Ω, we exclude the PML domain ($\beta_y(y) = 0$, for

For the inner discrete points, the 8th-order approximation of the derivatives is given by

$$\frac{\partial u_{i+\frac{1}{2}}}{\partial x} \approx \frac{1225}{1024\Delta x} \left(u_{i+1} - u_i - \frac{u_{i+2} - u_{i-1}}{15} \right.$$
$$\left. + \frac{u_{i+3} - u_{i-2}}{125} - \frac{u_{i+4} - u_{i-3}}{1715} \right) \tag{2}$$
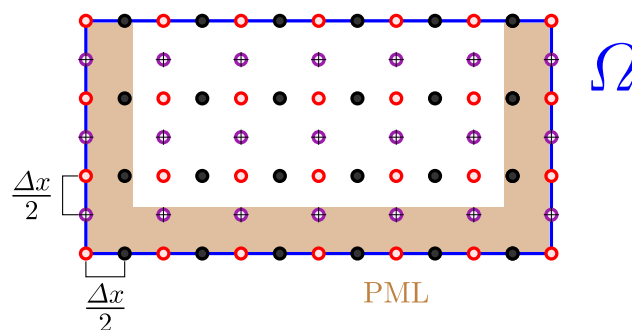
$$\frac{\partial^2 u_i}{\partial x^2} \approx -\frac{205}{72}u_i + \frac{8}{5}(u_{i+1} - u_{i-1}) - \frac{1}{5}(u_{i+2} - u_{i-2})$$
$$+ \frac{8}{315}(u_{i+3} - u_{i-3})$$
$$- \frac{1}{560}(u_{i+4} - u_{i-4}) \tag{3}$$

with analogous expressions for the y-coordinate in the 2D discretization.

The approximation of derivatives near the sides and bottom boundaries, where Dirichlet boundary conditions within a PML domain are applied, is performed using the formulas Eqs. 2 and 3. In these cases, the function values required outside the domain Ω are set to zero. However, this does not impact the accuracy of the numerical approxima-

tions because, within the PML domain, the wave amplitudes decrease to zero.

A different strategy is necessary to approximate derivatives at points near the upper boundary. Since there is no PML domain, and the boundary condition corresponds to a free surface.

2.2 Free surface

From the free-surface condition $\nabla u \cdot (0, 1) = 0$, we deduce the Neumann boundary condition $\frac{\partial u}{\partial y} = 0$. Additionally, by substituting $\beta_y = 0$ at the free surface in the third equation of Eq. 1, we obtain $w_y = 0$ at the free surface. Utilizing these values, we can approximate the required spatial derivatives of the functions in Eq. 1 concerning the variable y.

There are two main approaches for approximating the spatial derivatives concerning y. The first approach introduces artificial points outside Ω, assigning function values at these points to satisfy the conditions at the free surface. The second procedure involves approximating the derivatives at the free surface and its nearest points using only the function values within the domain Ω, without artificially extending the functions. According to Kristek et al. [32], the latter alternative brings greater accuracy to the numerical solution and is the approach employed throughout this work.

Next, assuming that the grid points lying on the free surface correspond to the evaluation of the displacement u (i.e., the free surface is at $y = 0$), we need 8th-order approximations for

1. the second derivative $\frac{\partial^2 u}{\partial y^2}$ at the points with $y = \{0, -\Delta x, -2\Delta x, -3\Delta x\}$.
2. the first derivative $\frac{\partial u}{\partial y}$ at the points with $y = \{-\frac{1}{2}\Delta x, -\frac{3}{2}\Delta x, -\frac{5}{2}\Delta x\}$.
3. the first derivative $\frac{\partial w_y}{\partial y}$ at the points with $y = \{0, -\Delta x, -2\Delta x, -3\Delta x\}$.

The referred approximations for the derivatives $\frac{\partial^2 u}{\partial y^2}$ and $\frac{\partial u}{\partial y}$, using Taylor expansions, can be found in the Appendix A.1. As for the derivative $\frac{\partial w_y}{\partial y}$, we apply the algorithm outlined in Fornberg [20]. This algorithm computes the derivative with

any approximation order and utilizes an arbitrary points distribution where the values of the derived function are known.

3 Time integration methods

After characterizing the spatial discretization and the approximation of the spatial derivatives, we obtain the following linear system of equations:

$$\frac{d}{dt}\boldsymbol{U}(t) = \boldsymbol{H}\boldsymbol{U}(t) + \boldsymbol{f}(t), \quad \boldsymbol{U}(t_0) = \boldsymbol{U}_0. \tag{4}$$

Here, $\boldsymbol{U}(t)$ is a vector comprising the discretized functions u, v, w_x, and w_y, while the matrix $\boldsymbol{H}$ represents the discretized spatial operator of the system Eq. 1. The vector $\boldsymbol{f}$ consists of the source function evaluated at each grid point.

Most of the numerical methods described in this section solve the first-order system of ordinary differential equations Eq. 4. Our primary focus lies in approximating the time dimension, leading to the classification of methods as either low or high order concerning time. The following subsections present the numerical schemes employed in the former classifications.

3.1 Low order methods

We consider four low-order methods that offer attractive features for approximating the solution of wave equations. Three of these methods are based on the Runge-Kutta (RK) approach, while the fourth is the Leap-frog scheme.

– **2nd order Runge-Kutta (RK3-2):** The RK3-2 method is a second-order RK scheme with three stages. It is a modification of the classical RK2-2 method designed to increase its stability region [18], enabling its application to hyperbolic problems. The scheme can be expressed as follows:

$$\begin{aligned}
\boldsymbol{k_1} &= H\boldsymbol{u}^n + \boldsymbol{f}(t_n), \\
\boldsymbol{k_2} &= H(\boldsymbol{u}^n + (\Delta t/2)\boldsymbol{k_1}) + \boldsymbol{f}(t_n + \Delta t/2), \\
\boldsymbol{k_3} &= H(\boldsymbol{u}^n + (\Delta t/2)\boldsymbol{k_2}) + \boldsymbol{f}(t_n + \Delta t/2), \\
\boldsymbol{u}^{n+1} &= \boldsymbol{u}^n + \Delta t\boldsymbol{k_3}.
\end{aligned}$$

– **4th order Runge-Kutta (RK4-4):** The classical RK4-4 scheme balances stability region and computational requirements [12].

– **7th order Runge-Kutta of nine stages (RK9-7):** This scheme has been specifically constructed for hyperbolic equations and exhibits favorable dispersion properties [15].

– **Two step method (Leap-frog):** The Leap-frog method is highly efficient for solving wave equations. It utilizes two time steps to approximate the second-order time derivative. The equations solved by the Leap-frog method are

$$\frac{\partial^2 u}{\partial t^2} = -\beta_x \beta_y u - (\beta_x + \beta_y)\frac{\partial u}{\partial t}$$
$$+ c^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial \omega_x}{\partial x} + \frac{\partial \omega_y}{\partial y}\right) + f$$
$$\frac{\partial w_x}{\partial t} = -\beta_x w_x + (\beta_y - \beta_x)\frac{\partial u}{\partial x}$$
$$\frac{\partial w_y}{\partial t} = -\beta_y w_y + (\beta_x - \beta_y)\frac{\partial u}{\partial y}$$

with the discrete approximations in time

$$\frac{\partial^2 u_i^n}{\partial t^2} \approx \frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2},$$
$$\frac{\partial w_{z_{i+1/2}}^n}{\partial t} \approx \frac{w_{z_{i+1/2}}^{n+1} - w_{z_{i+1/2}}^{n-1}}{2\Delta t}, \text{ with } z \in \{x, y\}.$$

3.2 High order methods

The methods presented in this section are of arbitrary orde4r and utilize exponential integrators based on Faber polynomials, Krylov subspaces, and a high-order Runge-Kutta method.

According to Hochbruck and Ostermann [25], an exponential integrator approximates the semi-analytic solution of Eq. 4 using the formula of constant variation

$$U(t) = e^{(t-t_0)H}U_0 + \int_{t_0}^{t} e^{(t-\tau)H}f(\tau)d\tau.$$

Expanding the function f in a Taylor series, the solution of (3.2) can be expressed as the matrix exponential [3]

$$u(t) = \begin{bmatrix} I_{n\times n} & 0 \end{bmatrix} e^{(t-t_0)\tilde{H}} \begin{bmatrix} u_0 \\ e_p \end{bmatrix}, \tag{5}$$

where $e_p \in \mathbb{R}^p$ is a vector with zeros in its first $p-1$ elements and one in its last element, $I_{n\times n}$ is the identity matrix of dimension n, and

$$\tilde{H} = \begin{pmatrix} H & W \\ 0 & J_{p-1} \end{pmatrix},$$

where the columns of the matrix W consist of the values of the function f and the approximations of the first $p-1$ derivatives of f, and J_{p-1} is a square matrix of dimensions $p \times p$ with ones in the upper diagonal and zeros elsewhere.

Equation 5 forms the basis for the exponential integrator methods implemented in this research, and the approach used to compute the matrix exponential in Eq. 5 determines each of the following exponential integrators.

– **Faber approximation (FA):** This method is an exponential integrator based on Faber polynomials. As presented in Ravelo et al. [47], the exponential approximation is carried on with the three-term recurrence Faber series

$$F_0(H) = I_{n\times n}, \quad F_1(H) = H/\gamma - c_0 I_{n\times n},$$
$$F_2(H) = F_1(H)F_1(H) - 2c_1 I_{n\times n},$$
$$F_j(H) = F_1(H)F_{j-1}(H) - c_1 F_{j-2}(H), \quad j \geq 3,$$

where the parameters c_0 and c_1 depend on the eigenvalues distribution of the operator H. Then, the solution in the next time instant is expressed as

$$u^{n+1} = \sum_{j=0}^{m} a_j F_j(H)u^n,$$

where a_j are the Faber coefficients.

– **Krylov subspace projection (KRY):** This method is an exponential integrator utilizing operator projections within the Krylov subspace. Various proposed algorithms involve adaptive time steps and different strategies for generating the subspace basis [22]. However, to ensure an impartial comparison among all the schemes, we opt for the traditional Arnoldi algorithm to establish the vector basis and perform the projection of H [21].

$$u_1 = u_0/\|u_0\|_2$$
Do j from 1 to m:
$\quad w = Hu_j$
$\quad$ Do k from 1 to j:
$\quad\quad A_{i,j} = w \cdot u_k$
$\quad\quad w = w - A_{i,j}u_k$
$\quad A_{j+1,j} = \|w\|_2$
$\quad u_{j+1} = w/A_{j+1,j}$
Then, $e^H u_0 \approx \|u_0\|_2[u_1|\ldots|u_m]e^A e_1$

Listing 1 Pseudocode of Arnoldi algorithm.

After constructing the matrix projection A, we compute the reduced matrix's exponential using the Padé polynomial approximation method, as outlined in Al-Mohy and Higham [3].

The Arnoldi algorithm to construct an orthonormal basis is very computationally intensive, and the amount of matrix-

vector operations does not represent its actual computational cost. Regarding this subject, the use of non-orthonormal bases has been proposed to greatly reduce this cost [22]. However, due to the non-orthogonality of the Krylov basis, the reduced matrix $\boldsymbol{A}$ does not represent an orthogonal projection of the linear transformation $\boldsymbol{H}$ onto the Krylov subspace [22]. This discrepancy may lead to numerical errors that differ from those encountered in the classical Arnoldi method. As we aim to use the classical Krylov method, we employ the Arnoldi algorithm without considering the cost of constructing the Krylov subspace, given the potential to significantly reduce the computational cost.

