

The Gautschi time stepping scheme for edge finite element discretizations of the Maxwell equations

M.A. Botchev ^{*,1}, D. Harutyunyan ², J.J.W. van der Vegt ¹

Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Received 19 April 2005; received in revised form 5 November 2005; accepted 2 January 2006
Available online 10 March 2006

Abstract

For the time integration of edge finite element discretizations of the three-dimensional Maxwell equations, we consider the Gautschi cosine scheme where the action of the matrix function is approximated by a Krylov subspace method. First, for the space-discretized edge finite element Maxwell equations, the dispersion error of this scheme is analyzed in detail and compared to that of two conventional schemes. Second, we show that the scheme can be implemented in such a way that a higher accuracy can be achieved within less computational time (as compared to other implicit schemes). We also analyzed the error made in the Krylov subspace matrix function evaluations. Although the new scheme is unconditionally stable, it is explicit in structure: as an explicit scheme, it requires only the solution of linear systems with the mass matrix.
© 2006 Elsevier Inc. All rights reserved.

Keywords: Maxwell equations; Gautschi cosine scheme; Dispersion analysis; Edge elements; Staggered leap frog scheme; Krylov subspace; Arnoldi process

1. Introduction

This paper deals with the numerical solution of the time dependent Maxwell equations. In particular, we are interested in time integration of the three-dimensional Maxwell equations discretized in space by Nedelec's edge finite elements [28,29]. Nedelec's edge and face elements have a number of attractive properties (as e.g. automatic satisfaction of the proper continuity requirements across the boundaries between different materials) and are a standard tool in the numerical treatment of the Maxwell equations [25]. We emphasize, however, that the time integration techniques presented in this paper are applicable to any space-discretized second order wave equation(s).

^{*} Corresponding author. Tel.: +31 53 489 3396; fax: +31 53 489 4833.

E-mail addresses: m.a.botchev@math.utwente.nl (M.A. Botchev), d.harutyunyan@math.utwente.nl (D. Harutyunyan), j.j.w.vandervegt@math.utwente.nl (J.J.W. van der Vegt).

¹ This research was supported by the Dutch government through the national program BSIK: knowledge and research capacity, in the ICT project BRICKS (<http://www.bsik-bricks.nl>), theme MSV1.

² Supported by NWO (Netherlands Organization for Scientific Research).

Many time stepping schemes exist for the time integration of the space-discretized Maxwell equations [41,12,24,22,23,5,19,20]. Often the time step in these schemes is restricted either due to stability restrictions or accuracy requirements, e.g. to resolve the waves. In practice, however, one often would like to have a step size free from stability restrictions since on nonuniform finite element meshes or in inhomogeneous media this restriction can be much more stringent than the wave resolution requirements. The need for better stability motivated the creation of a number of unconditionally stable schemes which proved successful in the finite element framework [12,24]. Stable time stepping schemes for the Maxwell equations have been also of importance in connection with finite difference spatial discretizations [22,23,5,19,20]. A scheme proposed by Gautschi [11] has recently received attention in the literature for the solution of second order highly oscillatory ODEs [18,17,15]. This scheme contains a matrix function, is exact for linear equations with constant inhomogeneity and thus unconditionally stable. In each time step the product of a matrix function with a given vector can be computed by Krylov subspace methods [37,6,21,32,7,16,8,18,36]. The time error of the scheme is of second order uniformly in the frequencies [17] and this allows to choose time steps larger than the smallest wave length.

In this paper we show that, using Krylov subspace techniques, the Gautschi cosine scheme can be efficiently implemented for the three-dimensional Maxwell equations discretized in space by edge elements. This yields a Gautschi–Krylov cosine scheme which proves to be very competitive, in terms of accuracy and CPU time, as compared to other implicit time-stable schemes for the time integration of the Maxwell equations.

Several authors study the dispersion properties of the discretized Maxwell equations. For the two-dimensional Maxwell equations discretized with the first order edge finite elements, Monk and Parrot compare dispersion properties of several conventional schemes [26]. A thorough analysis for three-dimensional problems with different boundary conditions on an unstructured tetrahedral mesh is carried out in [27]. For the dispersive properties of the higher order edge elements we refer to the paper of Ainsworth [1]. Dispersion properties of several high order time integration schemes for transient wave equations are considered by Cohen in [4]. In this paper the attractive properties of the new scheme are confirmed by a dispersion analysis done for the first order edge finite elements. For comparison purposes, the dispersion analysis is also presented for two other schemes, the conventional time-explicit leap frog scheme and an unconditionally stable scheme of Lee, Lee and Cangellaris often referred to as the Newmark β -scheme (in the sequel, the LLC scheme) [12,24].

To achieve high computational efficiency, it is crucial for the new Gautschi–Krylov scheme to properly choose the Krylov subspace dimension every time the action of the matrix function is computed. We propose a new simple strategy for controlling the Krylov subspace dimension.

The paper is organized as follows: Section 2 presents the Maxwell equations and their weak formulation, in Section 3 the Gautschi cosine scheme and two other time stepping schemes are described, the Krylov subspace error in the Gautschi–Krylov scheme and the stability of the scheme are analyzed in Section 4, and dispersion errors of the three schemes are investigated in Section 5. Finally, in the last section we demonstrate numerical results of a comparison of the schemes.

2. Maxwell equations

Consider the time-dependent Maxwell equations on a bounded lossless domain $\Omega \subset \mathbb{R}^3$:

$$\partial_t \mathbf{D}_s = \nabla \times \mathbf{H}_s - \mathbf{J}_s, \quad (2.1)$$

$$\partial_t \mathbf{B}_s = -\nabla \times \mathbf{E}_s, \quad (2.2)$$

$$\nabla \cdot \mathbf{D}_s = \rho_s, \quad (2.3)$$

$$\nabla \cdot \mathbf{B}_s = 0, \quad (2.4)$$

where \mathbf{E}_s and \mathbf{H}_s (\mathbf{D}_s and \mathbf{B}_s) are electric and magnetic fields (respectively, the electric and the magnetic flux densities). Furthermore, \mathbf{J}_s and ρ_s denote, respectively, the electric current and charge density (the latter is a space and time dependent function). The subscript s indicates that the SI units are used. Assume that the following boundary and initial conditions are given:

$$(\mathbf{n} \times \mathbf{E}_s)|_\Gamma = 0, \quad (2.5)$$

$$\mathbf{E}_s|_{t_s=0} = \bar{\mathbf{E}}_0, \quad \mathbf{H}_s|_{t_s=0} = \bar{\mathbf{H}}_0, \quad (2.6)$$

where \mathbf{n} is the outward normal vector to the domain boundary $\Gamma = \partial\Omega$. The following constitutive relations hold:

$$\mathbf{D}_s = \epsilon \mathbf{E}_s, \quad \mathbf{B}_s = \mu \mathbf{H}_s, \quad (2.7)$$

where the dielectric permittivity $\epsilon (= \epsilon_0 \epsilon_r)$ and the magnetic permeability $\mu (= \mu_0 \mu_r)$ are assumed to be space dependent tensors. The free space dielectric permittivity and magnetic permeability are defined by ϵ_0 and μ_0 , respectively. The dimensionless tensors ϵ_r and μ_r are material dependent and called relative permittivity and relative permeability, respectively.

2.1. Dimensionless Maxwell equations

To avoid problems with floating point arithmetic when working with very large numbers, we apply the following space and time scaling:

$$x = \frac{x_s}{L}, \quad t = \frac{c_0}{L} t_s, \quad (2.8)$$

where L is a reference length (expressed in meters), and $c_0 = (\epsilon_0 \mu_0)^{-1/2} \approx 3 \times 10^8$ m/s is the speed of light in vacuum. The scaling for y_s and z_s is done similarly to x_s . Furthermore, we normalize the fields as

$$\mathbf{E}_s(\mathbf{x}_s, t_s) = \frac{\tilde{H}_0}{Z_0} \mathbf{E}(\mathbf{x}, t), \quad \mathbf{H}_s(\mathbf{x}_s, t_s) = \tilde{H}_0 \mathbf{H}(\mathbf{x}, t), \quad \mathbf{J}_s(\mathbf{x}_s, t_s) = \frac{\tilde{H}_0}{L} \mathbf{J}(\mathbf{x}, t), \quad (2.9)$$

where $\mathbf{x}_s = (x_s, y_s, z_s)$, $\mathbf{x} = (x, y, z)$, $Z_0 = \sqrt{\mu_0/\epsilon_0}$ [Ohm] is the free space intrinsic impedance, and \tilde{H}_0 is a reference magnetic field strength [A/m]. Eqs. (2.1), (2.2) and constitutive relations (2.7) written for the scaled quantities yield the following dimensionless Maxwell equations:

$$\epsilon_r \partial_t \mathbf{E} = \nabla \times \mathbf{H} - \mathbf{J}, \quad (2.10)$$

$$\mu_r \partial_t \mathbf{H} = -\nabla \times \mathbf{E}. \quad (2.11)$$

Since the given boundary conditions are homogeneous, the dimensionless normalization leaves them unchanged.

By differentiating (2.10) in time and taking curl of (2.11), we eliminate \mathbf{H} from the system (2.10), (2.11) and obtain a second-order hyperbolic partial differential equation for \mathbf{E} ,

$$\epsilon_r \partial_{tt} \mathbf{E} + \nabla \times (\mu_r^{-1} \nabla \times \mathbf{E}) = -\partial_t \mathbf{J}. \quad (2.12)$$

Using (2.10) we obtain the initial condition for the derivative of \mathbf{E} :

$$\partial_t \mathbf{E}(x, 0) = \epsilon_r^{-1} (-\mathbf{J}(\mathbf{x}, 0) + \nabla \times \mathbf{H}(\mathbf{x}, 0)). \quad (2.13)$$

2.2. Weak formulation and finite element discretization

Defining the space

$$H_0(\text{curl}, \Omega) = \{\mathbf{u} \in L_2(\Omega)^3 \mid \nabla \times \mathbf{u} \in L_2(\Omega)^3, (\mathbf{n} \times \mathbf{u})|_\Gamma = 0\},$$

we arrive at the following Galerkin weak formulation of (2.12):

Find $\mathbf{E} \in H_0(\text{curl}, \Omega)$ such that $\forall \mathbf{w} \in H_0(\text{curl}, \Omega)$,

$$\partial_{tt}(\epsilon_r \mathbf{E}, \mathbf{w}) + (\mu_r^{-1} \nabla \times \mathbf{E}, \nabla \times \mathbf{w}) = -(\partial_t \mathbf{J}, \mathbf{w}). \quad (2.14)$$

Next, we introduce a tessellation of Ω (a hexahedral or tetrahedral mesh) with N internal edges and denote by \mathcal{W}_h the space of Nedelec's first order edge basis functions:

$$W_h = \text{span}\{\mathbf{w}_j(x) \mid \text{all internal edges } j = 1, \dots, N\},$$

where each basis function $\mathbf{w}_j(x)$ is defined with respect to the edge j as a linear polynomial such that [28,25]:

$$\alpha_i(\mathbf{w}_j) \equiv \int_{\text{edge } i} \mathbf{w}_j \cdot \mathbf{t}_i \, da = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j, \end{cases}$$

where $\alpha_i(\mathbf{w}_j)$ are the degrees of freedom associated with the edges and \mathbf{t}_i is the unit tangent vector along the edge i . The electric field \mathbf{E} is then approximated as

$$\mathbf{E} \approx \mathbf{E}_h = \sum_{j=1}^N e_j(t) \mathbf{w}_j.$$

The discretized version of (2.14) then reads:

Find $\mathbf{E}_h \in W_h$, such that $\forall \mathbf{W} \in W_h$,

$$\partial_t(\epsilon_r \mathbf{E}_h, \mathbf{W}) + (\mu_r^{-1} \nabla \times \mathbf{E}_h, \nabla \times \mathbf{W}) = -(\partial_t \mathbf{J}, \mathbf{W}). \tag{2.15}$$

Denoting by $\mathbf{e}(t)$ a vector function with the entries $e_j(t)$, we can write (2.15) in a matrix form as a system of ordinary differential equations (ODEs)

$$M_\epsilon \mathbf{e}'' + A_\mu \mathbf{e} = \mathbf{j}(t) \tag{2.16}$$

with

$$(M_\epsilon)_{ij} = (\epsilon_r \mathbf{w}_i, \mathbf{w}_j), \quad (\mathbf{j}(t))_i = -(\partial_t \mathbf{J}, \mathbf{w}_i), \quad (A_\mu)_{ij} = (\mu_r^{-1} \nabla \times \mathbf{w}_i, \nabla \times \mathbf{w}_j). \tag{2.17}$$

3. Time stepping schemes

In this section the Gautschi cosine time-stepping scheme is presented, along with two other conventional time-stepping schemes which we use for comparison with the Gautschi scheme. The first of the two schemes is the explicit staggered leap frog scheme and the second one is an implicit scheme designed for finite element discretizations of the Maxwell equations [24,12].

3.1. Leap frog scheme

The two-step staggered leap frog scheme for the semidiscrete Maxwell equations (2.16) reads

$$M_\epsilon \frac{\mathbf{e}^{n+1} - 2\mathbf{e}^n + \mathbf{e}^{n-1}}{\tau^2} + A_\mu \mathbf{e}^n = \mathbf{j}^n, \tag{3.1}$$

where τ is the time step size and the superscripts refer to the time levels $t_n = n\tau$. The scheme can be written in the form

$$M_\epsilon \mathbf{e}^{n+1} + (\tau^2 A_\mu - 2M_\epsilon) \mathbf{e}^n + M_\epsilon \mathbf{e}^{n-1} = \tau^2 \mathbf{j}^n. \tag{3.2}$$

If the matrices M_ϵ and A_μ are Hermitian, M_ϵ is positive definite and A_μ is positive semidefinite then the leap frog scheme is stable for

$$\tau^2 \leq \frac{4}{\lambda_{\max}},$$

where λ_{\max} is the maximum eigenvalue of the matrix $M_\epsilon^{-1} A_\mu$ (see Appendix A).

The computational work of the scheme per time step mainly consists of one matrix–vector multiplication with the matrix $M_\epsilon^{-1} A_\mu$. This can be efficiently done with the help of a sparse LU factorization of M_ϵ (see Remark 3.2.1 in Section 3.2.1).

3.2. Gautschi cosine scheme

3.2.1. Reduction of the semidiscrete Maxwell equations to the normal form

We first transform the ODE system (2.16) into the form

$$\mathbf{y}'' + \tilde{A}_{\epsilon,\mu}\mathbf{y} = \mathbf{f}(t), \quad (3.3)$$

which we call the normal form. Computing a sparse LU factorization of M_ϵ (see Remark 3.2.1), we obtain

$$M_\epsilon = L_\epsilon U_\epsilon.$$

Note that if ϵ is a symmetric positive definite tensor then the matrix M_ϵ is symmetric positive definite, too, and we can take $U_\epsilon = L_\epsilon^T$ (Cholesky factorization).

It is easy to see that the semidiscrete Maxwell equations (2.16) can be transformed to the form (3.3) with $\tilde{A}_{\epsilon,\mu}$ and \mathbf{y} defined in one of the following ways:

$$\tilde{A}_{\epsilon,\mu} = U_\epsilon^{-1} L_\epsilon^{-1} A_\mu, \quad \mathbf{y} = \mathbf{e}, \quad \mathbf{f} = U_\epsilon^{-1} L_\epsilon^{-1} \mathbf{j}, \quad (3.4)$$

$$\tilde{A}_{\epsilon,\mu} = L_\epsilon^{-1} A_\mu U_\epsilon^{-1}, \quad \mathbf{y} = U_\epsilon \mathbf{e}, \quad \mathbf{f} = L_\epsilon^{-1} \mathbf{j}, \quad (3.5)$$

$$\tilde{A}_{\epsilon,\mu} = A_\mu U_\epsilon^{-1} L_\epsilon^{-1}, \quad \mathbf{y} = L_\epsilon U_\epsilon \mathbf{e}, \quad \mathbf{f} = \mathbf{j}, \quad (3.6)$$

where the inverse matrices will normally never be computed explicitly (see Remark 3.2.1). Since we call (3.3) the normal form of (2.16), the transformations (3.4)–(3.6) can, respectively, be called the left, two-sided and right normalizations.

Remark 3.2.1. For the used edge finite element discretization a sparse LU (or Cholesky) factorization of the mass matrix can usually be efficiently computed even on fine meshes (at least, if the mesh is not too distorted [31] which is a general requirement for edge finite elements). In practice, matrices L_ϵ^{-1} and U_ϵ^{-1} will usually not be computed explicitly. This would be expensive because the inverses will not be sparse in general. In fact, we will only need to compute the action of the matrices L_ϵ^{-1} and U_ϵ^{-1} on a given vector and this can be done by solving a linear system with L_ϵ or U_ϵ , as is usually done in preconditioning (see e.g. Chapter 13.1 in [38] or Chapter 3.1 in [2]).

Note that the sparse LU factorization of the mass matrix is also required for explicit schemes. The factorization is performed only once for the complete time integration.

