Available online at www.sciencedirect.com
SCIENCE DIRECT®
Wave Motion 38 (2003) 279–293
WAVE MOTION
www.elsevier.com/locate/wavemoti

Solving seismic wave propagation in elastic media using the matrix exponential approach

J.S. Kole

Centre for Theoretical Physics and Materials Science Centre, University of Groningen, Nijenborgh 4, NL-9747 AG Groningen, The Netherlands

Received 22 January 2002; received in revised form 28 April 2003; accepted 1 May 2003

Abstract

Three numerical algorithms are proposed to solve the time-dependent elastodynamic equations in elastic solids. All algorithms are based on approximating the solution of the equations, which can be written as a matrix exponential. By approximating the matrix exponential with a product formula, an unconditionally stable algorithm is derived that conserves the total elastic energy density. By expanding the matrix exponential in Chebyshev polynomials for a specific time instance, a so-called "one-step" algorithm is constructed that is very accurate with respect to the time integration. By formulating the conventional velocity-stress finite-difference time-domain (VS-FDTD) algorithm in matrix exponential form, the staggered-in-time nature can be removed by a small modification, and higher-order in time algorithms can be easily derived. For two different seismic events the accuracy of the algorithms is studied and compared with the result obtained by using the conventional VS-FDTD algorithm.

1. Introduction

An important aid in the understanding of wave propagation in inhomogeneous media is seismic forward modeling. In all but the simplest cases, an analytical solution of the elastodynamic equations is not available, and one must resort to numerical solutions. For this, two main strategies can be followed: one solves the elastodynamic equations either in the strong formulation, where the equations of motion and boundary conditions are written in differential form, or in the weak formulation, where the equations of motion are given in integral form. The latter formulation, implemented by finite element [1,2], spectral element [3,4] or finite integral methods [5,6], may be preferred to deal with complex geometries, or non-trivial free surface boundary conditions.

The first strategy (the strong formulation) is followed in the finite-difference approach that is based on solving either the first-order velocity-stress differential equations [7–9], or the second-order wave equation [10–12]. In the original formulation of the velocity-stress finite-difference time-domain (VS-FDTD) approach [7,8], space and time are both discretized using second-order finite differences. Many enhancements have been introduced to increase the accuracy and treatment of special boundary conditions. For example, spectral methods have been employed to increase the accuracy of the approximation of the spatial derivative operators [13–15]; a rotated staggered spatial grid

E-mail address: j.s.kole@phys.rug.nl (J.S. Kole).

0165-2125/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0165-2125(03)00065-9

has been developed to surmount some instability and lack of spatial accuracy issues [16]; polynomial expansions of the time evolution operator have been used to increase the accuracy and efficiency of the time integration process [14,17].

Although each specific method cited above offers its own advantages and drawbacks, none of these methods feature unconditional stability and/or exact energy conservation, which can be useful for long integration times, and high material contrast situations, where instabilities have been noticed [18]. The algorithms that will be introduced in this paper, address this issue.

Starting from the velocity-stress first-order differential equations, it can be shown (see Section 2) that the solution of these equations can be written as the matrix exponential of a skew-symmetric matrix. This constitutes an orthogonal transformation, conserving the total energy density. Guided by recent results regarding the numerical solution of the time-dependent Maxwell equations [19–24], where the underlying skew-symmetry of the equations of motion is exploited in a matrix exponential approach, we apply this framework to the current problem. In Section 3 this framework is briefly repeated for convenience and three algorithms are derived to solve the time-dependent elastodynamic equations. The incorporation of the presence of a source is described in Section 4. For some typical examples, the performance and efficiency of the algorithms is studied in Section 5, and the conclusions are summarized in Section 6.

2. Theory

In the absence of body forces, the linearized equation of momentum conservation reads [25]

$$\rho \frac{\partial^2}{\partial t^2} u_i = \sum_j \frac{\partial}{\partial x_j} \sigma_{ij}, \tag{1}$$

where ρ is the density, u_i the displacement field and σ_{ij} the stress field ($i = x, y, z$). It can be recast into a coupled first-order velocity-stress equation [7], yielding in matrix form

$$\frac{\partial}{\partial t} \begin{pmatrix} \boldsymbol{\sigma} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} 0 & -CD^{\mathrm{T}} \\ \frac{1}{\rho} D & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\sigma} \\ \mathbf{v} \end{pmatrix}. \tag{2}$$

Here, $\boldsymbol{\sigma} = (\sigma_{xx}, \sigma_{xy}, \sigma_{xz}, \sigma_{yx}, \sigma_{yy}, \sigma_{yz}, \sigma_{zx}, \sigma_{zy}, \sigma_{zz})^{\mathrm{T}}$, $\mathbf{v} = (v_x, v_y, v_z)^{\mathrm{T}}$ is the velocity field and D the matrix containing the spatial derivatives operators,

$$D = \begin{pmatrix} \frac{\partial}{\partial x} & \frac{1}{2}\frac{\partial}{\partial y} & \frac{1}{2}\frac{\partial}{\partial z} & \frac{1}{2}\frac{\partial}{\partial y} & 0 & 0 & \frac{1}{2}\frac{\partial}{\partial z} & 0 & 0 \\ 0 & \frac{1}{2}\frac{\partial}{\partial x} & 0 & \frac{1}{2}\frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{1}{2}\frac{\partial}{\partial z} & 0 & \frac{1}{2}\frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{1}{2}\frac{\partial}{\partial x} & 0 & 0 & \frac{1}{2}\frac{\partial}{\partial y} & \frac{1}{2}\frac{\partial}{\partial x} & \frac{1}{2}\frac{\partial}{\partial y} & \frac{\partial}{\partial z} \end{pmatrix}. \tag{3}$$

The stiffness tensor $C = \{C_{ijkl}\}$ relates the stress and strain,

$$\sigma_{ij} = C_{ijkl} e_{kl} \tag{4}$$

and is symmetric and positive definite for elastic solids.

Some important symmetries in matrix equation (2) can be made explicit by introducing the fields

$$\mathbf{w} = \sqrt{\rho}\mathbf{v} \tag{5}$$

and

$$\mathbf{s} = \frac{1}{\sqrt{C}}\boldsymbol{\sigma}. \tag{6}$$

The expression $\sqrt{C}$ is valid since C is symmetric and positive definite.

By definition, the length of $\psi \equiv (\mathbf{s}, \mathbf{w})^{\mathrm{T}}$, given by

$$\|\psi\|^2 = \langle\psi|\psi\rangle \equiv \int_V \psi^{\mathrm{T}}\psi \,\mathrm{d}\mathbf{r} = \int_V (\mathbf{w}^2 + \mathbf{s}^2)\,\mathrm{d}\mathbf{r} = \int_V (\rho\mathbf{v}^2 + \boldsymbol{\sigma}^{\mathrm{T}}C^{-1}\boldsymbol{\sigma})\,\mathrm{d}\mathbf{r} \tag{7}$$

is related to the elastic energy density

$$w \equiv \frac{1}{2}(\rho\mathbf{v}^2 + \boldsymbol{\sigma}^{\mathrm{T}}C^{-1}\boldsymbol{\sigma}) = \frac{1}{2}\left(\rho\mathbf{v}^2 + \sum_{ijkl} C_{ijkl}e_{ij}e_{kl}\right) \tag{8}$$

of the fields.