– **High-order Runge-Kutta (HORK):** Runge-Kutta methods are extensively used for solving differential equations [14], and also in combination with exponential integrator schemes [18, 34] These methods are naturally extended to high-order schemes. They can be explicit and are easy to implement. For this research, we adopt the Runge-Kutta algorithm of arbitrary order proposed by Gottlieb [23], defined by the relation

$$
\begin{aligned}
\boldsymbol{k}_0 &= \boldsymbol{u}^n \\
\boldsymbol{k}_i &= (\boldsymbol{I}_{n \times n} + \Delta t \boldsymbol{H}) \boldsymbol{k}_{i-1}, \quad i = 1 \ldots m-1 \\
\boldsymbol{k}_m &= \sum_{i=0}^{m-2} \lambda_i \boldsymbol{k}_i + \lambda_{m-1} (\boldsymbol{I}_{n \times n} + \Delta t \boldsymbol{H}) \boldsymbol{k}_{m-1} \\
\boldsymbol{u}^{n+1} &= \boldsymbol{k}_m,
\end{aligned}
$$

where λ_i are the coefficients of the Runge-Kutta and have a straightforward computation. According to Gottlieb and Gottlieb [23], the Runge-Kutta method exhibits strong stability-preserving properties if the coefficients λ_i are non-negative.

3.3 Computational cost and memory usage

In addition to the accuracy of the numerical solution when discussing the different approaches, we are also interested in their resource consumption. Specifically, we focus on the computational operations required by each algorithm and their utilization of computational memory. Determining the exact number of computations performed by these methods is a complex task, further complicated by the fact that sparse matrix-vector multiplications are known to be bandwidth-limited in terms of performance [4, 26]. Therefore, we adopt a simplified model that focuses exclusively on counting the loading and storing of elements. We consider only the matrix-vector operations, as the other vector operations introduce, at most, small variations in the number of operations. Consequently, the cost of each method by time step will be its number of stages or matrix-vector operations (MVOs).

Therefore, the overall number of MVOs of a method for computing the solution up to a fixed time T and using a time step size Δt can be expressed as:

$$
\mathrm{N_{op}} = \#\mathrm{MVOs} \frac{T}{\Delta t} = \frac{\#\mathrm{MOVs}}{\Delta t} T,
$$

where the value of T can be disregarded when comparing the methods since it remains constant within a numerical experiment.

Memory consumption becomes a critical factor when solving the three-dimensional wave equation for seismic imaging applications. The primary concern is for the inverse problem, where the solution for each time step must be stored to be accessed later. Therefore, the number of time steps required for each method

$$
N_{\mathrm{mem}} = \frac{T}{\Delta t},
$$

is also an important variable that we will take into account afterward.

4 Analysis on homogeneous media

A common challenge arises when utilizing finite difference methods to solve wave equations due to numerical dispersion and dissipation. Numerical dispersion occurs when phase velocities depend on the frequency, leading to distortions in wave signals. On the other hand, numerical dissipation is associated with wave amplitude and is responsible for the emergence of high-frequency waves with small amplitudes in finite difference methods (Section 5.1 of Strikwerda [53]).

Since the continuous wave equation is non-dispersive and non-dissipative, it is essential to ensure that the numerical methods used to solve it do not introduce excessive dispersion and dissipation. In seismic imaging problems, these issues can lead to significant inaccuracies in estimating the velocity field. Therefore, special attention must be given to identifying and mitigating these errors.

In this section, we conduct a comparative analysis of the methods introduced in Section 3 within the context of a homogeneous velocity field and a single wave signal. We focus on evaluating their dispersion and dissipation errors and examining how these errors depend on the choice of time-step size.

4.1 Numerical dispersion and dissipation by Fourier transform

Our analysis investigates numerical dispersion and dissipation by quantifying variations in phase velocities of numerical approximations concerning a reference solution.

Fig. 2 Snapshots of the reference solution at times $t = 0.6s$ and $T = 1.3s$ within the homogeneous medium $\Omega = [0\text{km}, 6\text{km}] \times [0\text{km}, 5\text{km}]$. The Ricker signal source position (blue dot) and the receiver location (black square) are highlighted. During the time interval $t \in [0.6, 1.3]s$, the front wave propagates through the receiver location

(a) Wave propagation at time $t = 0.6s$.

(b) Wave propagation at time $T = 1.3s$.
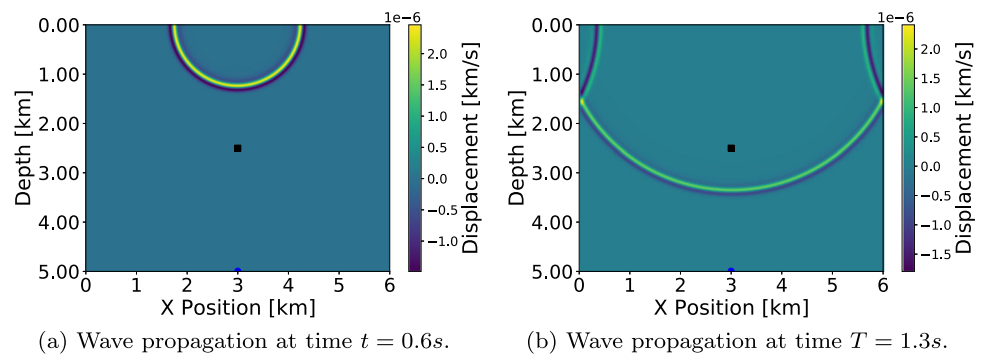
To achieve this, we conduct a comparison in the frequency domain and estimate velocity changes for each frequency. For this analysis, a Fourier transform is applied to the solution, consisting of a single signal of a Ricker wavelet [24]. Consequently, we consider a homogeneous medium with a source point and a receiver (a spatial position where the signal is recorded over time).

Let $\mathcal{F}\text{appr}(\omega)$ and $\mathcal{F}\text{ref}(\omega)$ denote the Fourier transforms of the approximated and reference signals, respectively, with ω representing the frequency. Thus, we establish the relationship as follows:

$$\mathcal{F}\text{ref}(\omega) = e^{k(\omega) + il(\omega)} \mathcal{F}\text{appr}(\omega),$$

where the real functions $k(\omega)$ and $l(\omega)$ account for the numerical dissipation and dispersion errors, respectively, present in the approximated solution.

It is important to note that minimizing dissipation and dispersion errors hinges upon the extent to which the functions $k(\omega)$ and $l(\omega)$ approaches zero. As the numerical solution is computed at a finite number of time instants, ω is also limited to a finite range. Then, we calculate the mean of the absolute values of $k(\omega)$ and $l(\omega)$, which can be considered an approximation of the integral of their absolute values. Hereafter, we refer to these metrics as the dissipation and dispersion error. Furthermore, to mitigate potential numerical errors arising from divisions by small quantities during the computation of dispersion and dissipation errors, we exclusively consider frequencies where the amplitudes in the reference or approximated solutions surpass 1% of the peak amplitude of the reference solution.

In the next section, we will outline the numerical features of the Ricker signal experiment. Following that, in the subsequent two sections, we will apply the criteria discussed here to assess the numerical dissipation and dispersion errors.

4.2 Single signal experiment

The numerical solutions for wave propagation equations are computed in the homogeneous medium $\Omega = [0\text{km}, 6\text{km}] \times [0\text{km}, 5\text{km}]$, with a velocity $c = 3\text{km/s}$. A Ricker source

is placed at position (3km, 4.99km) (with a delay of $t_0 = 0.18s$), and a receiver is positioned at (3km, 2.5km). The time integration is carried out until $T = 1.3s$ without applying any absorbing boundary conditions, as the reflections at the boundary have not yet reached the receiver by the final time. The spatial discretization size used for numerical solutions of the methods is $\Delta x = 10\text{m}$, while the reference solution is computed with $\Delta x = 2.5\text{m}$ and $\Delta t = 0.104\text{ms}$ using the RK9-7 scheme.

We are mainly interested in the largest time step allowed such that the error of the methods is under a fixed threshold. However, to ensure uniform wave sampling of the numerical approximations at the receiver, we use larger time steps up to the point when the wave closely approaches the receiver ($t = 0.6s$). Then, a uniform $\Delta t = 0.417\text{ms}$ is employed until the final time $T = 1.3s$ is reached. Figure 2 displays the homogeneous medium with the source and the receiver positions and the snapshots of the reference solution at times $t = 0.6s$ and $T = 1.3s$.

Although our primary focus lies in evaluating the time error of the methods, it is essential to acknowledge the influence of spatial discretization on numerical accuracy. To account for this spatial effect, convergence, dispersion, and dissipation are computed for all methods with a small timestep, $\Delta t = 0.417\text{ms}$ (see Fig. 10 in Appendix A.2). The
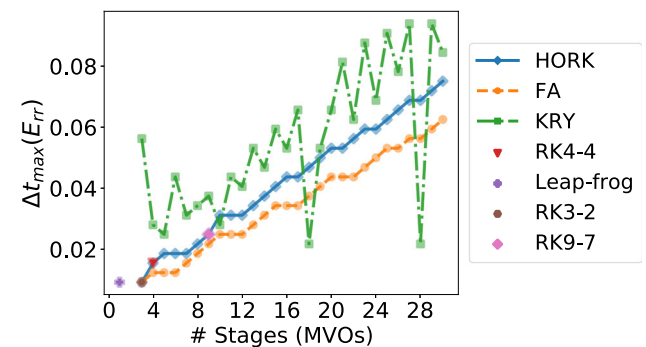

Fig. 3 Dependence of $\Delta t_{\max}$ on the approximation degree of the numerical scheme. A higher number of stages leads to an increase in the maximum allowable time step without significantly increasing the number of computations

Fig. 4 Variation of $\Delta t_{\max}$ (left) and N_{op}^{disp} (right) concerning the numerical scheme and the number of stages utilized, according to the numerical dispersion error for a Ricker source peak frequency of $f_M = 15$Hz. Generally, a higher number of stages leads to an increase in the maximum allowable time step size without significantly increasing the number of computations. * Here we neglect the computational complexity of creating the Krylov subspaces

(a) Maximum-time-step, $\Delta t_{\max}$.   (b) Number of MVOs by $\Delta t_{\max}$.
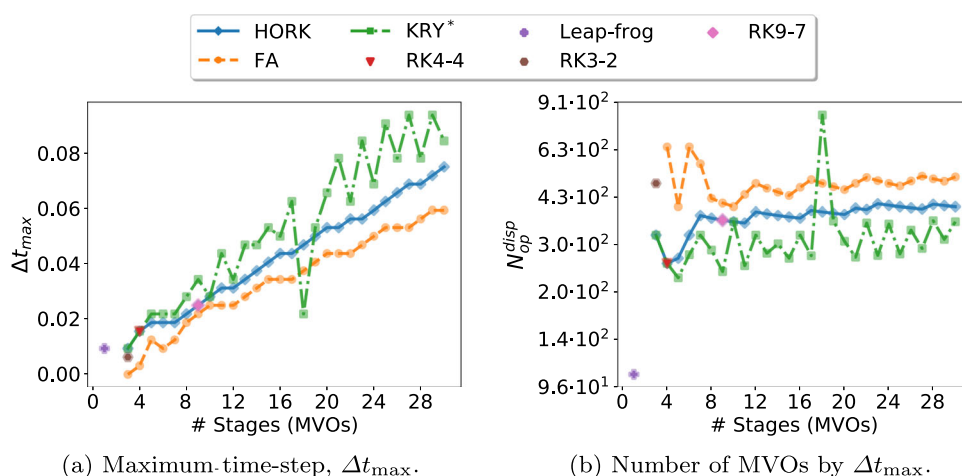
minimum convergence, dispersion, and dissipation errors obtained from this computation serve as an estimation of the spatial effect. Then, we determine the maximum Δt allowable for the methods such that the time error remains less or equal to 50% of the spatial error.

For this experiment, the approximation error due to the spatial discretization is approximately $3.9 \cdot 10^{-6}$ (see Fig. 10 in Section A.2). Based on this, we determine $\Delta t_{\max}$ as the maximum Δt such that the approximation error is less or equal to $E_{rr} = 5.9 \cdot 10^{-6}$. Then, the convergence can be analyzed by investigating the signal error at a specific receiver location (3km, 2.5km). It becomes clear that an increase in the number of stages leads to an increase in $\Delta t_{\max}$ (see Fig. 3).

