Department of Applied Mathematics
Faculty of EEMCS

University of Twente
The Netherlands

P.O. Box 217
7500 AE Enschede
The Netherlands
Phone: +31-53-4893400
Fax: +31-53-4893114
Email: memo@math.utwente.nl
www.math.utwente.nl/publications

Memorandum No. 1762

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

M.A. Botchev, D. Harutyunyan and
J.J.W. van der Vegt

May, 2005

ISSN 0169-2690

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

M.A. Botchev [a,*,1], D. Harutyunyan [a,2], J.J.W. van der Vegt [a,1]

[a] *Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands,*
[m.a.botchev, d.harutyunyan, j.j.w.vandervegt]@math.utwente.nl.

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). 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.

Key words: Maxwell equations, Gautschi cosine scheme, dispersion analysis, edge elements, staggered leap frog scheme.
MSC: 35Q60, 78M10, 65M12, 37M15, 15A15

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

* Corresponding author.

[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).

elements [19,20]. 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 [18]. We emphasize, however, that the time integration techniques presented in this paper are applicable to any space-discretized second order wave equation(s).

Many time stepping schemes exist for the time integration of the space-discretized Maxwell equations [31,8,17,15,16,3,13]. 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 [8,17]. Stable time stepping schemes for the Maxwell equations have been also of importance in connection with finite difference spatial discretizations [15,16,3,13]. A scheme proposed by Gautschi [7] has recently received attention in the literature for the solution of second order highly oscillatory ODE's [12,11]. 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 [27,4,14,22,5,10,6,12,26]. The time error of the scheme is of second order uniformly in the frequencies [11] 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.

Attractive properties of the new schemes are confirmed by a dispersion analysis done for the 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 (in the sequel, the LLC scheme) [8,17].

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, dispersion errors of the three schemes are investigated in Section 4, and 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 domain $\Omega \subset \mathbb{R}^3$:

$$\partial_t \boldsymbol{D}_s = \nabla \times \boldsymbol{H}_s - \boldsymbol{J}_s, \tag{2.1}$$
$$\partial_t \boldsymbol{B}_s = -\nabla \times \boldsymbol{E}_s, \tag{2.2}$$
$$\nabla \cdot \boldsymbol{D}_s = \rho_s, \tag{2.3}$$
$$\nabla \cdot \boldsymbol{B}_s = 0, \tag{2.4}$$

where $\boldsymbol{E}_s$ and $\boldsymbol{H}_s$ ($\boldsymbol{D}_s$ and $\boldsymbol{B}_s$) are electric and magnetic fields (respectively, the electric and the magnetic flux densities). Furthermore, $\boldsymbol{J}_s$ 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:

$$(\boldsymbol{n} \times \boldsymbol{E}_s)|_\Gamma = 0, \tag{2.5}$$
$$\boldsymbol{E}_s|_{t_s=0} = \bar{\boldsymbol{E}}_0, \qquad \boldsymbol{H}_s|_{t_s=0} = \bar{\boldsymbol{H}}_0, \tag{2.6}$$

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

$$\boldsymbol{D}_s = \epsilon \boldsymbol{E}_s, \quad \boldsymbol{B}_s = \mu \boldsymbol{H}_s, \tag{2.7}$$

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:

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

where L is a reference length (expressed in meters), and $c_0 = (\epsilon_0\mu_0)^{-1/2} \approx 3 \cdot 10^8$ 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

$$\boldsymbol{E}_s(\mathbf{x}_s, t_s) = \frac{\tilde{H}_0}{Z_0^{-1}}\boldsymbol{E}(\mathbf{x}, t), \quad \boldsymbol{H}_s(\mathbf{x}_s, t_s) = \tilde{H}_0\boldsymbol{H}(\mathbf{x}, t), \quad \boldsymbol{J}_s(\mathbf{x}_s, t_s) = \frac{\tilde{H}_0}{L}\boldsymbol{J}(\mathbf{x}, t), \tag{2.9}$$

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]. Equations (2.1),(2.2) and constitutive relations (2.7) written for the scaled quantities yield the following dimensionless Maxwell equations:

$$\epsilon_r\partial_t\boldsymbol{E} = \nabla \times \boldsymbol{H} - \boldsymbol{J}, \tag{2.10}$$

$$\mu_r\partial_t\boldsymbol{H} = -\nabla \times \boldsymbol{E}. \tag{2.11}$$

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 $\boldsymbol{H}$ from the system (2.10),(2.11) and obtain a second-order hyperbolic partial differential equation for $\boldsymbol{E}$

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

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

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

2.2 Weak formulation and finite element discretization

Defining the space

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

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

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

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

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

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

where each basis function $\boldsymbol{w}_j(x)$ is defined with respect to the edge j as a linear polynomial such that [19,18]:

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

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

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

The discretized version of (2.14) then reads:

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

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

Denoting by $\boldsymbol{e}(t)$ 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 (ODE's)

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

with

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

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 [17,8].

3.1 Leap frog scheme

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

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

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

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

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

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

where $\lambda_{\max}$ 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

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

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 $\boldsymbol{y}$ defined in one of the following ways:

$$\tilde{A}_{\epsilon,\mu} = U_\epsilon^{-1} L_\epsilon^{-1} A_\mu, \qquad \boldsymbol{y} = \boldsymbol{e}, \qquad \boldsymbol{f} = U_\epsilon^{-1} L_\epsilon^{-1} \boldsymbol{j}, \tag{3.4}$$

$$\tilde{A}_{\epsilon,\mu} = L_\epsilon^{-1} A_\mu U_\epsilon^{-1}, \qquad \boldsymbol{y} = U_\epsilon \boldsymbol{e}, \qquad \boldsymbol{f} = L_\epsilon^{-1} \boldsymbol{j}, \tag{3.5}$$

$$\tilde{A}_{\epsilon,\mu} = A_\mu U_\epsilon^{-1} L_\epsilon^{-1}, \qquad \boldsymbol{y} = L_\epsilon U_\epsilon \boldsymbol{e}, \qquad \boldsymbol{f} = \boldsymbol{j}, \tag{3.6}$$

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 transforma-

tions (3.4), (3.5), (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 [21] 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 [28] or Chapter 3.1 in [1]).*

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 [7,11] for an ODE system in the normal form (3.3):

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

where the function ψ is given by

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

For a derivation of the scheme we refer to [11].

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

Since the matrix $\tilde{A}_{\epsilon,\mu}$ is large and sparse, computation of $\psi(\tau^2 \tilde{A}_{\epsilon,\mu})\boldsymbol{v}$ by conventional methods (see e.g. [9], 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 [27,4,14,22,5,10,6,12], see also Chapter 11 in the recent book [28]).

Throughout this subsection we denote $A = \tau^2 \tilde{A}_{\epsilon,\mu}$, $A \in \mathbb{R}^{N\times N}$. Computation of $\psi(A)\boldsymbol{v}$ for a given vector $\boldsymbol{v}$ is based on the Arnoldi or, when $A = A^*$, on the Lanczos process (see e.g. [28,23]). 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 $\boldsymbol{v}$, the Arnoldi process generates after m steps orthonormal vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{m+1}$ (with $\boldsymbol{v}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|$) and a Hessenberg matrix $\bar{H}_m \in \mathbb{R}^{(m+1)\times m}$ such that (see [28,23])

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

where $V_{m+1} \in \mathbb{R}^{N\times m+1}$ is a matrix with column vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{m+1}$ (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

$$AV_m \approx V_m H_m, \tag{3.9}$$

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

$$\boldsymbol{v} = V_m y, \qquad y = \|\boldsymbol{v}\|e_1,$$

with e_1 being the first canonical basis vector in $\mathbb{R}^m$, and (cf. (3.9))

$$\psi(A)V_m y \approx V_m \psi(H_m)y, \qquad y = \|\boldsymbol{v}\|e_1,$$

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

$$\psi(A)\boldsymbol{v} \approx \|\boldsymbol{v}\|V_m\psi(H_m)e_1. \tag{3.10}$$

We emphasize that dependence of the orthonormal basis $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m$ on $\boldsymbol{v}$ is crucial to have a good approximation in (3.10).

In practice m is small (say 20), so that $\psi(H_m)$ in (3.10) can easily be computed by a standard method (see e.g. Chapter 11 in [9] and references therein). In the experiments presented in this paper, $\psi(H_m)$ was computed with Matlab's build-in functions `sqrtm` and `funm`.

An important question is when to stop the Arnoldi process. One stopping criterion is proposed in [12] 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

$$\left\| \frac{\boldsymbol{y}_{(m)}^{n+1} - \boldsymbol{y}_{(m-1)}^{n+1}}{\boldsymbol{y}_{(m)}^{n+1} - \boldsymbol{y}_{(0)}^{n+1}} \right\|_\infty \leqslant \texttt{TOL}, \tag{3.11}$$

where $\boldsymbol{y}_{(m)}^{n+1}$ 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 $\texttt{TOL} = 10^{-2}$, this value should be chosen according to the relative accuracy required for a specific problem). By $\boldsymbol{y}_{(0)}^{n+1}$ 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 $\boldsymbol{y}_{(0)}^{n+1}$ coincides with the solution of the leap frog scheme (cf. (3.1)) and, thus, is a second order time-consistent numerical solution. Stopping criterion (3.11) 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 $\boldsymbol{y}_{(0)}^{n+1}$.

The described steps lead to the algorithm for the Gautschi-Krylov time integration scheme presented in Figure 1.

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.

