Centrum Wiskunde & Informatica

REPORTRAPPORT

MAS

Modelling, Analysis and Simulation

Modelling, Analysis and Simulation

Unconditionally stable integration of Maxwell's equations

J.G. Verwer, M.A. Botchev

REPORT MAS-R0806 SEPTEMBER 2008

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

Unconditionally stable integration of Maxwell's equations

ABSTRACT
Numerical integration of Maxwell's equations is often based on explicit methods accepting a stability step size restriction. In literature evidence is given that there is also a need for unconditionally stable methods, as exemplified by the successful alternating direction implicit-finite difference time domain scheme. In this paper we discuss unconditionally stable integration for a general semi-discrete Maxwell system allowing non-Cartesian space grids as encountered in finite element discretizations. Such grids exclude the alternating direction implicit approach. Particular attention is given to the second-order trapezoidal rule implemented with preconditioned conjugate gradient iteration and to second-order exponential integration using Krylov subspace iteration for evaluating the arising phi-functions. A three-space dimensional test problem is used for numerical assessment and comparison with an economical second-order implicit-explicit integrator. We further pay attention to the Chebyshev series expansion for computing the exponential operator for skew-symmetric matrices.

2000 Mathematics Subject Classification: 65L05,65L20, 65M12, 65M20.
1998 ACM Computing Classification System: G.1.7, G.1.8.
Keywords and Phrases: Maxwell's equations, Implicit integration, Exponential integration, Conjugate gradient iteration, Krylov subspace iteration, Chebyshev expansion.

Unconditionally Stable Integration of Maxwell's Equations

J.G. Verwer
Center for Mathematics and Computer Science
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Jan.Verwer@cwi.nl

M.A. Botchev
University of Twente, Dept. Applied Mathematics, Faculty EEMCS
P.O. Box 217, Enschede, The Netherlands
m.a.botchev@math.utwente.nl

Dedicated to Henk van der Vorst for his numerous contributions to numerical mathematics

Abstract

Numerical integration of Maxwell's equations is often based on explicit methods accepting a stability step size restriction. In literature evidence is given that there is also a need for unconditionally stable methods, as exemplified by the successful alternating direction implicit – finite difference time domain scheme. In this paper we discuss unconditionally stable integration for a general semi-discrete Maxwell system allowing non-Cartesian space grids as encountered in finite element discretizations. Such grids exclude the alternating direction implicit approach. Particular attention is given to the second-order trapezoidal rule implemented with preconditioned conjugate gradient iteration and to second-order exponential integration using Krylov subspace iteration for evaluating the arising φ-functions. A three-space dimensional test problem is used for numerical assessment and comparison with an economical second-order implicit-explicit integrator. We further pay attention to the Chebyshev series expansion for computing the exponential operator for skew-symmetric matrices.

2000 Mathematics Subject Classification: Primary: 65L05, 65L20, 65M12, 65M20.
1998 ACM Computing Classification System: G.1.7, G.1.8.
Keywords and Phrases: Maxwell's equations, Implicit integration, Exponential integration, Conjugate gradient iteration, Krylov subspace iteration, Chebyshev expansion.

1 Introduction

Maxwell's equations from electromagnetism model interrelations between electric and magnetic fields. The equations form a time-dependent system of six first-order partial differential equations (PDEs). The equations appear in different forms, such as in the compact curl notation

$$\begin{aligned} \partial_t B &= -\nabla \times E\,, \\ \varepsilon \partial_t E &= \nabla \times (\mu^{-1}) B - \sigma E - J_E. \end{aligned} \tag{1.1}$$

Here B and E represent the magnetic and electric field, repectively. J_E is a given source term representing electric current density and ε, μ and σ are (tensor) coefficients representing, respectively, dielectric permittivity, magnetic permeability and conductivity. The equations are posed in a three-dimensional spatial domain and provided with appropriate boundary conditions. If the equations are posed in domains without conductors, the damping term $-\sigma E$ is absent. If in addition the source J_E is taken zero we have a prime example of a conservative wave equation system.

Numerical methods for time-dependent PDEs are often derived in two stages (the method of lines approach). First, the spatial operators are discretized on an appropriate grid covering the spatial domain, together with the accompanying boundary conditions. This leads to a time-continuous, semi-discrete problem in the form of an initial-value problem for a system of ordinary differential equations (ODEs). Second, a numerical integration method for this ODE system is chosen, which turns the semi-discrete solution into the desired fully discrete solution on the chosen space-time grid. In this paper we focus on the second numerical integration stage, as in [6]. While in [6] the focus was on methods treating the curl terms explicitly, here we address the question whether fully implicit and exponential time integration eliminating any temporal step size stability restriction can be feasible and efficient.

As in [6] we start from the general space-discretized Maxwell problem

$$\begin{pmatrix} M_u & 0 \\ 0 & M_v \end{pmatrix} \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} 0 & -K \\ K^T & -S \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} j_u \\ j_v \end{pmatrix}, \tag{1.2}$$

where $u = u(t)$ and $v = v(t)$ are the unknown vector (grid) functions approximating the values of the magnetic flux B and electric field E, respectively. The matrices K and K^T approximate the curl operator and the matrix S is associated with the dissipative conduction term. Throughout S can be assumed symmetric positive semi-definite. M_u and M_v are symmetric positive definite mass matrices possibly arising from a spatial finite element or compact finite difference discretization. The functions $j_u(t)$ and $j_v(t)$ are source terms. Typically, j_v represents the given source current J_E, but j_u and j_v may also contain boundary data. We do allow u and v to have different dimensions which can occur with certain finite-element methods, see e.g. [39], and assume $u \in \mathbb{R}^m, v \in \mathbb{R}^n$ with $n \geqslant m$ and $M_u \in \mathbb{R}^{m\times m}, M_v \in \mathbb{R}^{n\times n}, K \in \mathbb{R}^{m\times n}, S \in \mathbb{R}^{n\times n}$. The ODE system (1.2) is generic in the sense that spatial discretization of (H, E)-formulations of the Maxwell equations also lead to this form, see Section 4 of [6].