Referring to Fig. 3, it can be observed that the Krylov method displays a highly oscillatory pattern concerning its associated $\Delta t_{\max}$. Intriguingly, this pattern reaches its local peak values when the subspace dimension is an odd number. The general behavior of the methods convergence is not sensible to the cutting point of the error threshold, and for variations of $E_{rr} = 5.9 \cdot 10^{-6}$, they remain valid. So, we

expect a similar behavior when studying the dispersion and dissipation.

4.3 Dispersion results

The dispersion error arising from spatial discretization is estimated as 0.002. Consequently, we permit for the time integrator methods an error threshold of $1.5\times$ higher, equating to a maximum allowable dispersion error of 0.003. Then, we search for $\Delta t_{\max}$ such that the dispersion error remains below this limit.

In addition to $\Delta t_{\max}$, we introduce a computational cost measure denoted as N_{op}^{disp}, similar to the ideas of Section 3.3, defined as:

$$N_{op}^{disp} = \frac{\#\, MVOs}{\Delta t_{\max}}.$$

Based on Fig. 4, the Leap-frog algorithm is approximately two times faster than the other schemes but requires small

Fig. 5 Variation of $\Delta t_{\max}$ (left) and N_{op}^{disp} (right) concerning the numerical scheme and the number of stages utilized, according to the numerical dissipation error for a Ricker source peak frequency of $f_M = 15$Hz. Generally, a higher number of stages leads to an increase in the maximum allowable time step size without significantly increasing the number of computations. * Here we neglect the computational complexity of creating the Krylov subspaces

(a) Maximum-time-step, $\Delta t_{\max}$.   (b) Number of MVOs by $\Delta t_{\max}$.
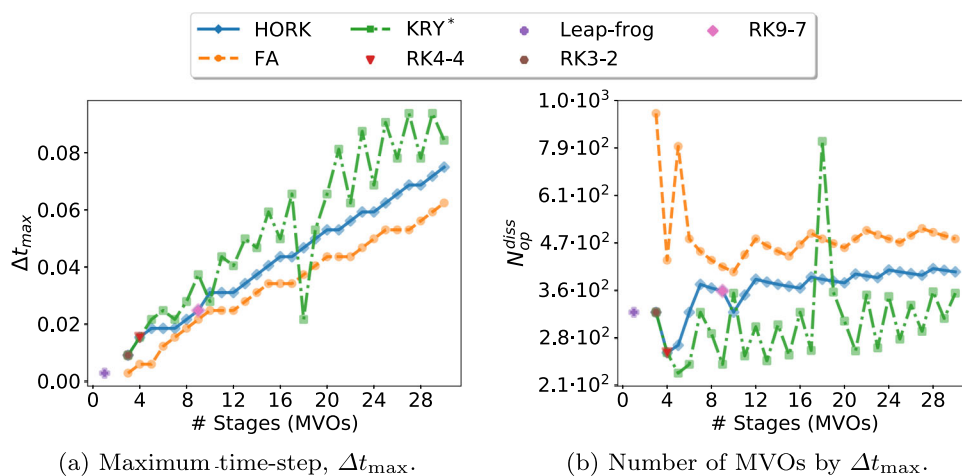
time steps. On the other hand, the explicit exponential methods exhibit an increase in their maximum time step as the number of stages used rises, without a significant increase in the number of matrix-vector operations required. Interestingly, the peak values of the Krylov methods for the largest Δt and the lower N_{op}^{disp} are consistently for the odd numbers of the subspace dimension Fig. 4.

To ensure the robustness of our analysis, we reproduce the previous results in Appendix A.2.1 using various peak frequencies of the Ricker source since wave frequencies influence dispersion.

4.4 Dissipation results

Similar to the previous section, we estimate the minimum dispersion error, independent of the time integrator used. The minimum dissipation error is approximately $2.4 \cdot 10^{-7}$. Thus, we once again compute the maximum time-step, Δt_{max}, such that the dispersion error remains below $3.6 \cdot 10^{-7}$. Besides of Δt_{max}, we define the computational cost measure as

$$N_{op}^{diss} = \frac{\# \, MVOs}{\Delta t_{max}},$$

similar to convergence and dispersion.

In Fig. 5, a similar trend is observed with the dispersion error, except that the performance of the exponential integrator is better in relation to the Leap-frog when comparing the dissipation error. Notably, the high-order methods display an increase in the time-step size with the number of stages used without significantly increasing the number of matrix-vector operations required.

As with the numerical dispersion, we reproduce the experiments for different Ricker source peak frequencies in Appendix (Section A.2.2).

5 Analysis on realistic seismic models

In this section, we describe the numerical experiments we will use to compare the accuracy of the approximations of the different methods. For comparison, we generated a reference solution using the RK9-7 scheme with a finer grid ($\Delta x = 5$m) and then estimated the error for each method using various time step sizes. To ensure a robust accuracy assessment, we employ two procedures. First, we compare the approximated solution across the entire physical space (excluding the PML domain) at a specific time instant. Second, we compare the seismogram data of the solution values at the upper boundary for all the simulation time. For each error evaluation, we determine the maximum time step size, Δt_{max}, that allows a scheme of a particular order to achieve a solution accuracy below a predefined threshold with the least number of MVOs. Additionally, we introduce an efficiency measure and an indicator of memory utilization derived from the number of MVOs and Δt_{max}, following the concepts outlined in Section 3.3.

5.1 Test cases

We consider four numerical scenarios with different velocity fields (see Fig. 6). The first is a synthetic example of a heterogeneous medium with high contrast velocities and a sharp

Fig. 6 Velocity fields of the test cases Corner Model, Santos Basin, Marmousi, and SEG/EAGE, used to study the numerical convergence
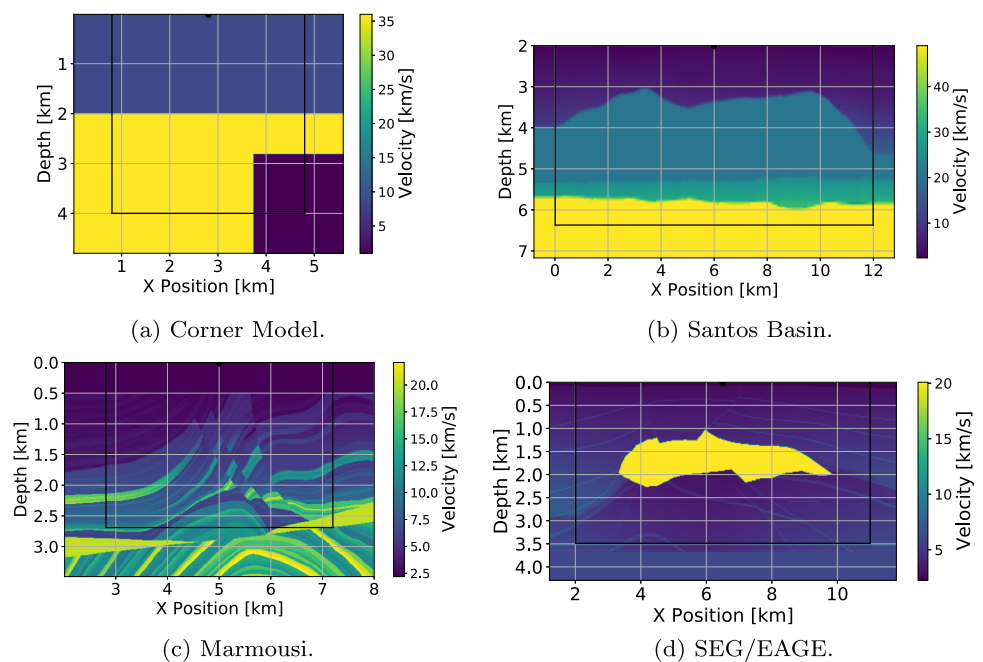

(a) Corner Model.

(b) Santos Basin.

(c) Marmousi.

(d) SEG/EAGE.

Table 1 Parameters of the four numerical simulations considered in this paper

Test cases	Corner Model	Santos Basin
Domain dimensions	$\Omega = [0km, 4km] \times [0km, 4km]$	$\Omega = [0km, 12km] \times [2km, 6.4km]$
Simulation time	$T = 1.1s$	$T = 1.5s$
Source position	(2km, 0.02km)	(6km, 2.02km)
PML thickness (δ)	1.0km	0.8km
Test cases	Marmousi	SEG/EAGE
Domain dimensions	$\Omega = [2km, 8km] \times [0km, 3.5km]$	$\Omega = [2km, 11km] \times [0km, 3.5km]$
Simulation time	$T = 1.5s$	$T = 2s$
Source position	(5km, 0.02km)	(6.5km, 0.02km)
PML thickness (δ)	0.8km	0.8km

corner. The second is a 2D slice of the velocity field of the Santos Basin[1] oil and gas exploration region. A 2D portion of Marmousi velocity field is the third example, and the final test is the 2D SEG/EAGE synthetic model.

In all the examples, we include a source and an arrangement of receivers near the surface of the medium. The specification of this construct and other parameters of the numerical simulations are specified in Table 1.

We save the solution at the upper boundary at each simulated time instant to construct the seismogram. We use a time span twice as long as specified in each experiment outlined in Table 1 to allow the reflected waves to reach the surface.

5.2 Maximum time-step

We need to calculate the maximum allowable time step, denoted as $\Delta t_{\max}$, for all time integration schemes. We initially consider the numerical error inherent to the spatial discretization in each numerical experiment (see Appendix A.3) since this error is independent of the time integration method. Next, we employ a tolerance level equivalent to 150% of the spatial discretization error in each experiment. Finally, we use that tolerance to compute the value of $\Delta t_{\max}$ for the numerical schemes described in Section 3.

We consider a spatial-step size of $\Delta x = 10m$ to compute the approximated solutions mentioned before. Figures 7 and 8 show the allowed $\Delta t_{\max}$ by all the methods for the numerical tests Corner Model, Santos Basin, Marmousi, and SEG/EAGE.

Figure 7 presents the maximum time step, $\Delta t_{\max}$, considering the spatial error of the solution at a time instant. Generally, when the approximation degree increases, we observe an increment in the allowed $\Delta t_{\max}$. Moreover, the Krylov subspace approximation exhibits the largest time steps among the studied methods, followed by the other high-order methods. In contrast, low-order methods such as Leap-frog and RK3-2 require smaller time steps.

---

[1] A typical velocity field of Santos Basin region, in Brazil.

The determination of $\Delta t_{\max}$ based on the seismogram data is illustrated in Fig. 8. Similar to Fig. 7, an increase in the number of stages leads to a higher maximum time step. Notably, the Krylov subspace method consistently demonstrates the highest $\Delta t_{\max}$ values, followed by other high-order methods.

Based on the insights gained from Figs. 7 and 8, we can conclude that the choice between using the error of the solution at a particular time instant or the seismogram data leads to similar values of $\Delta t_{\max}$ for the methods. Therefore, for the sake of simplicity, we estimate $\Delta t_{\max}$ with the error of the approximation in the physical domain at a specific time instant (as illustrated in Fig. 7). Next, we estimate the computational efficiency and memory consumption of each method using the concepts of Section 3.3.

5.3 Computational efficiency and memory consumption

From the previous section, we concluded that using a method with a large number of stages allows an increase in the maximum time step such that we have a solution with good accuracy. However, it is unclear if increasing the number of stages to use a larger Δt reduces the number of operations or how it helps in utilizing the memory. To answer this question, we apply the ideas discussed in Section 3.3 and define the measure of computational efficiency

$$N_{op}^{\Delta t} = \frac{\# \text{MVOs}}{\Delta t_{\max}},$$

and the indicator of memory consumption to store results for a backward propagation

$$N_{mem}^{\Delta t} = \frac{T}{\Delta t_{\max}},$$

where T is the simulation time defined by Table 1, for each numerical experiment.