In terms of ψ, matrix equation (2) becomes

$$\frac{\partial}{\partial t}\psi = \begin{pmatrix} 0 & -\sqrt{C}D^{\mathrm{T}}\dfrac{1}{\sqrt{\rho}} \\ \dfrac{1}{\sqrt{\rho}}D\sqrt{C} & 0 \end{pmatrix}\psi \equiv \mathcal{H}\psi. \tag{9}$$

Using the symmetric properties of ρ and $\sqrt{C}$, one can prove that the matrix $\mathcal{H}$ is skew-symmetric

$$\mathcal{H}^{\mathrm{T}} = \begin{pmatrix} 0 & \left(\dfrac{1}{\sqrt{\rho}}D\sqrt{C}\right)^{\mathrm{T}} \\ -\left(\sqrt{C}D^{\mathrm{T}}\dfrac{1}{\sqrt{\rho}}\right)^{\mathrm{T}} & 0 \end{pmatrix} = \begin{pmatrix} 0 & \sqrt{C}D^{\mathrm{T}}\dfrac{1}{\sqrt{\rho}} \\ -\dfrac{1}{\sqrt{\rho}}D\sqrt{C} & 0 \end{pmatrix} = -\mathcal{H} \tag{10}$$

with respect to the inner product as defined in Eq. (7). The formal solution of Eq. (9) is given by

$$\psi(t) = \mathrm{e}^{t\mathcal{H}}\psi(0) \equiv \mathcal{U}(t)\psi(0), \tag{11}$$

where $\psi(0)$ represents the initial state of the fields and the operator $\mathcal{U}$ determines their time evolution. And, since $\mathcal{H}$ is skew-symmetric the time evolution operator, $\mathcal{U}$ is an orthogonal transformation:

$$\mathcal{U}(t)^{\mathrm{T}} = \mathcal{U}(-t) = \mathcal{U}^{-1}(t) = \mathrm{e}^{-t\mathcal{H}}, \tag{12}$$

and it follows that

$$\langle\mathcal{U}(t)\psi(0)|\mathcal{U}(t)\psi(0)\rangle = \langle\psi(t)|\psi(t)\rangle = \langle\psi(0)|\psi(0)\rangle. \tag{13}$$

Hence, the time evolution operator $\mathcal{U}(t)$ rotates the vector $\psi(t)$ without changing its length $\|\psi\|$. In physical terms, this means that the total energy density of the fields does not change with time, as can be expected on physical grounds [25].

In practice, the construction of a numerical algorithm requires to discretize space and time. During both these procedures, the skew-symmetry of $\mathcal{H}$ (during the spatial discretization) and the orthogonality of $\mathcal{U}$ (during the time integration) should be conserved. For the discretization of space, this requirement can be met by choosing a staggered spatial grid [8] and a central difference approximation for the spatial derivative. This yields a skew-symmetric matrix H for the discrete analogue of $\mathcal{H}$. A unit cell of the grid is shown in Fig. 1. The explicit form of H is derived in Appendix A, in the case of a two-dimensional isotropic elastic solid. Accordingly, the discrete analogue of $\psi(t)$ is given by vector $\Psi(t)$.

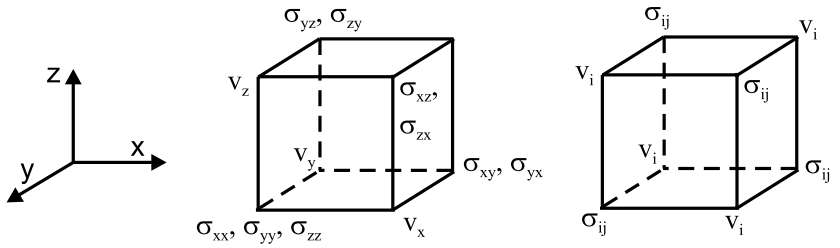

Fig. 1. Unit cell of the three-dimensional staggered grid onto which the continuous velocity and stress fields of the elastodynamic equations are mapped in order to conserve the skew-symmetry. Left: grid for elastic isotropic solids. Note: the Lamé constants λ and μ coincide with the stress field components, and the mass density is only defined on velocity field points. Right: grid for the general anisotropic case ($i, j = x, y, z$).

The continuous problem, defined by $\mathcal{H}$, is now translated to a lattice problem defined by H:

$$\Psi(t) = \exp(tH)\Psi(0) \equiv U(t)\Psi(0) \tag{14}$$

or, in the time-stepping approach, we have for a small time-step, τ

$$\Psi(t+\tau) = \exp(\tau H)\Psi(t) = U(\tau)\Psi(t). \tag{15}$$

At this point, we invoke three different strategies to perform the time integration, i.e. to approximate the matrix exponential $\exp(tH)$. Here, we closely follow the derivation of algorithms to solve the time-dependent Maxwell equations [19–24], where the problem to be solved is stated in a very similar form, although the underlying physics is different. The first algorithm is based on conserving the existing symmetries during the discretization of time, and is unconditionally stable. Here, the time integration is carried out by a time-stepping procedure. The second algorithm is based on approximating the solution itself for a particular time instance, by means of a Chebyshev expansion, and constitutes therefore a "one-step" algorithm. The last algorithm is based on recasting the original velocity-stress finite-difference algorithm into matrix exponential form. This allows to remove the staggered-in-time nature, and offers an elegant way to derive higher-order in time algorithms. The construction of the algorithms is briefly repeated in the next section.

3. The matrix exponential approach

3.1. Unconditionally stable algorithms

A sufficient condition for an algorithm to be unconditionally stable is that [26]

$$\|U(\tau)\Psi(t)\| \leq \|\Psi(t)\|. \tag{16}$$

Since $U(\tau)$ is an orthogonal transformation (see previous section), we have $\|U(\tau)\Psi(t)\| = \|\Psi(t)\|$, and it is sufficient to conserve the orthogonality of $U(t)$ for an approximation $\tilde{U}(t)$ to $U(t)$, in order to construct an unconditionally stable algorithm. One way to accomplish this is to make use of the Lie–Trotter–Suzuki formula [27,28] and generalizations thereof [29,30]. If the matrix H is decomposed, so that $H = \sum_{i=1}^{p} H_i$, then

$$U_1(\tau) = \mathrm{e}^{\tau H_1} \cdots \mathrm{e}^{\tau H_p} \tag{17}$$

is a first-order approximation to $U(\tau)$. More importantly, if each matrix H_i is skew-symmetric, then $U_1(\tau)$ is orthogonal by construction, and hence, algorithms based on $U_1(\tau)$, are unconditionally stable. Using the fact that

both $U(\tau)$ and $U_1(\tau)$ are orthogonal matrices, the error on $U_1(\tau)$ is subject to the upper bound [31]

$$\|U(\tau) - U_1(\tau)\| \leq \frac{\tau^2}{2} \sum_{i<j}^{p} \|[H_i, H_j]\|, \tag{18}$$

where $[H_i, H_j] = H_i H_j - H_j H_i$.

In Appendix A, the decomposition of H is carried out for two-dimensional isotropic elastic solids, for which $p = 12$, and it is shown that each matrix H_i is block diagonal. The computation of the matrix exponential of a block diagonal matrix H_i can be performed efficiently, as it is equal to the block diagonal matrix of the matrix exponentials of the individual blocks. Therefore, the numerical calculation of $\mathrm{e}^{\tau H_i}$ reduces to the calculation of matrix exponentials of 2×2 matrices, which are rotations.