We allow the space grids underlying (1.2) to be non-Cartesian. This has an important consequence in that it excludes the well-known unconditionally stable alternating direction implicit-finite difference time domain method attributed to [50, 51], see also [13, 17, 20, 35] and references therein. We will instead focus on conventional fully implicit integration (Section 3) and on exponential integration (Sections 4 and 5). This means that we need efficient solvers from the field of numerical linear algebra. For solving the systems of linear algebraic equations arising with implicit integrators we will use the conjugate gradient (CG) iterative method with preconditioning (Section 3). For exponential integration we will consider Krylov subspace iteration (Section 4) and a Chebyshev series expansion (Section 5). Both for the theory behind CG and Krylov iteration we refer to the text books [42, 48]. Seminal papers on Krylov subspace iteration for matrix functions are [14, 15, 22, 28, 40, 47]. The Chebyshev method is due to [45] and was recommended for Maxwell's equations in [12].

2 Stability and conservation properties

To begin with, we recall from [6] some stability and conservation properties of the semi-discrete system (1.2). Let $w \in \mathbb{R}^{n+m}$ denote the solution vector composed by $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$. A natural norm for establishing stability and conservation is the inner-product norm

$$\|w\|^2 = \|u\|^2_{M_u} + \|v\|^2_{M_v}\,, \qquad \|u\|^2_{M_u} = \langle M_u u, u\rangle\,, \quad \|v\|^2_{M_v} = \langle M_v v, v\rangle\,, \tag{2.1}$$

where $\langle\cdot\,,\cdot\rangle$ denotes the L_2 inner product. As S is symmetric positive semi-definite, for the homogeneous part of (1.2) follows

$$\frac{d}{dt}\|w\|^2 = -2\langle Sv, v\rangle \leqslant 0\,, \tag{2.2}$$

showing stability in the L_2 sense and (energy) conservation for a zero matrix S. It is desirable that integration methods respect these properties, either exactly or to sufficiently high accuracy.

For the purpose of analysis a formulation without mass matrices equivalent to (1.2) is obtained as follows. Introduce the Cholesky factorizations $L_{M_u} L_{M_u}^T = M_u$ and $L_{M_v} L_{M_v}^T = M_v$. Then

$$\begin{pmatrix} \tilde{u}' \\ \tilde{v}' \end{pmatrix} = \begin{pmatrix} 0 & -\tilde{K} \\ \tilde{K}^T & -\tilde{S} \end{pmatrix} \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} + \begin{pmatrix} \tilde{j}_u \\ \tilde{j}_v \end{pmatrix}, \tag{2.3}$$

where $\tilde{u} = L_{M_u}^T u$, $\tilde{v} = L_{M_v}^T v$ and

$$\tilde{K} = L_{M_u}^{-1} K L_{M_v}^{-T}, \quad \tilde{S} = L_{M_v}^{-1} S L_{M_v}^{-T} \quad \tilde{j}_u = L_{M_u}^{-1} j_u\,, \quad \tilde{j}_v = L_{M_v}^{-1} j_v\,. \tag{2.4}$$

Next introduce the transformed inner-product norm

$$\|\tilde{w}\|_2^2 = \|\tilde{u}\|_2^2 + \|\tilde{v}\|_2^2\,, \qquad \|\tilde{u}\|_2^2 = \langle \tilde{u}, \tilde{u} \rangle\,, \quad \|\tilde{v}\|_2^2 = \langle \tilde{v}, \tilde{v} \rangle\,, \tag{2.5}$$

for the vector $\tilde{w}$ composed of $\tilde{u}$ and $\tilde{v}$. For the homogeneous part of (2.3) then follows immediately

$$\frac{d}{dt} \|\tilde{w}\|_2^2 = -2 \langle \tilde{S}\tilde{v}, \tilde{v} \rangle \leqslant 0\,, \tag{2.6}$$

while the norm is preserved under the transformation, that is, $\|\tilde{w}\|_2 = \|w\|$ and $\langle \tilde{S}\tilde{v}, \tilde{v} \rangle = \langle Sv, v \rangle$. We note that the transformed system is introduced for analysis purposes only and that our numerical methods will be applied to system (1.2).

If in (1.1) the conductivity coefficient σ and the permittivity coefficient ε are constant scalars instead of space-dependent tensors (3×3 matrices), then the matrices M_v and S from (1.2) can be assumed identical up to a constant, implying that the matrix $\tilde{S}$ introduced in (2.3) becomes the constant diagonal matrix

$$\tilde{S} = \alpha I, \qquad \alpha = \frac{\sigma}{\varepsilon}\,. \tag{2.7}$$

This enables the derivation of a two-by-two system for the sake of further analysis. Introduce the singular-value decomposition $\tilde{K} = U \Sigma V^T$ where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal and Σ is a diagonal $m \times n$ matrix with nonnegative diagonal entries $s_1, \ldots, s_m$ satisfying

$$s_1 \geqslant s_2 \geqslant \cdots \geqslant s_r > s_{r+1} = \cdots = s_m = 0\,. \tag{2.8}$$