Fig. 7 Snapshots of the reference solution for Corner Model, Santos Basin, Marmousi, and SEG/EAGE numerical tests (left column), and the Δt_{max} of each method such that the error with the reference solution is under a fixed threshold (right column). An increase in the number of stages of the method leads to a larger Δt_{max}
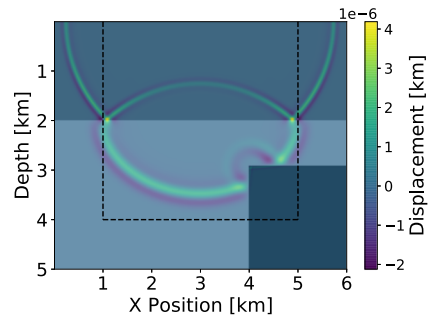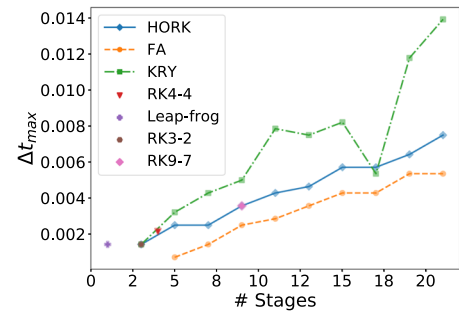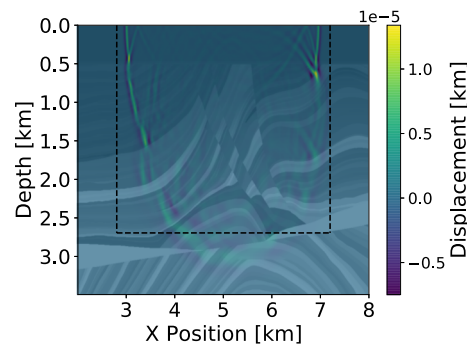

(a) Corner Model solution at time $T = 1.1s$.
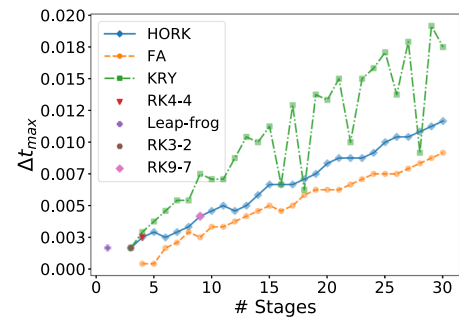
(b) Computing Δt_{max} for Corner Model using an error tolerance of $3.03 \cdot 10^{-7}$.
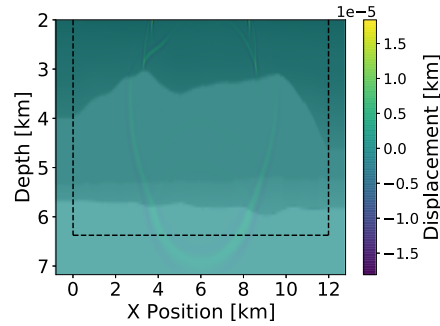
(c) Santos Basin solution at time $T = 1.5s$.

(d) Computing Δt_{max} for Santos Basin using an error tolerance of $8.33 \cdot 10^{-7}$.

(e) Marmousi solution at time $T = 1.5s$.
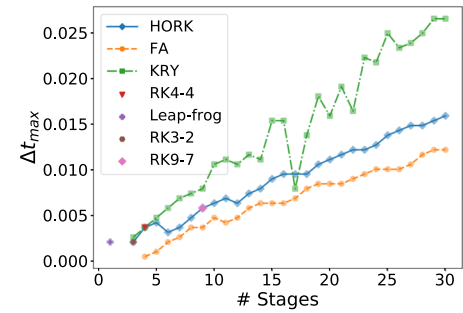
(f) Computing Δt_{max} for Marmousi using an error tolerance of $9.93 \cdot 10^{-7}$.
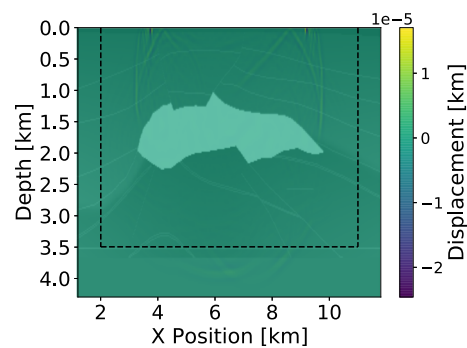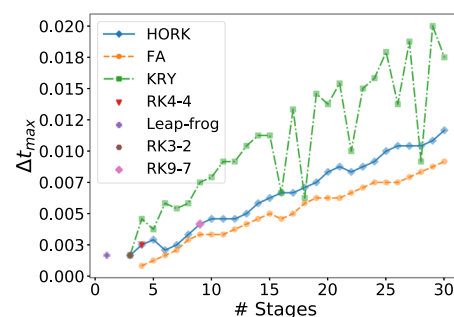
(g) SEG/EAGE solution at time $T = 2s$.
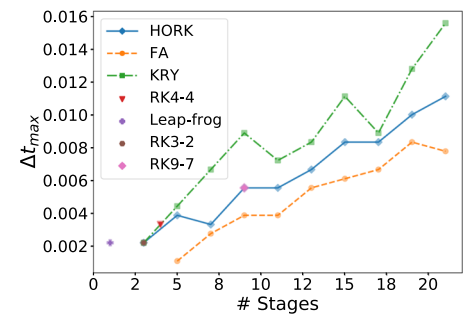
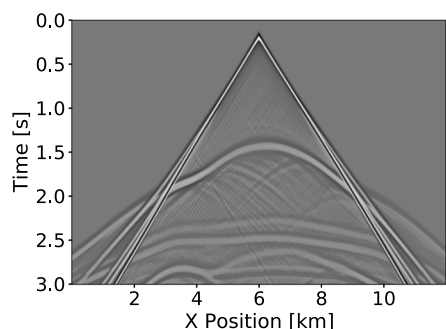(h) Computing Δt_{max} for SEG/EAGE using an error tolerance of $1.3 \cdot 10^{-6}$.

Fig. 8 Seismogram of the reference solution for Corner Model, Santos Basin, Marmousi, and SEG/EAGE numerical tests (left column), and the corresponding $\Delta t_{\max}$ of each method, ensuring the error remains below a fixed threshold (right column). An increase in the number of stages of the method leads to a larger $\Delta t_{\max}$
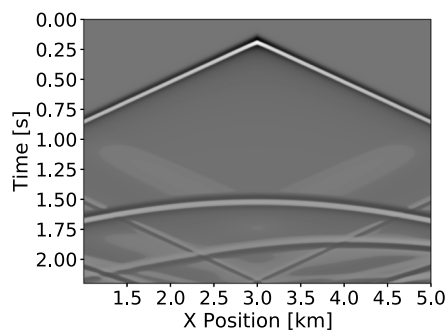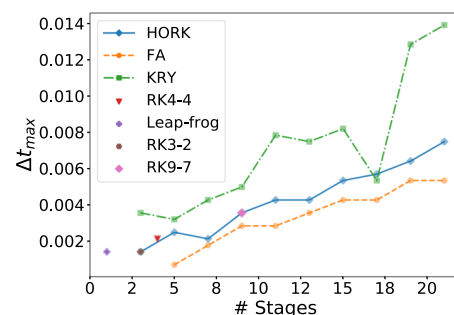

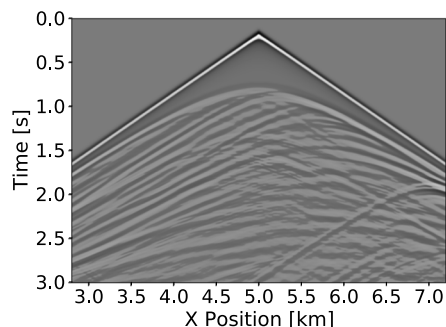(a) Corner Model seismogram until time $T = 2.2s$.

(b) Computing $\Delta t_{\max}$ for Corner Model using an error tolerance of $2.92 \cdot 10^{-7}$.
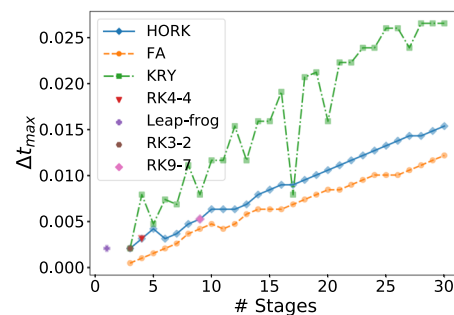

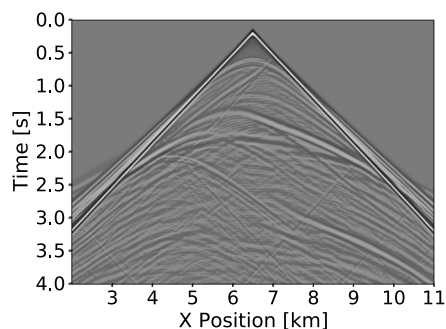(c) Santos Basin seismogram until time $T = 3s$.

(d) Computing $\Delta t_{\max}$ for Santos Basin using an error tolerance of $2.65 \cdot 10^{-6}$.
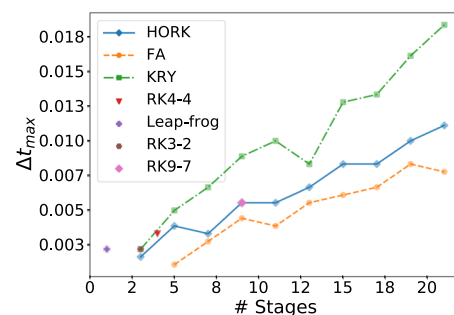

(e) Marmousi seismogram until time $T = 3s$.

(f) Computing $\Delta t_{\max}$ for Marmousi using an error tolerance of $1.3 \cdot 10^{-6}$.

(g) SEG/EAGE seismogram until time $T = 4s$.

(h) Computing $\Delta t_{\max}$ for SEG/EAGE using an error tolerance of $4.2 \cdot 10^{-6}$.

Figure 9 illustrates the number of MVOs and the memory usage for all the methods when solving the Marmousi numerical example. The Leap-frog algorithm proves the most efficient among the tested methods. However, in terms of memory utilization, this method requires a substantial amount of memory. On the other hand, among the high-order methods, the Krylov subspace approximations demonstrate the best performance, even comparable to the Leap-Frog scheme. However, we would like to point out that we are using a simplified model that doesn't consider the creation of the Krylov subspaces. Nevertheless, we observe a significant decrease in the number of stored vectors for high-order methods in general. Additionally, we notice that further increases in the approximation degree have an attenuated effect in reducing memory utilization, which is negligible for degrees larger than 20.

We observed similar behavior in the other numerical tests, and their corresponding graphs can be found in Appendix A.3.1.

6 Discussion

In this paper, we have implemented seven time-integration schemes, consisting of three arbitrary-order schemes based on exponential integrators and four classical low-order schemes. These algorithms have been compared through various numerical accuracy metrics, including stability, dispersion, and convergence. We have also studied the computational cost and memory requirements for each method across different approximation degrees.

The stability and dispersion analyses were conducted within a homogeneous domain by analyzing the Fourier transform of a single wave generated by a Ricker wavelet. We observed that the high-order methods were capable of using larger time steps as the polynomial degree increased. In general, we found that the Leap-frog method, although requiring smaller time steps, outperformed the high-order methods.

Yet, when considering the dissipation error, the high-order methods displayed competitiveness and even surpassed all the low-order methods.