3.4 LLC scheme

The following scheme proposed by J.-F. Lee, R. Lee, and A. Cangellaris (the LLC scheme, [17] and [8]) can be applied directly to the semidiscrete Maxwell equations (2.16):

$$M_\epsilon \frac{\boldsymbol{e}^{n+1} - 2\boldsymbol{e}^n + \boldsymbol{e}^{n-1}}{\tau^2} + A_\mu\left(\frac{1}{4}\boldsymbol{e}^{n-1} + \frac{1}{2}\boldsymbol{e}^n + \frac{1}{4}\boldsymbol{e}^{n+1}\right) = \boldsymbol{j}^n. \tag{3.12}$$

This scheme can be written in the form

$$(M_\epsilon + \frac{\tau^2}{4}A_\mu)\boldsymbol{e}^{n+1} = \tau^2\boldsymbol{j}^n - (\frac{\tau^2}{2}A_\mu - 2M_\epsilon)\boldsymbol{e}^n - (M_\epsilon + \frac{\tau^2}{4}A_\mu)\boldsymbol{e}^{n-1}, \tag{3.13}$$

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 [17].

$\boldsymbol{y}^n$ and $\boldsymbol{y}^{n-1}$ are given
$\boldsymbol{v} = \tau^2(-\tilde{A}_{\epsilon,\mu}\boldsymbol{y}^n + \boldsymbol{f}^n), \quad \beta = \|\boldsymbol{v}\|_2$
$\boldsymbol{y}_{(0)}^{n+1} = 2\boldsymbol{y}^n - \boldsymbol{y}^{n-1} + \boldsymbol{v}$
for $m = 1, \ldots,$
    extend the Krylov basis by one Arnoldi step:
        if$(m = 1)$ then
            $\boldsymbol{v}_1 = \boldsymbol{v}/\beta$
            initialize $\bar{H}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
        else
            extend $\bar{H}_{m-1}$ to $\bar{H}_m$ by adding
            zero column and zero row
        endif
        $\boldsymbol{w} = \tau^2 \tilde{A}_{\epsilon,\mu}\boldsymbol{v}_m$
        for $i = 1, \ldots, m$
            $h_{i,m} = \boldsymbol{w}^T \boldsymbol{v}_i$
            $\boldsymbol{w} = \boldsymbol{w} - h_{i,m}\boldsymbol{v}_i$
        endfor
        $h_{m+1,m} = \|\boldsymbol{w}\|_2$
        $\boldsymbol{v}_{m+1} = \boldsymbol{w}/h_{m+1,m}$
        $V_{m+1} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m, \boldsymbol{v}_{m+1}]$
    end of Arnoldi step
    compute matrix function $\psi(H_m)$
    $\boldsymbol{u} = V_m[\beta\psi(H_m)e_1]$
    $\boldsymbol{y}_{(m)}^{n+1} = 2\boldsymbol{y}^n - \boldsymbol{y}^{n-1} + \boldsymbol{u}$
    exit for-loop if condition (3.11) is fulfilled
endfor
$\boldsymbol{y}^{n+1} = \boldsymbol{y}_{(m)}^{n+1}$

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

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 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 $\boldsymbol{e}^{-1}$.

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. Introducing an auxiliary variable

$$\boldsymbol{u}^{n+1/2} = \frac{\boldsymbol{e}^{n+1} - \boldsymbol{e}^{n}}{\tau}, \tag{3.14}$$

we can write (3.12) as

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

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

$$\begin{aligned} M_\epsilon \frac{\boldsymbol{u}^{n} - \boldsymbol{u}^{n-1/2}}{\tau/2} + A_\mu \frac{\boldsymbol{e}^{n-1} + \boldsymbol{e}^{n}}{2} &= \boldsymbol{j}^{n}, \\ M_\epsilon \frac{\boldsymbol{u}^{n+1/2} - \boldsymbol{u}^{n}}{\tau/2} + A_\mu \frac{\boldsymbol{e}^{n} + \boldsymbol{e}^{n+1}}{2} &= \boldsymbol{j}^{n}. \end{aligned} \tag{3.15}$$

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

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

which, together with (3.14) and (3.15) leads to the following *one-step formulation of the LLC scheme*:

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

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

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

The one-step formulations for the leap frog and the Gautschi scheme can be

obtained along the same lines (see also [11]):

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

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

4 Dispersion Analysis

For PDE's 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) Equation (2.12) is given in an infinite source free ($\boldsymbol{J} \equiv 0$) region with periodic boundary conditions:

$$\epsilon_r \partial_{tt} \boldsymbol{E} + \nabla \times (\mu_r^{-1} \nabla \times \boldsymbol{E}) = 0. \tag{4.1}$$

(2) μ_r and ϵ_r are constant scalars.

A vector field

$$\boldsymbol{E}(x, y, z, t) = \boldsymbol{E}_0 \exp(i(\mathbf{k} \cdot \mathbf{x} - \omega t)), \quad \text{where } i = \sqrt{-1}, \tag{4.2}$$

is a solution of (4.1) if the dispersion relation

$$\omega^2 = c_r^2 k^2 \tag{4.3}$$

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.

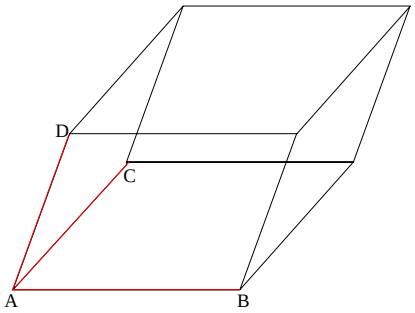

Fig. 2. Deformed element with deformation angles $\angle CAB$ and $\angle DAB$. The angle $\angle DAC = 90°$.

We consider the finite element discretization of (4.1) on a uniform parallelepiped mesh with elements of size $h \times h \times h$, see Figure 2. The angles $\angle DAB$ and $\angle CAB$ are called deformation angles.

Remark 4.0.1 *To avoid cumbersome expressions, we present many of the formulas for the cubic case $\angle DAB = \angle CAB = 90°$. 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 computatons 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 $A_\mu = \dfrac{1}{h\mu_r} A$, where the matrices M and A do not depend on the element size h. This results in the following system of ODE's

$$M\boldsymbol{e}'' + \frac{c_r^2}{h^2} A\boldsymbol{e} = 0. \tag{4.4}$$

The time exact dispersion equation is

$$-\omega^2 M\boldsymbol{e} + \frac{c_r^2}{h^2} A\boldsymbol{e} = 0. \tag{4.5}$$

We end up with an eigenvalue problem with large sparse matrices given in (4.4). 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 $e_j(t) = \displaystyle\int_{\text{edge } j} \boldsymbol{E}(\mathbf{x}, t) \cdot \boldsymbol{t}_j ds$. If the exact solution of (4.1) is given by (4.2) then for any two parallel edges p and j the expansion coefficients satisfy

$$e_p^{n+q} = \exp\left(i(\mathbf{k} \cdot \Delta_{pj} - \omega q \tau)\right) e_j^n, \tag{4.6}$$

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

4.1 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 5.3). Hence, we consider the time-accurate dispersion relation (4.5) for the system (4.4), which gives us the following generalized eigenvalue problem

$$-\omega^2 M\boldsymbol{e}^n + \frac{c_r^2}{h^2} A\boldsymbol{e}^n = 0. \tag{4.7}$$

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

$$\varphi(\omega) M\boldsymbol{e}^n + \eta A\boldsymbol{e}^n = 0. \tag{4.8}$$

Using the relations (4.6) it is not difficult to see that on a uniform grid the equations (4.8) are the same (up to a constant $\tilde{C}_{pj}$) for parallel edges, i. e. for any two parallel edges p and j holds:

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

where $M(a_j,:)$ denotes a_jth 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 Figure 3).

Let

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

then using (4.6) 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

$$\varphi(\omega) M(a_1,:)\boldsymbol{e}^n + \eta A(a_1,:)\boldsymbol{e}^n = 0. \tag{4.9}$$

The matrices M and A have a sparse structure because in (4.9) coefficients only of those basis functions are present which have nonempty common support

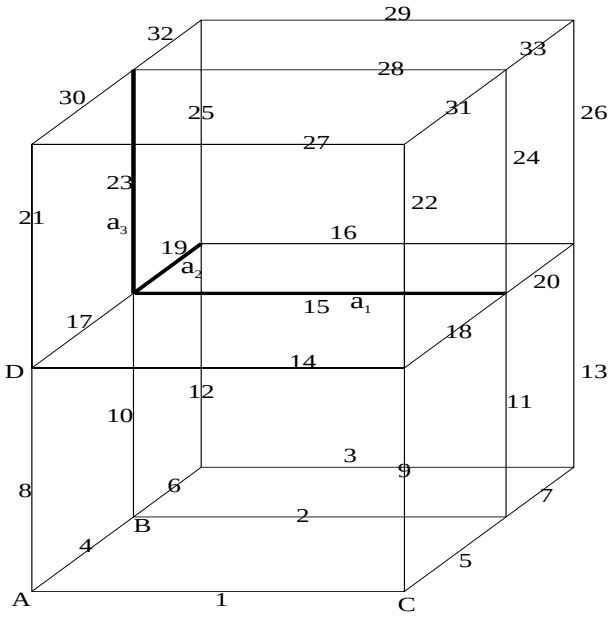

Fig. 3. Three nonparallel edges a_1, a_2, a_3 and the degrees of freedom (with a local numbering) that appear in equation (4.7) for edge a_1.

with the basis function corresponding to the edge a_1. On a cubic mesh we have

$$\begin{aligned} M(a_1,:)\boldsymbol{e}^n =& \frac{1}{36}(1,4,1,4,16,4,1,4,1)\cdot(\tilde{e}_1,\tilde{e}_2,\tilde{e}_3,\tilde{e}_{14},\tilde{e}_{15},\tilde{e}_{16},\tilde{e}_{27},\tilde{e}_{28},\tilde{e}_{29})^T,\\ A(a_1,:)\boldsymbol{e}^n =& \frac{1}{6}(-2,-2,-2,1,-1,-1,1,1,-1,4,-4,1,-1,-2,16,-2,\\ & 4,-4,-4,4,-1,1,-4,4,-1,1,-2,-2,-2,1,-1,-1,1)\\ & \cdot(\tilde{e}_1,\tilde{e}_2,\tilde{e}_3,\ldots,\tilde{e}_{32},\tilde{e}_{33})^T. \end{aligned} \tag{4.10}$$

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 (4.9) for edges a_2 and a_3 and using (4.6), we obtain a homogeneous system of equations

$$(\varphi(\omega)F + \eta G)\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = 0. \tag{4.11}$$

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

$$\begin{aligned} &\det\left(\varphi(\omega)F + \eta G\right) = 0, \qquad \text{or} \\ &\det\left(-\omega^2 F + \frac{c_r^2}{h^2}G\right) = 0, \end{aligned} \tag{4.12}$$

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 (4.12) satisfy

$$(\omega_h h)^2 = 18 \frac{4 - \cos\xi_3 \cos\xi_2 - \cos\xi_1 \cos\xi_2 \cos\xi_3 - \cos\xi_3 \cos\xi_1 - \cos\xi_1 \cos\xi_2}{(2 + \cos\xi_1)\,(2 + \cos\xi_2)\,(2 + \cos\xi_3)} c_r^2, \tag{4.13}$$

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 Figure 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 $\epsilon_r = \mu_r = 1$.