Here $r \leqslant m$ is the (row) rank of $\tilde{K}$ and the s_i are the singular values of the matrix $\tilde{K}$ (the square roots of the eigenvalues of $\tilde{K}\tilde{K}^T$). The transformed variables and source terms

$$\bar{u}(t) = U^T \tilde{u}(t)\,, \quad \bar{v}(t) = V^T \tilde{v}(t)\,, \quad \bar{j}_u(t) = U^T \tilde{j}_u(t)\,, \quad \bar{j}_v(t) = V^T \tilde{j}_v(t)\,, \tag{2.9}$$

satisfy the equivalent ODE system

$$\begin{pmatrix} \bar{u}' \\ \bar{v}' \end{pmatrix} = \begin{pmatrix} 0 & -\Sigma \\ \Sigma^T & -\alpha I \end{pmatrix} \begin{pmatrix} \bar{u} \\ \bar{v} \end{pmatrix} + \begin{pmatrix} \bar{j}_u \\ \bar{j}_v \end{pmatrix}, \tag{2.10}$$

where I is the $n \times n$ identity matrix. Note that the matrix transformation induced by (2.9) is a similarity transformation, so that the matrices of systems (2.3) and (2.10) have the same eigenvalues. Further, $\|\tilde{w}\|_2^2 = \|\bar{u}\|_2^2 + \|\bar{v}\|_2^2$ due to the orthogonality of U and V. Thus, if (2.7) applies, the stability of a time integration method may be studied for the homogeneous part of (2.10), provided also the method is invariant under the transformations leading to (2.10).

Since the matrix Σ is diagonal, (2.10) decouples into r two-by-two systems

$$\begin{pmatrix} \hat{u}' \\ \hat{v}' \end{pmatrix} = \begin{pmatrix} 0 & -s \\ s & -\alpha \end{pmatrix} \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} + \begin{pmatrix} \hat{j}_u \\ \hat{j}_v \end{pmatrix}, \tag{2.11}$$

with $s = s_k > 0$, $k = 1, \ldots, r$ and $n + m - 2r$ two-by-two systems

$$\begin{pmatrix} \hat{u}' \\ \hat{v}' \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & -\alpha \end{pmatrix} \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} + \begin{pmatrix} \hat{j}_u \\ \hat{j}_v \end{pmatrix}. \tag{2.12}$$

From the viewpoint of time integration, the first elementary two-by-two system (2.11) is canonical for Maxwell equation systems of which the conductivity coefficient σ and the permittivity coefficient ε are constant scalars. For stability analysis we may neglect the source terms, arriving at the two-by-two test model

$$\begin{pmatrix} \hat{u}' \\ \hat{v}' \end{pmatrix} = \begin{pmatrix} 0 & -s \\ s & -\alpha \end{pmatrix} \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix}, \qquad s \geqslant 0, \quad \alpha \geqslant 0. \tag{2.13}$$

Stability for this test model is equivalent to stability for (2.10), which in turn is equivalent to stability for the original semi-discrete Maxwell system (1.2), provided the conductivity coefficient σ and the permittivity coefficient ε are constant scalars. The eigenvalues of (2.13) are $(-\alpha \pm \sqrt{\alpha^2 - 4s^2})/2$. Assuming sufficiently small and large singular values s_k in (2.8), the spectra of (2.3) and (2.10) thus are cross-shaped with real eigenvalues between $-\alpha$ and 0 and complex eigenvalues with real part $-\alpha/2$ and imaginary parts $\pm\sqrt{4s_k^2 - \alpha^2}/2$.

3 Implicit integration

We will examine fully implicit time stepping for (1.2) for the second-order implicit trapezoidal rule (ITR). This method has the right stability and conservation properties for Maxwell's equations and shares the numerical algebra challenge with many other implicit methods, such as diagonally-implicit Runge-Kutta methods. So numerical algebra conclusions drawn for ITR are also valid for related higher-order methods. In this paper we focus on second-order methods because the order of the spatial discretization scheme for the 3D example problem used for testing is also limited to two. Before discussing ITR we first recall an economical second-order implicit-explicit method called CO2 (COmposition 2nd-order) in [6] which serves as a reference method.

3.1 The implicit-explicit method CO2

Method CO2 is given by

$$\begin{aligned} M_u \frac{u_{n+1/2} - u_n}{\tau} &= -\tfrac{1}{2} K v_n + \tfrac{1}{2} j_u(t_n), \\ M_v \frac{v_{n+1} - v_n}{\tau} &= K^T u_{n+1/2} - \tfrac{1}{2} S(v_n + v_{n+1}) + \tfrac{1}{2}(j_v(t_n) + j_v(t_{n+1})), \\ M_u \frac{u_{n+1} - u_{n+1/2}}{\tau} &= -\tfrac{1}{2} K v_{n+1} + \tfrac{1}{2} j_u(t_{n+1}). \end{aligned} \tag{3.1}$$

Like ITR this method is a one-step method stepping from (u_n, v_n) to (u_{n+1}, v_{n+1}) with step size τ. Here u_n denotes the approximation to the exact solution $u(t_n)$, etc., and $\tau = t_{n+1} - t_n$. The subindex n should not be confused with the length of the vector v in (1.2). CO2 is symmetric and treats the curl terms explicitly and the conduction term implicitly. Of practical importance is that the third-stage derivative computation can be copied to the first stage at the next time step to save computational work. Per time step this method thus is very economical. Apart from the mass matrices (see Remark 3.1), the method requires a single explicit evaluation of the full derivative per integration step which is the least possible.

In contrast to ITR method (3.1) is not unconditionally stable and a sharp step size bound for stability for the general system (1.2) is not known up to now. However, for Maxwell problems for which (2.7) holds stability can be concluded from the 2 × 2-model (2.11). Let $z_s = \tau s_{\max}$. The resulting step size bound is then valid for (1.2) and is given by

$$z_s < 2 \quad \text{if} \quad \alpha = 0 \quad \text{and otherwise} \quad z_s \leqslant 2. \tag{3.2}$$

Hence the conduction puts no limit on τ. Recall that $\alpha = 0$ in the absence of conduction and that $s_{\max}$ here is to be taken as the maximal square root of the eigenvalues of $\tilde{K}\tilde{K}^T$. Because K approximates the first-order curl operator these eigenvalues are proportional to h^{-2} where h represents a minimal spatial grid size. So for time stepping stability a relation $\tau \sim h$ for $h \to 0$ is required. On fine space grids and long time intervals this may lead to large amounts of time steps.