3.3. Formulation of Gautschi cosine scheme

We formulate the Gautschi cosine time stepping scheme [11,17] for an ODE system in the normal form (3.3):

$$\mathbf{y}^{n+1} - 2\mathbf{y}^n + \mathbf{y}^{n-1} = \tau^2 \psi(\tau^2 \tilde{A}_{\epsilon,\mu}) (-\tilde{A}_{\epsilon,\mu} \mathbf{y}^n + \mathbf{f}^n), \quad (3.7)$$

where the function ψ is given by

$$\psi(x^2) = 2 \frac{1 - \cos x}{x^2}. \quad (3.8)$$

For a derivation of the scheme we refer to [17].

3.3.1. Computation of $\psi(\tau^2 \tilde{A}_{\epsilon,\mu}) \mathbf{v}$

Since the matrix $\tilde{A}_{\epsilon,\mu}$ is large and sparse, computation of $\psi(\tau^2 \tilde{A}_{\epsilon,\mu}) \mathbf{v}$ by conventional methods (see e.g. [14], Chapter 11) is hardly feasible. However, the action of the matrix function ψ on a given vector at each time step can be efficiently computed by a Krylov subspace method. Algorithms for this have been developed and used in different contexts (we list in the chronological order [37,6,21,32,7,16,8,18], see also Chapter 11 in the recent book [38]).

Throughout this subsection we denote $A = \tau^2 \tilde{A}_{\epsilon,\mu}$, $A \in \mathbb{R}^{N \times N}$. Computation of $\psi(A) \mathbf{v}$ for a given vector \mathbf{v} is based on the Arnoldi or, when $A = A^*$, on the Lanczos process (see e.g. [38,33]). The Lanczos process involves

the three-term recurrences and is therefore cheaper, especially for large Krylov subspace dimensions m . Since in this case m is not too large we use the Arnoldi process which has better numerical stability properties.

Starting with A and \mathbf{v} , the Arnoldi process generates after m steps orthonormal vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{m+1}$ (with $\mathbf{v}_1 = \mathbf{v}/\|\mathbf{v}\|$) and a Hessenberg matrix $\bar{H}_m \in \mathbb{R}^{(m+1) \times m}$ such that (see [38,33])

$$AV_m = V_{m+1}\bar{H}_m, \tag{3.9}$$

where $V_{m+1} \in \mathbb{R}^{N \times (m+1)}$ is a matrix with column vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{m+1}$ (and, correspondingly, V_m is V_{m+1} with the last column skipped). Denote by H_m a matrix obtained from \bar{H}_m by deleting its last row. We have

$$AV_m \approx V_m H_m, \tag{3.10}$$

where the approximation improves as m grows. Krylov subspace approximations to $\psi(A)\mathbf{v}$ are based on the last relation: since in the Arnoldi process by construction $\mathbf{v}_1 = \mathbf{v}/\|\mathbf{v}\|$ we have

$$\mathbf{v} = V_m \mathbf{y}, \quad \mathbf{y} = \|\mathbf{v}\| \mathbf{e}_1,$$

with \mathbf{e}_1 being the first canonical basis vector in \mathbb{R}^m , and (cf. (3.10))

$$\psi(A)V_m \mathbf{y} \approx V_m \psi(H_m) \mathbf{y}, \quad \mathbf{y} = \|\mathbf{v}\| \mathbf{e}_1,$$

so that the action of the matrix function on the given vector \mathbf{v} is computed as

$$\psi(A)\mathbf{v} \approx \|\mathbf{v}\| V_m \psi(H_m) \mathbf{e}_1. \tag{3.11}$$

We emphasize that dependence of the orthonormal basis $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ on \mathbf{v} is crucial to have a good approximation in (3.11).

In practice m is small (say 20), so that $\psi(H_m)$ in (3.11) can easily be computed by a standard method (see e.g. Chapter 11 in [14] and references therein). In the experiments presented in this paper, $\psi(H_m)$ was computed with Matlab’s built-in functions `sqrtm` and `funm`.

An important question is when to stop the Arnoldi process. One stopping criterion is proposed in [18] and is based on controlling a norm of a generalized residual. Unfortunately, in our experiments this approach appeared to be very sensitive to the given tolerance which had to be tuned for every test problem. For this reason we use another simple strategy: the Arnoldi process was stopped as soon as

$$\left\| \frac{\mathbf{y}_{(m)}^{n+1} - \mathbf{y}_{(m-1)}^{n+1}}{\mathbf{y}_{(m)}^{n+1} - \mathbf{y}_{(0)}^{n+1}} \right\|_{\infty} \leq \text{TOL}, \tag{3.12}$$

where $\mathbf{y}_{(m)}^{n+1}$ is the numerical solution of the scheme (3.7) obtained with m steps of the Arnoldi process, the division of the vectors is understood elementwise and `TOL` is a tolerance (in all our experiments we used the value `TOL` = 10^{-2} , this value should be chosen according to the relative accuracy required for a specific problem). By $\mathbf{y}_{(0)}^{n+1}$ we denote the solution obtained by (3.7) with $\psi(\tau^2 \tilde{A}_{\epsilon,\mu})$ set to the identity matrix (so that no Arnoldi steps are done). Note that $\mathbf{y}_{(0)}^{n+1}$ coincides with the solution of the leap frog scheme (cf. (3.1)) and, thus, is a second order time-consistent numerical solution. Stopping criterion (3.12) means that the further increase of the Krylov subspace dimension m leads to no further improvement in the accuracy as compared to the accuracy already obtained with respect to the leap frog solution $\mathbf{y}_{(0)}^{n+1}$. Note that this stopping criterion can be shown to be a controller of the Krylov subspace error (see Section 4.2).

The described steps lead to the algorithm for the Gautschi–Krylov time integration scheme presented in Fig. 1. The analysis of the Krylov subspace error made in the matrix function evaluations and the stability of the new scheme are presented in Section 4.

Since the work to compute the matrix function of the small matrix H_m is negligible, the overall computational work of the Gautschi scheme per time step is dominated by $m + 1$ matrix–vector multiplications with the matrix $\tilde{A}_{\epsilon,\mu}$ (m of which are required by the Arnoldi process). This means an increase by a factor of $m + 1$ as compared to the work per time step in the leap frog scheme.

```

 $\mathbf{y}^n$  and  $\mathbf{y}^{n-1}$  are given
 $\mathbf{v} = \tilde{A}_{\epsilon,\mu}\mathbf{y}^n - \mathbf{f}^n, \quad \beta = \|\mathbf{v}\|_2$ 
 $\mathbf{y}_{(0)}^{n+1} = 2\mathbf{y}^n - \mathbf{y}^{n-1} - \tau^2\mathbf{v}$ 
for  $m = 1, \dots,$ 
    extend the Krylov basis by one Arnoldi step:
    if( $m = 1$ ) then
         $\mathbf{v}_1 = \mathbf{v}/\beta$ 
        initialize  $\tilde{H}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ 
    else
        extend  $\tilde{H}_{m-1}$  to  $\tilde{H}_m$  by adding
        a zero column and a zero row
    endif
     $\mathbf{w} = \tau^2\tilde{A}_{\epsilon,\mu}\mathbf{v}_m$ 
    for  $i = 1, \dots, m$ 
         $h_{i,m} = \mathbf{w}^T\mathbf{v}_i$ 
         $\mathbf{w} = \mathbf{w} - h_{i,m}\mathbf{v}_i$ 
    endfor
     $h_{m+1,m} = \|\mathbf{w}\|_2$ 
     $\mathbf{v}_{m+1} = \mathbf{w}/h_{m+1,m}$ 
     $V_{m+1} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m, \mathbf{v}_{m+1}]$ 
end of Arnoldi step
compute matrix function  $\psi(H_m)$ 
 $\mathbf{u} = V_m[\beta\psi(H_m)\mathbf{e}_1]$ 
 $\mathbf{y}_{(m)}^{n+1} = 2\mathbf{y}^n - \mathbf{y}^{n-1} - \tau^2\mathbf{u}$ 
exit for-loop if condition (3.12) is fulfilled
endifor
 $\mathbf{y}^{n+1} = \mathbf{y}_{(m)}^{n+1}$ 

```

Fig. 1. The Gautschi scheme with the Krylov subspace matrix function evaluation and adaptive choice of the Krylov dimension.

3.4. LLC scheme

The following scheme proposed by J.-F. Lee, R. Lee, and A. Cangelaris (the LLC scheme [24,12]) can be applied directly to the semidiscrete Maxwell equations (2.16):

$$M_\epsilon \frac{\mathbf{e}^{n+1} - 2\mathbf{e}^n + \mathbf{e}^{n-1}}{\tau^2} + A_\mu \left(\frac{1}{4}\mathbf{e}^{n-1} + \frac{1}{2}\mathbf{e}^n + \frac{1}{4}\mathbf{e}^{n+1} \right) = \mathbf{j}^n. \quad (3.13)$$

This scheme can be written in the form

$$\left(M_\epsilon + \frac{\tau^2}{4}A_\mu \right) \mathbf{e}^{n+1} = \tau^2 \mathbf{j}^n - \left(\frac{\tau^2}{2}A_\mu - 2M_\epsilon \right) \mathbf{e}^n - \left(M_\epsilon + \frac{\tau^2}{4}A_\mu \right) \mathbf{e}^{n-1}, \quad (3.14)$$

revealing that a linear system with matrix $M_\epsilon + \frac{\tau^2}{4}A_\mu$ has to be solved at every time step. For discretizations obtained on relatively coarse grids this can be done by a sparse direct solver, by computing the LU factorization once and reusing it at every time step. If a direct solution is not feasible, a preconditioned Krylov iterative solver can be used.

The LLC scheme is unconditionally (regardless of the time step τ) stable [24].

3.5. One-step formulations of the three schemes

Each of the three schemes described in this section is a two-step scheme (i.e. it requires numerical solutions on both n and $n - 1$ time levels to get the next time level solution) but can be written in a one-step form. This is

normally done by introducing an auxiliary time derivative variable. These one-step formulations can be used at the first time step where the two-step formulation would have required the normally unknown value of e^{-1} .

In the context of the Maxwell equations, a natural way to obtain a one-step formulation of a time integration scheme is to consider the Maxwell equations as the two first order equations. A possible drawback of this approach is that one has to work with both fields and, hence, build up appropriate spatial discretizations for each of the fields. Thus, one of the benefits of treating the Maxwell equations as a second order equation for one of the fields is then lost.

In this section we give the one-step formulations for all schemes. We derive it for the LLC scheme. The other two one-step formulations can be obtained in a similar way. The formulations are given for an auxiliary variable but directly applicable to the two first order Maxwell equations, too. Introducing the time-derivative auxiliary variable as

$$\mathbf{u}^{n+1/2} = \frac{\mathbf{e}^{n+1} - \mathbf{e}^n}{\tau}, \tag{3.15}$$

we can write (3.13) as

$$M_\epsilon \frac{\mathbf{u}^{n+1/2} - \mathbf{u}^{n-1/2}}{\tau} + \frac{1}{2} A_\mu \frac{\mathbf{e}^{n-1} + \mathbf{e}^n}{2} + \frac{1}{2} A_\mu \frac{\mathbf{e}^n + \mathbf{e}^{n+1}}{2} = \frac{1}{2} \mathbf{j}^n + \frac{1}{2} \mathbf{j}^n,$$

or, formally introducing the variable \mathbf{u}^n , as

$$\begin{aligned} M_\epsilon \frac{\mathbf{u}^n - \mathbf{u}^{n-1/2}}{\tau/2} + A_\mu \frac{\mathbf{e}^{n-1} + \mathbf{e}^n}{2} &= \mathbf{j}^n, \\ M_\epsilon \frac{\mathbf{u}^{n+1/2} - \mathbf{u}^n}{\tau/2} + A_\mu \frac{\mathbf{e}^n + \mathbf{e}^{n+1}}{2} &= \mathbf{j}^n. \end{aligned} \tag{3.16}$$

Writing the first half-step update here for the next time level (i.e. replacing n with $n + 1$) we have

$$M_\epsilon \frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1/2}}{\tau/2} + A_\mu \frac{\mathbf{e}^n + \mathbf{e}^{n+1}}{2} = \mathbf{j}^{n+1},$$

which, together with (3.15) and (3.16) leads to the following *one-step formulation of the LLC scheme*:

$$\begin{aligned} M_\epsilon \frac{\mathbf{u}^{n+1/2} - \mathbf{u}^n}{\tau/2} + A_\mu \frac{\mathbf{e}^n + \mathbf{e}^{n+1}}{2} &= \mathbf{j}^n, \\ \frac{\mathbf{e}^{n+1} - \mathbf{e}^n}{\tau} &= \mathbf{u}^{n+1/2}, \\ M_\epsilon \frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1/2}}{\tau/2} + A_\mu \frac{\mathbf{e}^n + \mathbf{e}^{n+1}}{2} &= \mathbf{j}^{n+1}. \end{aligned} \tag{3.17}$$

In this form the sequence of computations for the scheme is not immediately clear and we rewrite it as

$$\begin{aligned} \left(M_\epsilon + \frac{\tau^2}{4} A_\mu \right) \mathbf{e}^{n+1} &= \frac{\tau^2}{2} \mathbf{j}^n + \left(M_\epsilon - \frac{\tau^2}{4} A_\mu \right) \mathbf{e}^n + \tau M_\epsilon \mathbf{u}^n, \\ M_\epsilon \mathbf{u}^{n+1} &= \frac{\tau}{2} \mathbf{j}^{n+1} - \frac{\tau}{4} A_\mu (\mathbf{e}^n + \mathbf{e}^{n+1}) + M_\epsilon \frac{\mathbf{e}^{n+1} - \mathbf{e}^n}{\tau}. \end{aligned}$$

The one-step formulations for the leap frog and the Gautschi scheme can be obtained along the same lines (see also [17]):

$$\text{One-step leap frog: } \begin{cases} M_\epsilon \frac{\mathbf{u}^{n+1/2} - \mathbf{u}^n}{\tau/2} + A_\mu \mathbf{e}^n = \mathbf{j}^n, \\ \frac{\mathbf{e}^{n+1} - \mathbf{e}^n}{\tau} = \mathbf{u}^{n+1/2}, \\ M_\epsilon \frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1/2}}{\tau/2} + A_\mu \mathbf{e}^{n+1} = \mathbf{j}^{n+1}. \end{cases}$$

$$\text{One-step Gautschi: } \begin{cases} \frac{u^{n+1/2}-u^n}{\tau/2} = \psi(\tau^2 \tilde{A}_{\epsilon,\mu})(-\tilde{A}_{\epsilon,\mu} \mathbf{y}^n + \mathbf{f}^n), \\ \frac{y^{n+1}-y^n}{\tau} = u^{n+1/2}, \\ \frac{u^{n+1}-u^{n+1/2}}{\tau/2} = \psi(\tau^2 \tilde{A}_{\epsilon,\mu})(-\tilde{A}_{\epsilon,\mu} \mathbf{y}^{n+1} + \mathbf{f}^{n+1}). \end{cases}$$

4. Analysis of the Gautschi–Krylov scheme

4.1. Krylov subspace approximation error

Theorem 4.1.1. Consider the homogeneous ODE system $\mathbf{y}'' + A\mathbf{y} = 0$. Then, the solution of the Gautschi–Krylov scheme has the form:

$$\mathbf{y}^{n+1} = -\mathbf{y}^{n-1} + 2 \cos(\tau A^{1/2}) \mathbf{y}^n + \underbrace{\int_0^\tau A^{-1/2} \sin((\tau-s)A^{1/2}) \tilde{\mathbf{g}}(s) ds}_{=:\delta^n, \text{ Krylov error}} \tag{4.1}$$

$$\tilde{\mathbf{g}}(s) = -\beta h_{m+1,m} \mathbf{v}_{m+1} e_m^T \psi(s^2 H_m) e_1,$$

where τ is the step size, m is the Krylov dimension, $\beta = \|A\mathbf{y}^n\|$, $h_{m+1,m}$ is the $(m+1, m)$ entry of the matrix \bar{H}_m . The matrices \bar{H}_m , H_m , and the vector \mathbf{v}_{m+1} are defined in (3.9), (3.10), e_1 and e_m are, respectively, the first and the last canonical basis vectors in \mathbb{R}^m , and ψ is given by (3.8).

For the exact Gautschi scheme (where the matrix function evaluations are done exactly) relation (4.1) holds with $\delta^n \equiv 0$.

Proof. The proof (inspired by the analysis given in Section 4 of [36]) consists of showing that the solution of the Gautschi–Krylov scheme is the exact solution of a perturbed (inhomogeneous) ODE system.