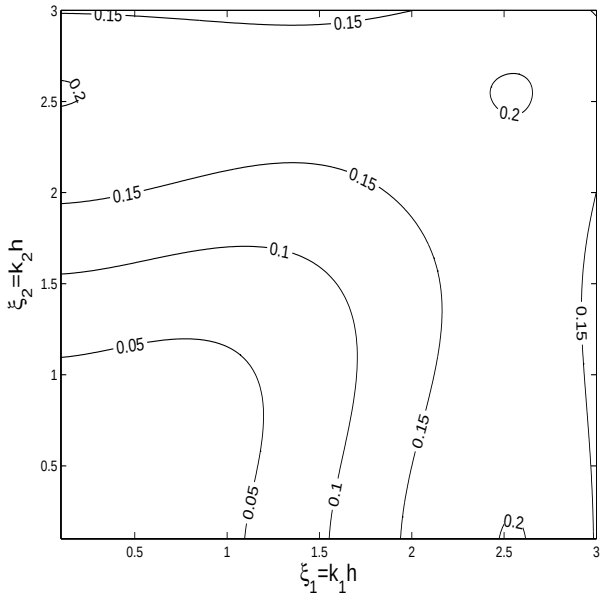

Fig. 4. The phase velocity error of the Gautschi scheme for cubic elements.

Under the assumption $|kh| \ll 1$ the Taylor expansion of (4.13) shows

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

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

Applying relation (4.6) to the leap frog scheme (3.1), we have

$$\begin{aligned}\frac{\boldsymbol{e}^{n+1}-2\boldsymbol{e}^{n}+\boldsymbol{e}^{n-1}}{\tau^2} &= \frac{\exp(-i\omega\tau)\boldsymbol{e}^{n}-2\boldsymbol{e}^{n}+\exp(i\omega\tau)\boldsymbol{e}^{n}}{\tau^2}\\ &= \frac{2(\cos(\omega\tau)-1)}{\tau^2}\boldsymbol{e}^{n}. \end{aligned} \tag{4.14}$$

Then the generalized eigenvalue problem of the leap frog scheme is

$$\frac{2(\cos(\omega\tau)-1)}{\tau^2}M\boldsymbol{e}^{n}+\frac{c_r^2}{h^2}A\boldsymbol{e}^{n}=0. \tag{4.15}$$

Introducing $\varphi(\omega)=\dfrac{2(\cos(\omega\tau)-1)}{\tau^2}$ and $\eta=\dfrac{c_r^2}{h^2}$ in (4.8) we obtain the dispersion equation for the leap frog scheme

$$\det\left(\frac{2(\cos(\omega\tau)-1)}{\tau^2}F+\frac{c_r^2}{h^2}G\right)=0, \tag{4.16}$$

with the 3×3 matrices F and G defined as in (4.12). There are 3 roots, one is zero which is non physical. The solution of (4.16) satisfies (on a cubic mesh)

$$\cos(\omega\tau)=1-2\frac{\chi_1(\tau,h,\mathbf{k})}{\chi_2(\tau,h,\mathbf{k})}, \tag{4.17}$$

where

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

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

According to the exact dispersion relation (4.3), we would like to have only real solutions ω of (4.17). Otherwise, as it is clear from (4.2), the imaginary part of ω will contribute to dissipation of the solution (damping if $Im(\omega)<0$ or amplification if $Im(\omega)>0$, see e.g. [30]). The value of ω is real if and only if

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

or, equivalently,

$$\frac{c_r\tau}{h} \leqslant \frac{1}{3}\sqrt{\frac{2(2+\cos\xi_1)(2+\cos\xi_2)(2+\cos\xi_3)}{4-\cos\xi_1\cos\xi_2\cos\xi_3-\cos\xi_1\cos\xi_2-\cos\xi_2\cos\xi_3-\cos\xi_3\cos\xi_1}}. \tag{4.18}$$

Since it is always true that

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

for the inequality (4.18) to hold true it is sufficient to require that

$$\frac{c_r\tau}{h} \leqslant \frac{1}{3}, \tag{4.19}$$

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 (4.17) shows

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

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 Figures 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 Figure 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.

4.3 LLC scheme

The generalized eigenvalue problem for the LLC scheme (3.12) is

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

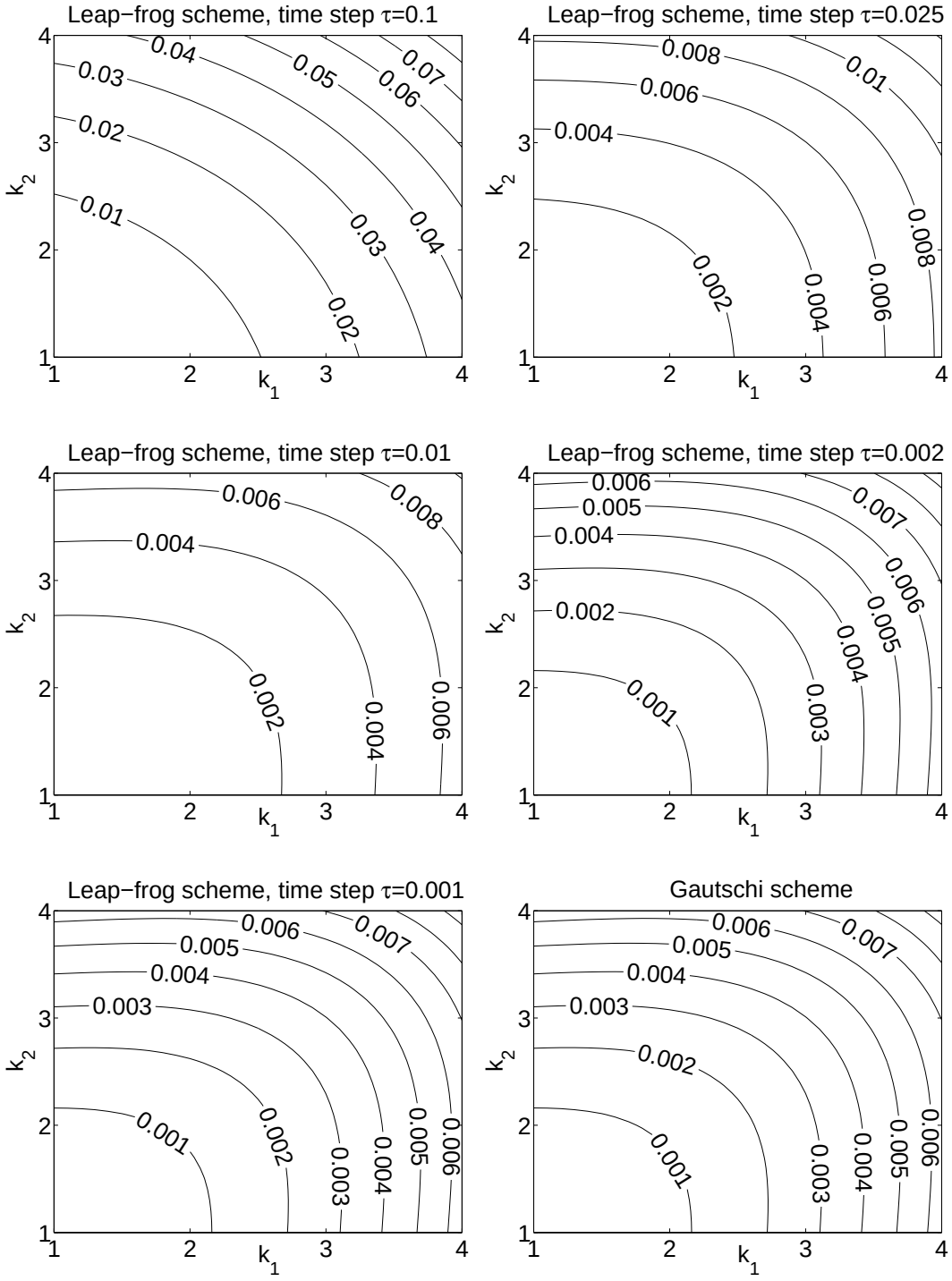

Fig. 5. Absolute value of the angular frequency errors for the leap frog scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/2$.