In practice, implementation of the first-order algorithm is all that is required to construct higher-order algorithms. This is due to the fact that in the product formula approach, the accuracy of an approximation can be improved in a systematic way by reusing lower order approximations, without changing the fundamental symmetries. For example, the orthogonal matrix

$$U_2(\tau) = U_1\left(\frac{-\tau}{2}\right)^{\mathrm{T}} U_1\left(\frac{\tau}{2}\right) = \mathrm{e}^{\tau H_p/2} \cdots \mathrm{e}^{\tau H_1/2}\, \mathrm{e}^{\tau H_1/2} \cdots \mathrm{e}^{\tau H_p/2} \tag{19}$$

is a second-order approximation to $U(\tau)$ [29,30]. A particularly useful fourth-order approximation (applied in, for example, [19,20,28–40]) is given by [29]

$$U_4(\tau) = U_2(a\tau) U_2(a\tau) U_2((1-4a)\tau) U_2(a\tau) U_2(a\tau), \tag{20}$$

where $a = 1/(4 - 4^{1/3})$.

3.2. One-step algorithm

A well-known alternative for time-stepping is to use Chebyshev polynomials to construct approximations to time-evolution operators [41–45]. This approach has also been successfully applied to the problem of seismic wave propagation [14,17]. However, the main differences between these implementations and the algorithm explained below, besides the spatial discretization (which is based here on a central difference approximation, instead of a spectral method), is that in the present case, matrix H is explicitly anti-symmetrized, which gives rise to purely imaginary eigenvalues [46]. This is an important property to justify the validity of the expansion. In the derivation of the current algorithm, we follow [21–23], a recent implementation of the Chebyshev algorithm to solve electromagnetic wave propagation.

The basic idea is to expand the time evolution matrix $U(t) = \exp(tH)$ for a specific time instance t in matrix valued Chebyshev polynomials on the domain of eigenvalues of H, which lies entirely on the imaginary axis since H is skew-symmetric. For proper application of the expansion, the domain of eigenvalues is rescaled to $[-1, 1]$, by considering the matrix $B = -\mathrm{i}H/\|H\|_1$, where $\|H\|_1$ denotes the 1-norm of the matrix. It is given by $\|H\|_1 \equiv \max_j \sum_i |H_{ij}|$, see [46], and is easy to compute since the matrix H is sparse. Operating on state $\Psi(0)$, the expansion becomes

$$\Psi(t) = \exp(tH)\Psi(0) = \exp(\mathrm{i}zB)\Psi(0) = \left[J_0(z) I + 2 \sum_{n=1}^{\infty} J_n(z) \tilde{T}_n(B) \right] \Psi(0), \tag{21}$$

where I is the identity matrix, $z = t\|H\|$, J_n is the nth-order Bessel function and $\tilde{T}_n(B) = \mathrm{i}^n T_n(B)$ are the modified Chebyshev polynomials, defined by the recursion relation

$$\tilde{T}_0(B)\Psi(0) = \Psi(0), \tag{22a}$$

$$\tilde{T}_1(B)\Psi(0) = \mathrm{i}B\Psi(0), \tag{22b}$$

$$\tilde{T}_{n+1}(B)\Psi(0) = 2\mathrm{i}B\tilde{T}_n(B)\Psi(0) + \tilde{T}_{n-1}(B)\Psi(0) \quad \text{for } n \geq 1. \tag{22c}$$

Due to the fact that the matrix B is purely imaginary, it follows from the above recursion relation (22a)–(22c) that $\tilde{T}_n(B)\Psi(0)$ and thus $\Psi(t)$ will be real valued and no complex arithmetic is involved, as should be the case.

In practice, the summation in Eq. (21) will be truncated at some expansion index m. This number depends on the value of z, since the amplitude of the coefficients $J_n(z)$ decrease exponentially for $n > z$; this is explained in more detail in [21–23]. Consequently, the computation of one time-step amounts to carrying out m repetitions of recursion relation Eq. (22a)–(22c) to obtain the final state. This is a simple procedure: only the multiplication of a vector with a sparse matrix and the summation of vectors are involved.

3.3. A modified VS-FDTD algorithm

In Section 3.1, it was shown that in the product formula formalism, higher-order in time algorithms can be constructed by reusing the lower order algorithms. This elegant technique to increase the accuracy of time integration can also be applied to FDTD algorithms (in case of the Maxwell equations, see [24]), and hence the conventional VS-FDTD algorithm, if it is recast into an exponent operator form.

The update equations of the VS-FDTD [7,8] algorithm can be written as

$$\begin{pmatrix} \boldsymbol{\sigma}(t+\tau) \\ \mathbf{v}(t+\frac{1}{2}\tau) \end{pmatrix} = (I+\tau A)(I+\tau B) \begin{pmatrix} \boldsymbol{\sigma}(t) \\ \mathbf{v}(t-\frac{1}{2}\tau) \end{pmatrix} \equiv U_1(\tau) \begin{pmatrix} \boldsymbol{\sigma}(t) \\ \mathbf{v}(t-\frac{1}{2}\tau) \end{pmatrix}, \tag{23}$$

where

$$A = \begin{pmatrix} 0 & -CD^{\mathrm{T}} \\ 0 & 0 \end{pmatrix}, \tag{24}$$

and

$$B = \begin{pmatrix} 0 & 0 \\ \frac{1}{\rho}D & 0 \end{pmatrix}. \tag{25}$$

Since $A^2 = B^2 = 0$, the time evolution operator $U_1(\tau)$ is equal to

$$U_1(\tau) = \exp(\tau A)\exp(\tau B) \tag{26}$$

and second-order accurate in time operating on fields that are defined staggered-in-time. However, if $U_1(\tau)$ is interpreted as an approximation to the operator $\exp(\tau A + \tau B)$, working on fields non-staggered-in-time, it is a first-order approximation. Furthermore, since the time evolution operator is now expressed in matrix exponential form, the accuracy can be increased by the same procedure as was used for the unconditionally stable algorithms (cf. Eqs. (19) and (20)). Therefore, the operator

$$U_2(\tau) = \exp(\tfrac{1}{2}\tau A)\exp(\tau B)\exp(\tfrac{1}{2}\tau A) \tag{27}$$

constitutes a second-order approximation to the exact time evolution operator. And similarly, a fourth-order algorithm can be derived, see Eq. (20).

So, by introducing a small modification to the original VS-FDTD algorithm, that would require a minimal change in existing numerical codes, the staggered-in-time nature is removed, and higher-order in time algorithms are derived.

4. Sources

In the presence of an explosive initial condition, or another time-dependent body force, the equation of motion reads

$$\frac{\partial}{\partial t}\psi(t) = \mathcal{H}\psi(t) + \phi(t), \tag{28}$$

where the time-dependent source is denoted by the term $\phi(t)$. The formal solution is given by

$$\psi(t) = \exp(t\mathcal{H})\psi(0) + \int_0^t \exp((t-u)\mathcal{H})\phi(u)\,\mathrm{d}u. \tag{29}$$

In the time-stepping approach (e.g. the unconditionally stable algorithms), the source term—if its time dependence is known explicitly—can be integrated for each time-step. For example, a standard quadrature formula can be employed to compute the integral over u, like the fourth-order accurate Simpson rule [47]

$$\int_t^{t+\tau} \mathrm{e}^{(t+\tau-u)\mathcal{H}}\phi(u)\,\mathrm{d}u \approx \frac{\tau}{6}\left(\mathrm{e}^{\tau H}\phi(t) + 4\,\mathrm{e}^{\tau H/2}\phi\left(t+\frac{\tau}{2}\right) + \phi(t+\tau)\right). \tag{30}$$

In case of the one-step algorithm, this approach of incorporating the source term is not efficient, as for each value of $t-u$ the recursion (22a)–(22c) would have to be performed, and a different route is taken. Instead, each of the two terms on the right-hand side of Eq. (29) is expanded in modified Chebyshev polynomials separately. The expansion of the first term is discussed in Section 3.2. For the second term, the source-term integral, the procedure is carried out here for a source with Gaussian time dependence,

$$\phi(t) = f(t)S(\mathbf{r}) = \exp(-\alpha(t-t_0)^2)S(\mathbf{r}), \tag{31}$$

where $S(\mathbf{r})$ denotes the spatial dependence of the source.