Without loss of generality, we shift for convenience the time variable such that $t_n = 0$, $t_{n+1} = t$ and the Gautschi scheme can be written as

$$\mathbf{y}(t) - 2\mathbf{y}(0) + \mathbf{y}(-t) = -t^2 \psi(t^2 A) A \mathbf{y}(0).$$

Substituting here function ψ as it is defined in (3.8) leads to relation (4.1) with $\delta^n \equiv 0$ which thus indeed holds for the exact Gautschi scheme. In the Gautschi–Krylov scheme the right-hand side is computed approximately with Arnoldi or Lanczos process as

$$-t^2 \psi(t^2 A) A \mathbf{y}(0) = -\beta t^2 \psi(t^2 A) V_m e_1 \approx -\beta t^2 V_m \psi(t^2 H_m) e_1,$$

where the matrix V_m is defined in (3.9), (3.10). The Gautschi–Krylov scheme can thus be written as

$$\mathbf{y}(t) - 2\mathbf{y}(0) + \mathbf{y}(-t) = -\beta t^2 V_m \psi(t^2 H_m) e_1. \tag{4.2}$$

Denote $(\cdot)' = d(\cdot)/dt$. Since

$$(t^2 \psi(t^2 H_m))'' = (2H_m^{-1} - 2 \cos(tH_m^{1/2}) H_m^{-1})'' = 2 \cos(tH_m^{1/2}),$$

differentiating equality (4.2) twice with respect to t yields

$$[\mathbf{y}(t) + \mathbf{y}(-t)]'' = -2\beta V_m \cos(tH_m^{1/2}) e_1.$$

We now use the Arnoldi relation (3.9) rewritten as

$$AV_m = V_m H_m + h_{m+1,m} \mathbf{v}_{m+1} e_m^T \tag{4.3}$$

and write

$$-2\beta V_m \cos(tH_m^{1/2}) e_1 = -2\beta V_m H_m H_m^{-1} \cos(tH_m^{1/2}) e_1 = -2\beta (AV_m - h_{m+1,m} \mathbf{v}_{m+1} e_m^T) H_m^{-1} \cos(tH_m^{1/2}) e_1,$$

so that

$$[\mathbf{y}(t) + \mathbf{y}(-t)]'' = -2\beta(AV_m - h_{m+1,m}\mathbf{v}_{m+1}e_m^T)H_m^{-1} \cos(tH_m^{1/2})e_1. \tag{4.4}$$

On the other hand, the right-hand side of (4.2) can be transformed as

$$-\beta t^2 V_m \psi(t^2 H_m) e_1 = -2\beta V_m (I - \cos(tH_m^{1/2})) H_m^{-1} e_1 = -2\beta V_m H_m^{-1} e_1 + 2\beta V_m \cos(tH_m^{1/2}) H_m^{-1} e_1. \tag{4.5}$$

Here the term $V_m H_m^{-1}$ reads

$$V_m H_m^{-1} = A^{-1} V_m + h_{m+1,m} A^{-1} \mathbf{v}_{m+1} e_m^T H_m^{-1},$$

this follows from the Arnoldi relation (4.3). Substituting the last expression into (4.5) we get the following relation for the right-hand side of the Gautschi–Krylov scheme (4.2):

$$-2\beta A^{-1} V_m e_1 - 2\beta h_{m+1,m} A^{-1} \mathbf{v}_{m+1} e_m^T H_m^{-1} e_1 + 2\beta V_m \cos(tH_m^{1/2}) H_m^{-1} e_1.$$

Note that since the starting vector of the Arnoldi process is $A\mathbf{y}(0) = \beta v_1$ (see Fig. 1 and recall that $\mathbf{y}(0) = \mathbf{y}''$), for the first term holds:

$$-2\beta A^{-1} V_m e_1 = -2\beta A^{-1} \mathbf{v}_1 = -2A^{-1} A\mathbf{y}(0) = -2\mathbf{y}(0)$$

and the Gautschi–Krylov scheme thus reads (cf. (4.2))

$$\mathbf{y}(t) - 2\mathbf{y}(0) + \mathbf{y}(-t) = -2\mathbf{y}(0) - 2\beta h_{m+1,m} A^{-1} \mathbf{v}_{m+1} e_m^T H_m^{-1} e_1 + 2\beta V_m \cos(tH_m^{1/2}) H_m^{-1} e_1.$$

Here multiplication of both sides with A results in

$$A(\mathbf{y}(t) + \mathbf{y}(-t)) = -2\beta h_{m+1,m} \mathbf{v}_{m+1} e_m^T H_m^{-1} e_1 + 2\beta A V_m \cos(tH_m^{1/2}) H_m^{-1} e_1$$

or, taking into account that $\cos(tH_m^{1/2}) H_m^{-1} = H_m^{-1} \cos(tH_m^{1/2})$,

$$-2\beta A V_m H_m^{-1} \cos(tH_m^{1/2}) e_1 = -A(\mathbf{y}(t) + \mathbf{y}(-t)) - 2\beta h_{m+1,m} \mathbf{v}_{m+1} e_m^T H_m^{-1} e_1.$$

Replacing the first term of the right-hand side in (4.4) by the right-hand side of the last relation, we obtain

$$[\mathbf{y}(t) + \mathbf{y}(-t)]'' = -A(\mathbf{y}(t) + \mathbf{y}(-t)) - 2\beta h_{m+1,m} \mathbf{v}_{m+1} e_m^T H_m^{-1} e_1 + 2\beta h_{m+1,m} \mathbf{v}_{m+1} e_m^T H_m^{-1} \cos(tH_m^{1/2}) e_1,$$

and, using (3.8),

$$[\mathbf{y}(t) + \mathbf{y}(-t)]'' = -A(\mathbf{y}(t) + \mathbf{y}(-t)) - \underbrace{\beta h_{m+1,m} \mathbf{v}_{m+1} e_m^T \psi(t^2 H_m) e_1}_{=: \tilde{\mathbf{g}}(t)}. \tag{4.6}$$

We now can get an analytic expression for $u(t) \equiv \mathbf{y}(t) + \mathbf{y}(-t)$ by solving the following initial-value problem:

$$\mathbf{u}'' = -A\mathbf{u} + \tilde{\mathbf{g}}(t), \quad \mathbf{u}(0) = 2\mathbf{y}(0), \quad \mathbf{u}'(0) = 0, \tag{4.7}$$

where the initial condition $u'(0) = 0$ holds because function $u(t)$ is even. Applying a variation-of-constants formula to this initial-value problem gives

$$\mathbf{u}(t) = \cos(tA^{1/2})\mathbf{u}(0) + A^{-1/2} \sin(tA^{1/2})\mathbf{u}'(0) + \int_0^t A^{-1/2} \sin((t-s)A^{1/2})\tilde{\mathbf{g}}(s)ds,$$

$$\mathbf{y}(t) + \mathbf{y}(-t) = 2 \cos(tA^{1/2})\mathbf{y}(0) + \int_0^t A^{-1/2} \sin((t-s)A^{1/2})\tilde{\mathbf{g}}(s)ds,$$

which, after changing the time variable back (so that $\mathbf{y}(0) = \mathbf{y}''$, $\mathbf{y}(\pm t) = \mathbf{y}''^{\pm 1}$) yields the required relation (4.1). \square

4.2. Stopping criterion for the Arnoldi process

The proposed stopping criterion for the Arnoldi process (cf. (3.12)) can be shown to be a controller of the Krylov subspace error specified by (4.1). To see this, we assume that one time step is done with both the Gautschi–Krylov and the exact Gautschi schemes and rewrite (4.1) as

$$\begin{aligned} \mathbf{y}_{(m)}^{n+1} &= -\mathbf{y}^{n-1} + 2 \cos(\tau A^{1/2}) \mathbf{y}^n + \boldsymbol{\delta}_{(m)}^n, \\ \mathbf{y}_{\text{ex}}^{n+1} &= -\mathbf{y}^{n-1} + 2 \cos(\tau A^{1/2}) \mathbf{y}^n. \end{aligned}$$

where m is the Krylov subspace dimension, $\mathbf{y}_{(m)}^{n+1}$ and $\mathbf{y}_{\text{ex}}^{n+1}$ are, respectively, solutions of the Gautschi–Krylov and the exact Gautschi schemes and the Krylov subspace error $\boldsymbol{\delta}_{(m)}^n$ is given by (4.1):

$$\boldsymbol{\delta}_{(m)}^n = -\beta h_{m+1,m} \int_0^\tau A^{-1/2} \sin((\tau - s)A^{1/2}) \mathbf{v}_{m+1} e_m^T \psi(s^2 H_m) e_1 ds. \tag{4.8}$$

This expression cannot be readily used in practice for the evaluation of $\boldsymbol{\delta}_{(m)}^n$ due to the presence of the term $A^{-1/2} \sin((\tau - s)A^{1/2}) \mathbf{v}_{m+1}$. Computation of this matrix–vector product with the large matrix A is too expensive and an approximation should be used. This can be done in different ways. For example, one might take several first terms of the following series [10] as an approximation:

$$A^{-1/2} \sin((\tau - s)A^{1/2}) = (\tau - s)I - \frac{1}{3!}(\tau - s)^3 A + \frac{1}{5!}(\tau - s)^5 A^2 - \dots \tag{4.9}$$

Note that substituting this relation in (4.8) we could obtain another, more detailed expression for the Krylov subspace error $\boldsymbol{\delta}_{(m)}^n$ (for a similar analysis see Lemma 4.1 in [36]). Instead of (4.9) one might also use some other approximations based, e.g., on Chebyshev polynomials. A more natural and efficient way for estimating the Krylov subspace error is to use the same continued Arnoldi process to get a reference solution (for a different time integration scheme, this was proposed in [36]). More specifically, assume that, in addition to the m steps of the Arnoldi process, another j steps of the process are done. Then

$$A^{-1/2} \sin((\tau - s)A^{1/2}) \mathbf{v}_{m+1} = A^{-1/2} \sin((\tau - s)A^{1/2}) V_{m+j} e_{m+1}^{(m+j)} \approx V_{m+j} H_{m+j}^{-1/2} \sin((\tau - s)H_{m+j}^{1/2}) e_{m+1}^{(m+j)}, \tag{4.10}$$

where $e_{m+1}^{(m+j)}$ is the $(m + 1)$ th canonical basis vector in \mathbb{R}^{m+j} . This approximation is accurate if $|h_{m+j+1,m+j}|$ is small enough (see (4.3) with m replaced by $m + j$). Since $h_{m+j+1,m+j} \approx 0$ implies $\boldsymbol{\delta}_{(m+j)}^n \approx 0$, the solution $\mathbf{y}_{(m+j)}^{n+1}$ of the Gautschi–Krylov scheme after $m + j$ steps is then also accurate:

$$\mathbf{y}_{(m+j)}^{n+1} \approx \mathbf{y}_{\text{ex}}^{n+1}.$$

Hence, the value of $\boldsymbol{\delta}_{(m)}^n$ with approximation (4.10) can be estimated as

$$\boldsymbol{\delta}_{(m)}^n = \mathbf{y}_{(m)}^{n+1} - \mathbf{y}_{\text{ex}}^{n+1} \approx \mathbf{y}_{(m)}^{n+1} - \mathbf{y}_{(m+j)}^{n+1}.$$

In the proposed stopping criterion of the Arnoldi process (cf. (3.12)), the difference $\mathbf{y}_{(m)}^{n+1} - \mathbf{y}_{(m+j)}^{n+1}$ is evaluated in a special relative norm suitable for the time stepping process. The choice $j = 1$ (also made in [36]) is appropriate since in most cases the Arnoldi process for matrix function evaluations exhibits a superlinear convergence [16,36].

4.3. Stability of the Gautschi–Krylov scheme

The original Gautschi scheme (where the matrix function evaluations are performed exactly) is exact for the linear ODE system $\mathbf{y}'' + A\mathbf{y} = 0$ and hence is trivially stable. To show stability of the Gautschi–Krylov scheme, we follow approach of [18] and consider perturbations $\boldsymbol{\varepsilon}^n \equiv \mathbf{y}^n - \mathbf{y}_{\text{ex}}^n$ with respect to the solution \mathbf{y}_{ex}^n of the exact Gautschi scheme. Theorem 4.1.1 states that

$$\begin{aligned} \mathbf{y}_{\text{ex}}^{n+1} &= -\mathbf{y}_{\text{ex}}^{n-1} + 2 \cos(\tau A^{1/2}) \mathbf{y}_{\text{ex}}^n, \quad \text{or} \\ \begin{bmatrix} \mathbf{y}_{\text{ex}}^{n+1} \\ \mathbf{y}_{\text{ex}}^n \end{bmatrix} &= \begin{bmatrix} 2 \cos(\tau A^{1/2}) & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_{\text{ex}}^n \\ \mathbf{y}_{\text{ex}}^{n-1} \end{bmatrix}. \end{aligned} \tag{4.11}$$

Subtracting this relation from (4.1) we arrive at

$$\begin{aligned} \mathbf{e}^{n+1} &= -\mathbf{e}^{n-1} + 2 \cos(\tau A^{1/2}) \mathbf{e}^n + \boldsymbol{\delta}^n, \quad \text{or} \\ \begin{bmatrix} \mathbf{e}^{n+1} \\ \mathbf{e}^n \end{bmatrix} &= \begin{bmatrix} 2 \cos(\tau A^{1/2}) & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} \mathbf{e}^n \\ \mathbf{e}^{n-1} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\delta}^n \\ 0 \end{bmatrix}. \end{aligned}$$

For $\boldsymbol{\delta}^n \equiv 0$ this recursion coincides with the exact solution recursion (4.11) and thus is stable if and only if the ODE system to be solved is stable. One may understand stability in different ways [30,13], for instance, we may require that

$$\|G^n\| \leq K, \quad \text{for } n \geq 0, \quad n\tau \leq T, \quad G = \begin{bmatrix} 2 \cos(\tau A^{1/2}) & -I \\ I & 0 \end{bmatrix}, \tag{4.12}$$

where K does not depend on τ and T is the final time, for some operator norm $\|\cdot\|$. We now assume that the exact Gautschi scheme is stable in this sense and thus (4.12) holds true.

Stability of the Gautschi–Krylov scheme follows immediately as it does for perturbed (inhomogeneous) difference schemes (see e.g. [30, Chapter 4] or [13, Section 14]). Although the Krylov approximation error $\boldsymbol{\delta}^n$ can formally be made arbitrarily small, the Gautschi–Krylov scheme remains stable even if we allow a linear growth of the norm of $\boldsymbol{\delta}^n$ with respect to the time step τ :

$$\left\| \begin{bmatrix} \boldsymbol{\delta}^n \\ 0 \end{bmatrix} \right\| \leq C\tau,$$

with C independent on τ . Denoting

$$\mathcal{E}^n = \begin{bmatrix} \mathbf{e}^n \\ \mathbf{e}^{n-1} \end{bmatrix}, \quad \hat{\boldsymbol{\delta}}^n = \begin{bmatrix} \boldsymbol{\delta}^n \\ \boldsymbol{\delta}^{n-1} \end{bmatrix},$$

one can obtain a standard expression for two-level schemes

$$\mathcal{E}^n = G^n \mathcal{E}^0 + G^{n-1} \hat{\boldsymbol{\delta}}^0 + G^{n-2} \hat{\boldsymbol{\delta}}^1 + \dots + \hat{\boldsymbol{\delta}}^{n-1},$$

from which the stability estimate follows:

$$\|\mathcal{E}^n\| \leq \|G^n\| \|\mathcal{E}^0\| + \|G^{n-1}\| n \max_{0 \leq i \leq n-1} \|\hat{\boldsymbol{\delta}}^i\| \leq K \|\mathcal{E}^0\| + KnC\tau \leq K \|\mathcal{E}^0\| + KCT.$$

5. Dispersion analysis

For PDEs of the wave type dispersion analysis is an important tool to understand the error behavior of the scheme.

In this section we analyze and compare, for the edge finite element spatial discretization on a uniform mesh, the numerical dispersion error for the three schemes introduced in Section 3. For the analysis, we make the following two assumptions:

- (1) Eq. (2.12) is given in an infinite source free ($\mathbf{J} \equiv 0$) region with periodic boundary conditions:

$$\epsilon_r \partial_{tt} \mathbf{E} + \nabla \times (\mu_r^{-1} \nabla \times \mathbf{E}) = 0. \tag{5.1}$$

- (2) μ_r and ϵ_r are constant scalars.

A vector field

$$\mathbf{E}(x, y, z, t) = \mathbf{E}_0 \exp(i(\mathbf{k} \cdot \mathbf{x} - \omega t)), \quad \text{where } i = \sqrt{-1}, \tag{5.2}$$

is a solution of (5.1) if the dispersion relation

$$\omega^2 = c_r^2 k^2 \tag{5.3}$$

holds, where $\mathbf{k} = (k_1, k_2, k_3)$ is the wave vector, $\mathbf{x} = (x, y, z)$, $k = \|\mathbf{k}\|_2 \equiv \sqrt{k_1^2 + k_2^2 + k_3^2}$ is the wave number, $c_r = 1/(\sqrt{\epsilon_r \mu_r})$ is the scaled speed of light, and ω is the angular frequency.

We consider the finite element discretization of (5.1) on a uniform parallelepiped mesh with elements of size $h \times h \times h$, see Fig. 2. The angles $\angle DAB$ and $\angle CAB$ are called deformation angles.

Remark 5.0.1. To avoid cumbersome expressions, we present many of the formulas for the cubic case $\angle DAB = \angle CAB = 90^\circ$. If a formula is valid only for the cubic elements, this is explicitly reported. However, the whole analysis is valid for the general case and the resulting plots of the dispersion errors are given also for the deformed mesh. Part of computations for the dispersion analysis were done in Maple.