Introducing $\varphi(\omega) = \dfrac{2(\cos(\omega\tau) - 1)}{\tau^2}$ and $\eta = \dfrac{\cos(\omega\tau) + 1}{2}\dfrac{c_r^2}{h^2}$ in (4.8) we obtain the dispersion equation for the LLC scheme

$$\det\left(\frac{2(\cos(\omega\tau) - 1)}{\tau^2}F + \frac{(\cos(\omega\tau) + 1)}{2}\frac{c_r^2}{h^2}G\right) = 0, \qquad (4.20)$$

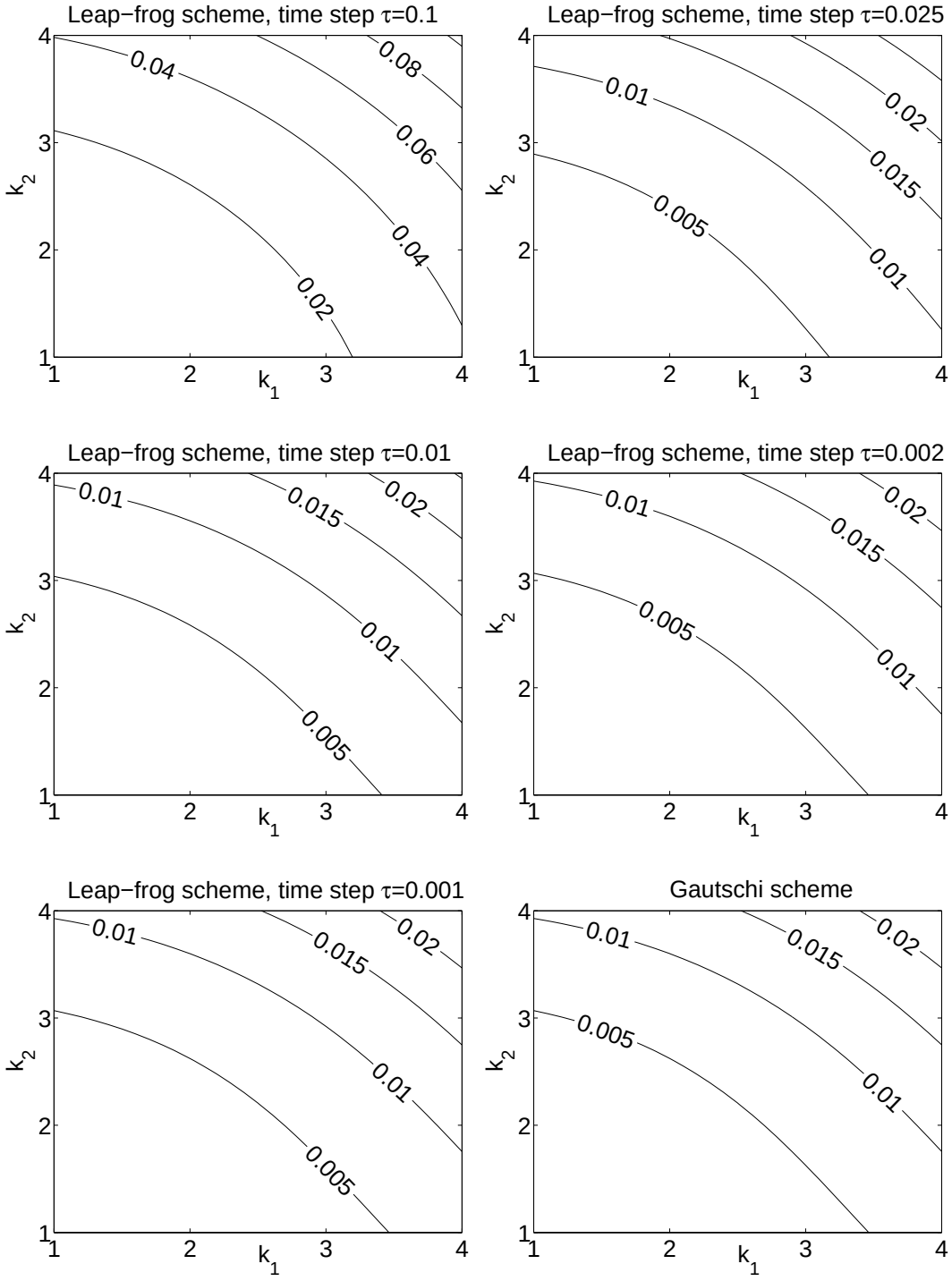

Fig. 6. Absolute value of the angular frequency errors for the leap frog scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/3$.

where the 3×3 matrices F and G are given as in (4.12). There are 3 roots, one is zero. The solution of (4.20) satisfies (on a cubic mesh)

$$\cos(\omega\tau) = \frac{\chi_2(\tau, h, \mathbf{k}) - \chi_1(\tau, h, \mathbf{k})}{\chi_2(\tau, h, \mathbf{k}) + \chi_1(\tau, h, \mathbf{k})}, \tag{4.21}$$

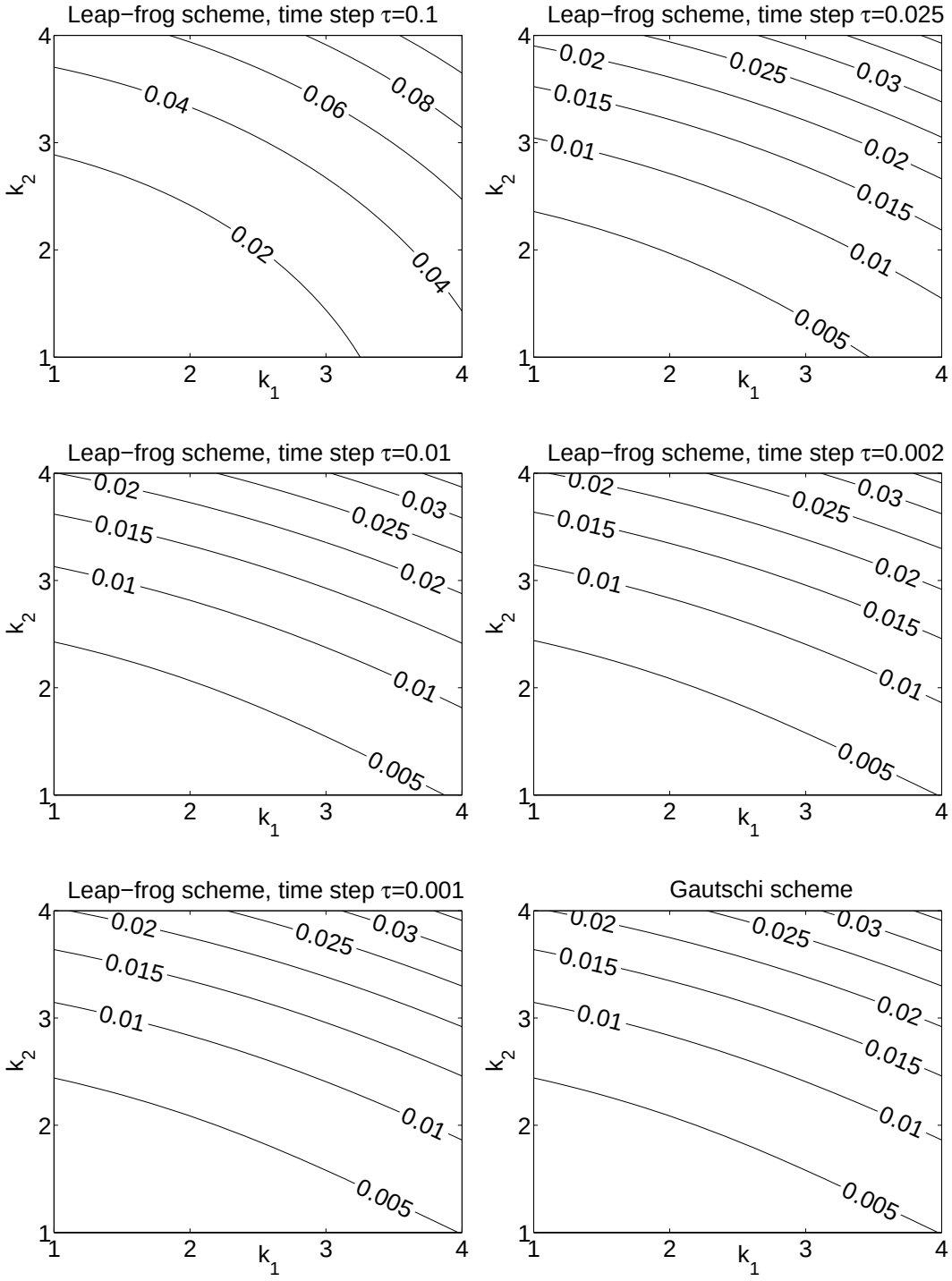

Fig. 7. Absolute value of the angular frequency errors for the leap frog scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/4$.

where

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

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

Under the assumption $|kh| \ll 1$ the Taylor expansion of (4.21) shows

$$\omega_\tau = c_r k(1 - \frac{1}{12}c_r^2 k^2 \tau^2 + \frac{1}{24}\frac{k_1^4 + k_2^4 + k_3^4}{k^2}h^2 + O(h^4) + O(\tau^4) + O(\tau^2 h^2) + \text{higher order terms}), \tag{4.22}$$

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

$$\tau = \sqrt{\frac{1}{2c_r^2}\frac{k_1^4 + k_2^4 + k_3^4}{k^4}}\, h, \tag{4.23}$$

which can be called *an optimum time step*. We note that (4.22),(**??**) are only valid on a cubic mesh.

In Figures 8–10, the absolute error of angular frequency of 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 Figure 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 Figure 8 differs significantly from the the other plots in the figure due to the increase in the error order observed in (4.22) (cf. (4.23) with $k_3 = 0$ and $k_1 \approx k_2$).

5 Numerical experiments

5.1 Test problem 1

This test problem is obtained by choosing an arbitrary vector field function $\boldsymbol{E}_{\mathsf{an}}(x, y, z, t)$ satisfying the boundary conditions, projecting it onto the finite element subspace and substituting the projection into the semidiscrete system (2.16). The source function $\boldsymbol{j}(t)$ is then chosen such that the finite element projection of $\boldsymbol{E}_{\mathsf{an}}$ 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).