It is this observation which underlies the question whether implicit or exponential integration is feasible and competitive so as to enhance time stepping efficiency. For the derivation and further discussion of this method we refer to [6] where it was called CO2 as it is of second order and based on COmposition of a certain partitioned Euler rule. One of the results in [6] states that the second order behavior is maintained in the presence of time-dependent boundary conditions (stiff source terms). A similar result will be proven in the appendix (Section A.2) for the exponential integrator EK2 derived in Section 5. Finally, with regard to time stepping CO2 bears a close resemblance to the popular time-staggered Yee-scheme [49] and as such is a natural candidate for a reference method.

Remark 3.1 The mass matrices naturally give rise to implicitness such that we encounter at each integration step one inversion of M_u and $M_v + \frac{1}{2}\tau S$. Systems with mass matrices can be (approximately) solved in an efficient way. This can be achieved either by fast solvers (sparse direct or preconditioned iterative) or by special mass lumping techniques. For mass lumping of the finite-element discretization used in Section 3.5, see e.g. [2, 19]. For keeping our assessments as general as possible we will use the original non-lumped form. Throughout this paper (so also for the other integration methods) we will use sparse Cholesky factorization to realize the mass matrix inversions. For constant τ the factorization should only be carried out once at the start of the integration leaving only sparse forward-backward substitutions during the time stepping. $\diamond$

3.2 The implicit trapezoidal rule ITR

Denote (1.2) by

$$Mw' = Aw + g(t)\,, \tag{3.3}$$

where

$$w = \begin{pmatrix} u \\ v \end{pmatrix}, \quad M = \begin{pmatrix} M_u & 0 \\ 0 & M_v \end{pmatrix}, \quad A = \begin{pmatrix} 0 & -K \\ K^T & -S \end{pmatrix}, \quad g(t) = \begin{pmatrix} j_u \\ j_v \end{pmatrix}. \tag{3.4}$$

ITR then reads

$$M\,\frac{w_{n+1} - w_n}{\tau/2} = Aw_{n+1} + Aw_n + g(t_n) + g(t_{n+1})\,. \tag{3.5}$$

This classical implicit method mimics the stability and conservation property (2.2). That is, for zero sources,

$$\frac{\|w_{n+1}\|^2 - \|w_n\|^2}{\tau} = -2\langle S\,\frac{v_{n+1} + v_n}{2}, \frac{v_{n+1} + v_n}{2}\rangle\,, \qquad \forall \tau > 0\,. \tag{3.6}$$

Hence (3.5) is unconditionally stable (and conservative for zero S). Like for CO2 the method is second-order consistent, also for stiff source terms emanating from time-dependent boundary functions. From that perspective the method is ideal, however at the costs of solving each time step the linear system

$$(M - \tfrac{1}{2}\tau A)\,w_{n+1} = (M + \tfrac{1}{2}\tau A)\,w_n + \tfrac{1}{2}\tau g(t_n) + \tfrac{1}{2}\tau g(t_{n+1}) \tag{3.7}$$

for the matrix

$$M - \tfrac{1}{2}\tau A = \begin{pmatrix} M_u & \frac{1}{2}\tau K \\ -\frac{1}{2}\tau K^T & M_v + \frac{1}{2}\tau S \end{pmatrix}. \tag{3.8}$$

Sparse LU-decomposition will become too costly in memory for large-scale 3D simulations. We therefore focus on iteration whereby we rewrite (Schur complement) the linear system (3.7) to an equivalent form which is significantly more amenable for iterative solution.

Let r_u, r_v denote the righthand sides of (3.7) belonging to u, v. Suppressing the time index $n+1$ in u_{n+1}, v_{n+1} this system then reads

$$\begin{aligned} M_u u + \tfrac{1}{2}\tau K v &= r_u \,, \\ -\tfrac{1}{2}\tau K^T u + M_v v + \tfrac{1}{2}\tau S v &= r_v \,. \end{aligned} \tag{3.9}$$

Since the mass matrix M_u is symmetric positive definite, we can multiply the first equation from left by $\frac{1}{2}\tau K^T M_u^{-1}$. Then adding the two equations yields the equivalent system

$$\begin{aligned} M_u u + \tfrac{1}{2}\tau K v &= r_u \,, \\ \mathcal{M} v &= r_v + \tfrac{1}{2}\tau K^T M_u^{-1} r_u \,, \end{aligned} \tag{3.10}$$

wherein u has been eliminated from the second equation. The $n \times n$-matrix $\mathcal{M}$ is given by

$$\mathcal{M} = M_v + \frac{1}{2}\tau S + \frac{1}{4}\tau^2 K^T M_u^{-1} K \,. \tag{3.11}$$

So we can first solve v from the second equation and subsequently u from the first. Hereby we assume that the three inversions for M_u are carried out through sparse Cholesky decomposition, entirely similar as for method CO2. Of main interest is that $\mathcal{M}$ is symmetric positive definite which calls for the iterative conjugate gradient (CG) method.