We conducted extensive tests to evaluate convergence using four distinct velocity fields: three realistic fields and one with sharp interfaces. We assessed the approximation error both in the physical space at a specific time instant and using seismogram data. Remarkably, our results proved consistent and robust across both types of errors and all four numerical experiments. Moreover, the Krylov method presented the largest time step size in all the tests, resulting in the least amount of solution vectors required to save for the inverse problem. As a drawback, the Krylov method requires at each time instant to access as many vectors (with the dimensions of the solution of the wave equations) as stages of the method are used. This greatly hinders using the method to solve the direct problem. In general, high approximation degrees allowed for larger time steps, a finding that significantly impacts the number of saved vectors needed for solving the inverse problem. These results provide a different strategy to approach the memory challenges associated with the inverse problem.

This research addresses a gap in the existing literature, as most previous studies on high-order methods have predominantly focused on the spatial dimension [13, 37, 59, 60]. Additionally, no prior work has comprehensively examined high-order exponential integrators in the context of the wave equation applied to seismic imaging, scrutinizing the performance across a wide range of approximation degrees. Nonetheless, we acknowledge that our implementation of high-order approximations using exponential integrators is naive. Substantial enhancements are possible, particularly in terms of implementing adaptive time-stepping strategies to mitigate the hump phenomena associated with the matrix exponential [42]. Indeed, adaptive algorithms have been proposed, such as the KIOPS algorithm for the Krylov subspace projections, which significantly outperforms the classical Krylov method used in our study.

Fig. 9 Dependence of the number of MVOs and amount of stored solution vectors on the polynomial degree for the Marmousi numerical test. As the number of stages increases, the number of computations stabilizes, and memory usage decreases. * Here we neglect the computational complexity of creating the Krylov subspaces

(a) Computational cost.

(b) Memory utilization.
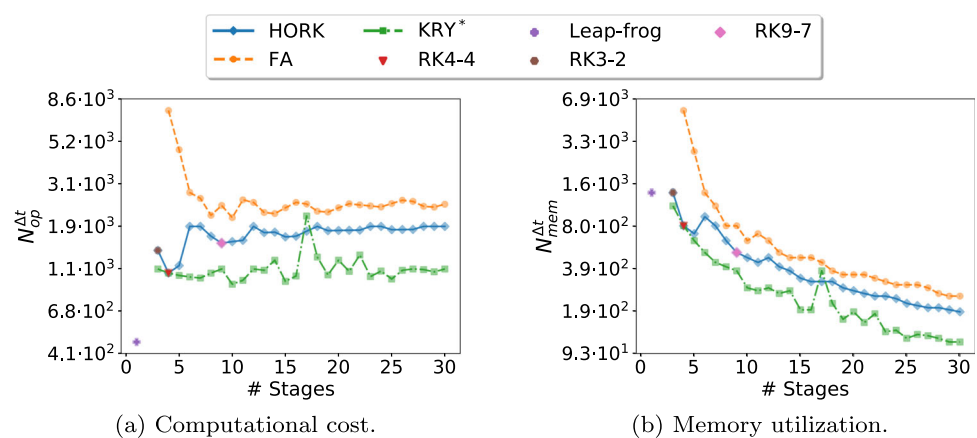
A Appendix

## A.1 Approximations at the free-surface

We present the finite difference approximations of 8th order for the required derivatives of the functions at the points near the free surface. To simplify the notation, we define $u_i = u(x, -i\Delta x)$, and $w_i = w_y\left(x, -(i+\frac{1}{2})\Delta x\right)$. Since we are considering a uniform grid, we have that $\Delta y = \Delta x$, and so, only Δx will be used.

$$\frac{\partial^2 u}{\partial y^2}(x, 0) \approx \left(-\frac{3144919}{352800}u_0 + 16u_1 - 14u_2 + \frac{112}{9}u_3 - \frac{35}{4}u_4\right.$$
$$\left. + \frac{112}{25}u_5 - \frac{14}{9}u_6 + \frac{16}{49}u_7 - \frac{1}{32}u_8\right)\frac{1}{\Delta x^2},$$

$$\frac{\partial^2 u}{\partial y^2}(x, -\Delta x) \approx \left(\frac{271343}{156800}u_0 - \frac{1991}{630}u_1 + \frac{57}{40}u_2 + \frac{13}{60}u_3 - \frac{109}{288}u_4\right.$$
$$\left. + \frac{6}{25}u_5 - \frac{11}{120}u_6 + \frac{179}{8820}u_7 - \frac{9}{4480}u_8\right)\frac{1}{\Delta x^2},$$

$$\frac{\partial^2 u}{\partial y^2}(x, -2\Delta x) \approx \left(-\frac{18519}{78400}u_0 + \frac{58}{35}u_1 - \frac{251}{90}u_2 + \frac{22}{15}u_3 - \frac{1}{16}u_4\right.$$
$$\left. - \frac{14}{225}u_5 + \frac{1}{30}u_6 - \frac{2}{245}u_7 + \frac{17}{20160}u_8\right)\frac{1}{\Delta x^2},$$

$$\frac{\partial^2 u}{\partial y^2}(x, -3\Delta x) \approx \left(\frac{74801}{1411200}u_0 - \frac{37}{140}u_1 + \frac{67}{40}u_2 - \frac{263}{90}u_3 - \frac{23}{100}u_5\right.$$

$$\left. + \frac{13}{360}u_6 - \frac{1}{245}u_7 + \frac{1}{4480}u_8\right)\frac{1}{\Delta x^2}, + \frac{53}{32}u_4$$

$$\frac{\partial u}{\partial y}\left(x, -\frac{1}{2}\Delta x\right) \approx \left(\frac{5034629}{3763200}u_0 - \frac{23533}{15360}u_1 + \frac{4259}{15360}u_2 - \frac{1103}{9216}u_3\right.$$
$$\left. + \frac{151}{3072}u_4 - \frac{1171}{76800}u_5 + \frac{139}{46080}u_6 - \frac{211}{752640}u_7\right)\frac{1}{\Delta x},$$

$$\frac{\partial u}{\partial y}\left(x, -\frac{3}{2}\Delta x\right) \approx \left(-\frac{363509}{3763200}u_0 + \frac{6297}{5120}u_1 - \frac{6147}{5120}u_2 + \frac{211}{3072}u_3\right.$$
$$\left. + \frac{3}{1024}u_4 - \frac{153}{25600}u_5 + \frac{29}{15360}u_6 - \frac{57}{250880}u_7\right)\frac{1}{\Delta x},$$

$$\frac{\partial u}{\partial y}\left(x, -\frac{5}{2}\Delta x\right) \approx \left(\frac{4631}{250880}u_0 - \frac{305}{3072}u_1 + \frac{1245}{1024}u_2 - \frac{3725}{3072}u_3\right.$$
$$\left. + \frac{275}{3072}u_4 - \frac{69}{5120}u_5 + \frac{5}{3072}u_6 - \frac{5}{50176}u_7\right)\frac{1}{\Delta x},$$

A.2 Homogeneous medium

This section complements the results in Section 4. First, we show the convergence, dispersion, and dissipation errors associated with the eighth-order spatial discretization scheme using $\Delta x = 10m$ (Fig. 10). Additionally, we present how varying the peak frequencies as $f_M = 10, 15, 20, 25$, impact the maximum allowable time-step Δt_{max} and the number of matrix-vector operations (MVOs) for different schemes and approximation degrees.

Fig. 10 Convergence, dispersion, and dissipation errors using the time-step $\Delta t = \frac{\Delta x}{8c}$ for different numerical methods, with a peak frequency of $f_M = 15$Hz. The approximation order does not matter, since there is an error associated to the spatial discretization

(a) Convergence error.

(b) Dispersion error.

(c) Dissipation error.
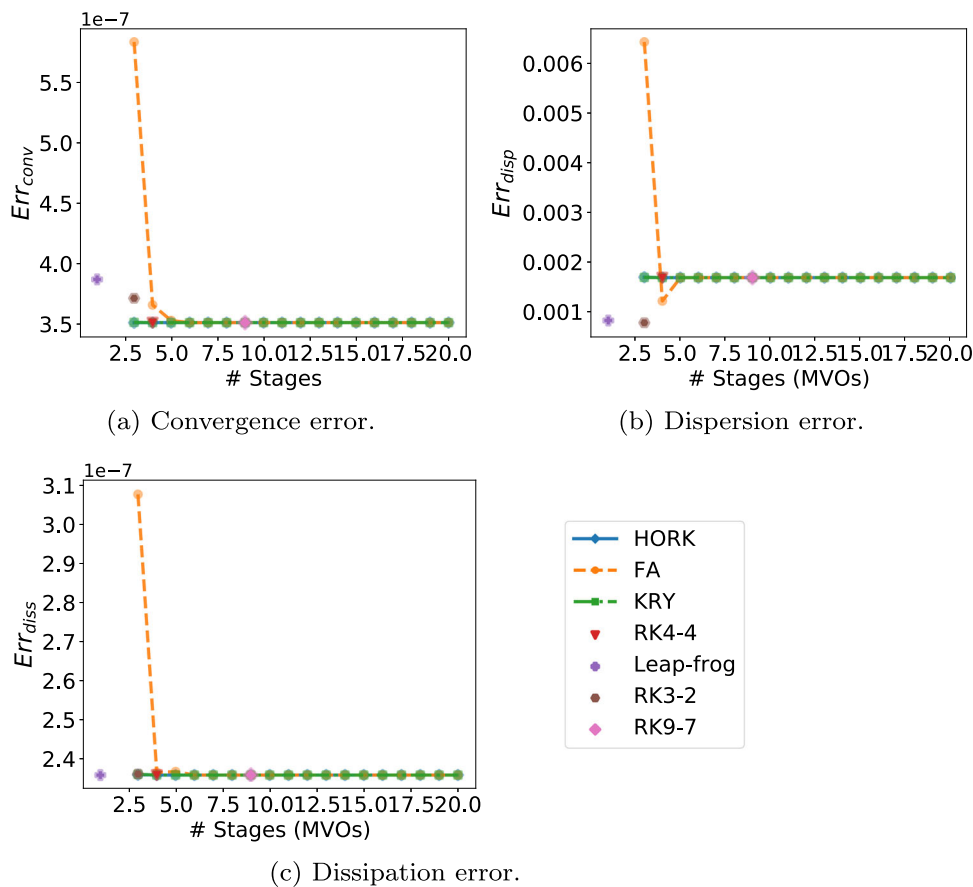
A.2.1 Dispersion results

From Fig. 11, we perceive that the general behavior is maintained independent of the peak frequencies. With the difference that when the peak frequency increases, the results for the Krylov method are more oscillatory, and the high-

Fig. 11 Maximum time step (Δt_{max}) while controlling the time dispersion error of each method to be below 50% of the spatial dispersion error concerning different peak frequencies of the Ricker wavelet. A grater number of stages generally allows larger time steps
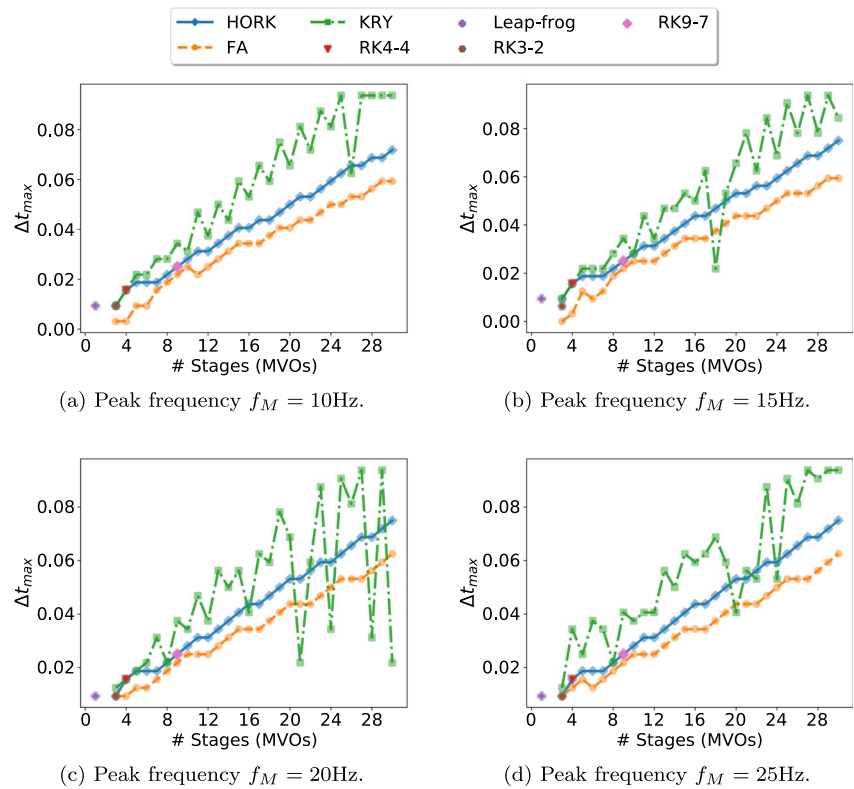

(a) Peak frequency $f_M = 10$Hz.