More specifically, we consider the dimensionless Maxwell equations (2.12) in

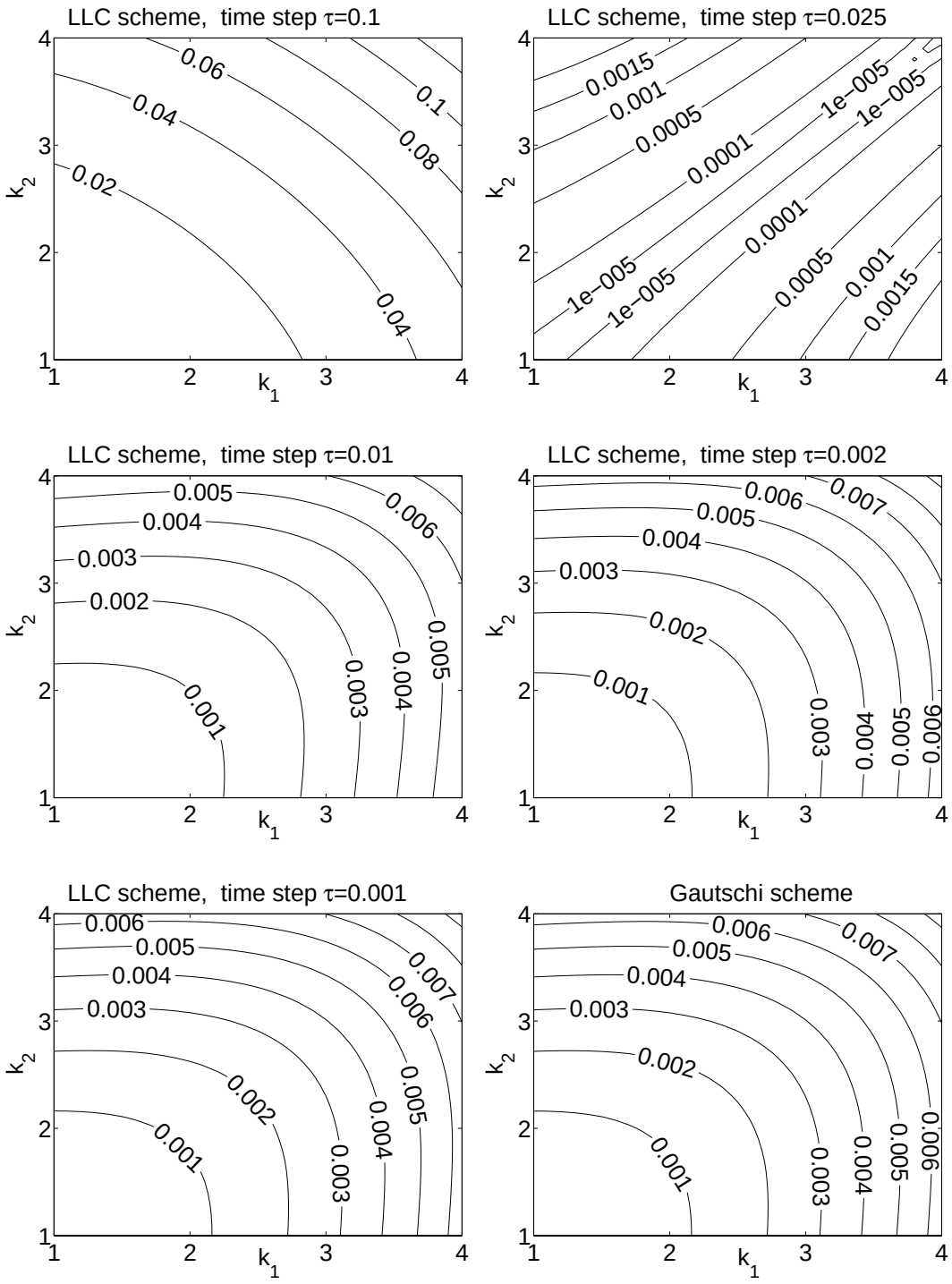

Fig. 8. Absolute value of the angular frequency errors for the LLC scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/2$. The plot for the time step $\tau = 0.025$ reflects the increase in the error order (cf. (4.23) with $k_3 = 0$ and $k_1 \approx k_2$).

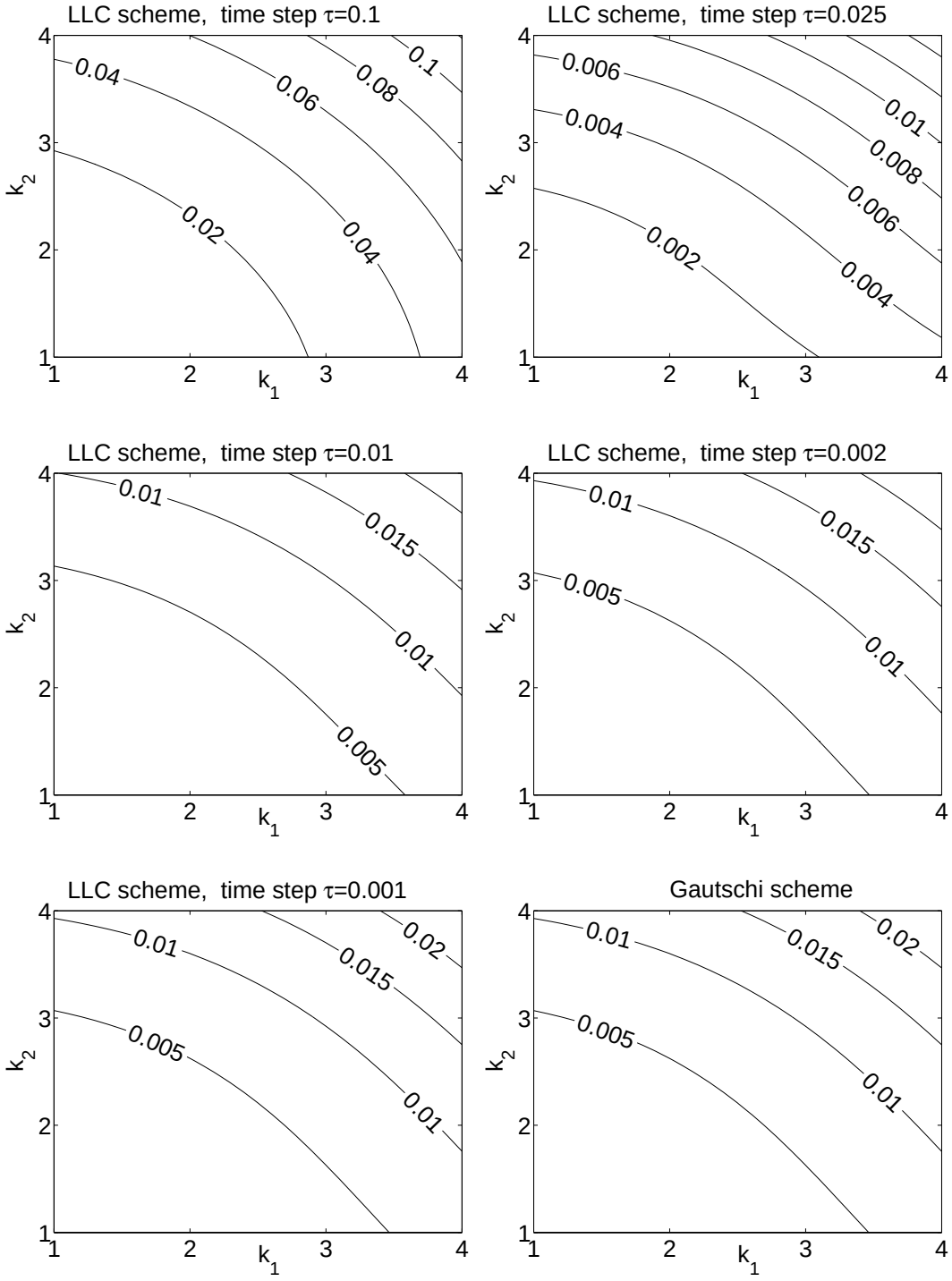

Fig. 9. Absolute value of the angular frequency errors for the LLC scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/3$.