The source term

$$h(t, t_0, \alpha, H) = \int_0^t \mathrm{d}u\,\mathrm{e}^{(t-u)H} f(u) \tag{32}$$

is expanded in modified Chebyshev polynomials,

$$h(t, t_0, \alpha, H)S(\mathbf{r}) = \left(\frac{1}{2}b_0 I + \sum_{k=0}^{K} b_k \tilde{T}_k\right) S(\mathbf{r}), \tag{33}$$

where the expansion coefficients are given by

$$b_k = \mathrm{i}^{-k}\frac{2}{\pi}\int_0^{\pi} h(t, t_0, \alpha, \cos\theta)\,\cos k\theta\,\mathrm{d}\theta. \tag{34}$$

The replacement of H by $\cos\theta$ emphasizes that H should be normalized such that all eigenvalues lie in the range $[-1, 1]$. We proceed by evaluating $h(t, t_0, \alpha, H)$. After substitution of $z = \|H\|_1$, $z_0 = t_0\|H\|_1$, $\beta = \alpha/\|H\|_1^2$ and $x = -\mathrm{i}H/\|H\|_1$ to normalize the matrix H, we obtain

$$h(z, z_0, \beta, x) = \frac{1}{2}\sqrt{\frac{\pi}{\alpha}}\exp\left((z-z_0)\mathrm{i}x - \frac{x^2}{4\beta}\right) \times \left[\mathrm{erf}\left(z_0\sqrt{\beta} - \frac{\mathrm{i}x}{2\sqrt{\beta}}\right) + \mathrm{erf}\left((z-z_0)\sqrt{\beta} + \frac{\mathrm{i}x}{2\sqrt{\beta}}\right)\right]. \tag{35}$$

Now we put $x = \cos\theta$ and the remaining integral over θ in Eq. (34) is computed by a Fast Fourier transformation:

$$b_k = 2\mathrm{i}^{-k}\sum_{n=0}^{N-1} \mathrm{e}^{2\pi\mathrm{i}nk/N} h\left(z, z_0, \beta, \cos\frac{2\pi n}{N}\right). \tag{36}$$

The derivation of the Chebyshev expansion coefficients for a source defined by equation

$$g(t) = \frac{\partial}{\partial t} f(t) = -2\alpha(t - t_0)\exp(-\alpha(t - t_0)^2) \tag{37}$$

is very similar, and will not be treated explicitly here.

5. Results

The performance and accuracy of the algorithms introduced in the previous sections is studied by comparing the results with a reference solution generated by the one-step algorithm, denoted by $\hat{\Psi}(t)$. This choice is motivated by the fact that the latter, considering the time integration, produces numerically exact results [14,17,42]. Furthermore, there are rigorous bounds on the error of the unconditionally stable algorithm (cf. Eq. (18)): in the presence of a source $\phi(t)$, the difference between the exact solution $\Psi(t)$ and the approximate solution $\tilde{\Psi}(t)$, obtained by using the fourth-order unconditionally stable algorithm is bounded by [24]

$$\|\Psi(t) - \tilde{\Psi}(t)\| \leq c_4 t\tau^4 \left(\|\Psi(0)\| + \int_0^t \mathrm{d}u \|\phi(u)\| \right), \tag{38}$$

where c_4 is a constant. For the difference between the exact solution and the solution obtained by using the one-step algorithm, we can write using the triangle inequality

$$\|\Psi(t) - \hat{\Psi}(t)\| \leq \|\Psi(t) - \tilde{\Psi}(t)\| + \|\tilde{\Psi}(t) - \hat{\Psi}(t)\|. \tag{39}$$

Using Eqs. (38) and (39) and the fact that the difference $\|\tilde{\Psi}(t) - \hat{\Psi}(t)\|$ vanishes with τ^4, as we will show below, we can be confident that the one-step algorithm indeed produces numerically exact results. This justifies to define the error as $\|\tilde{\Psi}(t) - \hat{\Psi}(t)\|/\|\hat{\Psi}(t)\|$.

The performance of the following algorithms is considered: the original VS-FDTD algorithm, denoted by Vir; the unconditionally stable algorithms, denoted by LTS-2 and LTS-4, for respectively, second- and fourth-order accuracy in time; and the non-staggered-in-time modified VS-FDTD algorithm, denoted by VNS-2 and VNS-4, depending on the accuracy in time.

Consider a rectangular system consisting of two different materials, displayed in Fig. 2. This system is also studied in [8,10], and proved to be a good testing situation for the performance of an algorithm solving the elastodynamic equations. The time evolution of the velocity and stress fields is computed using an explosion as initial condition, modeled by Eq. (37), up to $t = 6$ s, with the six different algorithms. In Fig. 3, the kinetic-energy density distribution is shown for two different time instances.

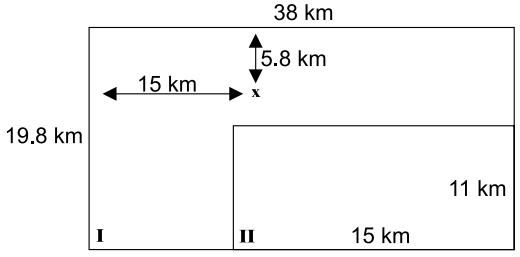

Fig. 2. The corner-edge system, consisting of two different materials. The system size and location of the source (**x**) are indicated in the picture. At the top, a free-surface boundary condition is imposed, the other boundaries are rigid. The overall density is $\rho = 2500\,\mathrm{kg\,m^{-3}}$, and the mesh size is $\delta = 100$ m. In the bulk material (I), the wave velocities are $v_p = 6$ and $v_s = 2\,\mathrm{km\,s^{-1}}$, whereas in the inner material (II), they are $v_p = 9$ and $v_s = 3\,\mathrm{km\,s^{-1}}$. The source excites the s_{xx} and s_{yy} stress fields with time dependence $g(t)$ from Eq. (37) and parameters $\alpha = 40$ and $t_0 = 1.5$ s.

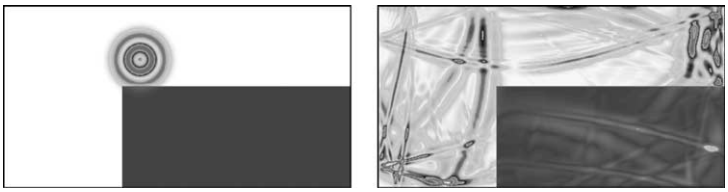

Fig. 3. Two snapshots of the kinetic-energy density distribution of the corner-edge system of Fig. 2. Left: state at $t = 1.8$ s, right: state at $t = 6.0$ s.