(b) Peak frequency $f_M = 15$Hz.

(c) Peak frequency $f_M = 20$Hz.

(d) Peak frequency $f_M = 25$Hz.

Fig. 12 Dependence of the number of matrix-vector operations and the maximum time-step required to compute the solution on the polynomial degree, considering different peak frequencies. While increasing the number of stages generally leads to a slight increment in computations. * Here we neglect the computational complexity of creating the Krylov subspaces

(a) Peak frequency $f_M = 10$Hz.

(b) Peak frequency $f_M = 15$Hz.

(c) Peak frequency $f_M = 20$Hz.

(d) Peak frequency $f_M = 25$Hz.
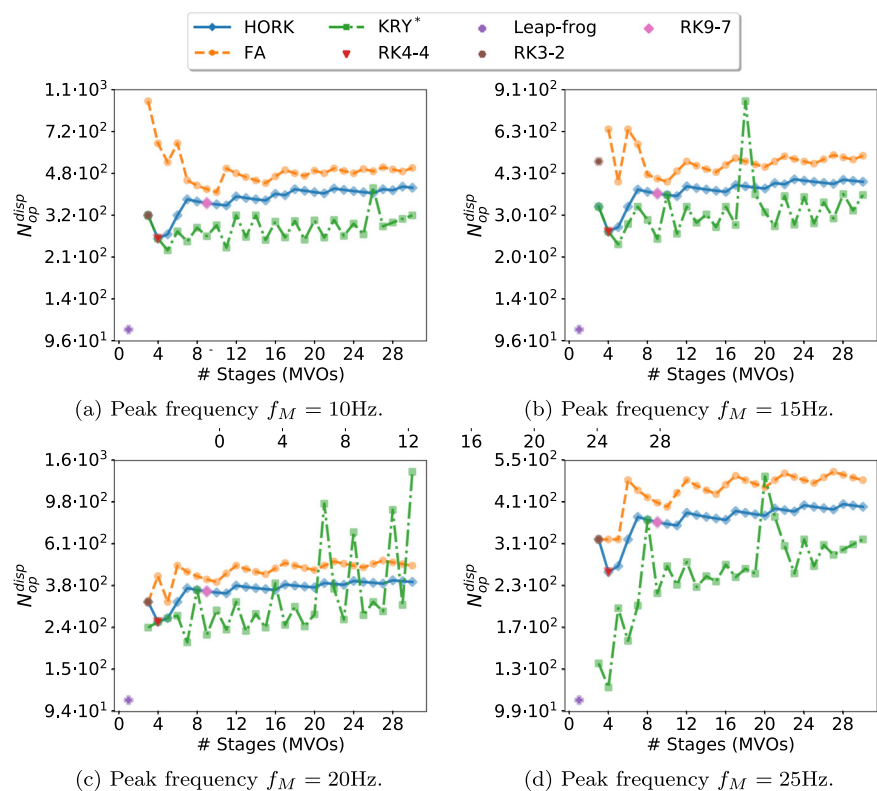
degree approximations using Faber polynomials suffer from more round-off errors.

In Fig. 12, we still observe that the Leap-frog algorithm requires the least amount of MVOs. The FA and HORK methods share a similar number of computations independent of the peak frequency.

A.2.2 Dissipation results

A similar trend of Fig. 11 is observed in Fig. 13, as with the dispersion error. The Krylov method still has the worst performance for the different peak frequencies. However, it is noteworthy that the RK9-7 method (red triangle) displays an even better performance concerning the dissipation error.

Fig. 13 Maximum time step such that the time dissipation error of each method is less than 50% of the spatial dispersion error for different peak frequencies of the Ricker wavelet. In general, more stages allow larger time steps, except for the Krylov method, where Δt_{max} reach a limit

(a) Peak frequency $f_M = 10$Hz.

(b) Peak frequency $f_M = 15$Hz.

(c) Peak frequency $f_M = 20$Hz.
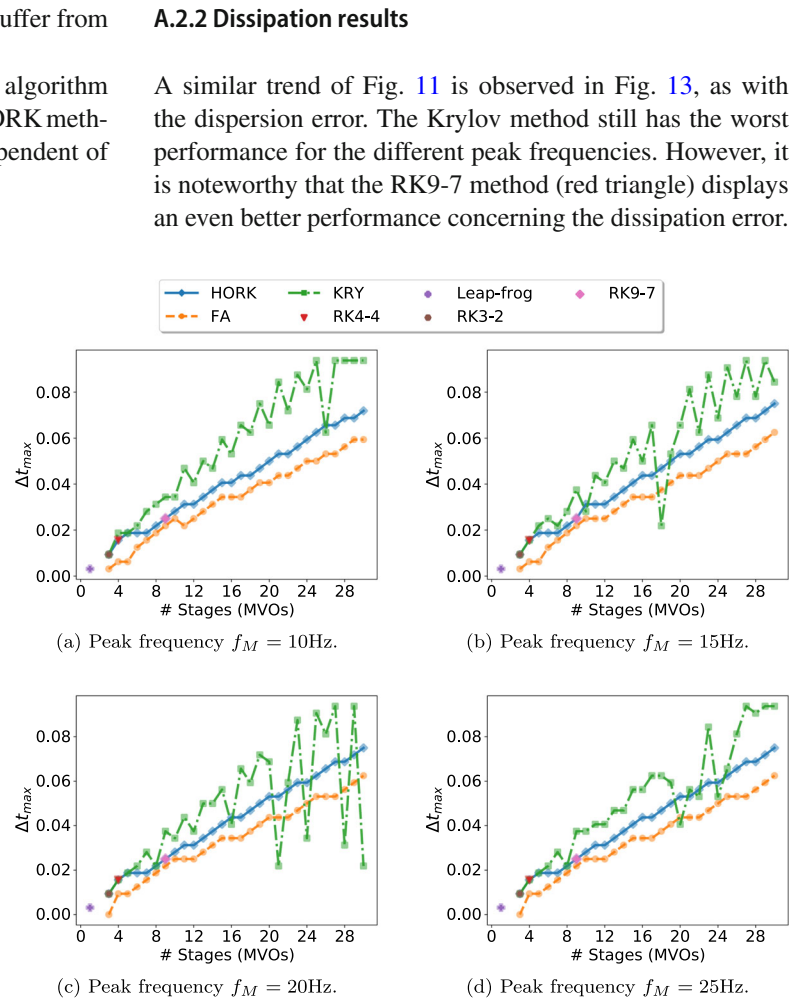
(d) Peak frequency $f_M = 25$Hz.

Fig. 14 Dependence on the polynomial degree of the number of matrix-vector operations by maximum time-step required to compute the solution for different peak frequencies. When the number of stages increases, the number of computations increases slightly. * Here we neglect the computational complexity of creating the Krylov subspaces

(a) Peak frequency $f_M = 10$Hz.

(b) Peak frequency $f_M = 15$Hz.

(c) Peak frequency $f_M = 20$Hz.

(d) Peak frequency $f_M = 25$Hz.
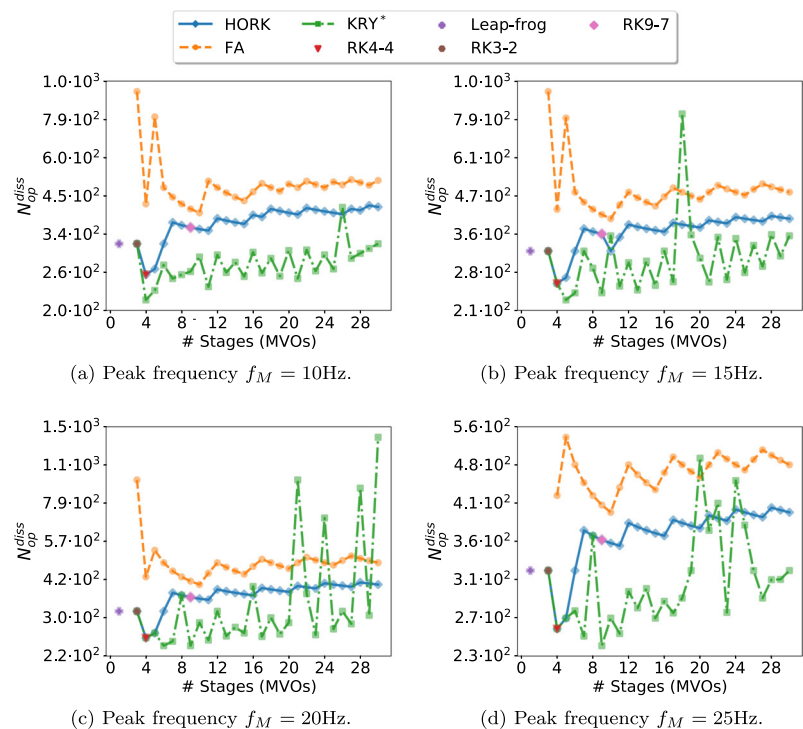
Regarding computational efficiency in the analysis of the dispersion error, the RK9-7 scheme still maintains an efficient computational performance. The FA and HORK exhibit similar behavior among the high-order methods, with a decline in efficiency for high-order Faber polynomials as the peak frequency increases. Nonetheless, the Krylov method exhibits the best performance in general, but with a very marked oscillatory behavior (Fig. 14).

A.3 Convergence and computational efficiency

In this section, we complement the results of the numerical experiments of Section 5. First, we show the error graphics using the minimum time-step of $\Delta t = \frac{\Delta x}{8c_{max}}$, where c_{max} is the medium maximum velocity. These graphs account for all the methods discussed in Section 3 and several approximation degrees for the high-order schemes. Following that, we present the graphics of the estimation of Δt_{max}, the computational efficiency, and the memory utilization.

Based on Fig. 15, we observe an approximation error in all the numerical examples that do not decrease with the order of the method or with the selected method. This error is independent of the time integration strategy and is produced by the spatial discretization operator. While the dependence of the spatial error on the numerical experiment is weak, it is important to estimate it accurately for a reliable computation of Δ_{max}, as quantified in Table 2.