3.3 CG convergence

Let us first assess the convergence of the CG method. For this purpose we employ the transformation underlying system (2.3) which can be shown to be equivalent to Cholesky factorization preconditioning with the mass matrix M_v, see also Section 3.4. The counterpart of (3.10) then reads

$$\begin{aligned} \tilde{u} + \tfrac{1}{2}\tau \tilde{K}\tilde{v} &= \tilde{r}_u \,, \\ \tilde{\mathcal{M}}\tilde{v} &= \tilde{r}_v \,, \end{aligned} \tag{3.12}$$

with the straightforward definition of the new righthand sides and

$$\tilde{\mathcal{M}} = I + \frac{1}{2}\tau \tilde{S} + \frac{1}{4}\tau^2 \tilde{K}^T \tilde{K} \,. \tag{3.13}$$

CG is a natural choice as it optimal in the following sense [48]: for any initial guess $\tilde{v}_0$ it computes iterants $\tilde{v}_i$ which satisfy the polynomial relation [1)]

$$\tilde{v}_i - \tilde{v} = P_i(\tilde{\mathcal{M}})\,(\tilde{v}_0 - \tilde{v}) \,, \tag{3.14}$$

such that in the $\tilde{\mathcal{M}}$-norm the iteration error $\|\tilde{v}_i - \tilde{v}\|_{\tilde{\mathcal{M}}}$ is minimal over the set of all polynomials P_i of degree i satisfying $P_i(0) = 1$. This iteration error satisfies the well-known bound

$$\|\tilde{v}_i - \tilde{v}\|_{\tilde{\mathcal{M}}} \leqslant 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i \|\tilde{v}_0 - \tilde{v}\|_{\tilde{\mathcal{M}}} \,, \tag{3.15}$$

where κ is the spectral condition number of $\tilde{\mathcal{M}}$, that is, $\kappa = \lambda_{\max}/\lambda_{\min}$ being the quotient of the maximal and minimal eigenvalue of $\tilde{\mathcal{M}}$. This upper bound does not reflect the celebrated superlinear convergence [44] of CG which makes it a truly successful solver. However, the bound does provide a rough assessment of the potential of the combination ITR-CG in relation to CO2. Hereby it is noted that in good approximation a single CG iteration with matrix $\mathcal{M}$ is cost wise equal to a single CO2 step.

Would $\tilde{S}$ and $\tilde{K}^T\tilde{K}$ commute, the condition number can be derived directly from the spectra of $\tilde{S}$ and $\tilde{K}^T\tilde{K}$. In the general case commutation will be rare. Therefore we next assume that we

[1)] The subindex i should not be confused with the subindex n used to denote a time level t_n.

do have a Maxwell problem for which condition (2.7) holds. Then we have commutation and the eigenvalues λ of $\tilde{\mathcal{M}}$ are given by

$$\lambda = 1 + \frac{1}{2}\tau\alpha + \frac{1}{4}\tau^2 s^2 \,, \tag{3.16}$$

where s^2 is an eigenvalue of $\tilde{K}^T\tilde{K}$ the square root of which also features in (2.11). Hence

$$\lambda_{\text{min}} = 1 + \frac{1}{2}\tau\alpha + \frac{1}{4}\tau^2 s_{\text{min}}^2 \,, \qquad \lambda_{\text{max}} = 1 + \frac{1}{2}\tau\alpha + \frac{1}{4}\tau^2 s_{\text{max}}^2 \,. \tag{3.17}$$

Regarding ITR we are only interested in step sizes τ such that $z_s = \tau s_{\text{max}} \gg 2$ because otherwise method CO2 will be more efficient, see the step size bound (3.2). Since s_{max} is proportional to h which represents the minimal spatial grid size, we then may neglect the contribution $\tau\alpha$ and approximate κ by

$$\kappa \approx 1 + \frac{1}{4}z_s^2 \,, \tag{3.18}$$

showing that one CG iteration reduces the initial iteration error in the $\tilde{\mathcal{M}}$-norm by

$$\nu(z_s) \approx \frac{\sqrt{1+\frac{1}{4}z_s^2} - 1}{\sqrt{1+\frac{1}{4}z_s^2} + 1} \sim 1 - \frac{4}{z_s} \sim e^{-\frac{4}{z_s}} \,, \quad z_s \to \infty \,. \tag{3.19}$$

Unfortunately, this reduction factor is by far too low for ITR implemented with CG to become a competitive method. To see this the following argument suffices. For $z_s \gg 2$ the number of CG iterations for an overall reduction factor ϵ is approximately $j = -\frac{1}{4}z_s \ln(\epsilon)$. Because each iteration is in good approximation as expensive as a single integration step with method CO2, we can afford j steps with CO2 with step size τ/j provided we have stability of CO2, that is, if $z_s/j \leqslant 2$. Inserting j this appears to hold for all $\epsilon \leqslant e^{-2} \approx 10^{-1}$. When iterating with CG an error reduction of the initial error by a factor ten is of course quite poor and we can conclude that the computational effort is better spent in applying CO2 with a step size τ/j. This will lead to a notable smaller time stepping error for comparable effort since ITR and CO2 are both of second order. Clearly, ITR will not be competitive to CO2 unless superlinear CG-convergence, not reflected by (3.15), takes place and/or CG is applied with a more efficient preconditioner.

3.4 CG implementation

CG was implemented for the following reformulation of the ITR scheme (3.7):

$$\begin{pmatrix} M_u & \frac{\tau}{2}K \\ \frac{\tau}{2}K^T & -(M_v + \frac{\tau}{2}S) \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix} = \begin{pmatrix} b_u \\ b_v \end{pmatrix} , \tag{3.20}$$

where

$$\begin{pmatrix} b_u \\ b_v \end{pmatrix} = \begin{pmatrix} 0 & -\tau K \\ -\tau K^T & \tau S \end{pmatrix} \begin{pmatrix} u_n \\ v_n \end{pmatrix} + \frac{\tau}{2} \begin{pmatrix} j_u(t_n) + j_u(t_{n+1}) \\ -j_v(t_n) - j_v(t_{n+1}) \end{pmatrix} , \tag{3.21}$$

and $u_{n+1} = u_n + \Delta u, v_{n+1} = v_n + \Delta v$. Hereby the saddle point system was treated with the Schur complement as described above. Writing (3.7) in this form is beneficious since this makes the zero vector a natural initial guess for the iteration process and saves one matrix-vector multiplication which is otherwise needed for the initial residual.[2)]

For an efficient usage it is important to choose a proper stopping criterion for CG. Too many iterations would mean a waste of effort, whereas too few might cause loss of stability[3)]. Using, for

[2)] Other initial vectors can be considered to assure that each Krylov iterate has truly second-order temporal consistency [4].