Table 1
Error as function of time-step for all time-stepping algorithms, for the system defined in Fig. 2[a]

Time-step (s)	Vir	VNS-2	VNS-4	LTS-2	LTS-4
5×10^{-2}	∞	∞	∞	3.0×10^{0}	3.1×10^{0}
5×10^{-3}	1.7×10^{-2}	1.7×10^{-2}	4.1×10^{-6}	5.9×10^{-1}	1.6×10^{-4}
5×10^{-4}	1.7×10^{-4}	1.7×10^{-4}	4.1×10^{-10}	6.1×10^{-3}	1.6×10^{-8}
5×10^{-5}	1.7×10^{-6}	1.7×10^{-6}	2.0×10^{-13}	6.1×10^{-5}	2.0×10^{-10}
5×10^{-6}	1.7×10^{-8}	1.7×10^{-8}	2.8×10^{-13}	6.1×10^{-7}	–

[a] An infinite symbol (∞) denotes that the algorithm was not stable, whereas in one case (–) the computation was not performed.

For the corner-edge system, the errors are listed in Table 1, and are shown in Fig. 4. The error in the staggered-in-time Vir algorithm is determined by averaging the error in the kinetic and potential energy density at, respectively, the final time instance and the final time instance shifted by half a time-step. The time shifting procedure is carried out by the Chebyshev algorithm and also applied to prepare the initial condition. For the largest time-step, the VS-FDTD algorithms (Vir, VNS-2, VNS-4) are unstable. This can be expected, since the maximum time-step is limited by the

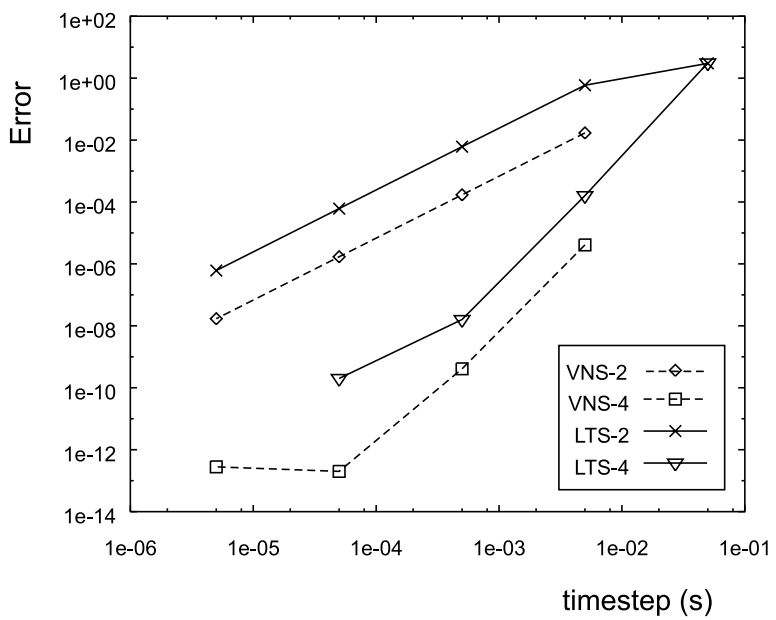

Fig. 4. Error as function of time-step for all time-stepping algorithms, for the system defined in Fig. 2, using the data from Table 1. Note: the graphs for Vir and VNS-2 overlap completely and therefore both are represented by the graph of VNS-2.

Table 2
Error as function of time-step for all time-stepping algorithms[a]

Time-step (s)	Vir	VNS-2	VNS-4	LTS-2	LTS-4
5×10^{-2}	∞	∞	∞	1.2×10^{0}	5.8×10^{-2}
5×10^{-3}	1.3×10^{-1}	1.3×10^{-1}	1.6×10^{-5}	5.4×10^{-2}	1.3×10^{-5}
5×10^{-4}	1.3×10^{-3}	1.3×10^{-3}	1.6×10^{-9}	5.4×10^{-4}	1.4×10^{-9}
5×10^{-5}	1.3×10^{-5}	1.3×10^{-5}	1.6×10^{-13}	5.4×10^{-6}	1.7×10^{-11}
5×10^{-6}	1.3×10^{-7}	1.3×10^{-7}	1.1×10^{-13}	5.4×10^{-8}	1.7×10^{-10}

[a] In this case, the one-step algorithm employs $m = 321$ expansion terms. The system measures $L_x = L_y = 10$, with a mesh of $\delta = 0.1$, and the material parameters ρ, λ, μ vary randomly in space with values distributed randomly in the interval $[1, 3]$. The error is determined at $t = 3$ (all quantities are expressed here in dimensionless units).

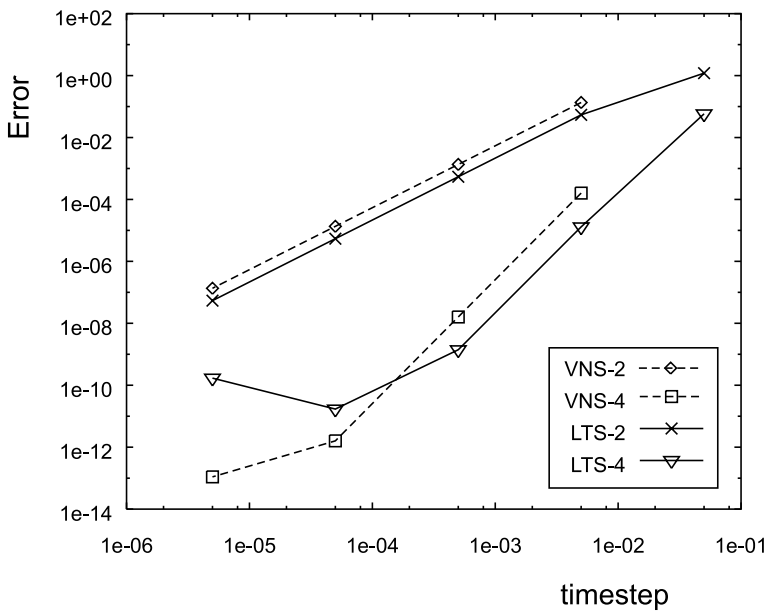

Fig. 5. Error as function of time-step for all time-stepping algorithms, for a random medium and random initial conditions, using the data from Table 2. Note: the graphs for Vir and VNS-2 overlap completely and therefore both are represented by the graph of VNS-2.

largest velocity, the v_p velocity, and the mesh size, through the Courant limit [8]

$$\tau < \frac{\delta}{v_p \sqrt{2}}. \tag{40}$$

Furthermore, from Table 1 it is clear that for all algorithms the error scales according to the order of accuracy in time. We also see that for the corner-edge system, the VS-FDTD algorithms perform much better than the energy-conserving LTS algorithms, as long as the time-step is smaller than the Courant limit. For time-steps larger than the Courant limit, the LTS algorithms (LTS-2 and LTS-4) are stable, although the error does not (yet) scale according to the order of accuracy in time.[1] For very accurate results (errors below 10^{-10}), the number of operations the achieve this accuracy becomes so large that the error does not scale systematic anymore.

With respect to the efficiency of the algorithms, we note that the number of matrix-vector operations W, necessary to perform one time-step, is 1 for the Vir algorithm, 1.5 for the second-order VNS-2 and LTS-2 algorithms, and

[1] This behavior is also noticed when unconditionally stable algorithms are used to solve the time-dependent Maxwell equations, see [20].