Table 2 Numerical error at a time instant in the physical domain produced by the spatial discretization

Numerical experiment	Spatial error	Error tolerance
Corner Model	$2.02 \cdot 10^{-7}$	$3.03 \cdot 10^{-7}$
Santos Basin	$5.55 \cdot 10^{-7}$	$8.33 \cdot 10^{-7}$
Marmousi	$6.62 \cdot 10^{-7}$	$9.93 \cdot 10^{-7}$
SEG/EAGE	$8.65 \cdot 10^{-7}$	$1.3 \cdot 10^{-6}$

Table 3 Numerical error utilizing the seismogram data produced by the spatial discretization

Numerical experiment	Seismogram error	Error tolerance
Corner Model	$2.92 \cdot 10^{-7}$	$4.38 \cdot 10^{-7}$
Santos Basin	$2.65 \cdot 10^{-6}$	$3.97 \cdot 10^{-6}$
Marmousi	$1.3 \cdot 10^{-6}$	$1.95 \cdot 10^{-6}$
SEG/EAGE	$4.2 \cdot 10^{-6}$	$6.3 \cdot 10^{-6}$

Fig. 15 Error at a time instant in the physical space achieved by each time integrator and several approximation degrees, for all the numerical experiments described in Section 5.1, using a time step size of $\Delta t = \frac{\Delta x}{8c_{max}}$. Regardless of the order of the method, there is an inferior limit for the error due to spatial discretization step-size size and scheme
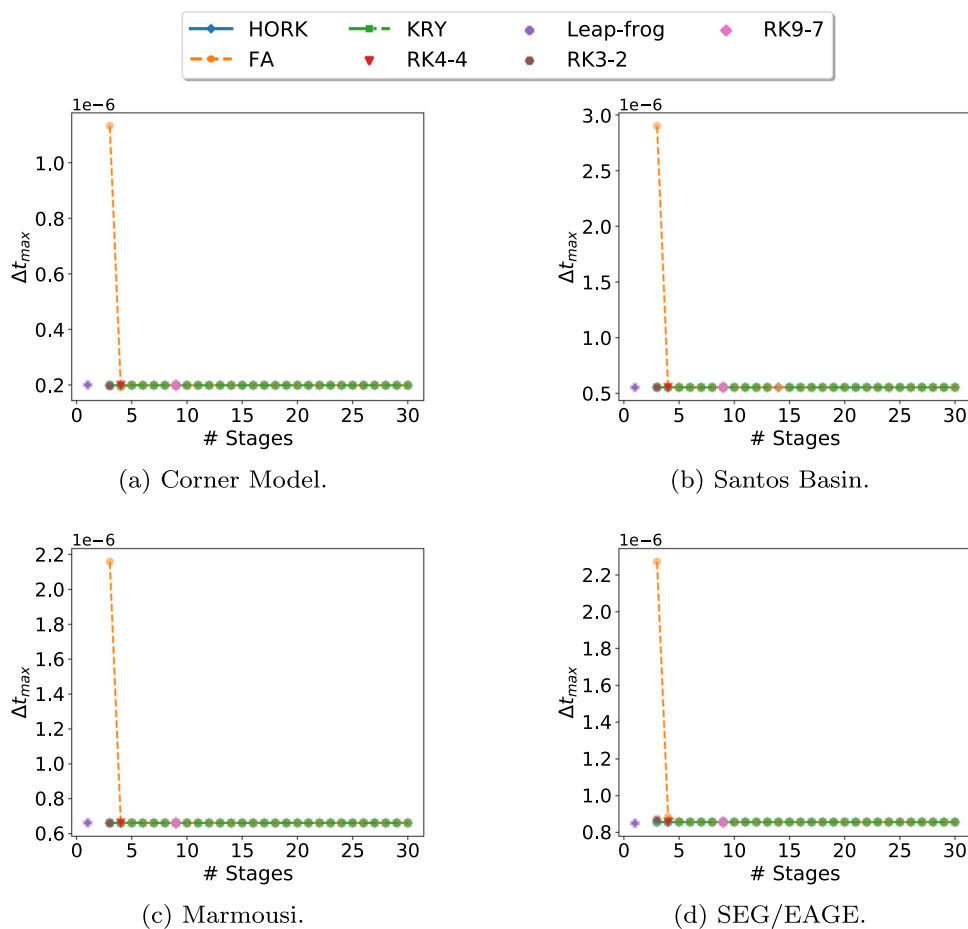

(a) Corner Model.

(b) Santos Basin.

(c) Marmousi.

(d) SEG/EAGE.

Table 2 contains two key columns of information. The first column, labeled "Spatial error", represents the error stemming from the spatial discretization. Meanwhile, the second column, labeled "Error tolerance", accounts for the error tolerance of 150% of the spatial error we defined for the numerical experiment.

For the minimum error using the seismogram data, we have the respective error graphics and tolerance for each numerical test (Fig. 16). Additionally, Table 3 summarizes the error values.
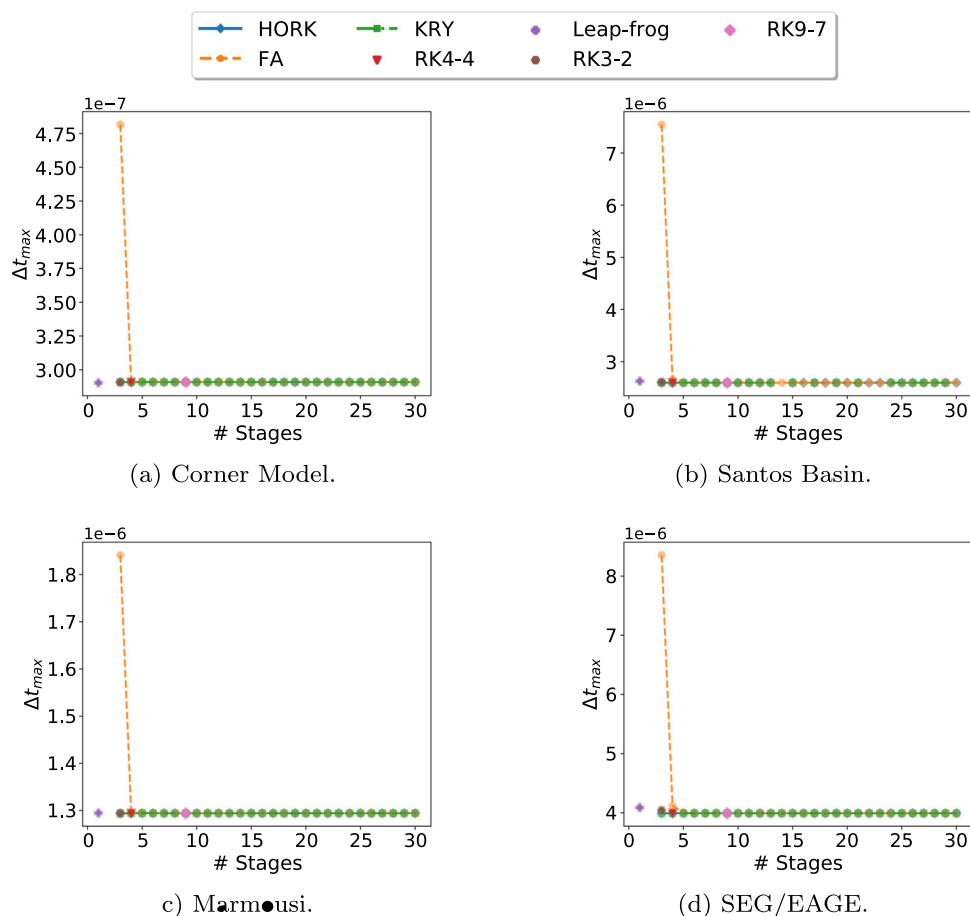
Fig. 16 Error using the seismogram data achieved by each time integrator and several approximation degrees, for all the numerical experiments described in Section 5.1, using a time step size of $\Delta t = \frac{\Delta x}{8c_{\max}}$. Regardless of the order of the method, there is an inferior limit for the error due to spatial discretization step-size size and scheme

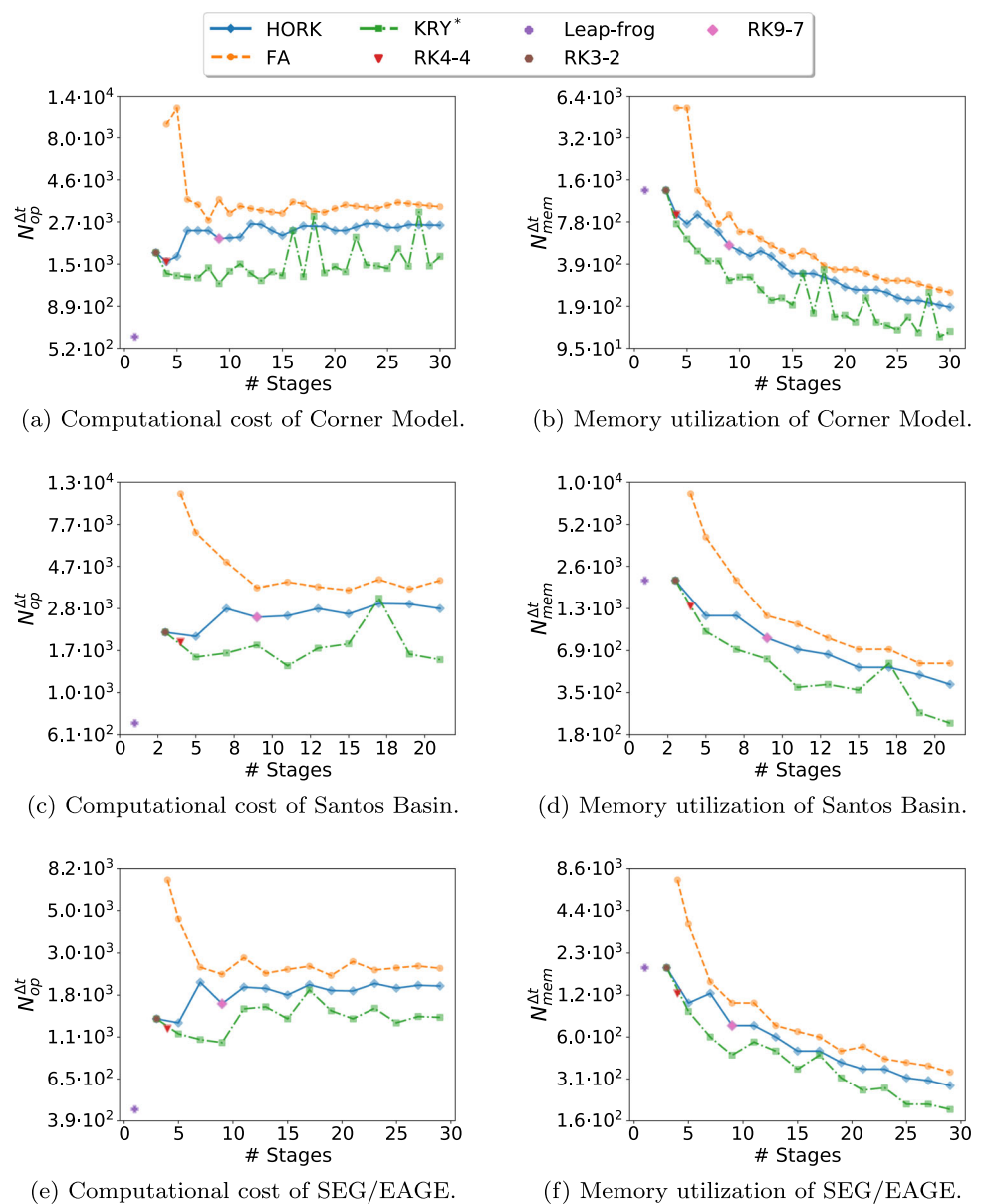### A.3.1 Computational efficiency and memory consumption

Figure 17 displays each time the integrator's computational cost and memory utilization for the numerical tests Corner Model, Santos Basin, and SEG/EAGE. Although there are some variations between the experiments, the general behavior remains consistent. High-order methods require significantly less memory; in some cases, they are competitive with low-order methods, such as the Leap-Frog scheme.

The relationship between the number of MVOs and the quantity of stored solution vectors concerning the polynomial degree is illustrated for the Corner Model (first line), Santos Basin (second line), and SEG/EAGE (third line) numerical tests. As the number of stages increases, there is a stabi-

lization in the number of computations, and memory usage decreases.