On this regular mesh the finite element matrices (2.17) take the form $M_\epsilon = \epsilon_r h M$ and $A_\mu = \frac{1}{h_\mu} A$, where the matrices M and A do not depend on the element size h . This results in the following system of ODEs:

$$M\mathbf{e}'' + \frac{c_r^2}{h^2} A\mathbf{e} = 0. \tag{5.4}$$

The time exact dispersion equation is

$$-\omega^2 M\mathbf{e} + \frac{c_r^2}{h^2} A\mathbf{e} = 0. \tag{5.5}$$

We end up with an eigenvalue problem with large sparse matrices given in (5.4). Since we are working on a uniform mesh, it is possible to reduce the problem size as follows:

The expansion coefficients of the finite element approximation are $e_j(t) = \int_{\text{edge } j} \mathbf{E}(\mathbf{x}, t) \cdot \mathbf{t}_j ds$. If the exact solution of (5.1) is given by (5.2) then for any two parallel edges p and j the expansion coefficients satisfy

$$e_p^{n+q} = \exp(i(\mathbf{k} \cdot \Delta_{pj} - \omega q \tau)) e_j^n, \tag{5.6}$$

where the superscript indicates the time level, the subscript indicates the number of the edge to which the coefficient belongs, and Δ_{pj} is a vector from the midpoint of edge p to the midpoint of edge j .

5.1. Gautschi method

We analyze the Gautschi scheme under the assumption that the action of the matrix function (3.8) on a given vector can be computed exactly (or very accurately) so that the scheme is exact in time. This assumption is realistic (see Section 6.3). Hence, we consider the time-accurate dispersion relation (5.5) for the system (5.4), which gives us the following generalized eigenvalue problem:

$$-\omega^2 M\mathbf{e}'' + \frac{c_r^2}{h^2} A\mathbf{e}'' = 0. \tag{5.7}$$

Denoting $\varphi(\omega) = -\omega^2$ and $\eta = \frac{c_r^2}{h^2}$, we have

$$\varphi(\omega) M\mathbf{e}'' + \eta A\mathbf{e}'' = 0. \tag{5.8}$$

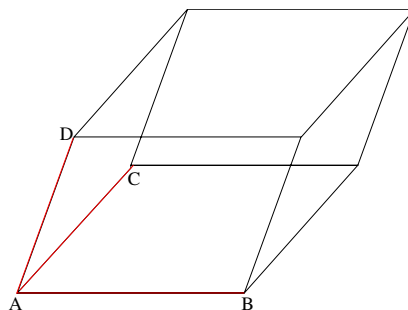


Fig. 2. Deformed element with deformation angles $\angle CAB$ and $\angle DAB$. The angle $\angle DAC = 90^\circ$.

Using the relations (5.6) it is not difficult to see that on a uniform grid Eq. (5.8) are the same (up to a constant \tilde{C}_{pj}) for parallel edges, i.e. for any two parallel edges p and j holds:

$$\varphi(\omega)M(a_p, :)\mathbf{e}^n + \eta A(a_p, :)\mathbf{e}^n = \tilde{C}_{pj}(\varphi(\omega)M(a_j, :)\mathbf{e}^n + \eta A(a_j, :)\mathbf{e}^n) = 0,$$

where $M(a_j, :)$ denotes a_j th row of matrix M , and similarly for A . Therefore it is sufficient to consider the equations corresponding to any three edges a_1, a_2, a_3 among which there are no parallel edges (see Fig. 3).

Let

$$X(t) = \int_{a_1} \mathbf{E}(\mathbf{x}, t) \cdot \mathbf{t} da, \quad Y(t) = \int_{a_2} \mathbf{E}(\mathbf{x}, t) \cdot \mathbf{t} da, \quad Z(t) = \int_{a_3} \mathbf{E}(\mathbf{x}, t) \cdot \mathbf{t} da,$$

then using (5.6) all the other degrees of freedom (coefficients) in the whole mesh can be expressed in terms of X, Y, Z .

The corresponding equation of edge a_1 is

$$\varphi(\omega)M(a_1, :)\mathbf{e}^n + \eta A(a_1, :)\mathbf{e}^n = 0. \tag{5.9}$$

The matrices M and A have a sparse structure because in (5.9) coefficients only of those basis functions are present which have nonempty common support with the basis function corresponding to the edge a_1 . On a cubic mesh we have

$$\begin{aligned} M(a_1, :)\mathbf{e}^n &= \frac{1}{36}(1, 4, 1, 4, 16, 4, 1, 4, 1) \cdot (\tilde{e}_1, \tilde{e}_2, \tilde{e}_3, \tilde{e}_{14}, \tilde{e}_{15}, \tilde{e}_{16}, \tilde{e}_{27}, \tilde{e}_{28}, \tilde{e}_{29})^T, \\ A(a_1, :)\mathbf{e}^n &= \frac{1}{6}(-2, -2, -2, 1, -1, -1, 1, 1, -1, 4, -4, 1, -1, -2, 16, -2, \\ &\quad 4, -4, -4, 4, -1, 1, -4, 4, -1, 1, -2, -2, -2, 1, -1, -1, 1) \cdot (\tilde{e}_1, \tilde{e}_2, \tilde{e}_3, \dots, \tilde{e}_{32}, \tilde{e}_{33})^T. \end{aligned} \tag{5.10}$$

Here the tilde sign is used to distinguish the local index with the global index, for example $\tilde{e}_{15} = e_{a_1}, \tilde{e}_{19} = e_{a_2}$. Writing the relations similar to (5.9) for edges a_2 and a_3 and using (5.6), we obtain a homogeneous system of equations

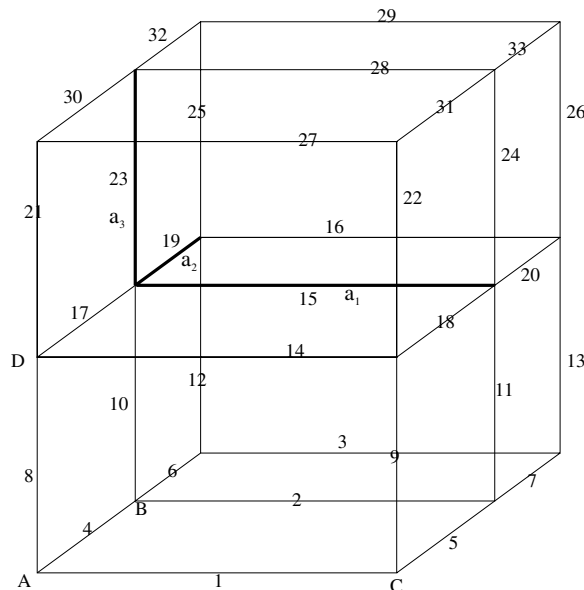


Fig. 3. Three nonparallel edges a_1, a_2, a_3 and the degrees of freedom (with a local numbering) that appear in Eq. (5.7) for edge a_1 .

$$(\varphi(\omega)F + \eta G) \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = 0. \tag{5.11}$$

On both cubic and deformed meshes the numerical dispersion relation of the Gautschi scheme is

$$\det(\varphi(\omega)F + \eta G) = 0, \quad \text{or} \tag{5.12}$$

$$\det \left(-\omega^2 F + \frac{c_r^2}{h^2} G \right) = 0,$$

where the 3×3 matrices F and G depend on the wave vector \mathbf{k} and the mesh size (entries of F and G are specified for the cubic mesh in [Appendix B](#)). One of the solutions of the dispersion relation is $\omega = 0$, which does not represent anything physical. The other solutions of (5.12) satisfy

$$(\omega_h h)^2 = 18 \frac{4 - \cos \xi_3 \cos \xi_2 - \cos \xi_1 \cos \xi_2 \cos \xi_3 - \cos \xi_3 \cos \xi_1 - \cos \xi_1 \cos \xi_2}{(2 + \cos \xi_1)(2 + \cos \xi_2)(2 + \cos \xi_3)} c_r^2, \tag{5.13}$$

where $\xi_i = hk_i$, $i = 1, 2, 3$, and ω_h denotes the numerical angular frequency. The exact phase velocity is given by $c_r = \omega/k$ and the numerical phase velocity is $v = \omega_h/k$. In [Fig. 4](#) a plot of the phase velocity error is given for cubic elements with $k_3 = 0$. For all the numerical experiments throughout this section we assume that $\epsilon_r = \mu_r = 1$.

Under the assumption $|kh| \ll 1$ the Taylor expansion of (5.13) shows

$$\omega_h = c_r k \left(1 + \frac{1}{24} \frac{k_1^4 + k_2^4 + k_3^4}{k^2} h^2 + \text{higher order terms} \right),$$

which means that the dispersion relation for the Gautschi scheme is satisfied up to second order.

5.2. Leap frog scheme

Applying relation (5.6) to the leap frog scheme (3.1), we have

$$\frac{e^{n+1} - 2e^n + e^{n-1}}{\tau^2} = \frac{\exp(-i\omega\tau)e^n - 2e^n + \exp(i\omega\tau)e^n}{\tau^2} = \frac{2(\cos(\omega\tau) - 1)}{\tau^2} e^n. \tag{5.14}$$

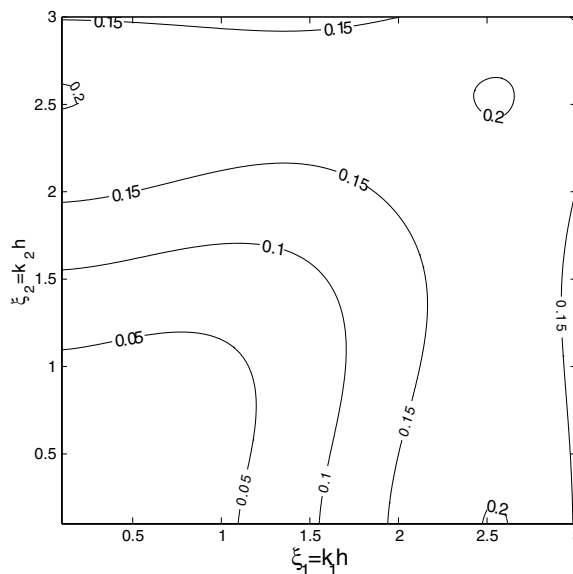


Fig. 4. The phase velocity error of the Gautschi scheme for cubic elements.

Then the generalized eigenvalue problem of the leap frog scheme is

$$\frac{2(\cos(\omega\tau) - 1)}{\tau^2} M e^n + \frac{c_r^2}{h^2} A e^n = 0. \tag{5.15}$$

Introducing $\varphi(\omega) = \frac{2(\cos(\omega\tau)-1)}{\tau^2}$ and $\eta = \frac{c_r^2}{h^2}$ in (5.8) we obtain the dispersion equation for the leap frog scheme

$$\det \left(\frac{2(\cos(\omega\tau) - 1)}{\tau^2} F + \frac{c_r^2}{h^2} G \right) = 0, \tag{5.16}$$

with the 3×3 matrices F and G defined as in (5.12). There are 3 roots, one is zero which is non physical. The solution of (5.16) satisfies (on a cubic mesh)

$$\cos(\omega\tau) = 1 - 2 \frac{\chi_1(\tau, h, \mathbf{k})}{\chi_2(\tau, h, \mathbf{k})}, \tag{5.17}$$

where

$$\begin{aligned} \chi_1(\tau, h, \mathbf{k}) &= 9c_r^2\tau^2(4 - \cos \xi_1 \cos \xi_2 \cos \xi_3 - \cos \xi_1 \cos \xi_2 - \cos \xi_2 \cos \xi_3 - \cos \xi_3 \cos \xi_1), \\ \chi_2(\tau, h, \mathbf{k}) &= 2h^2(2 + \cos \xi_1)(2 + \cos \xi_2)(2 + \cos \xi_3), \end{aligned}$$

and $\xi_i = hk_i, i = 1, 2, 3$.

According to the exact dispersion relation (5.3), we would like to have only real solutions ω of (5.17). Otherwise, as it is clear from (5.2), the imaginary part of ω will contribute to dissipation of the solution (damping if $\text{Im}(\omega) < 0$ or amplification if $\text{Im}(\omega) > 0$, see e.g. [40]). The value of ω is real if and only if

$$\left| 1 - 2 \frac{\chi_1(\tau, h, \mathbf{k})}{\chi_2(\tau, h, \mathbf{k})} \right| \leq 1,$$

or, equivalently,

$$\frac{c_r\tau}{h} \leq \frac{1}{3} \sqrt{\frac{2(2 + \cos \xi_1)(2 + \cos \xi_2)(2 + \cos \xi_3)}{4 - \cos \xi_1 \cos \xi_2 \cos \xi_3 - \cos \xi_1 \cos \xi_2 - \cos \xi_2 \cos \xi_3 - \cos \xi_3 \cos \xi_1}}. \tag{5.18}$$

Since it is always true that

$$\sqrt{\frac{2(2 + \cos \xi_1)(2 + \cos \xi_2)(2 + \cos \xi_3)}{4 - \cos \xi_1 \cos \xi_2 \cos \xi_3 - \cos \xi_1 \cos \xi_2 - \cos \xi_2 \cos \xi_3 - \cos \xi_3 \cos \xi_1}} \geq 1,$$

for the inequality (5.18) to hold true it is sufficient to require that

$$\frac{c_r\tau}{h} \leq \frac{1}{3}, \tag{5.19}$$

which gives stability condition on the uniform mesh. A more general stability condition is given in Appendix A.

Under the assumption $|kh| \ll 1$ the Taylor expansion of (5.17) shows

$$\omega_\tau = c_r k \left(1 + \frac{1}{24} c_r^2 k^2 \tau^2 + \frac{1}{24} \frac{k_1^4 + k_2^4 + k_3^4}{k^2} h^2 + \text{higher order terms} \right),$$

where ω_τ is the numerical angular frequency. In order to have spatial and temporal error terms of the same order, we should take $\tau = O(h)$. This is a clear disadvantage of leap frog compared to Gautschi.

In Figs. 5–7, the absolute error of the angular frequency for the leap frog scheme is shown in comparison with the Gautschi scheme for different values of the time step τ and deformation angles θ ($\angle DAC = \angle BAC = \theta$, see Fig. 2). Here, for simplicity, we assume $k_3 = 0$. Note that in all figures the plots of the leap frog scheme become increasingly similar (as τ decreases) to the plot of the time-exact Gautschi scheme. We observe that reduction of the time step beyond 0.002 does not give more accurate results because the spatial error is dominant.

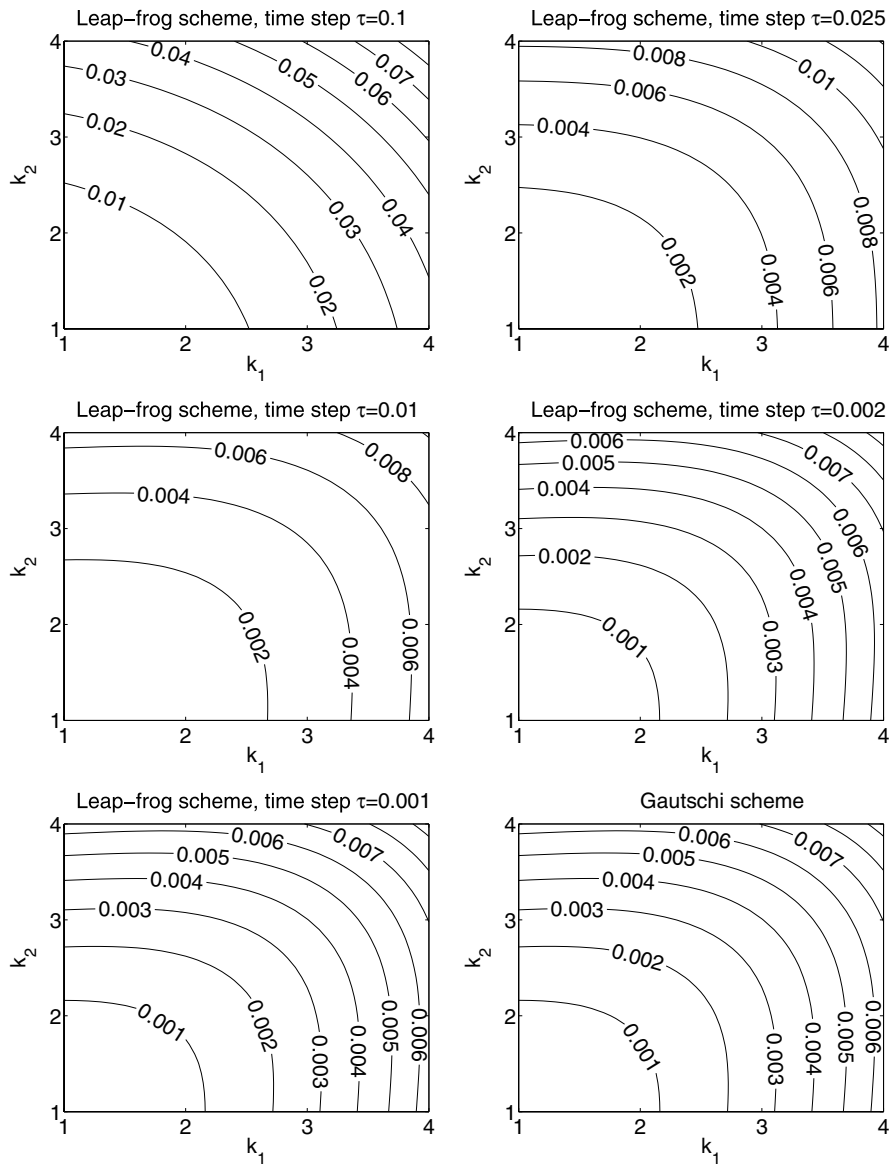


Fig. 5. Absolute value of the angular frequency errors for the leap frog scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/2$.