the domain $\Omega = [0,1] \times [0,1] \times [0,1]$ and we take

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

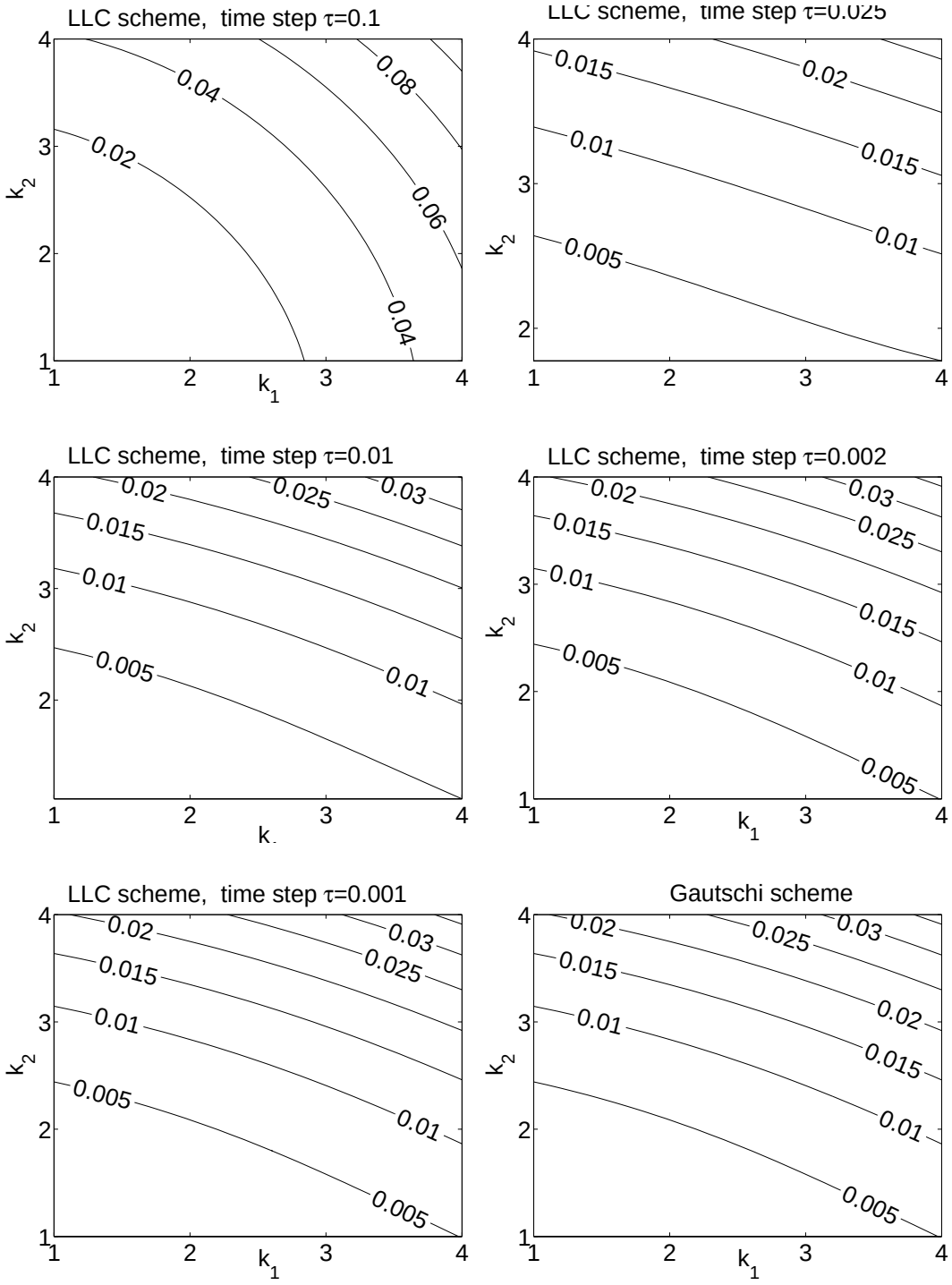

Fig. 10. Absolute value of the angular frequency errors for the LLC scheme with different time steps and for the Gautschi scheme, mesh size $h = 1/20$, deformation angle $\theta = \pi/4$.

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

$$\boldsymbol{e}_{\mathsf{an}}(t) = v(t)\bar{\boldsymbol{e}}$$

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

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

In our experiments we took

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

$$v(t) = \sum_{i=1}^{N_\omega} \cos \omega_i t, \qquad \bar{\boldsymbol{E}}(x, y, z) = \begin{bmatrix} \sin \pi y \, \sin \pi z \\ \sin \pi x \, \sin \pi z \\ \sin \pi x \, \sin \pi y \end{bmatrix}. \tag{5.1}$$

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.

5.2 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 PDE's (see e.g. [25]). This test problem is convenient when one wants to know the error at the final time.

5.3 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 $\boldsymbol{j}(t)$ (cf. 5.1) are chosen as

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

The results are presented in Figure 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 (3.11). 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 Euclidian norm of the difference

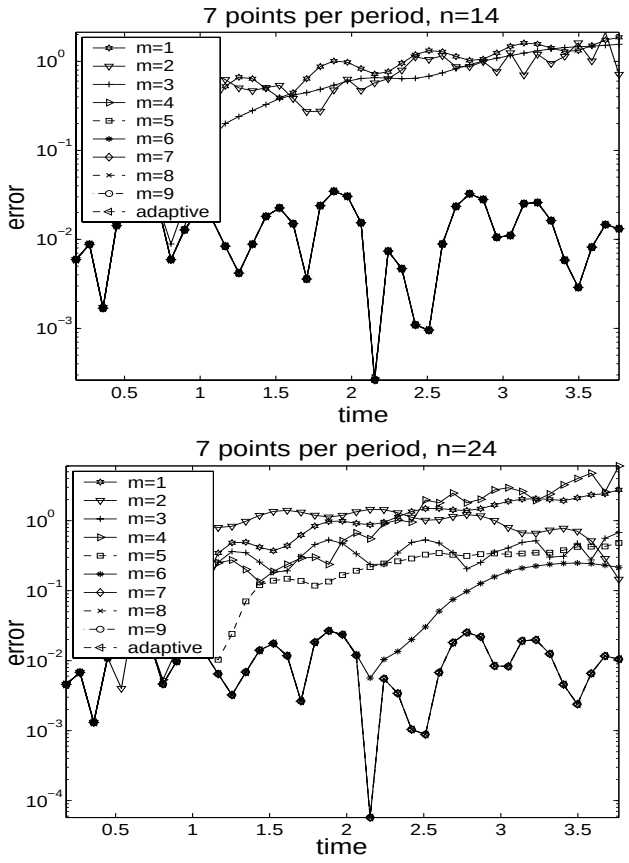

Fig. 11. Error evolution of the Gautschi scheme for $14\times14\times14$ (top) and $24\times24\times24$ (bottom) meshes for different Krylov subspace dimensions m. The step size corresponds to 7 points per time period.

between the coefficients of the finite element basis expansions of the numerical and the exact solutions.

Inspection of the plots in Figure 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 $m \geqslant \tilde{m}$ we have a scheme where the error caused by the Krylov subspace approximation is negligible as compared to the time error. 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.

5.4 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

$$\texttt{error} = \left\| \frac{\boldsymbol{y}^{\bar{n}} - \boldsymbol{y}^{\bar{n}}_{\text{exact}}}{\boldsymbol{y}^{\bar{n}}_{\text{exact}} + \epsilon_C} \right\|_\infty ,$$

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

5.4.1 Computational work per time step

We recall that the computational work per time step in the Gautschi scheme is a factor $m+1$ (with m being the Krylov dimension) more than for the leap frog scheme. The Krylov dimension of the Gautschi scheme grows mildly when the mesh gets finer: in the experiments presented below typical values of m varied between 3 and 7 on the $10 \times 10 \times 10$ mesh and between 10 and 15 on the $20 \times 20 \times 20$ mesh.

The cost of the LLC scheme is difficult to compare explicitly with those of the other two schemes. The LLC scheme requires the solution of a linear system with the matrix $M_\epsilon + \frac{\tau^2}{4} A_\mu$ whose sparse LU factorization is less efficient than that of M_ϵ. This makes the 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. One could use an iterative solver in the LLC scheme. However, a simple analysis based on the number of required matrix-vector multiplications shows that the costs would still be higher than for the Gautschi scheme (which needs only upto 15 matrix-vector multiplications on the $20 \times 20 \times 20$ mesh). Note that performance of the iterative Krylov processes in both the LLC and the Gautschi scheme can be improved by a suitable preconditioning (see [26] for preconditioning of the Krylov subspace matrix function evalutions). On the other hand, the use of approximate implicit schemes [2] or stabilized explicit schemes [29,24,25] might be a good option here, too.

5.4.2 Uniform cubic mesh

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

$$N_\omega = 101, \qquad w_i = 1 + \frac{9}{100}(i-1),\ i = 1, \ldots, 101. \tag{5.2}$$

The results are presented in Figure 12. We see that all the schemes clearly exhibit second order time accuracy. The peculiar drop in the error-versus-τ

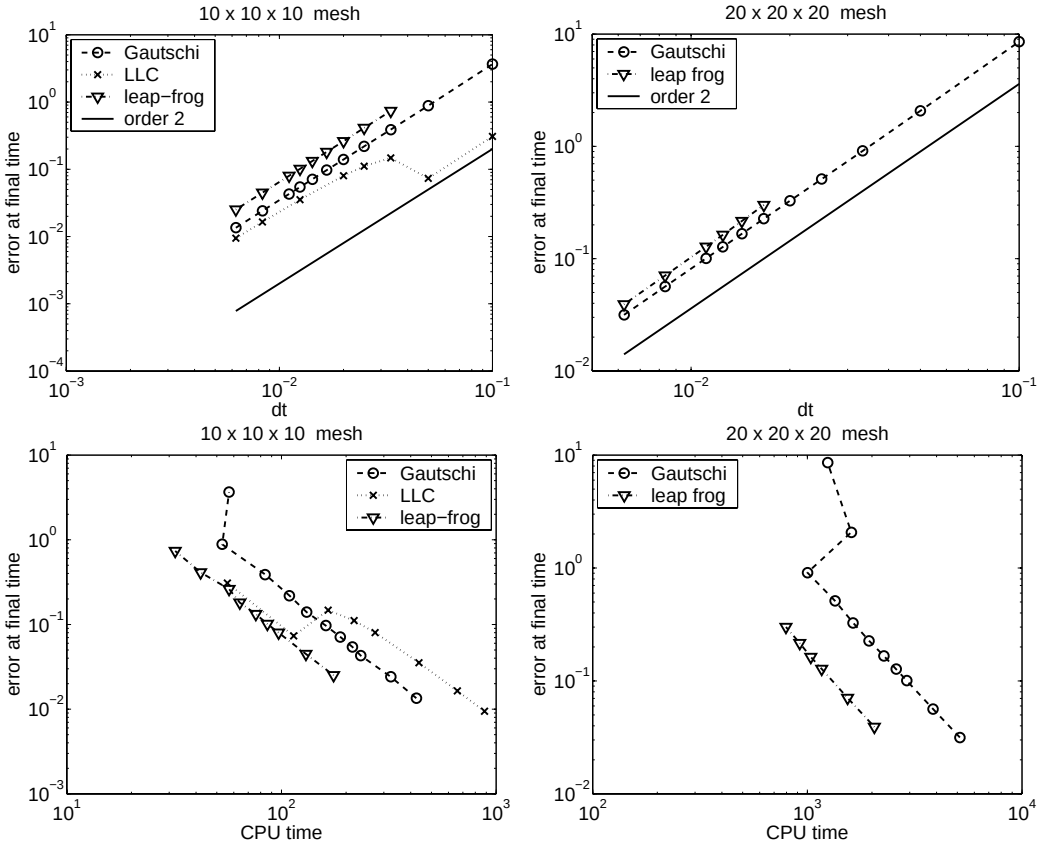

Fig. 12. Errors at the final time against the corresponding step sizes and the required CPU times for the homogeneously distributed frequencies in the source term (cf. (5.2)). Uniform mesh.

plot of the LLC scheme, is caused by the increase in the error order observed in (4.22), (4.23).