10 for the fourth-order VNS-4 and LTS-4 algorithms. For the specific example here, the corner-edge system, the one-step algorithm employs $m = 1514$ expansion terms. At $t = 6$ and $\tau = 0.005$, the VNS-2 algorithm already uses more (namely 1800) matrix-vector operations. Therefore, we draw the conclusion that one-step algorithm should be preferred to be used to solve the time evolution for this problem. Note that in general, the choice of which algorithm to use depends heavily on which degree of error is acceptable. In this specific case, there are values for τ for which the VNS-2 algorithm uses less matrix-vector operations than the one-step algorithm, but then the error will be larger than the error for $\tau = 0.005$, and maybe unacceptably high. On the other hand, one VNS-2 or LTS-2 matrix-vector operation is carried out (in practice) faster than one Chebyshev recursion iteration, although this may depend on the actual implementation.

It is important to note that the initial condition plays an important role in the error of the solution produced by a specific algorithm. From the results of the corner-edge system, one might draw the conclusion that the VS-FDTD algorithms achieve better results than the LTS algorithms for all systems. This is not true. In Table 2, the error is listed as a function of time-step for all algorithms, as compared with the one-step algorithm, for a system consisting of a random medium (also studied in, for example, [48]) and starting from a random initial condition. The results are also shown in Fig. 5. From the table and the figure, it is clear that in this case, the LTS family of algorithms perform better than the VS-FDTD algorithms. Again we see that the error scales according to the order of accuracy in time, except for very small errors (below 10^{-10}). Especially for the LTS-4 algorithm, we see that accumulation of rounding errors, due to large number of operations, increases the error.

6. Conclusions

In this paper, we have introduced algorithms to solve the time-dependent elastodynamic equations, based on a matrix exponential approach. The conservation of the underlying skew-symmetry of the first-order partial differential equations while discretizing the spatial operators and fields, offers a sound starting point to expand the time evolution operator in Chebyshev polynomials. The resulting one-step algorithm is accurate up to machine precision, and this statement is justified by rigorous bounds on the error of the unconditionally stable algorithms. The latter class of algorithms proved particularly useful if the total energy should be conserved or if a random initial condition is used.

Finally, the original VS-FDTD algorithm is modified by recasting it into an exponent operator form. In this new formulation, the staggered-in-time nature is removed and higher-order in time algorithms are derived, based on the lower-order algorithms. Existing VS-FDTD codes can be modified with minor effort to benefit from these advantages.

In this paper, only free and rigid boundary conditions are considered. Future research is aimed at incorporating absorbing boundary conditions, and the presence of visco-elastic materials. These circumstances will give rise to negative real eigenvalues in the time evolution matrix. Early study indicates that incorporating ABCs into the unconditionally stable time-stepping methods require minimal change, and these changes will not affect stability, as the eigenmodes will be attenuated. However, for the one-step method, a different approach must be taken, for example, by choosing a different polynomial basis that accounts for the presence of real negative eigenvalues, see [17].

More sophisticated discretization schemes, conserving the skew-symmetry, can be easily incorporated [20], and do not require conceptual changes.

Acknowledgements

The author acknowledges H. De Raedt, M.T. Figge and K. Michielsen for useful discussions. This work is partially supported by the Dutch ‘Stichting Nationale Computer Faciliteiten’ (NCF).

Appendix A. Discretization and decomposition of $\mathcal{H}$ for two-dimensional isotropic elastic solids

Here it is shown how the matrix

$$\mathcal{H} = \begin{pmatrix} 0 & -\sqrt{C}D^{\mathrm{T}}\dfrac{1}{\sqrt{\rho}} \\ \dfrac{1}{\sqrt{\rho}}D\sqrt{C} & 0 \end{pmatrix} \tag{A.1}$$

is discretized conserving its skew-symmetric properties, for two-dimensional isotropic elastic solids. In two-dimensional P-SV wave propagation, no dependency upon y is assumed. For isotropic elastic solids, the stiffness matrix is given in terms of the Lamé coefficients λ and μ by [25]

$$C = \begin{pmatrix} \lambda+2\mu & \lambda & 0 \\ \lambda & \lambda+2\mu & 0 \\ 0 & 0 & \mu \end{pmatrix}, \tag{A.2}$$

in the basis $\boldsymbol{\sigma} = (\sigma_{xx}, \sigma_{zz}, \sigma_{xz})^{\mathrm{T}}$. In the skew-symmetric basis, using the variables $\mathbf{w} = \sqrt{\rho}\mathbf{v}$ and $\mathbf{s} = C^{-1/2}\boldsymbol{\sigma}$, one needs $\sqrt{C}$, which reads

$$\sqrt{C} = \begin{pmatrix} \alpha & \beta & 0 \\ \beta & \alpha & 0 \\ 0 & 0 & \sqrt{\mu} \end{pmatrix}, \tag{A.3}$$

where

$$\alpha = \frac{1}{\sqrt{2}}(\sqrt{\lambda+\mu}+\sqrt{\mu}), \tag{A.4}$$

$$\beta = \frac{1}{\sqrt{2}}(\sqrt{\lambda+\mu}-\sqrt{\mu}). \tag{A.5}$$

This gives for the explicit form of the matrix $\mathcal{H}$

$$\mathcal{H} = \left(\begin{array}{ccc|cc} 0 & 0 & 0 & \alpha\frac{\partial}{\partial x}\frac{1}{\sqrt{\rho}} & \beta\frac{\partial}{\partial z}\frac{1}{\sqrt{\rho}} \\ 0 & 0 & 0 & \beta\frac{\partial}{\partial x}\frac{1}{\sqrt{\rho}} & \alpha\frac{\partial}{\partial z}\frac{1}{\sqrt{\rho}} \\ 0 & 0 & 0 & \sqrt{\mu}\frac{\partial}{\partial z}\frac{1}{\sqrt{\rho}} & \sqrt{\mu}\frac{\partial}{\partial x}\frac{1}{\sqrt{\rho}} \\ \hline \frac{1}{\sqrt{\rho}}\frac{\partial}{\partial x}\alpha & \frac{1}{\sqrt{\rho}}\frac{\partial}{\partial x}\beta & \frac{1}{\sqrt{\rho}}\frac{\partial}{\partial z}\sqrt{\mu} & 0 & 0 \\ \frac{1}{\sqrt{\rho}}\frac{\partial}{\partial z}\beta & \frac{1}{\sqrt{\rho}}\frac{\partial}{\partial z}\alpha & \frac{1}{\sqrt{\rho}}\frac{\partial}{\partial x}\sqrt{\mu} & 0 & 0 \end{array}\right), \tag{A.6}$$

in the basis $\psi = (s_{xx}, s_{zz}, s_{xz}, w_x, w_z)^{\mathrm{T}}$. The discrete analogue of ψ, the vector Ψ, is obtained by mapping the fields onto the two-dimensional staggered grid [8] that is shown in Fig. 6. We adopt the convention that the s_{xx} and s_{zz} stress fields are located in the $(1, 1)$ corner of the grid. The values of the discretized fields f are related to their continuous counterparts g by

$$f(i, j, t) = g(\tfrac{1}{2}i\delta, \tfrac{1}{2}j\delta, t). \tag{A.7}$$

Therefore, using the unit vector $\mathbf{e}(i, j, k)$, fields within the vector Ψ can be indexed on the grid by

$$\Psi(i, j, s_{xx}, t) \equiv \mathbf{e}^{\mathrm{T}}_{i,j,s_{xx}}\Psi(t) \equiv s_{xx}(i, j, t) \tag{A.8}$$

and analogous equations apply for indexing the other stress and velocity fields within Ψ.