5.3. LLC scheme

The generalized eigenvalue problem for the LLC scheme (3.13) is

$$\frac{2(\cos(\omega\tau) - 1)}{\tau^2} M e^n + \frac{(\cos(\omega\tau) + 1)}{2} \frac{c_r^2}{h^2} A e^n = 0.$$

Introducing $\varphi(\omega) = \frac{2(\cos(\omega\tau)-1)}{\tau^2}$ and $\eta = \frac{\cos(\omega\tau)+1}{2} \frac{c_r^2}{h^2}$ in (5.8) we obtain the dispersion equation for the LLC scheme

$$\det \left(\frac{2(\cos(\omega\tau) - 1)}{\tau^2} F + \frac{(\cos(\omega\tau) + 1)}{2} \frac{c_r^2}{h^2} G \right) = 0, \tag{5.20}$$

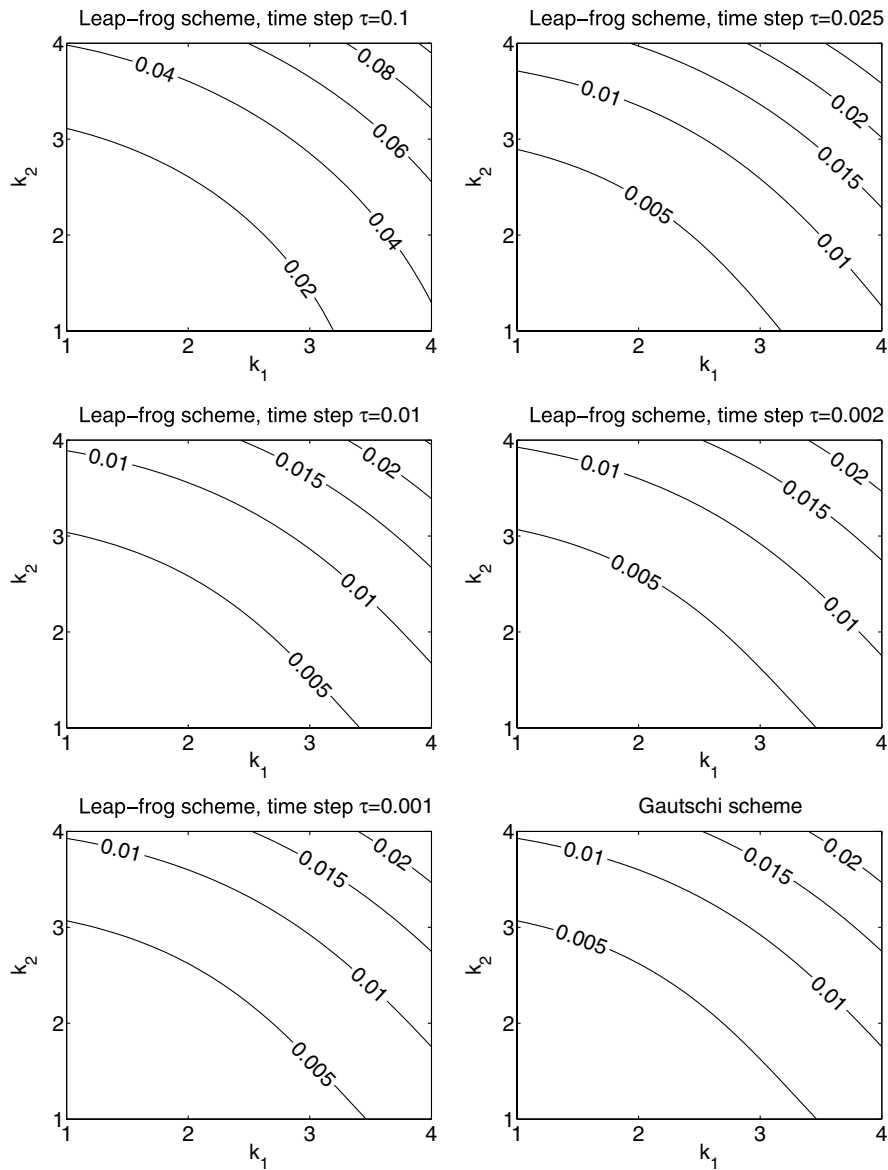


Fig. 6. Absolute value of the angular frequency errors for the leap frog scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/3$.

where the 3×3 matrices F and G are given as in (5.12). There are 3 roots, one is zero. The solution of (5.20) satisfies (on a cubic mesh)

$$\cos(\omega\tau) = \frac{\chi_2(\tau, h, \mathbf{k}) - \chi_1(\tau, h, \mathbf{k})}{\chi_2(\tau, h, \mathbf{k}) + \chi_1(\tau, h, \mathbf{k})}, \tag{5.21}$$

where

$$\begin{aligned} \chi_1(\tau, h, \mathbf{k}) &= 9c_r^2\tau^2(4 - \cos \xi_1 \cos \xi_2 \cos \xi_3 - \cos \xi_1 \cos \xi_2 - \cos \xi_2 \cos \xi_3 - \cos \xi_3 \cos \xi_1), \\ \chi_2(\tau, h, \mathbf{k}) &= 2h^2(2 + \cos \xi_1)(2 + \cos \xi_2)(2 + \cos \xi_3), \end{aligned}$$

and $\xi_i = hk_i$, $i = 1, 2, 3$.

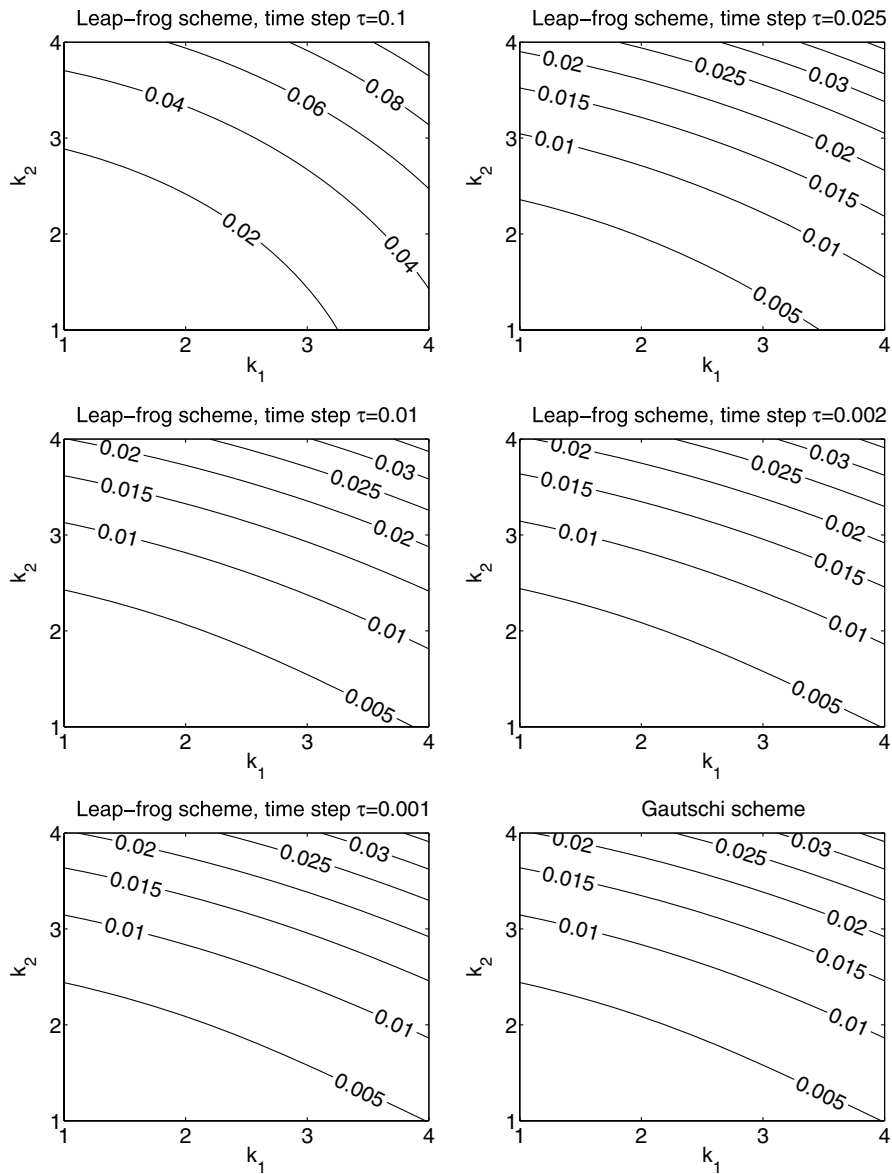


Fig. 7. Absolute value of the angular frequency errors for the leap frog scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/4$.

Under the assumption $|kh| \ll 1$ the Taylor expansion of (5.21) shows

$$\omega_\tau = c_r k \left(1 - \frac{1}{12} c_r^2 k^2 \tau^2 + \frac{1}{24} \frac{k_1^4 + k_2^4 + k_3^4}{k^2} h^2 + O(h^4) + O(\tau^4) + O(\tau^2 h^2) + \text{higher order terms} \right), \quad (5.22)$$

where ω_τ denotes the numerical angular frequency. In order to make the spatial and temporal error terms of the same order, we should take $\tau = O(h)$. We note that the dispersion error of the LLC scheme becomes fourth order accurate if we choose

$$\tau = \sqrt{\frac{1}{2c_r^2} \frac{k_1^4 + k_2^4 + k_3^4}{k^4}} h, \quad (5.23)$$

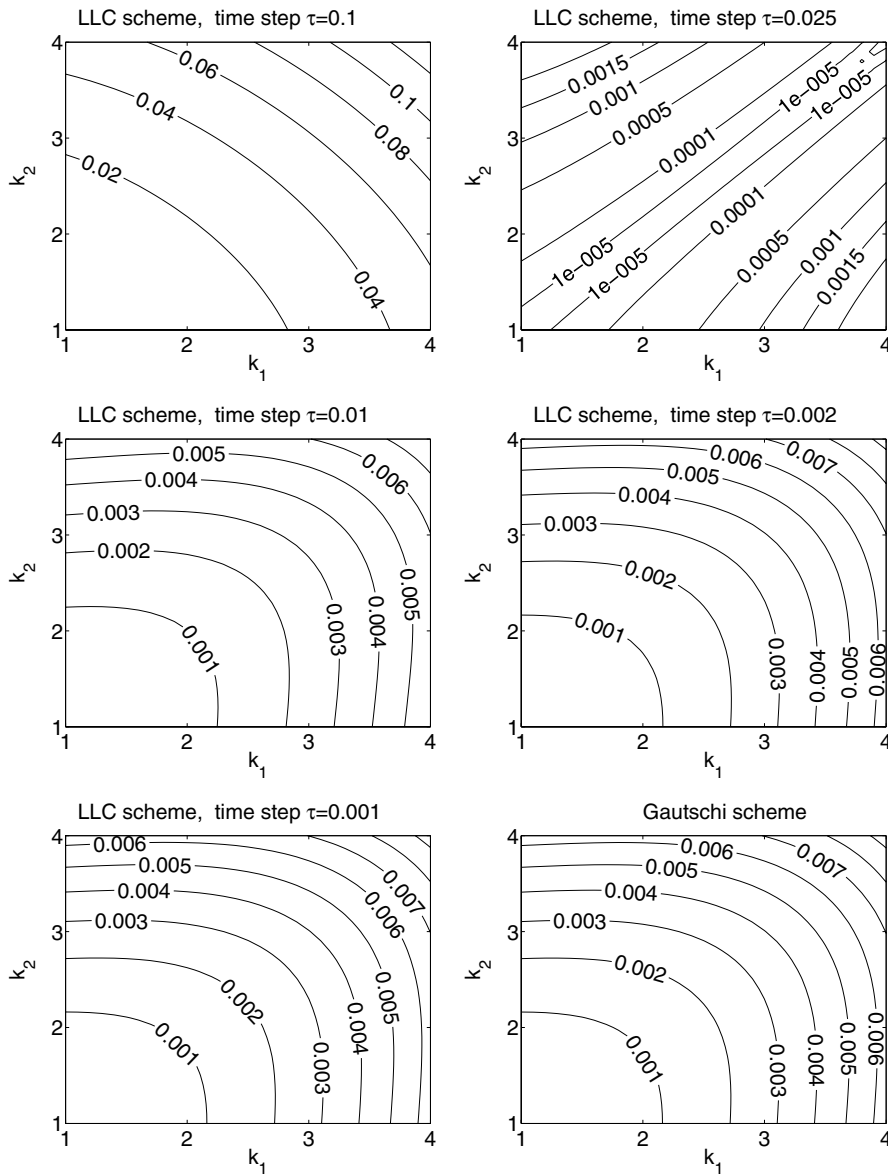


Fig. 8. Absolute value of the angular frequency errors for the LLC scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/2$. The plot for the time step $\tau = 0.025$ reflects the increase in the error order (cf. (5.23) with $k_3 = 0$ and $k_1 \approx k_2$).

which can be called an optimum time step. We note that (5.22), (5.23) are only valid on a cubic mesh.

In Figs. 8–10, the absolute error of the angular frequency for the LLC scheme is shown in comparison with the time-accurate Gautschi scheme for different values of time step τ and deformation angles θ ($\angle DAC = \angle BAC = \theta$, see Fig. 2). Here again we assume for simplicity $k_3 = 0$.

For the LLC scheme we observe a similar convergence behavior as for the leap frog scheme. Note that the plot for the step size $\tau = 0.025$ in Fig. 8 differs significantly from the other plots in the figure due to the increase in the error order observed in (5.22) (cf. (5.23) with $k_3 = 0$ and $k_1 \approx k_2$).

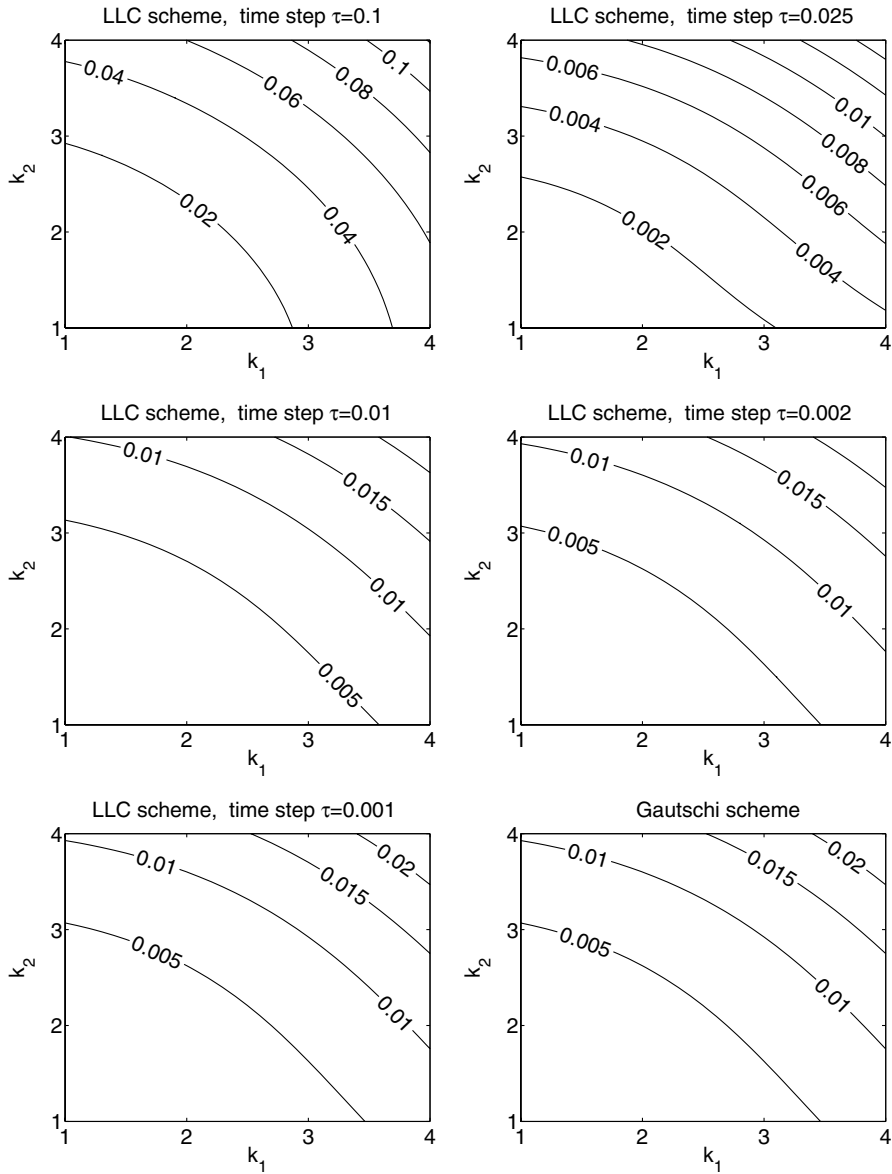


Fig. 9. Absolute value of the angular frequency errors for the LLC scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/3$.