[3)] An alternative approach for the iterative linear system solution in implicit time integration is to use a fixed number of iterations per time step and to control stability of the approximate implicit scheme by adjusting the time step size [4, 5] convenience, the same notation for Δu and Δv, solving system (3.7) approximately with residual r_{cg} effectively means that the found Δv is a solution of the perturbed linear system

$$\mathcal{M}\Delta v = \frac{\tau}{2}K^T M_u^{-1} b_u - b_v - r_{\mathsf{cg}}\,, \tag{3.22}$$

where $\mathcal{M}$ is defined in (3.11) and the approximate solution Δu, Δv of (3.20) satisfies

$$\begin{pmatrix} M_u & \frac{\tau}{2}K \\ \frac{\tau}{2}K^T & -(M_v + \frac{\tau}{2}S) \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix} = \begin{pmatrix} b_u \\ b_v \end{pmatrix} + \begin{pmatrix} 0 \\ r_{\mathsf{cg}} \end{pmatrix}. \tag{3.23}$$

We stop CG as soon as for a certain constant δ [4)]

$$\|r_{\mathsf{cg}}\|_2 \leqslant \tau\, \|\frac{\tau}{2}K^T M_u^{-1} b_u - b_v\|_2\, \delta\,, \tag{3.24}$$

which means that the inexact ITR-CG scheme (3.23) can be seen as a perturbed form of the exact ITR scheme (3.20) where the perturbations are kept bounded. The purpose of this inequality is to enforce r_{cg} to be a fraction of the local truncation error of ITR for component v which we aim by means of an educated guess for δ. Note that r_{cg} just becomes the local truncation error upon substitution of the exact ODE solution. Choosing δ too large implies of course loss of ITR accuracy, whereas a too small δ wastes matvecs. We will give actual values of δ when we report our test results.

For the CG solution of the Schur complement system with the matrix $\mathcal{M}$ we have used two preconditioners. The first one is the sparse Cholesky factorization of the mass matrix M_v, the second is the incomplete Cholesky (IC) factorization with the drop tolerance $\varepsilon = 10^{-6}$ [32, 41] applied to the matrix

$$M_v + \frac{\tau}{2}S + \frac{\tau^2}{4}K^T K \tag{3.25}$$

obtained from $\mathcal{M}$ by deleting M_u^{-1}. The mass matrix preconditioner is readily available and as argued earlier, for ITR the costs of one mass matrix preconditioned CG iteration are roughly the same as the costs of one time step with CO2. This is because one CG iteration requires just one matvec with the preconditioned matrix (and several vector updates).

The IC(ε) preconditioner requires significant set up time. For example, for the fourth grid of Table 3.1 given in Section 3.5 the preparation cost required a CPU time sufficient for performing 90 to 100 matvecs with the preconditioned matrix $\mathcal{M}$. An attractive property of the IC(ε) preconditioner compared to the mass matrix preconditioner is a higher level of sparsity. For example, for $\varepsilon = 10^{-6}$ the sparsity is at least twice as large as for the Cholesky factors of the mass matrix. During integration the IC(ε) preconditioner therefore is slightly cheaper due to the higher level of sparsity. In our experiments we found little differences between numbers of iterations for the mass matrix and IC(ε) preconditioner. We therefore will report only iteration numbers for the first one. Note that the eigenvalues of the mass-matrix preconditioned $\mathcal{M}$ are given by (3.16) if we do have a Maxwell problem for which condition (2.7) holds.

3.5 Comparing ITR and CO2

In this section we compare the fully implicit integrator ITR, equipped with the above described preconditioned CG implementation, to method CO2.

3.5.1 A 3D Maxwell test problem

The comparison is based on tests with a three-dimensional (3D) Maxwell problem we earlier used in [6]. This problem is given in the (H, E) formulation

$$\begin{aligned} \mu\partial_t H &= -\nabla \times E\,, \\ \varepsilon\partial_t E &= \nabla \times H - \sigma E - J, \end{aligned} \tag{3.26}$$

[4)] Here and in the remainder $\|\cdot\|_2$ denotes the discrete inner product (L_2) norm.

Table 3.1: Grid parameters and time step size information for CO2.

grid	number of edges	longest edge $h_{\max}$	shortest edge $h_{\min}$	CO2 time step restriction	CO2 time step used
1	105	0.828	0.375	0.47	0.2
2	660	0.661	0.142	0.18	0.1
3	4632	0.359	0.0709	0.079	0.05
4	34608	0.250	0.0063	0.028	0.025
5	85308	0.118	0.0139	0.014	0.0125

with independent variables $(x, y, z) \in \Omega \subset \mathbb{R}^3$, $t \in [0, T]$, and initial and boundary conditions

$$E|_{t=0} = E_0(x, y, z), \qquad H|_{t=0} = H_0(x, y, z), \tag{3.27a}$$

$$(\vec{n} \times E)|_{\partial\Omega} = E_{\mathsf{bc}}, \qquad (\vec{n} \times H)|_{\partial\Omega} = H_{\mathsf{bc}}\,. \tag{3.27b}$$