The zigzags 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 Figure 12, the implicit Gautschi and LLC schemes lose to the leap frog schemes in performance. This is to be expected since we work on a uniform mesh in a domain with homogeneous ϵ_r and μ_r. Very similar results were obtained for the case where

$$N_\omega = 101, \quad \omega_i \text{ evenly distributed in } [1,2],\ i = 1,\ldots,100, \quad \omega_{101} = 10. \tag{5.3}$$

Here all the schemes yield errors which are approximately a factor 10^3 smaller than for the homogeneous distribution of ω_i (5.2).

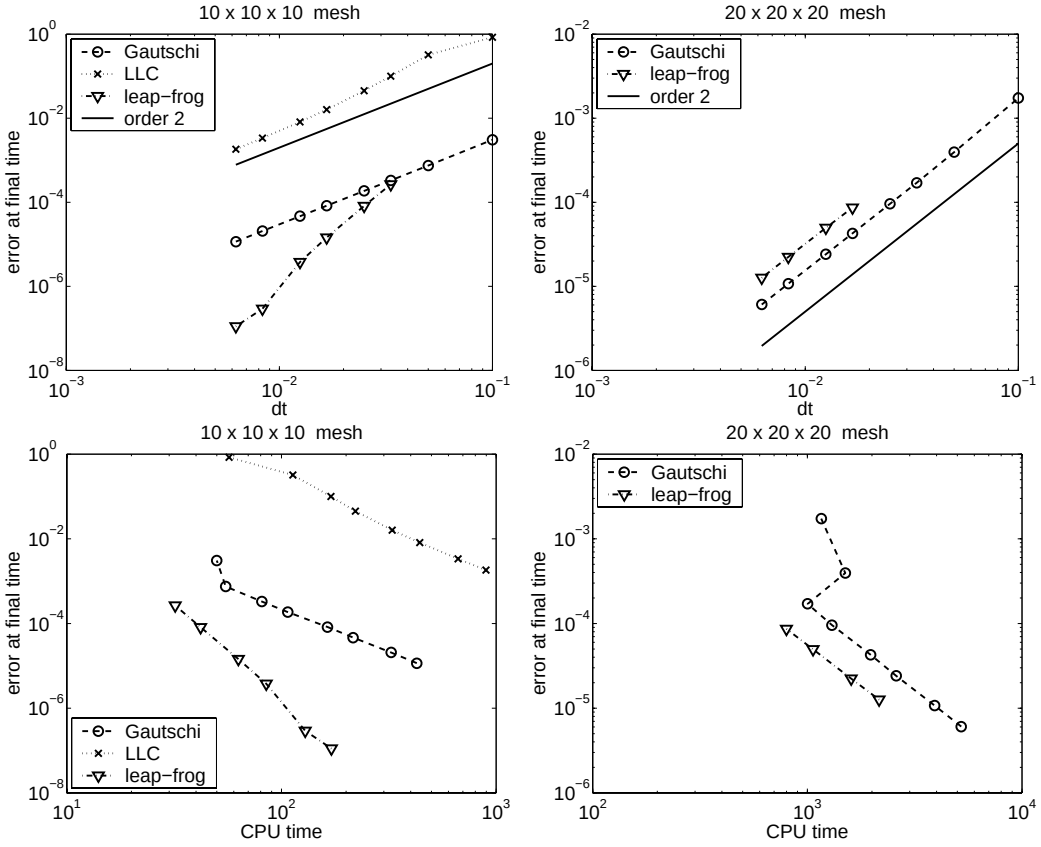

Fig. 13. Errors at the final time against the corresponding step sizes and the required CPU times for the clustered distribution of the frequencies in the source term (cf. (5.3)). Uniform mesh.

5.4.3 Unstructured tetrahedral mesh

In this example, Test problem 2 with the homogeneously distributed frequencies in the source term (cf. (5.2)) is solved on a unstructured tetrahedral mesh generated by the Centaur mesh generator. In the mesh used (see Figure 14), 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 a factor two smaller than the stability time step restriction of a uniform mesh with approximately the same number of degrees of freedom).

The results of the experiment are given in Figure 15. 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.

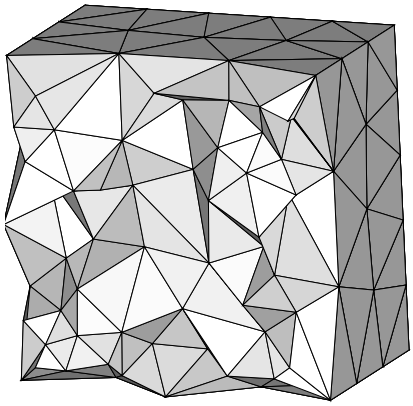

Fig. 14. A cut of the unstructured mesh used for the experiment.

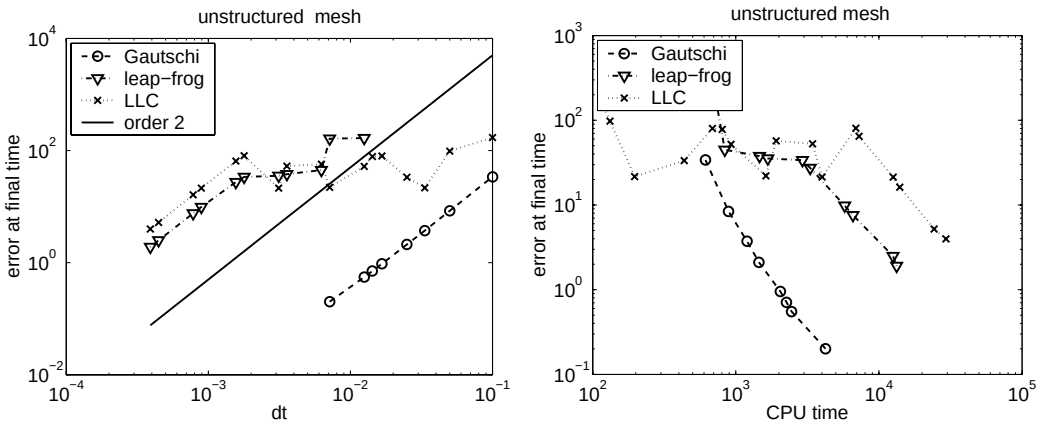

Fig. 15. Unstructured mesh. Errors at the final time against the corresponding step sizes and the required CPU times for the homogeneously distributed frequencies in the source term (cf. (5.2)).

5.4.4 Exactness of the Gautschi scheme for the slowly varying inhomogeneous term

The Gautschi scheme is known to be exact for the constant inhomogeneous term $\boldsymbol{j}(t)$ [7,11]. 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_i:

$$\text{(i)} \quad N_\omega = 1, \qquad \omega_1 = 0, \tag{5.4}$$

$$\text{(i)} \quad N_\omega = 3, \qquad \omega_1 = 10^{-5},\ \omega_2 = 2.23 \cdot 10^{-5},\ \omega_3 = 8 \cdot 10^{-6}. \tag{5.5}$$

The results (obtained for the uniform cubic mesh) are presented in Figures 16 and 17. 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.

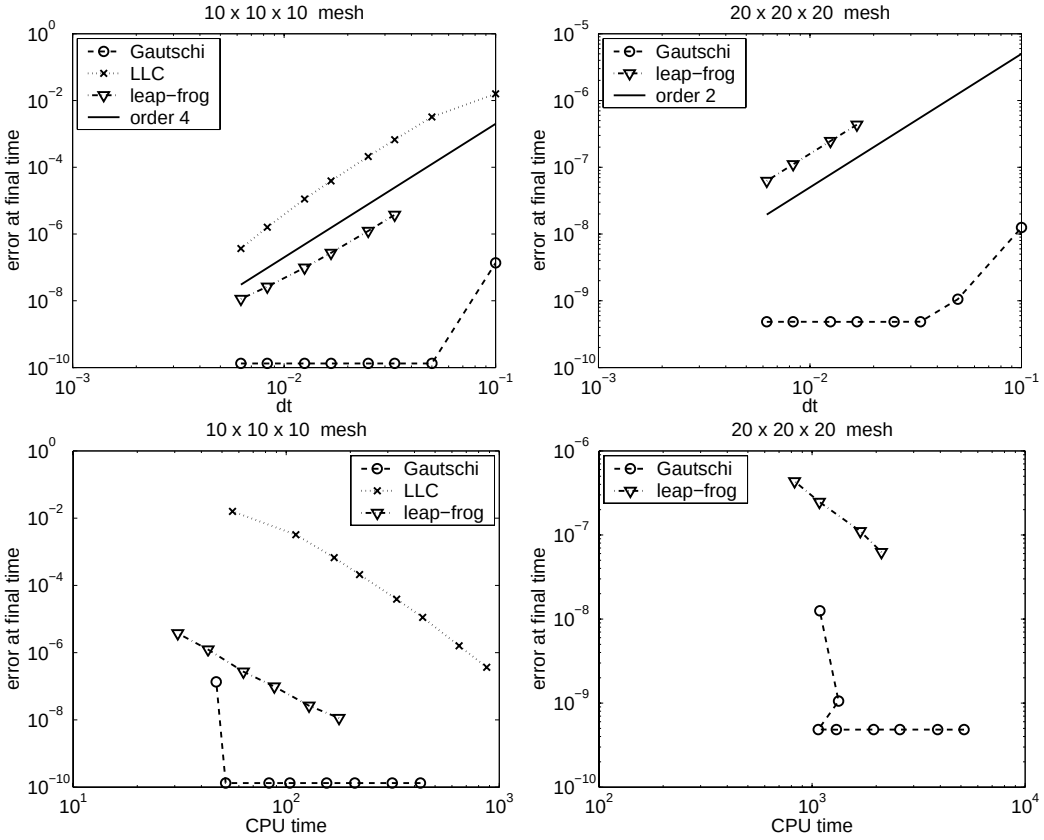

Fig. 16. Errors at the final time against the corresponding step sizes and the required CPU times for the constant source term (cf. (5.4)).

6 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.