6. Numerical experiments

6.1. Test problem 1

This test problem is obtained by choosing an arbitrary vector field function $\mathbf{E}_{\text{an}}(x, y, z, t)$ satisfying the boundary conditions, projecting it onto the finite element subspace and substituting the projection into the semidiscrete system (2.16). The source function $\mathbf{j}(t)$ is then chosen such that the finite element projection of \mathbf{E}_{an} is the exact solution of (2.16). Note that it is important to use the exact solution of the *semidiscrete* system because the difference of this solution with the computed numerical solution represents then *solely* the time error (without the spatial discretization error).

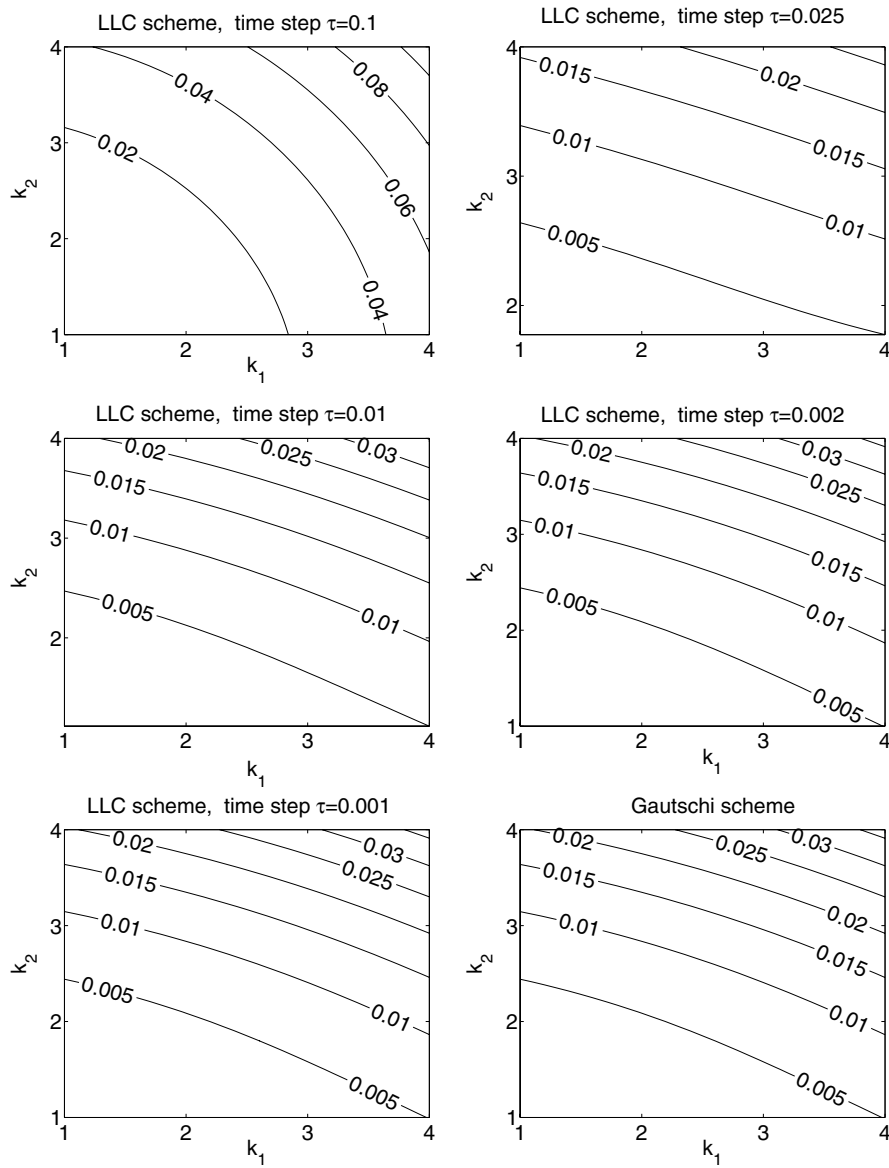


Fig. 10. Absolute value of the angular frequency errors for the LLC scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/4$.

More specifically, we consider the dimensionless Maxwell equations (2.12) in the domain $\Omega = [0, 1] \times [0, 1] \times [0, 1]$ and we take

$$\mathbf{E}_{\text{an}}(x, y, z, t) = v(t)\bar{\mathbf{E}}(x, y, z).$$

If $\bar{\mathbf{e}}$ is the finite element projection of the field $\bar{\mathbf{E}}$ then

$$\mathbf{e}_{\text{an}}(t) = v(t)\bar{\mathbf{e}}$$

is the exact solution of the semidiscrete ODE system (2.16) with

$$\mathbf{j}(t) = (v''M_\epsilon + vA_\mu)\bar{\mathbf{e}}.$$

In our experiments we took

$$\epsilon_r = 1, \quad \mu_r = 1.$$

$$v(t) = \sum_{i=1}^{N_\omega} \cos \omega_i t, \quad \bar{\mathbf{E}}(x, y, z) = \begin{bmatrix} \sin \pi y \sin \pi z \\ \sin \pi x \sin \pi z \\ \sin \pi x \sin \pi y \end{bmatrix}. \quad (6.1)$$

where the values of ω_i are reported later separately for each of the test runs. This test problem is well suited for studying the evolution of the time error, since the exact solution is readily computable for any moment of time t .

6.2. Test problem 2

This test problem differs from the previous one only by the choice of the exact (reference) solution. The exact solution is obtained by any of the available schemes run with an extremely small time step size τ . With this τ all schemes produce numerical solutions which are practically exact in time but with the same spatial error as the numerical solutions obtained for realistically large τ . Such a testing approach is common in numerical time integration of space-discretized PDEs (see e.g. [35]). This test problem is convenient when one wants to know the error at the final time.

6.3. The Krylov subspace dimension and the time error

Here we investigate how the choice of the Krylov subspace dimension in the Gautschi scheme influences its time integration error. We are interested in the evolution of the error in time and therefore use Test problem 1. The frequencies ω_i of the inhomogeneous term $\mathbf{j}(t)$ (cf. (6.1)) are chosen as

$$\omega_1 = 1, \quad \omega_2 = 10.$$

The results are presented in Fig. 11. Here, the time error evolution of the Gautschi scheme is shown for different fixed Krylov subspace dimensions m and for the adaptive choice of m based on the condition (3.12). The time integration was done up to the final time $T = 6 \frac{2\pi}{\max_i \{\omega_i\}}$ corresponding to the 6 periods of time. The shown error is the Euclidean norm of the difference between the coefficients of the finite element basis expansions of the numerical and the exact solutions.

Inspection of the plots in Fig. 11 shows that there is a certain value of $m = \tilde{m}$ such that increasing the Krylov subspace dimension beyond \tilde{m} does not lead to any improvement in time accuracy. In other words, even if we compute the action of the matrix function on vectors very accurately the error does not decrease. Thus, for $m \geq \tilde{m}$ we have a scheme where the error caused by the Krylov subspace approximation is negligible as compared to the time error of the exact Gautschi scheme. The adaptive choice of m is able to catch the value of \tilde{m} very accurately: for example, for the upper plot ($14 \times 14 \times 14$ mesh) we can see that $\tilde{m} \approx 4$ whereas the adaptive choice gave values m between 3 and 5.

The typical dimensions of the Krylov subspace, observed in practice, depend on the time step size used. For the step sizes up to a factor two larger than the CFL number (which is the maximal possible step size of the explicit leap frog scheme) the Krylov dimension is usually 2. For larger realistic time step sizes values of m up to 12 can be observed. The values of m mildly grow as the spatial mesh gets finer.

6.4. Computational work

We recall that on the uniform meshes the computational work per time step in the Gautschi scheme is a factor $m + 1$ (with m being the Krylov subspace dimension) more than for the leap frog scheme.

On uniform meshes the computational work of the LLC scheme is difficult to compare explicitly with those of the leap frog and Gautschi schemes. This is because on uniform meshes the sparse LU factorization of the matrix $M_\epsilon + \frac{\epsilon^2}{4} A_\mu$ in the LLC scheme is more expensive than that of M_ϵ , since the matrix M_ϵ is sparser due to orthogonality of some basis functions on the cubic elements. This makes the LLC scheme very expensive on finer meshes as compared to the other two schemes. For this reason the results for the LLC scheme in this section are shown only for a coarser $10 \times 10 \times 10$ mesh.

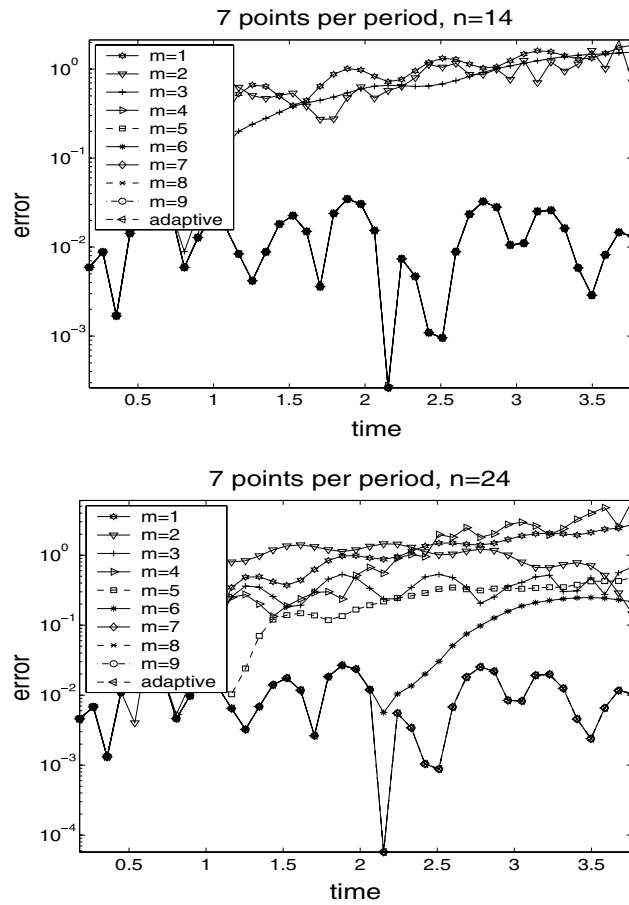


Fig. 11. Error evolution of the Gautschi scheme for $14 \times 14 \times 14$ (top) and $24 \times 24 \times 24$ (bottom) meshes for different Krylov subspace dimensions m . The step size corresponds to 7 points per time period.

On the uniform meshes let us denote the computational work required for the LU factorizations of the matrices $M_\epsilon + \frac{\tau^2}{4}A_\mu$ and M_ϵ as *lu_fac_LLC* and *lu_fac_lf*, respectively. The computational work for one matrix–vector multiplication with the matrices $M_\epsilon - \frac{\tau^2}{4}A_\mu$ and A_μ involved in the LLC and leap frog schemes is defined as *mat_vec_LLC* and *mat_vec_lf*, respectively. The computational work required for the LU solver for the schemes LLC and leap frog is denoted as *lu_sol_LLC* and *lu_sol_lf*, respectively.

In contrast to the situation on the uniform meshes, the matrices M_ϵ and $M_\epsilon + \frac{\tau^2}{4}A_\mu$ have the same sparsity structure on unstructured meshes, hence require the same computational work for the LU factorization. Although the computational work per time step in the Gautschi scheme is larger than in the LLC or the leap frog scheme, the Gautschi scheme appears to be more efficient (see results of Section 6.5.2). Let us define a relative work required for one LU factorization as *lu_fac*, one matrix–vector multiplication as *mat_vec* and one LU solver as *lu_sol*. It is clear that per time step the LLC and the leap frog schemes require *mat_vec* + *lu_sol* and the Gautschi scheme requires $(m + 1)(mat_vec + lu_sol)$ operations.

If we denote the required computational work per time step for the cases described above as

$$\begin{aligned}
 Q_{lf} &= mat_vec_lf + lu_sol_lf, \\
 Q_{LLC} &= mat_vec_LLC + lu_sol_LLC, \\
 Q &= mat_vec + lu_sol,
 \end{aligned}$$

then the overall computational work for all the schemes on the uniform and unstructured meshes is given in Table 1, where T is the final time and τ is the time step size.

Table 1
Computational work for the three schemes

	Uniform mesh	Unstructured mesh
Gautschi	$\frac{T}{\tau}(m+1) \cdot Q_{lf} + lu_fac_lf$	$\frac{T}{\tau}(m+1) \cdot Q + lu_fac$
LLC	$\frac{T}{\tau} \cdot Q_{LLC} + lu_fac_LLC$	$\frac{T}{\tau} \cdot Q + lu_fac$
Leap frog	$\frac{T}{\tau} \cdot Q_{lf} + lu_fac_lf$	$\frac{T}{\tau} \cdot Q + lu_fac$

On finer uniform or unstructured meshes the LU factorizations may require too much computational efforts. In this case one could use an iterative solver for the three schemes. In the context of the Arnoldi process used in the Gautschi scheme this would mean that the action of M_ϵ^{-1} is computed by an inner iterative solver. Note that the matrix $M_\epsilon + \frac{\epsilon^2}{4}A_\mu$ appearing in the LLC scheme usually requires more iterations of an iterative solver than the well-conditioned mass matrix M_ϵ [9]. Performance of the iterative solvers in all the schemes can be improved by a suitable preconditioning (see [36] for preconditioning of the Krylov subspace matrix function evaluations). On the other hand, the use of approximate implicit schemes [3] or stabilized explicit schemes [39,34,35] might be a good option here, too.

6.5. Comparisons of the three schemes

We compare now the time stepping errors at the final time and the CPU times of the three schemes presented in Section 3. Since we are interested in time errors at the final time, we use Test problem 2. The presented error values are computed as

$$\text{error} = \left\| \frac{\mathbf{y}^{\bar{n}} - \mathbf{y}_{\text{exact}}^{\bar{n}}}{\mathbf{y}_{\text{exact}}^{\bar{n}} + \epsilon_C} \right\|_\infty, \quad (6.2)$$

where the division of the vectors is understood element-wise, $\mathbf{y}^{\bar{n}}$ and $\mathbf{y}_{\text{exact}}^{\bar{n}}$ are the numerical and the exact (reference) solutions at the final time $T = \bar{n}\tau = 50$, and ϵ_C is the machine epsilon.

6.5.1. Uniform cubic mesh

In the experiments presented in this section, a uniform cubic mesh was used. In the first test, the frequencies ω_i of the source term $\mathbf{j}(t)$ were taken to be homogeneously distributed:

$$N_\omega = 101, \quad w_i \text{ evenly distributed in } [1, 10], \quad i = 1, \dots, 101. \quad (6.3)$$

The results are presented in Fig. 12. We see that all the schemes clearly exhibit second order time accuracy. The peculiar drop in the error-versus- τ plot of the LLC scheme, is caused by the increase in the error order observed in (5.22), (5.23).

The nonmonotonicity seen on the error-versus-CPU time plots of the Gautschi scheme are characteristic for the scheme: smaller time step sizes result in reduction of the Krylov dimension m which makes the scheme significantly cheaper. There is, thus, an optimal time step size for which the overall computational work is minimal. As one can see in Fig. 12, the Gautschi and LLC schemes lose to the leap frog scheme in performance. This is to be expected since we work on a uniform mesh in a domain with homogeneous ϵ_r and μ_r .

Because of different sparsity patterns of the matrices $M_\epsilon + \frac{\epsilon^2}{4}A_\mu$ and M_ϵ the plots versus the computational work in Fig. 13 are presented only for the leap frog and the Gautschi schemes.

Very similar results were obtained for the case where

$$N_\omega = 101, \quad \omega_i \text{ evenly distributed in } [1, 2], \quad i = 1, \dots, 100, \quad \omega_{101} = 10. \quad (6.4)$$

Here all the schemes yield errors which are approximately a factor 10^3 smaller than for the homogeneous distribution of ω_i (6.3). In this the case the error-versus- τ dependence of the LLC scheme is monotone.