Fig. 17 Dependence of the number of MVOs and amount of stored solution vectors on the polynomial degree, for the Corner Model (first line), Santos Basin (second line), and SEG/EAGE (third line) numerical tests. As the number of stages increases, the number of computations stabilizes, and memory usage decreases. * Here we neglect the computational complexity of creating the Krylov subspaces

(a) Computational cost of Corner Model.

(b) Memory utilization of Corner Model.

(c) Computational cost of Santos Basin.

(d) Memory utilization of Santos Basin.

(e) Computational cost of SEG/EAGE.

(f) Memory utilization of SEG/EAGE.

References

1. Acebron, J.A.: A monte carlo method for computing the action of a matrix exponential on a vector. Appl. Math. Comput. **362**, 124545 (2019)

2. Al-Mohy, A.H., Higham, N.J.: A new scaling and squaring algorithm for the matrix exponential. SIAM J. Matrix Anal. Appl. **31**(3), 970–989 (2010)

3. Al-Mohy, A.H., Higham, N.J.: Computing the action of the matrix exponential, with an application to exponential integrators. SIAM J. Sci. Comput. **33**(2), 488–511 (2011)

4. Alappat, C., Hager, G., Schenk, O., Wellein, G.: Level-based blocking for sparse matrices: Sparse matrix-power-vector multiplication. IEEE Trans. Parallel Distrib. Syst. **34**(2), 581–597 (2022)

5. Alkhadhr, S., Liu, X., Almekkawy, M.: Modeling of the forward wave propagation using physics-informed neural networks. In: 2021 IEEE International Ultrasonics Symposium (IUS), pp. 1–4. IEEE, (2021)

6. Alonso, J.M., Ibáñez, J., Defez, E., Alonso-Jordá, P.: Euler polynomials for the matrix exponential approximation. J. Comput. Appl. Math. **425**, 115074 (2023)

7. Assi, H., Cobbold, R.S.: Compact second-order time-domain perfectly matched layer formulation for elastic wave propagation in two dimensions. Math. Mech. Solids. **22**(1), 20–37 (2017)

8. Bader, P., Blanes, S., Casas, F.: Computing the matrix exponential with an optimized Taylor polynomial approximation. Mathematics **7**(12), 1174 (2019)

9. Bergamaschi, L., Vianello, M.: Efficient computation of the exponential operator for large, sparse, symmetric matrices. Numer. Linear Algebr. **7**(1), 27–45 (2000)

10. Bergamaschi, L., Caliari, M., Vianello, M.: The relpm exponential integrator for fe discretizations of advection-diffusion equations. In: International Conference on Computational Science, pp. 434–442. Springer, (2004)

11. Brachet, M., Debreu, L., Eldred, C.: Comparison of exponential integrators and traditional time integration schemes for the shallow water equations. Appl. Numer. Math. (2022)

12. Burden, R.L, Faires, J.D, Burden, A.M.: Numerical analysis. Cengage learning (2015)

13. Burman, E., Duran, O., Ern, A.: Hybrid high-order methods for the acoustic wave equation in the time domain. Commun. Appl. Math. Comput. **4**(2), 597–633 (2022)

14. Butcher, J.C.: A history of runge-kutta methods. Appl. Numer. Math. **20**(3), 247–260 (1996)

15. Calvo, M., Franco, J., Montijano, J., Rández, L.: Explicit runge-kutta methods for initial value problems with oscillating solutions. J. Comput. Appl. Math. **76**(1–2), 195–212 (1996)

16. Chern, A.: A reflectionless discrete perfectly matched layer. J. Comput. Phys. **381**, 91–109 (2019)

17. Cohen, D., Dujardin, G.: Exponential integrators for nonlinear schrödinger equations with white noise dispersion. Stoch. Partial Differ. Equ.: Anal. Comput. **5**(4), 592–613 (2017)

18. Crouseilles, N., Einkemmer, L., Massot, J.: Exponential methods for solving hyperbolic problems with application to collisionless kinetic equations. J. Comput. Phys. **420**, 109688 (2020)

19. Deka, P.J., Einkemmer, L., Tokman, M.: Lexint: Package for exponential integrators employing leja interpolation. SoftwareX **21**, 101302 (2023)

20. Fornberg, B.: Generation of finite difference formulas on arbitrarily spaced grids. Math. Comput. **51**(184), 699–706 (1988)

21. Gallopoulos, E., Saad, Y.: Efficient solution of parabolic equations by krylov approximation methods. SIAM J. Sci. Stat. Comput. **13**(5), 1236–1264 (1992)

22. Gaudreault, S., Rainwater, G., Tokman, M.: Kiops: A fast adaptive krylov subspace solver for exponential integrators (vol 372, pg 236, 2018). J. Comput. Phys. **441** (2021)

23. Gottlieb, S., Gottlieb, L.A.J.: Strong stability preserving properties of runge-kutta time discretization methods for linear constant coefficient operators. J. Sci. Comput. **18**(1), 83–109 (2003)

24. Harold, R.: Ricker, ormsby, klander, butterworth – a choice of wavelets. Hi-Res Geoconsulting **19**(7), 8–9 (1994)

25. Hochbruck, M., Ostermann, A.: Exponential integrators. Acta Numer. **19**, 209–286 (2010)

26. Huber, D., Schreiber, M., Yang, D., Schulz, M.: Cache-aware matrix polynomials. In: Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part I 20, pp. 132–146. Springer, (2020)

27. Ikelle, L.T., Amundsen, L.: Introduction to petroleum seismology. Society of Exploration Geophysicists (2018)

28. Iyiola, O.S., Wade, B.A.: Exponential integrator methods for systems of non-linear space-fractional models with super-diffusion processes in pattern formation. Comput. Math. Appl. **75**(10), 3719–3736 (2018)

29. Jing, H., Chen, Y., Wang, J., Xue, W.: A highly efficient time-space-domain optimized method with lax-wendroff type time discretization for the scalar wave equation. J. Comput. Phys. **393**, 1–28 (2019)

30. Kole, J.: Solving seismic wave propagation in elastic media using the matrix exponential approach. Wave Motion **38**(4), 279–293 (2003)

31. Kosloff, D., Filho, A.Q., Tessmer, E., Behle, A.: Numerical solution of the acoustic and elastic wave equations by a new rapid expansion method 1. Geophys. Prospect. **37**(4), 383–394 (1989)

32. Kristek, J., Moczo, P., Archuleta, R.J.: Efficient methods to simulate planar free surface in the 3d 4th-order staggered-grid finite-difference schemes. Stud. Geophys. Geod. **46**, 355–381 (2002)

33. Kwon, S.B., Bathe, K.J., Noh, G.: An analysis of implicit time integration schemes for wave propagations. Comput. Struct. **230**, 106188 (2020)

34. Lawson, J.D.: Generalized runge-kutta processes for stable systems with large lipschitz constants. SIAM J. Numer. Anal. **4**(3), 372–380 (1967)

35. Lee, J.H.: Consistent transmitting boundaries for time-domain analyses of wave propagation in layered anisotropic waveguides. Int. J. Numer. Methods. Eng. **124**(8), 1883–1907 (2023)

36. Li, K., Liao, W.: An efficient and high accuracy finite-difference scheme for the acoustic wave equation in 3d heterogeneous media. J. Comput. Soc. Sci. **40**, 101063 (2020)

37. Liu, Y., Sen, M.K.: A new time-space domain high-order finite-difference method for the acoustic wave equation. J. Comput. Phys. **228**(23), 8779–8806 (2009)

38. Loffeld, J., Tokman, M.: Comparative performance of exponential, implicit, and explicit integrators for stiff systems of odes. J. Comput. Appl. Math. **241**, 45–67 (2013)

39. Lu, Y.Y.: Computing a matrix function for exponential integrators. J. Comput. Appl. Math. **161**(1), 203–216 (2003)

40. Miao, Z., Zhang, J.: Optimizing finite-difference scheme in multidirections on rectangular grids based on the minimum norm. Geophysics **87**(4), F41–F54 (2022)

41. Moczo, P., Kristek, J., Halada, L.: 3d fourth-order staggered-grid finite-difference schemes: Stability and grid dispersion. Bull. Seismol. Soc. Am. **90**(3), 587–603 (2000)

42. Moler, C., Van Loan, C.: Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. SIAM Rev. **45**(1), 3–49 (2003)

43. Mossaiby, F., Ghaderian, M., Rossi, R.: Implementation of a generalized exponential basis functions method for linear and non-linear problems. Int. J. Numer. Methods Eng. **105**, 221–240 (2015)

44. Niesen J, Wright W (2009) A krylov subspace algorithm for evaluating the φ-functions in exponential integrators. arXiv:0907.4631

45. Peixoto, P.S., Schreiber, M.: Semi-lagrangian exponential integration with application to the rotating shallow water equations. SIAM J. Sci. Comput. **41**(5), B903–B928 (2019)

46. Pototschnig, M., Niegemann, J., Tkeshelashvili, L., Busch, K.: Time-domain simulations of the nonlinear maxwell equations using operator-exponential methods. IEEE Trans. Antennas Propag. **57**(2), 475–483 (2009)

47. Ravelo, F.V., Peixoto, P.S., Schreiber, M.: An explicit exponential integrator based on faber polynomials and its application to seismic wave modeling. J. Sci. Comput. **98**(2), 32 (2024)

48. Robertsson, J.O., Blanch, J.O.: Numerical methods, finite difference. Encyclopedia Solid Earth Geophys. 1–9 (2020)

49. Ruud, B., Hestholm, S.: 2d surface topography boundary conditions in seismic wave modelling. Geophys. Prospect. **49**(4), 445–460 (2001)

50. Schmelzer, T., Trefethen, L.: Evaluating matrix functions for exponential integrators via carathéodory-fejér approximation and contour integrals. - -:– (2006)

51. Schreiber, M., Schaeffer, N., Loft, R.: Exponential integrators with parallel-in-time rational approximations for the shallow-water equations on the rotating sphere. Parallel Comput. **85**, 56–65 (2019)

52. Sidje, R.B.: Expokit: A software package for computing matrix exponentials. ACM Trans. Math. Softw. (TOMS) **24**(1), 130–156 (1998)

53. Strikwerda, J.C.: Finite difference schemes and partial differential equations. SIAM (2004)

54. Tago, J., Cruz-Atienza, V., Chaljub, E., Brossier, R., Coutant, O., Garambois, S., Prieux, V., Operto, S., Mercerat, D., Virieux, J., et al.: Modelling seismic wave propagation for geophysical imaging. In: Seismic waves-research and analysis, IntechOpen (2012)

55. Tal-Ezer, H., Kosloff, D., Koren, Z.: An accurate scheme for seismic forward modelling. Geophys. Prospect. **35**(5), 479–490 (1987)

56. Tessmer, E.: Using the rapid expansion method for accurate time-stepping in modeling and reverse-time migration. Geophysics **76**(4), S177–S185 (2011)

57. Thomas, J.W.: Numerical partial differential equations: finite difference methods, vol 22. Springer Science & Business Media (2013)

58. Walters, S., Forbes, L., Reading, A.: Analytic and numerical solutions to the seismic wave equation in continuous media. Proc. R. Soc. Lond. A. **476**(2243), 20200636 (2020)

59. Weber, I., Kreiss, G., Nazarov, M.: Stability analysis of high order methods for the wave equation. J. Comput. Appl. Math. **404**, 113900 (2022)

60. Wilcox, L.C., Stadler, G., Burstedde, C., Ghattas, O.: A high-order discontinuous galerkin method for wave propagation through coupled elastic-acoustic media. J. Comput. Phys. **229**(24), 9373–9396 (2010)

61. Zhang, X., Yang, D., Song, G.: A nearly analytic exponential time difference method for solving 2d seismic wave equations. Earthq. Sci. **27**(1), 57–77 (2014)

62. Zingg, D.W., Lomax, H., Jurgens, H.: High-accuracy finite-difference schemes for linear wave propagation. SIAM J. Sci. Comput. **17**(2), 328–346 (1996)