The coefficients μ, ε and σ are taken constant in time and space and $\vec{n}$ denotes the outward unit normal vector to the boundary $\partial\Omega$. The boundary functions E_{bc} and H_{bc} vary in space and time. Specifically, $\Omega = [0, 1]^3$ and $T = 10$ and we choose the source current $J = J(x, y, z, t)$ such that the Maxwell system (3.26) allows a specific exact solution

$$E(x, y, z, t) = \alpha(t) E_{\mathsf{stat}}(x, y, z), \qquad H(x, y, z, t) = \beta(t) H_{\mathsf{stat}}(x, y, z), \tag{3.28}$$

where the scalar functions α, β and the vector functions E_{stat}, H_{stat} satisfy $\mu\beta'(t) = -\alpha(t)$ and $H_{\mathsf{stat}} = \nabla \times E_{\mathsf{stat}}$. The source function J is then defined as

$$J(x, y, z, t) = -(\varepsilon\alpha'(t) + \sigma\alpha(t))\, E_{\mathsf{stat}}(x, y, z) \;+\; \beta(t) \nabla \times H_{\mathsf{stat}}(x, y, z)\,, \tag{3.29}$$

and to satisfy (3.28) we choose

$$\begin{gathered} E_{\mathsf{stat}}(x, y, z) = \begin{pmatrix} \sin \pi y \, \sin \pi z \\ \sin \pi x \, \sin \pi z \\ \sin \pi x \, \sin \pi y \end{pmatrix}, \quad H_{\mathsf{stat}}(x, y, z) = \begin{pmatrix} \sin \pi x (\cos \pi y - \cos \pi z) \\ \sin \pi y (\cos \pi z - \cos \pi x) \\ \sin \pi z (\cos \pi x - \cos \pi y) \end{pmatrix}, \\ \alpha(t) = \sum_{k=1}^{3} \cos \omega_k t\,, \quad \beta(t) = -\frac{1}{\mu} \sum_{k=1}^{3} \frac{\sin \omega_k t}{\omega_k}\,, \end{gathered} \tag{3.30}$$

with $\omega_1 = 1$, $\omega_2 = 1/2$, $\omega_3 = 1/3$. Further, we take $\mu = 1$, $\varepsilon = 1$ and either $\sigma = 0$ or $\sigma = 60\pi$ (this corresponds with values encountered in real applications).

This 3D Maxwell problem is spatially discretized with first-order, first-type Nédélec edge finite elements on tetrahedral unstructured grids [34, 36, 37]. Although it is formulated with H and E as primary variables, the resulting semi-discrete system belongs to class (1.2). In [6] we observed first-order spatial convergence for E and second order for H. We have used the grids numbered four and five listed in Table 3.1 which displays grid parameters and step size information for CO2. To save space we refer to [6] and references therein for a more complete description of this test problem and its spatial discretization.

3.5.2 Test results

Table 3.2 reports computational costs in terms of matvecs for CO2 and ITR-CG for the fourth and fifth grid mentioned in Table 3.1. Two cases are distinguished, the zero conduction coefficient $\sigma = 0$ and the nonzero conduction coefficient $\sigma = 60\pi$, see Section 3.5.1. For both cases we have chosen $\delta = 0.05$ in the stopping criterion (3.24) and step sizes τ for ITR-CG much larger than the limit step size of CO2. For the chosen values the temporal errors remain smaller than the spatial ones, justifying the use of ITR-CG with respect to the full discretization error.

Table 3.2: Computational costs of CO2 (applied with maximal τ) versus the costs of ITR-CG (applied with different τ); stopping criterion parameter $\delta = 0.05$.

		τ	$\sigma = 0$ # matvecs per t.step	$\sigma = 0$ total # matvecs	$\sigma = 60\pi$ # matvecs per t.step	$\sigma = 60\pi$ total # matvecs
grid 4	CO2	0.025	1	400	1	400
	ITR/mass	0.0625	4.94	790	2.00	320
	ITR/mass	0.125	8.99	719	2.01	161
	ITR/mass	0.25	15.95	638	2.98	119
	ITR/mass	0.5	25.4	508	3.85	77
	ITR/mass	1.0	29.6	296	4.60	46
grid 5	CO2	0.0125	1	800	1	800
	ITR/mass	0.25	31.52	1261	5.3	212
	ITR/mass	0.5	47.5	950	6.65	133
	ITR/mass	1.0	57.8	578	7.6	76

Our first observation is that the number of CG iterations per ITR time step grows only sublinearly with the time step size τ, in particular for $\sigma = 60\pi$. For this reason ITR can become faster than CO2 for sufficiently large τ if δ is chosen properly (which appears to hold for $\delta = 0.05$). Taking δ ten times smaller results for the fourth grid and $\sigma = 0$ in the matvec sequence $(1088, 1020, 945, 827, 668)$, showing a greater expense than CO2 for the larger step sizes. Likewise, for $\sigma = 60\pi$ we find the sequence $(345, 250, 158, 117, 69)$, showing only a small expense growth for δ ten times smaller. As anticipated, the numbers increase as the grid gets finer. However, as the grid gets finer, the maximum allowable time step for CO2 does decrease too. This is also the case on the finest fifth grid even though it is more uniform than the fourth one, see Table 3.1.

Our second observation concerning Table 3.2 is that the number of CG iterations per time step for $\sigma = 60\pi$ is significantly smaller than for $\sigma = 0$. The reason is that for the current test problem M_v and S are identical up to a constant, see Section 2. Hence, for growing σ, the eigenvalues of the mass-preconditioned matrix $\mathcal{M}$ given by (3.16) get more clustered around $1 + \alpha\tau/2$ and the condition number $\lambda_{\max}/\lambda_{\min}$ decreases.