We now present the performance of the Gautschi scheme on a finer mesh $40 \times 40 \times 40$ with higher, as compared to (6.3) and (6.4), frequencies in the source term:

$$\begin{aligned} N_\omega &= 103, \quad w_i \text{ evenly distributed in } [1, 2], \quad i = 1, \dots, 100, \\ \omega_{101} &= 10, \quad \omega_{102} = 24, \quad \omega_{103} = 25. \end{aligned} \quad (6.5)$$

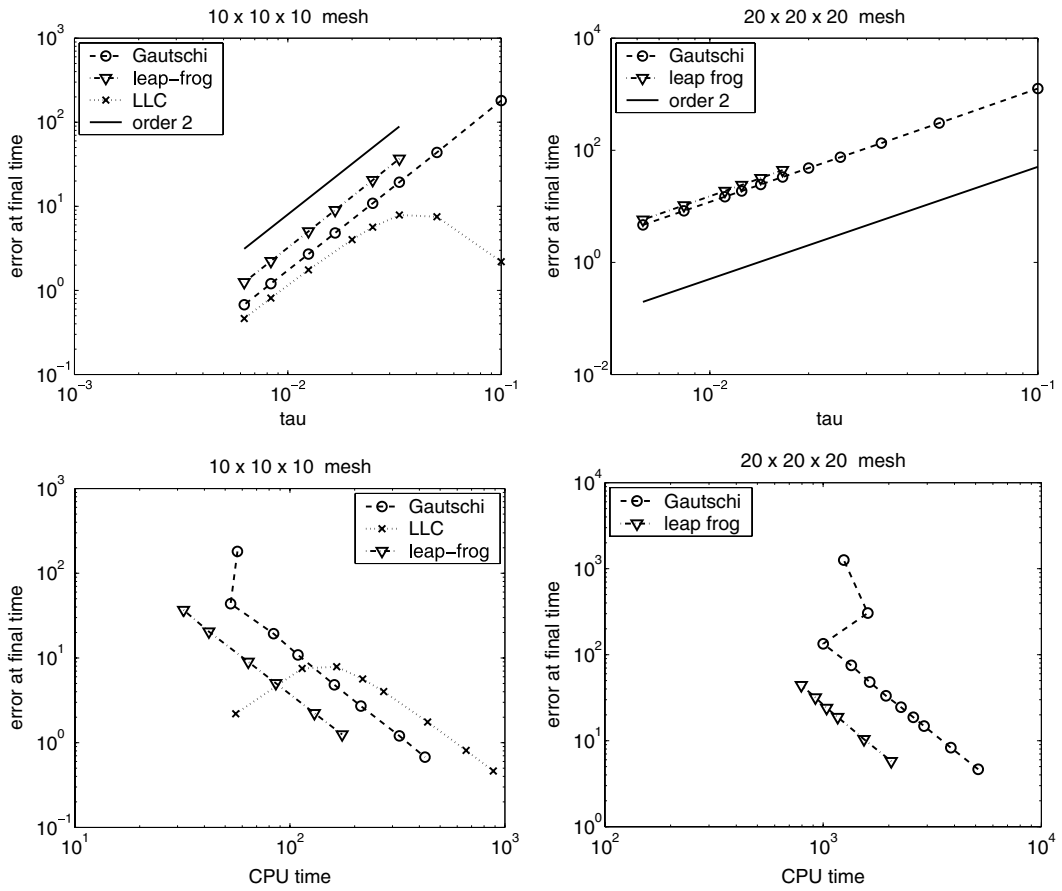


Fig. 12. Uniform mesh. Errors at the final time against the corresponding step sizes and the required CPU times for the homogeneously distributed frequencies in the source term (cf. (6.3)).

In Fig. 14 the errors at the final time are given against the corresponding step sizes and computational work. For this mesh, the sparse LU factorization of the matrix $M_\epsilon + \frac{\tau^2}{4}A_\mu$ in the LLC scheme is prohibitively expensive and the conjugate gradient iterative solver is used.

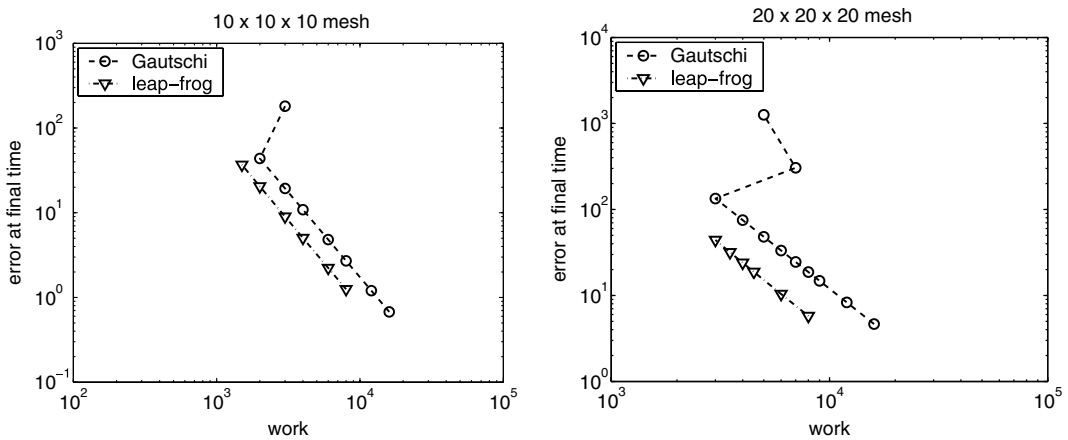


Fig. 13. Uniform mesh. Errors at the final time against the corresponding computational work for homogeneously distributed frequencies in the source term. The work is measured in the Q_{lf} units (see Section 6.4).

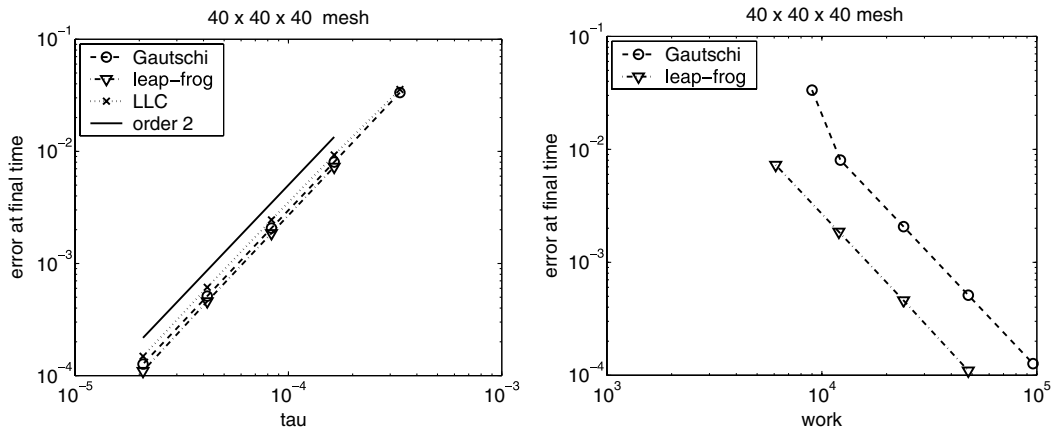


Fig. 14. Uniform mesh. Errors at the final time against the corresponding step sizes for clustered distribution of frequencies in the source term, see (6.5). The work is measured in the Q_{lf} units (see Section 6.4).

6.5.2. Unstructured tetrahedral mesh

In this example, Test problem 2 with the homogeneously distributed frequencies in the source term (cf. (6.3)) is solved on a unstructured tetrahedral mesh generated by the Centaur mesh generator. In the mesh used (see Fig. 15), the ratio between longest and shortest edge is about 17. Although the mesh is rather coarse, the time step of the leap frog scheme is restricted for stability reasons to the relatively small time step 0.0155 (which is approximately a factor 2 smaller than the stability time step restriction of a uniform mesh with roughly the same number of degrees of freedom).

The results of the experiment are given in Fig. 16. Note the irregular convergence pattern of the LLC scheme which is apparently caused by effects of the MATLAB/UMFPACK sparse direct solver used in the scheme (the accuracy of the solver is compromised to retain sparsity in the LU factors).

It is evident that to achieve the same accuracy both the explicit leap frog scheme and the implicit LLC scheme require much smaller time steps than the Gautschi scheme and their computational times are bigger than that of the Gautschi scheme.

In Fig. 17 we compare accuracies delivered by the schemes versus required computational work (see Section 6.4). It is clear from this figure that on the unstructured mesh the Gautschi scheme appears to be the most efficient.

6.5.3. Exactness of the Gautschi scheme for the slowly varying inhomogeneous term

The Gautschi scheme is known to be exact for the constant inhomogeneous term $j(t)$ [11,17]. To see whether this is the case for our Krylov subspace implementation of the scheme, we take in these two tests (i) zero and (ii) very small values of ω_j :

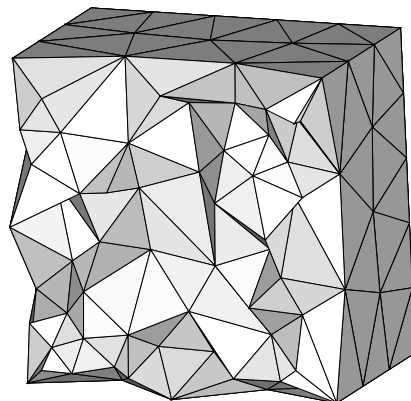


Fig. 15. A cut of the unstructured mesh used for the experiment.

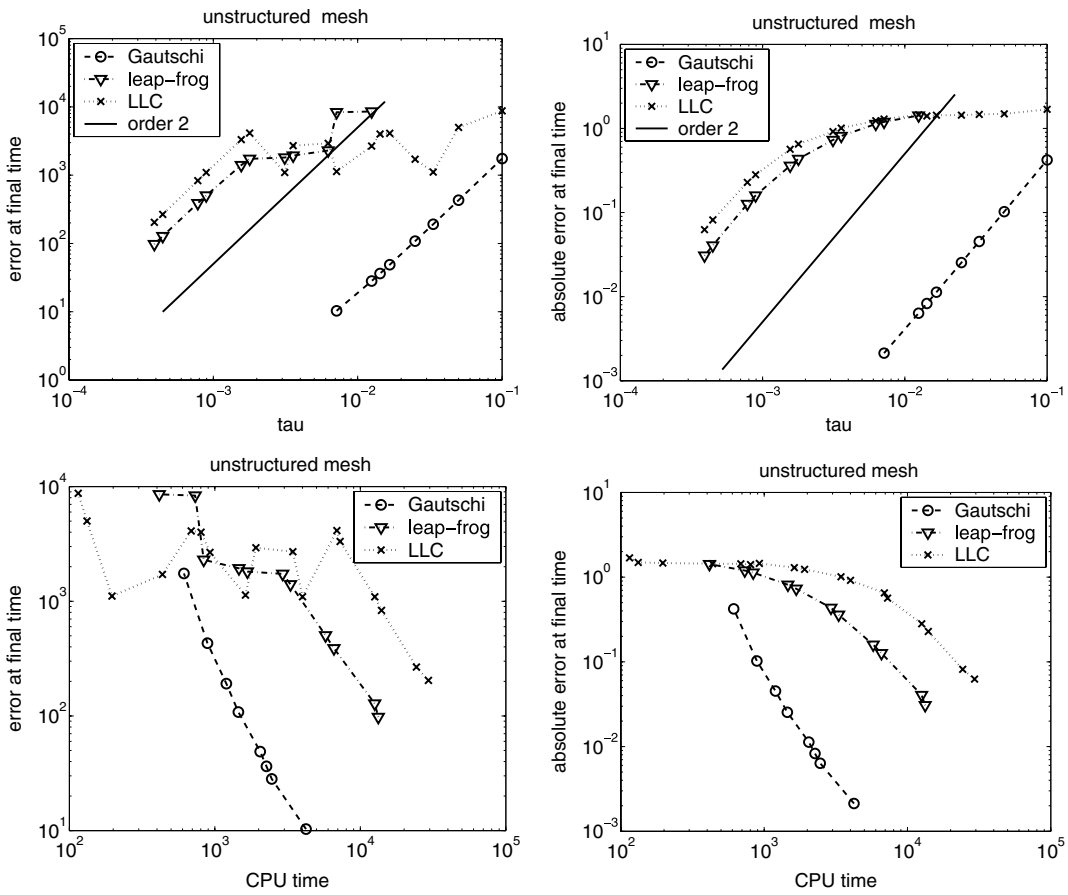


Fig. 16. Unstructured mesh. Errors at the final time against the corresponding step sizes and the required CPU times for the homogeneously distributed frequencies in the source term (cf. (6.3)). Left plots: the error is measured as in (6.2). Right plots: the error is measured as $\|y^n - y_{\text{exact}}^n\| / \|y_{\text{exact}}^n\|$.

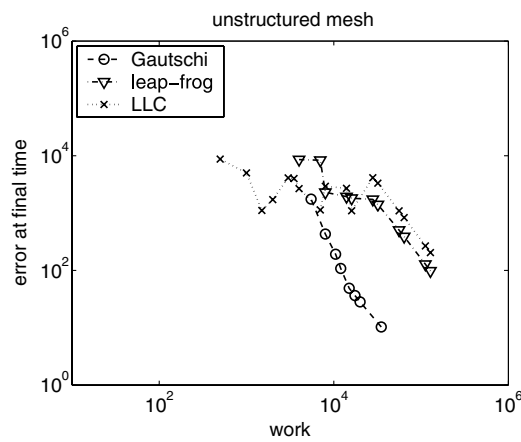


Fig. 17. Unstructured mesh. Errors at the final time against the corresponding computational work for homogeneously distributed frequencies in the source term. The work is measured in the Q units (see Section 6.4).

$$(i) N_\omega = 1, \quad \omega_1 = 0, \tag{6.6}$$

$$(ii) N_\omega = 3, \quad \omega_1 = 10^{-5}, \quad \omega_2 = 2.23 \times 10^{-5}, \quad \omega_3 = 8 \times 10^{-6}. \tag{6.7}$$

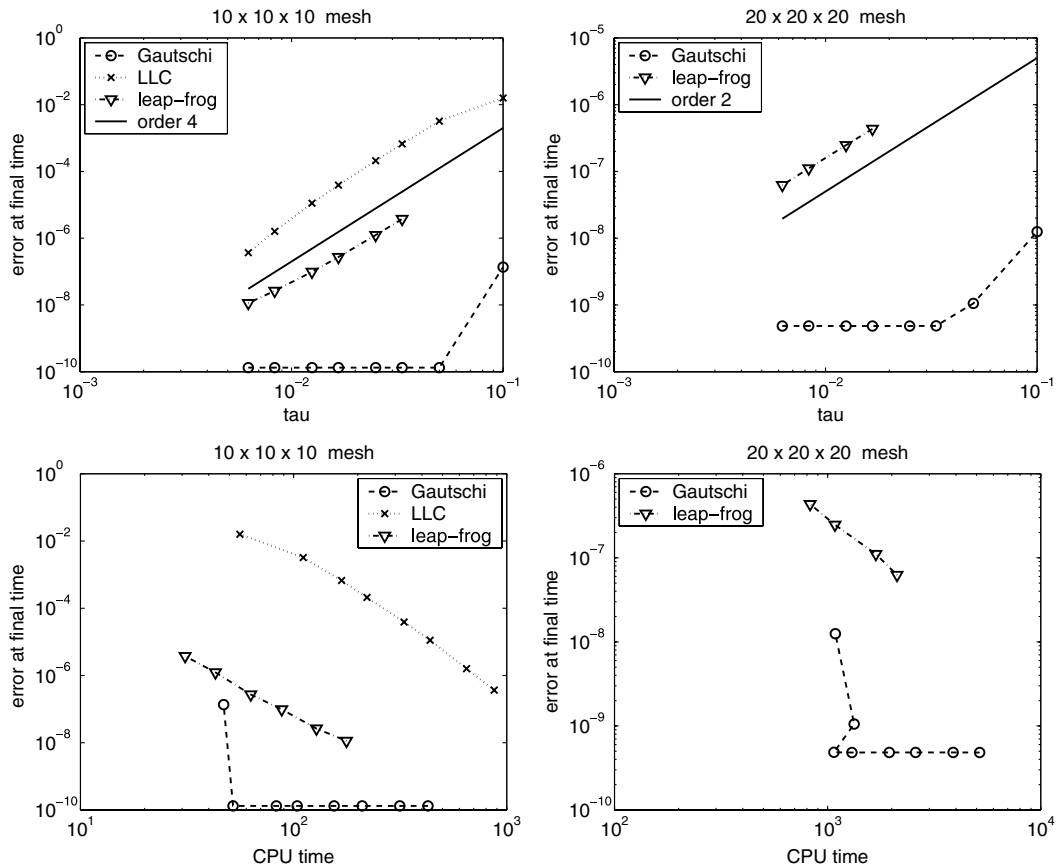


Fig. 18. Errors at the final time against the corresponding step sizes and the required CPU times for a constant source term (cf. (6.6)).

The results obtained on the uniform cubic mesh for zero values of ω_i are presented in Fig. 18. Similar, practically undistinguishable plots were obtained for the very small frequencies (6.7). Note the superconvergence effects observed for the leap frog and the LLC schemes on the $10 \times 10 \times 10$ mesh: the schemes are almost fourth order accurate. The results clearly show that the Gautschi scheme with adaptive choice of the Krylov subspace dimension is practically exact for these problems.

7. Conclusions and suggestions for future research

It is shown that the Gautschi cosine scheme can be efficiently implemented for edge finite element discretizations of the three-dimensional Maxwell equations. The implementation involves a sparse LU (or Cholesky) factorization of the mass matrix which is also required for explicit time stepping schemes and in most cases can be done efficiently. When the direct solution is not feasible the action of the inverse of the mass matrix could also be computed by an iterative solver.