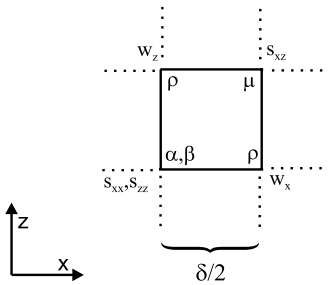

Fig. 6. Unit cell of the two-dimensional staggered grid onto which the continuous velocity and stress fields of the elastodynamic equations are mapped in order to conserve the skew-symmetry.

Due to the staggered nature of the grid and the choice of the origin, the s_{xx} and s_{zz} stress fields are only defined on the $x =$ odd and $z =$ odd lattice points. Similarly, the w_x and w_z velocity fields are defined at, respectively, $x =$ even/$z =$ odd and $x =$ odd/$z =$ even lattice points, and the s_{xz} stress field is given at the $x =$ even and $z =$ even grid entries. Note that all for simplicity of notation the fields are indexed on the full grid, despite the fact that they are not defined on each point.

It is assumed that the total number of lattice points in each direction is odd, and also that the boundary is located at the first and last rows/columns of the grid. The free or rigid boundary conditions themselves are implemented by excluding the field points that are located at the boundary and should remain zero during the time integration.

Using this grid and the central-difference approximation to the spatial derivative, we obtain the spatially discretized analogue of Eq. (A.6), for example

$$\frac{\partial}{\partial t}s_{xx}(i,\,j,\,t) = \alpha(i,\,j)\frac{1}{\delta}\left[\frac{w_x(i+1,\,j,\,t)}{\sqrt{\rho(i+1,\,j)}} - \frac{w_x(i-1,\,j,\,t)}{\sqrt{\rho(i-1,\,j)}}\right] + \beta(i,\,j)\frac{1}{\delta}\left[\frac{w_z(i,\,j+1,\,t)}{\sqrt{\rho(i,\,j+1)}} - \frac{w_z(i,\,j-1,\,t)}{\sqrt{\rho(i,\,j-1)}}\right] \tag{A.9}$$

and

$$\begin{aligned}\frac{\partial}{\partial t}w_x(i,\,j,\,t) = {} & \frac{\alpha(i+1,\,j)s_{xx}(i+1,\,j,\,t) - \alpha(i-1,\,j)s_{xx}(i-1,\,j,\,t)}{\delta\sqrt{\rho(i,\,j)}} \\ & + \frac{\beta(i+1,\,j)s_{zz}(i+1,\,j,\,t) - \beta(i-1,\,j)s_{zz}(i-1,\,j,\,t)}{\delta\sqrt{\rho(i,\,j)}} \\ & + \frac{\sqrt{\mu(i,\,j+1)}s_{xz}(i,\,j+1,\,t) - \sqrt{\mu(i,\,j-1)}s_{xz}(i,\,j-1,\,t)}{\delta\sqrt{\rho(i,\,j)}}.\end{aligned} \tag{A.10}$$

Similar equations hold for s_{zz}, s_{xz} and w_z. Using this notation, the matrix H can be decomposed into

$$H = H^{(x,s_{xx},w_x)} + H^{(x,s_{zz},w_x)} + H^{(x,s_{xz},w_z)} + H^{(z,s_{xx},w_z)} + H^{(z,s_{zz},w_z)} + H^{(z,s_{xz},w_x)}, \tag{A.11}$$

where, for instance, the explicit form of $H^{(x,s_{xx},w_x)}$ is given by

$$\begin{aligned}H^{(x,s_{xx},w_x)} = {} & \sum_{j=1}^{n_y'}\sum_{i=1}^{n_x-2'}\frac{\alpha(i+1,\,j)}{\delta\sqrt{\rho(i,\,j)}}[\mathbf{e}_{i,\,j,\,s_{xx}}\,\mathbf{e}_{i+1,\,j,\,w_x}^{\mathrm{T}} - \mathbf{e}_{i+1,\,j,\,w_x}\,\mathbf{e}_{i,\,j,\,s_{xx}}^{\mathrm{T}}] \\ & + \sum_{j=1}^{n_y'}\sum_{i=2}^{n_x-1'}\frac{\alpha(i,\,j)}{\delta\sqrt{\rho(i+1,\,j)}}[\mathbf{e}_{i,\,j,\,s_{xx}}\,\mathbf{e}_{i+1,\,j,\,w_x}^{\mathrm{T}} - \mathbf{e}_{i+1,\,j,\,w_x}\,\mathbf{e}_{i,\,j,\,s_{xx}}^{\mathrm{T}}]\end{aligned} \tag{A.12}$$

$$H^{(x,s_{xx},w_x)} = H_1^{(x,s_{xx},w_x)} + H_2^{(x,s_{xx},w_x)}. \tag{A.13}$$

Here, the prime in the summation indicates that the summation index is increased with strides of two. It is easy to convince oneself that the matrices $H_1^{(x,s_{xx},w_x)}$ and $H_2^{(x,s_{xx},w_x)}$ are block diagonal and skew-symmetric. The other matrices in Eq. (A.11) have similar explicit forms and can also be decomposed into block diagonal parts. Therefore, a first-order approximation to the matrix exponent, as given by Eq. (17), reads

$$\begin{aligned}\exp(\tau H) = {} & \exp(\tau H_1^{(x,s_{xx},w_x)}) \exp(\tau H_2^{(x,s_{xx},w_x)}) \exp(\tau H_1^{(x,s_{zz},w_x)}) \times \exp(\tau H_2^{(x,s_{zz},w_x)}) \exp(\tau H_1^{(x,s_{xz},w_z)}) \\ & \times \exp(\tau H_2^{(x,s_{xz},w_z)}) \times \exp(\tau H_1^{(z,s_{xx},w_z)}) \exp(\tau H_2^{(z,s_{xx},w_z)}) \exp(\tau H_1^{(z,s_{zz},w_z)}) \times \exp(\tau H_2^{(z,s_{zz},w_z)}) \\ & \times \exp(\tau H_1^{(z,s_{xz},w_x)}) \exp(\tau H_2^{(z,s_{xz},w_x)}) + \mathcal{O}(\tau^2). \end{aligned} \tag{A.14}$$