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

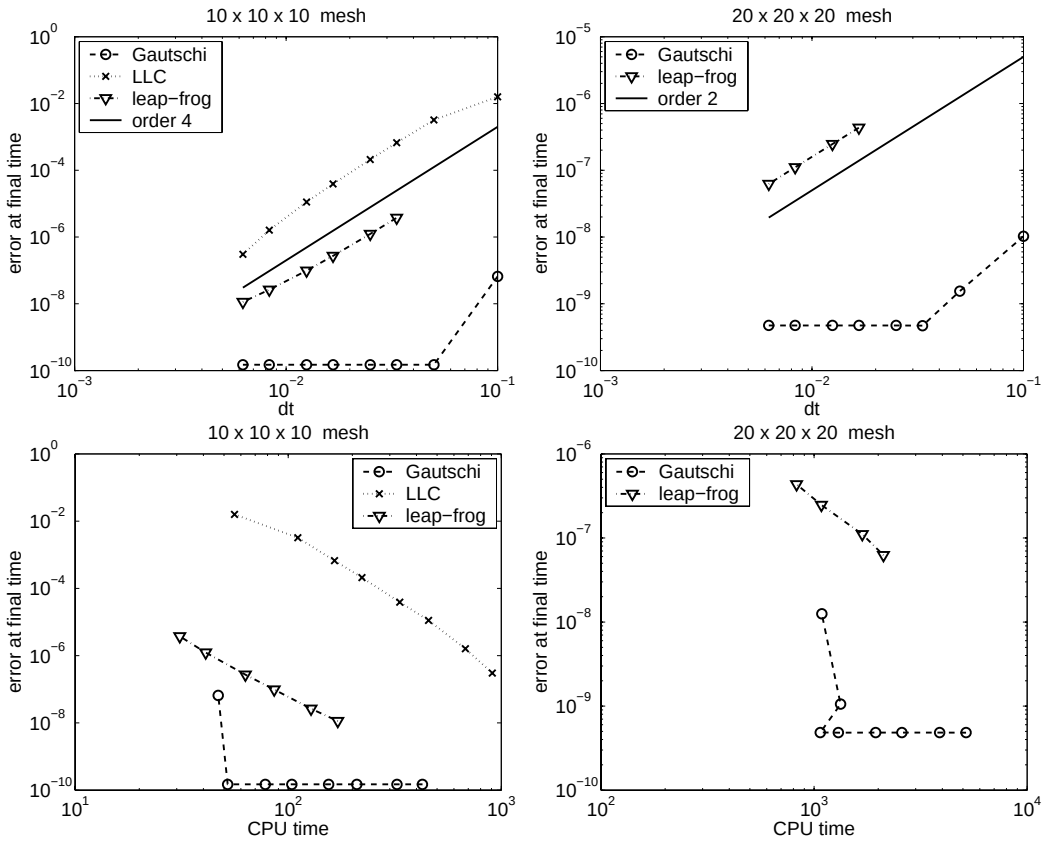

Fig. 17. Errors at the final time against the corresponding step sizes and the required CPU times for the slowly varying source term (cf. (5.5)).

The presented numerical experiments demonstrate that the Gautschi scheme is more efficient (in terms of the achieved accuracy and the required CPU time) 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. It would also be interesting to see how the Gautschi-Krylov scheme performs with the recently developed matrix function preconditioning technique [26].

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

Acknowledgements

The first author thanks Jason Frank for useful discussions.

References

[1] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. A. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, PA, 1994. Available at URL http:// www. netlib .org/ templates/.

[2] M. A. Botchev, G. L. G. Sleijpen, and H. A. van der Vorst. Stability control for approximate implicit time stepping schemes with minimum residual iterations. *Appl. Numer. Math.*, 31(3):239–253, 1999.

[3] H. De Raedt, K. Michielsen, J. S. Kole, and M. T. Figge. One-step finite-difference time-domain algorithm to solve the Maxwell equations. *Phys. Rev. E*, 67:056706, 2003.

[4] V. L. Druskin and L. A. Knizhnerman. Two polynomial methods of calculating functions of symmetric matrices. *U.S.S.R. Comput. Maths. Math. Phys.*, 29(6):112–121, 1989.

[5] V. L. Druskin and L. A. Knizhnerman. Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic. *Numer. Lin. Alg. Appl.*, 2:205–217, 1995.

[6] V. L. Druskin and L. A. Knizhnerman. Extended Krylov subspaces: approximation of the matrix square root and related functions. *SIAM J. Matrix Anal. Appl.*, 19(3):755–771 (electronic), 1998.

[7] W. Gautschi. Numerical integration of ordinary differential equations based on trigonometric polynomials. *Numer. Math*, 3:381–397, 1961.

[8] S. D. Gedney and U. Navsariwala. An unconditionally stable finite element time-domain solution of the vector wave equation. *IEEE Microwave and Guided Wave Letters*, 5(10):332–334, Oct. 1995.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore and London, third edition, 1996.

[10] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 34(5):1911–1925, 1997.

[11] M. Hochbruck and C. Lubich. A Gautschi-type method for oscillatory second-order differential equations. *Numer. Math.*, 83:403–426, 1999.

[12] M. Hochbruck, C. Lubich, and H. Selhofer. Exponential integrators for large systems of differential equations. *SIAM J.Sci. Comp.*, 15(5):1552–1574, 1998.

[13] R. Horváth. Uniform treatment of numerical time-integrations of the Maxwell equation. In W. H. A. Schilders, J. W. ter Maten, and S. H. M. J. Houben, editors, *Proc. of the Conference "Scientific Computing in Electrical Engineering" SCEE-2002, Eindhoven*, Springer Series Mathematics in Industry, pages 231–239. Springer Verlag, 2004.

[14] L. A. Knizhnerman. Calculation of functions of unsymmetric matrices using Arnoldi's method. *U.S.S.R. Comput. Maths. Math. Phys.*, 31(1):1–9, 1991.

[15] J. S. Kole, M. T. Figge, and H. De Raedt. Unconditionally stable algorithms to solve the time-dependent Maxwell equations. *Phys. Rev. E*, 64:066705, 2001.

[16] J. S. Kole, M. T. Figge, and H. De Raedt. Higher-order unconditionally stable algorithms to solve the time-dependent Maxwell equations. *Phys. Rev. E*, 65:066705, 2002.

[17] J.-F. Lee, R. Lee, and A. Cangellaris. Time-domain finite-element methods. *IEEE transactions on antennas and propagation*, 45(3):430–442, 1997.

[18] P. Monk. *Finite Element Methods for Maxwell's Equations*. Oxford University Press, 2003.

[19] J.-C. Nédélec. Mixed finite elements in $\mathbf{R}^3$. *Numer. Math.*, 35(3):315–341, 1980.

[20] J.-C. Nédélec. A new family of mixed finite elements in $\mathbf{R}^3$. *Numer. Math.*, 50(1):57–81, 1986.

[21] G. Rodrigue and D. White. A vector finite element time-domain method for solving Maxwell's equations on unstructured hexahedral grids. *SIAM J. Sci. Comput.*, 23(3):683–706 (electronic), 2001.

[22] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 29(1):209–228, 1992.

[23] Y. Saad. Iterative methods for sparse linear systems. Book out of print, 2000. Available at URL http:// www-users .cs .umn .edu/ ~saad /books.html.

[24] A. S. Shvedov and V. T. Zhukov. Explicit iterative difference schemes for parabolic equations. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 13(2):133–148, 1998.

[25] B. P. Sommeijer, L. F. Shampine, and J. G. Verwer. RKC: An explicit solver for parabolic PDEs. *J. Comput. Appl. Math.*, 88:315–326, 1997.

[26] J. van den Eshof and M. Hochbruck. Preconditioning Lanczos approximations to the matrix exponential. Preprint of the Dept. of Mathematics, Heinrich Heine University, Duesseldorf, Germany, March, 2004, March 2004. To appear in SIAM J. Sci. Comput.

[27] H. A. van der Vorst. An iterative solution method for solving $f(A)x = b$, using using Krylov subspace information obtained for the symmetric positive definite matrix A. *J. Comput. Appl. Math.*, 18:249–263, 1987.

[28] H. A. van der Vorst. *Iterative Krylov methods for large linear systems*. Cambridge University Press, 2003.

[29] J. G. Verwer. Explicit Runge–Kutta methods for parabolic partial differential equations. *Appl. Num. Math.*, 22:359–379, 1996.

[30] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer, 2001.

[31] K. S. Yee. Numerical solution of initial boundary value problems involving Maxwells equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, 14(3):302–307, March 1966.

[32] O. C. Zienkiewicz and I. Morgan. *Finite elements and approximations*. John Wiley & Sons, 1983.

A Stability of the leap frog scheme

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

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

or in its equivalent form

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

In our analysis, we follow the standard approach based on diagonalizing the matrices involved in the scheme (see e.g. [32]). Any solution of (A.2) can be written as

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

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

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

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

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

which, due to linear independence of $\boldsymbol{\alpha}_m$'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. $|\gamma_m^n| \leqslant 1$) if and only if the roots $\nu_{1,2}$ of its characteristic equation

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

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

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

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

$$(1 - \frac{\tau^2}{2}\lambda_m)^2 - 1 \leqslant 0, \tag{A.9}$$

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

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

Then the stability condition for the leap frog scheme is

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

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

B Dispersion relation matrices F and G

The matrices F and G in (4.12) on a cubic mesh with element size $h \times h \times h$ are given as:

the matrix F is diagonal, with entries

$$\begin{aligned}
F_{11} &= \frac{1}{9}\cos(hk_2)\cos(hk_3) + \frac{2}{9}\cos(k_3 h) + \frac{2}{9}\cos(k_2 h) + \frac{4}{9}, \\
F_{22} &= \frac{1}{9}\cos(hk_1)\cos(hk_3) + \frac{2}{9}\cos(k_3 h) + \frac{2}{9}\cos(k_1 h) + \frac{4}{9}, \\
F_{33} &= \frac{1}{9}\cos(hk_1)\cos(hk_2) + \frac{2}{9}\cos(k_2 h) + \frac{2}{9}\cos(k_1 h) + \frac{4}{9},
\end{aligned}$$

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

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