We also proposed a simple strategy for the adaptive choice of the Krylov dimension. This strategy proves to be successful in our experiments, in particular, the error triggered by the Krylov subspace approximation appears negligible to the time error. Moreover, the exactness of the Gautschi scheme for the constant inhomogeneous term was observed in practice for our Gautschi–Krylov implementation. A backward error analysis of the Krylov subspace error was done leading to an explicit formula for the error. This also provided an insight for the stopping criterion used in the Arnoldi process. Furthermore, the stability of the new scheme was proved.

Dispersion analysis presented in the paper revealed superior properties of the Gautschi scheme as compared to the leap frog and the LLC scheme.

The presented numerical experiments demonstrate that the Gautschi scheme is more efficient (in terms of the achieved accuracy and the required computational work) than the implicit LLC scheme. The Gautschi scheme is much more efficient than the explicit leap frog scheme and the LLC scheme (i) on nonuniform meshes or (ii) when the inhomogeneous source term is a slowly varying function of time.

A relevant future research topic would be an extension of the Gautschi–Krylov scheme to the Maxwell equations with nonzero conductivity terms or absorbing boundary conditions. In both cases the weak formulation (2.15) will contain a first order time derivative. A possible approach here would be to use splitting methods.

It would also be interesting to see how the Gautschi–Krylov scheme performs with the recently developed matrix function preconditioning technique [36].

The presented results indicate that the Gautschi–Krylov scheme is a promising tool for efficient time integration of the Maxwell equations.

Acknowledgement

The first author thanks Jason Frank and Gerard Sleijpen for useful discussions.

Appendix A. Stability of the leap frog scheme

To derive a stability condition for the leap frog scheme we consider the homogeneous case $\mathbf{j}(t) = 0$:

$$M_\epsilon \mathbf{e}^{n+1} + (\tau^2 A_\mu - 2M_\epsilon) \mathbf{e}^n + M_\epsilon \mathbf{e}^{n-1} = 0, \tag{A.1}$$

or in its equivalent form

$$\mathbf{e}^{n+1} + (\tau^2 M_\epsilon^{-1} A_\mu - 2I) \mathbf{e}^n + \mathbf{e}^{n-1} = 0. \tag{A.2}$$

In our analysis, we follow the standard approach based on diagonalizing the matrices involved in the scheme (see e.g. [42]). Any solution of (A.2) can be written as

$$\mathbf{e}^n = \sum_m \gamma_m^n \boldsymbol{\alpha}_m, \tag{A.3}$$

where $\boldsymbol{\alpha}_m$'s are the eigenvectors corresponding to the eigenvalues (λ_m) of the following eigenvalue problem

$$M_\epsilon^{-1} A_\mu \mathbf{x} = \lambda \mathbf{x}. \tag{A.4}$$

We assume that matrices M_ϵ and A_μ are Hermitian, M_ϵ is positive definite and A_μ is positive semidefinite. This is guaranteed by the finite element discretization provided that μ and ϵ have corresponding properties. The eigenvalues of (A.4) are then nonnegative. Substitution of (A.3) into (A.2) yields

$$\begin{aligned} & \sum_m \gamma_m^{n+1} \boldsymbol{\alpha}_m + (\tau^2 M_\epsilon^{-1} A_\mu - 2I) \sum_m \gamma_m^n \boldsymbol{\alpha}_m + \sum_m \gamma_m^{n-1} \boldsymbol{\alpha}_m \\ &= \sum_m \gamma_m^{n+1} \boldsymbol{\alpha}_m + \sum_m \gamma_m^n (\tau^2 \lambda_m - 2) \boldsymbol{\alpha}_m + \sum_m \gamma_m^{n-1} \boldsymbol{\alpha}_m = 0. \end{aligned} \tag{A.5}$$

which, due to the linear independence of the $\boldsymbol{\alpha}_m$'s, implies

$$\gamma_m^{n+1} + (\tau^2 \lambda_m - 2) \gamma_m^n + \gamma_m^{n-1} = 0 \quad \text{for all } m. \tag{A.6}$$

This recurrence is stable (i. e. $|\gamma_m^n| \leq 1$) if and only if the roots $v_{1,2}$ of its characteristic equation

$$v^2 + (\tau^2 \lambda_m - 2)v + 1 = 0 \tag{A.7}$$

do not exceed one in absolute value. The solution of (A.7) is

$$v_{1,2} = 1 - \frac{\tau^2}{2} \lambda_m \pm \sqrt{\left(1 - \frac{\tau^2}{2} \lambda_m\right)^2 - 1}. \tag{A.8}$$

A straightforward computation shows that the stability condition $|v_{1,2}| \leq 1$ is fulfilled if and only if

$$\left(1 - \frac{\tau^2}{2} \lambda_m\right)^2 - 1 \leq 0, \quad (\text{A.9})$$

which, together with (A.8), necessarily means that $|v_{1,2}| = 1$. The solutions of (A.9) satisfy

$$\tau^2 \leq \frac{4}{\lambda_m}, \quad \text{for all } m, (\lambda_m \geq 0).$$

Then the stability condition for the leap frog scheme is

$$\tau^2 \leq \frac{4}{\lambda_{\max}},$$

where λ_{\max} is the maximum eigenvalue of the matrix $M_\epsilon^{-1}A_\mu$.

Appendix B. Dispersion relation matrices F and G

The matrices F and G in (5.12) on a cubic mesh with element size $h \times h \times h$ are given as the matrix F is diagonal, with entries

$$\begin{aligned} F_{11} &= \frac{1}{9} \cos(hk_2) \cos(hk_3) + \frac{2}{9} \cos(k_3h) + \frac{2}{9} \cos(k_2h) + \frac{4}{9}, \\ F_{22} &= \frac{1}{9} \cos(hk_1) \cos(hk_3) + \frac{2}{9} \cos(k_3h) + \frac{2}{9} \cos(k_1h) + \frac{4}{9}, \\ F_{33} &= \frac{1}{9} \cos(hk_1) \cos(hk_2) + \frac{2}{9} \cos(k_2h) + \frac{2}{9} \cos(k_1h) + \frac{4}{9}, \end{aligned}$$

the matrix G is complex Hermitian with entries

$$G = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ \bar{g}_{12} & g_{22} & g_{23} \\ \bar{g}_{13} & \bar{g}_{23} & g_{33} \end{pmatrix},$$

where \bar{g} denotes the complex conjugate of g and

$$\begin{aligned} g_{11} &= \frac{8}{3} - \frac{2}{3} \cos(h(k_2 - k_3)) - \frac{2}{3} \cos(hk_2) - \frac{2}{3} \cos(hk_3) - \frac{2}{3} \cos(h(k_2 + k_3)), \\ g_{12} &= -\frac{2}{3} + \frac{1}{6} e^{-ih(k_2+k_3)} - \frac{1}{6} e^{-ih(-k_1+k_2+k_3)} - \frac{2}{3} e^{-ih(-k_1+k_2)} + \frac{2}{3} e^{-ihk_2} + \frac{1}{6} e^{-ih(-k_1+k_3)} - \frac{1}{6} e^{-ihk_3} \\ &\quad - \frac{1}{6} e^{-ih(-k_1+k_2-k_3)} + \frac{1}{6} e^{-ih(k_2-k_3)} + \frac{1}{6} e^{ih(k_1+k_3)} - \frac{1}{6} e^{ihk_3} + \frac{2}{3} e^{ik_1h}, \\ g_{13} &= -\frac{2}{3} + \frac{1}{6} e^{-ih(k_2+k_3)} - \frac{1}{6} e^{-ih(-k_1+k_2+k_3)} + \frac{1}{6} e^{-ih(-k_1+k_2)} - \frac{1}{6} e^{-ihk_2} - \frac{2}{3} e^{-ih(-k_1+k_3)} \\ &\quad + \frac{2}{3} e^{-ihk_3} + \frac{1}{6} e^{ih(k_1+k_2)} - \frac{1}{6} e^{ih(k_1+k_2-k_3)} + \frac{1}{6} e^{ih(k_2-k_3)} - \frac{1}{6} e^{ihk_2} + \frac{2}{3} e^{ik_1h}, \\ g_{22} &= -\frac{2}{3} \cos(k_1h) + \frac{8}{3} - \frac{2}{3} \cos(h(-k_1 + k_3)) - \frac{2}{3} \cos(hk_3) - \frac{2}{3} \cos(h(k_1 + k_3)), \\ g_{23} &= -\frac{2}{3} - \frac{1}{3} \cos(k_1h) + \frac{1}{6} e^{-ih(-k_1+k_3)} + \frac{2}{3} e^{-ihk_3} + \frac{1}{6} e^{ih(k_1+k_2)} - \frac{1}{6} e^{ih(k_1+k_2-k_3)} - \frac{2}{3} e^{ih(k_2-k_3)} \\ &\quad + \frac{2}{3} e^{ihk_2} + \frac{1}{6} e^{ih(-k_1+k_2)} - \frac{1}{6} e^{ih(-k_1+k_2-k_3)} + \frac{1}{6} e^{-ih(k_1+k_3)}, \\ g_{33} &= -\frac{2}{3} \cos(h(k_1 + k_2)) + \frac{8}{3} - \frac{2}{3} \cos(k_1h) - \frac{2}{3} \cos(hk_2) - \frac{2}{3} \cos(h(-k_1 + k_2)). \end{aligned}$$

References

- [1] M. Ainsworth, Dispersive properties of high order Nédélec/edge element approximation of time-harmonic Maxwell equations, *Philos. Trans. Roy. Soc. Ser. A* 362 (1816) (2004) 471–493.
- [2] R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H.A. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, PA, 1994. Available from: <http://www.netlib.org/templates/>.
- [3] M.A. Botchev, G.L.G. Sleijpen, H.A. van der Vorst, Stability control for approximate implicit time stepping schemes with minimum residual iterations, *Appl. Numer. Math.* 31 (3) (1999) 239–253.
- [4] G.C. Cohen, *Higher-Order Numerical Methods for Transient Wave Equations*, Springer, Berlin, 2002.
- [5] H. De Raedt, K. Michielsen, J.S. Kole, M.T. Figge, One-step finite-difference time-domain algorithm to solve the Maxwell equations, *Phys. Rev. E* 67 (2003) 056706.
- [6] V.L. Druskin, L.A. Knizhnerman, Two polynomial methods of calculating functions of symmetric matrices, *U.S.S.R. Comput. Math. Math. Phys.* 29 (6) (1989) 112–121.
- [7] V.L. Druskin, L.A. Knizhnerman, Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic, *Numer. Linear Algebra Appl.* 2 (1995) 205–217.
- [8] V.L. Druskin, L.A. Knizhnerman, Extended Krylov subspaces: approximation of the matrix square root and related functions, *SIAM J. Matrix Anal. Appl.* 19 (3) (1998) 755–771 (electronic).
- [9] A. Fisher, R.N. Rieban, G.H. Rodrigue, D.A. White, A generalized mass lumping technique for vector finite-element solutions of the time-dependent Maxwell equations, *IEEE Trans. Antennas Propagat.* 53 (9) (2005) 2900–2910.
- [10] F.R. Gantmacher, *The Theory of Matrices*, vol. 1, AMS Chelsea Publishing, Providence, RI, 1998, Translated from the Russian by K.A. Hirsch, Reprint of the 1959 translation.
- [11] W. Gautschi, Numerical integration of ordinary differential equations based on trigonometric polynomials, *Numer. Math.* 3 (1961) 381–397.
- [12] S.D. Gedney, U. Navsariwala, An unconditionally stable finite element time-domain solution of the vector wave equation, *IEEE Microwave Guided Wave Lett.* 5 (10) (1995) 332–334.
- [13] S.K. Godunov, V.S. Ryabenkii, *Difference Schemes. An Introduction to the Underlying Theory*, Elsevier Science, Amsterdam, 1987.
- [14] G.H. Golub, C.F. Van Loan, *Matrix Computations*, third ed., The Johns Hopkins University Press, Baltimore and London, 1996.
- [15] V. Grimm, On error bounds for the Gautschi-type exponential integrator applied to oscillatory second-order differential equations, *Numer. Math.* 100 (2005) 71–89.
- [16] M. Hochbruck, C. Lubich, On Krylov subspace approximations to the matrix exponential operator, *SIAM J. Numer. Anal.* 34 (5) (1997) 1911–1925.
- [17] M. Hochbruck, C. Lubich, A Gautschi-type method for oscillatory second-order differential equations, *Numer. Math.* 83 (1999) 403–426.
- [18] M. Hochbruck, C. Lubich, H. Selhofer, Exponential integrators for large systems of differential equations, *SIAM J. Sci. Comput.* 19 (5) (1998) 1552–1574.
- [19] R. Horváth, Uniform treatment of numerical time-integrations of the Maxwell equation, in: W.H.A. Schilders, J.W. ter Maten, S.H.M.J. Houben (Eds.), *Proceedings of the Conference “Scientific Computing in Electrical Engineering” SCCE-2002*, Eindhoven, Springer Series Mathematics in Industry, Springer, Berlin, 2004, pp. 231–239.
- [20] R. Horváth, I. Faragó, W. Schilders, Investigation of numerical time-integrations of Maxwell’s equations using the staggered grid spatial discretization, *Int. J. Numer. Model.* 18 (2005) 149–169.
- [21] L.A. Knizhnerman, Calculation of functions of unsymmetric matrices using Arnoldi’s method, *U.S.S.R. Comput. Math. Math. Phys.* 31 (1) (1991) 1–9.
- [22] J.S. Kole, M.T. Figge, H. De Raedt, Unconditionally stable algorithms to solve the time-dependent Maxwell equations, *Phys. Rev. E* 64 (2001) 066705.
- [23] J.S. Kole, M.T. Figge, H. De Raedt, Higher-order unconditionally stable algorithms to solve the time-dependent Maxwell equations, *Phys. Rev. E* 65 (2002) 066705.
- [24] J.-F. Lee, R. Lee, A. Cangellaris, Time-domain finite-element methods, *IEEE Trans. Antennas Propagat.* 45 (3) (1997) 430–442.
- [25] P. Monk, *Finite Element Methods for Maxwell’s Equations*, Oxford University Press, Oxford, 2003.
- [26] P.B. Monk, A.K. Parrott, A dispersion analysis of finite element methods for Maxwell’s equations, *SIAM J. Sci. Comput.* 15 (4) (1994) 916–937.
- [27] P.B. Monk, A.K. Parrott, Phase-accuracy comparisons and improved far-field estimates for 3-D edge elements on tetrahedral meshes, *J. Comput. Phys.* 170 (2) (2001) 614–641.
- [28] J.-C. Nédélec, Mixed finite elements in \mathbf{R}^3 , *Numer. Math.* 35 (3) (1980) 315–341.
- [29] J.-C. Nédélec, A new family of mixed finite elements in \mathbf{R}^3 , *Numer. Math.* 50 (1) (1986) 57–81.
- [30] R.D. Richtmyer, K.W. Morton, *Difference Methods for Initial-Value Problems*, second ed., Interscience Publishers John Wiley, New York–London–Sydney, 1967.
- [31] G. Rodrigue, D. White, A vector finite element time-domain method for solving Maxwell’s equations on unstructured hexahedral grids, *SIAM J. Sci. Comput.* 23 (3) (2001) 683–706 (electronic).
- [32] Y. Saad, Analysis of some Krylov subspace approximations to the matrix exponential operator, *SIAM J. Numer. Anal.* 29 (1) (1992) 209–228.

- [33] Y. Saad, Iterative methods for sparse linear systems, Book out of print, 2000. Available from: <<http://www-users.cs.umn.edu/~saad/books.html>>.
- [34] A.S. Shvedov, V.T. Zhukov, Explicit iterative difference schemes for parabolic equations, Russian J. Numer. Anal. Math. Modell. 13 (2) (1998) 133–148.
- [35] B.P. Sommeijer, L.F. Shampine, J.G. Verwer, RKC: an explicit solver for parabolic PDEs, J. Comput. Appl. Math. 88 (1997) 315–326.
- [36] J. van den Eshof, M. Hochbruck, Preconditioning Lanczos approximations to the matrix exponential, Preprint of the Department of Mathematics, Heinrich Heine University, Duesseldorf, Germany, March, 2004, SIAM J. Sci. Comput. 27 (4) (2006) 1438–1457.
- [37] H.A. van der Vorst, An iterative solution method for solving $f(A)x = b$, using Krylov subspace information obtained for the symmetric positive definite matrix A , J. Comput. Appl. Math. 18 (1987) 249–263.
- [38] H.A. van der Vorst, Iterative Krylov Methods for Large Linear Systems, Cambridge University Press, 2003.
- [39] J.G. Verwer, Explicit Runge–Kutta methods for parabolic partial differential equations, Appl. Numer. Math. 22 (1996) 359–379.
- [40] P. Wesseling, Principles of Computational Fluid Dynamics, Springer, Berlin, 2001.
- [41] K.S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE Trans. Antennas Propagat. 14 (3) (1966) 302–307.
- [42] O.C. Zienkiewicz, I. Morgan, Finite Elements and Approximations, John Wiley, New York, 1983.