References

[1] J. Lysmer, L. Drake, A finite element method for seismology, Methods in Computational Physics, Academic Press, New York, 1972.
[2] H. Boa, et al., Large-scale simulation of elastic wave propagation in heterogenous media on parallel computers, Comp. Meth. Appl. Mech. Eng. 152 (1998) 85–102.
[3] A.T. Patera, A spectral method for fluid dynamics: laminar flow in a channel expansion, J. Comput. Phys. 54 (1984) 468–488.
[4] D. Komatitsch, Méthodes spectrales et éléments spectraux pour l'équation de l'élastodynamique 2D et 3D en milieu hétérog lsquo;ene, Ph.D. Thesis, Institute de Physique du Globe de Paris, Paris, 1997.
[5] P. Fellinger, et al., Numerical modeling of elastic wave propagation and scattering with EFIT—elastodynamic finite integration technique, Wave Motion 21 (1995) 47–66.
[6] G.V. Konyukh, Y.V. Kritsov, B.G. Mikhailenko, Numerical–analytical algorithm of seismic wave propagation in inhomogeneous media, Appl. Math. Lett. 11 (1998) 99–104.
[7] J. Virieux, SH-wave propagation in heterogeneous media: velocity-stress finite difference method, Geophysics 49 (1984) 1933–1957.
[8] J. Virieux, P-SV wave propagation in heterogeneous media: velocity-stress finite difference method, Geophysics 51 (1986) 889–901.
[9] A.R. Levander, Fourth-order finite-difference P-SV seismograms, Geophysics 53 (1988) 1425–1436.
[10] K.R. Kelly, et al., Synthetic seismograms: a finite difference approach, Geophysics 41 (1976) 2–27.
[11] M.A. Dablain, The application of high-order differencing to the scalar wave equation, Geophysics 51 (1987) 54–66.
[12] H. Igel, P. Mora, B. Riollet, Anisotropic wave propagation through finite-difference grids, Geophysics 60 (1995) 1203–1216.
[13] D. Kosloff, E. Baysal, Forward modeling by a Fourier method, Geophysics 47 (1982) 1402–1412.
[14] H. Tal-Ezer, D. Kosloff, Z. Koren, An accurate scheme for seismic forward modelling, Geophys. Prosp. 35 (1987) 479–490.
[15] H. Tal-Ezer, D. Kosloff, An modified pseudospectral method with an O(N^{-1}) time step restriction, J. Comp. Phys. 104 (1993) 457–469.
[16] E.H. Saenger, N. Gold, S.A. Shapiro, Modeling the propagation of elastic waves using a modified finite-difference grid, Wave Motion 31 (2000) 77–92.
[17] H. Tal-Ezer, J.M. Carcione, D. Kosloff, An accurate and efficient scheme for wave propagation in linear viscoelastic media, Geophysics 55 (1990) 1366–1379.
[18] I. Opršal, J. Zahradník, From unstable to stable seismic modeling by finite-difference method, Phys. Chem. Earth A 24 (1999) 247–252.
[19] J.S. Kole, M.T. Figge, H. De Raedt, Unconditionally stable methods to solve the time-dependent Maxwell equations, Phys. Rev. E 64 (2001) 066705-1–066705-14.
[20] J.S. Kole, M.T. Figge, H. De Raedt, Higher-order conditionally stable methods to solve the time-dependent Maxwell equations, Phys. Rev. E 65 (2002) 066705-1–066705-12.
[21] H. De Raedt, K. Michielsen, J.S. Kole, M.T. Figge, Solving the Maxwell equations by the Chebyshev method: a one-step finite-difference time-domain algorithm, IEEE Transactions on Antennas and Propagation, in press.
[22] H. De Raedt, J.S. Kole, K. Michielsen, M.T. Figge, Solving the Maxwell equations by the Chebyshev method: a one-step finite-difference time-domain algorithm. http://arxiv.org/abs/physics/0208060.
[23] H. De Raedt, J.S. Kole, K. Michielsen, M.T. Figge, One-step Algorithm to solve the time-dependent Maxwell equations, Phys. Rev. E 67 (2003) 056706-1–056706-12.
[24] H. De Raedt, J.S. Kole, K. Michielsen, M.T. Figge, Numerical methods for solving the time-dependent Maxwell equations. http://arxiv.org/abs/physics/02010035.
[25] L.D. Landau, E.M. Lifshitz, Theory of Elasticity, Pergamon Press, Oxford, 1986.
[26] G.D. Smith, Numerical Solution of Partial Differential Equations, Clarendon Press, Oxford, 1985.
[27] H.F. Trotter, On the product of semi-groups of operators, Proc. Am. Math. Soc. 10 (1959) 545–551.
[28] M. Suzuki, S. Miyashita, A. Kuroda, New method of Monte Carlo simulations and phenomenological theory, Prog. Theor. Phys. 58 (1977) 1377–1387.
[29] M. Suzuki, Decomposition formulas of exponential operators and Lie exponentials with some applications to quantum physics, J. Math. Phys. 26 (1985) 601–612;
M. Suzuki, General theory of fractal path integrals with applications to many-body theories and statistical physics, J. Math. Phys. 32 (1991) 400–407.

[30] H. De Raedt, B. De Raedt, Applications of the generalized Trotter formula, Phys. Rev. A 28 (1983) 3575–3580.
[31] H. De Raedt, Product formula algorithms for solving the time dependent Schrödinger equation, Comp. Phys. Rep. 7 (1987) 1–72.
[32] A.J. Chorin, T.J.R. Hughes, M.F. McCracken, J.E. Marsden, Product formulas and numerical algorithms, Comm. Pure Appl. Math. 31 (1978) 205–256.
[33] H. Kobayashi, N. Hatano, M. Suzuki, Study of correction terms for higher-order decompositions of exponential operators, Physica A 211 (1994) 234–254.
[34] H. De Raedt, K. Michielsen, Algorithm to solve the time-dependent Schrödinger equation for a charged particle in an inhomogeneous magnetic field: application to the Aharonov–Bohn effect, Comp. Phys. 8 (1994) 600–607.
[35] A. Rouhi, J. Wright, Spectral implementation of a new operator splitting method for solving partial differential equations, Comp. Phys. 9 (1995) 554–563.
[36] B.A. Shadwick, W.F. Buell, Unitary integration: a numerical technique preserving the structure of the quantum Liouville equation, Phys. Rev. Lett. 79 (1997) 5189–5193.
[37] M. Krech, A. Bunker, D.P. Landau, Fast spin dynamics algorithms for classical spin systems, Comp. Phys. Comm. 111 (1998) 1–13.
[38] P. Tran, Solving the time-dependent Schrödinger equation: suppression of reflection from the grid boundary with a filtered split-operator approach, Phys. Rev. E 58 (1998) 8049–8051.
[39] K. Michielsen, H. De Raedt, J. Przeslawski, N. Garcia, Computer simulation of time-resolved optical imaging of objects hidden in turbid media, Phys. Rep. 304 (1998) 89–144.
[40] H. De Raedt, A.H. Hams, K. Michielsen, K. De Raedt, Quantum computer emulator, Comp. Phys. Comm. 132 (2000) 1–20.
[41] C. Leforestier, R.H. Bisseling, et al., A comparison of different propagation schemes for the time-dependent Schrödinger equation, J. Comp. Phys. 94 (1991) 59–80.
[42] H. Tal-Ezer, Spectral methods in time for hyperbolic equations, SIAM. J. Numer. Anal. 23 (1986) 11–26.
[43] Y.L. Loh, S.N. Taraskin, S.R. Elliot, Fast time-evolution method for dynamical systems, Phys. Rev. Lett. 84 (2000) 2290–2293.
[44] T. Itaka, et al., Calculating the linear response functions of noninteracting electrons with a time-dependent Schrödinger equation, Phys. Rev. E 56 (1997) 1222–1229.
[45] R.N. Silver, H. Röder, Calculation of densities of states and spectral functions by Chebyshev recursion and maximum entropy, Phys. Rev. E 56 (1997) 4822–4829.
[46] J.H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.
[47] K.E. Atkinson, An Introduction to Numerical Analysis, Wiley, New York, 1989.
[48] G. Kneip, C. Kerner, Accurate and efficient seismic modeling in random media, Geophysics 58 (1993) 576–588.