Note that in the ITR scheme one needs to repeatedly solve the linear system (3.20) where the matrix remains the same and only the right-hand side changes per time step. This suggests that computational effort can be saved by reusing the information generated by CG. One possible way of doing this is Method 2 of [18] which essentially consists of storing an orthonormal basis spanning the successive CG-solutions and starting every new CG process with a projection on the stored subspace. As evidenced in [18], Method 2 can lead to a significant saving in the total number of iterations. We have tested the method for this problem but have not observed any improvement. This is because the right-hand side of (3.20) changes quite significantly from one time step to another, thus making the projection procedure futile.

For $\delta = 0.05$, $\sigma = 0$ and the fourth grid, Figure 3.1 shows the time evolution of full (space and time) errors in $\|\cdot\|_2$ for CO2 and ITR. We see that the errors are comparable and more or less independent of τ which illustrates that the spatial error dominates. This is the sort of situation where implicit time stepping can be competitive. Our test with $\sigma = 0$ (undamped case) shows no distinct advantage when counting numbers of matvecs. On the other hand, the test with $\sigma = 60\pi$ is no doubt promising and warrants further investigation with a more sophisticated CG implementation, finer space grids and more test examples including variable conduction.

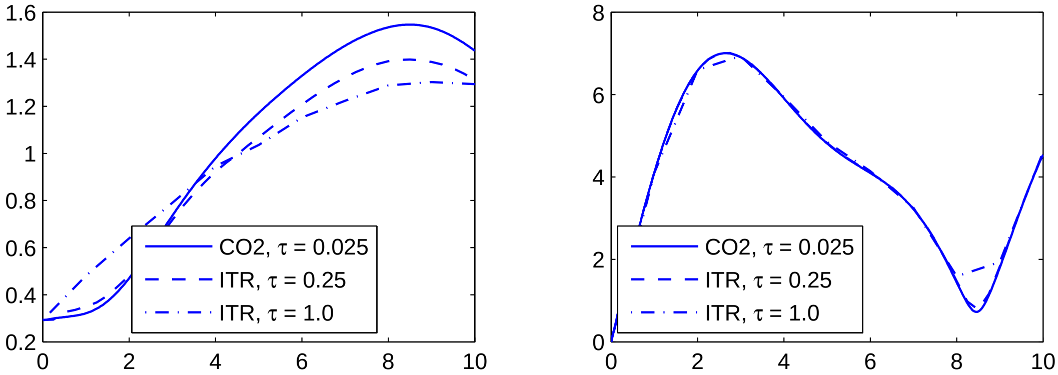

Figure 3.1: The full L_2 error (left E-field, right H-field) versus time for CO2 and ITR on the fourth grid, $\sigma = 0$, $\delta = 0.05$. ITR uses the mass matrix preconditioner. The costs for these runs are 400 matvecs for CO2 and 371 for ITR with $\tau = 0.25$ and 299 for ITR with $\tau = 1.0$ (see Table 3.2).

4 Exponential integration

The implicit trapezoidal rule ITR is a conventional method in the sense that it is a representative of the Runge-Kutta and linear multistep methods. The so-called exponential integration methods form another class being built on linearization and direct use of accurate, unconditionally stable approximations to the exponential operator. For this reason they are of potential interest to the Maxwell equations. Exponential integrators do have a long history [9, 21, 26, 29, 31, 38, 46] and have undergone a revival during the last decade, see e.g. [3, 8, 10, 23, 25, 33]. An important reason for this revival is the availability of the Krylov subspace iteration technique for approximating the exponential and the so-called derived φ-functions. In this section we will also use Krylov subspace iteration.

4.1 The exponential integrator EK2

For formulating our exponential integrator we rewrite the semi-discrete system (1.2) as

$$w' = F(t, w)\,, \qquad F(t, w) = Jw + f(t)\,, \tag{4.1}$$

where $J = M^{-1}A$ and $f(t) = M^{-1}g(t)$ and w, M, A and $g(t)$ are defined as in (3.3). For this ODE system we consider the second-order exponential integrator

$$w_{n+1} = w_n + \tau\varphi_1(\tau J)F(t_n, w_n) + \tau\varphi_2(\tau J)\left(f(t_{n+1}) - f(t_n)\right)\,, \tag{4.2}$$

where $\varphi_1(z) = (e^z - 1)/z$ and $\varphi_2(z) = (\varphi_1(z) - 1)/z$. This second-order method follows from linearly interpolating f over $[t_n, t_{n+1}]$ in the variation of constants formula

$$w(t_{n+1}) = e^{\tau J}w(t_n) + \int_0^\tau e^{(\tau - s)J}f(t_n + s)ds \tag{4.3}$$

and subsequently computing the resulting integrals analytically. The earliest papers we know of where this interpolating approach with exact, analytic computation of integrals has been used are [9] and [31]. In the recent literature this approach is sometimes called exponential time differencing, see e.g. [10]. A second-order method closely related to (4.2) where f' is used reads

$$w_{n+1} = w_n + \tau\varphi_1(\tau J)F(t_n, w_n) + \tau^2\varphi_2(\tau J)f'(t_n)\,. \tag{4.4}$$

This method belongs to a class of exponential Runge-Kutta-Rosenbrock methods [7, 25].

In the literature many formulas of higher order are proposed. Here we restrict ourselves to using the second-order method (4.2) because we wish to compare to the second-order method CO2 and the spatial discretization of our test example does not exceed order two either. Per integration step this method requires the approximation of two vectors $\varphi(\tau J)b$ representing $\varphi_1(\tau J)F(t_n, w_n)$ and $\varphi_2(\tau J)\,(f(t_{n+1}) - f(t_n))$ for which we use Krylov subspace iteration, similar as in [23] and [25] and in related work on exponential integration. In the remainder of the paper we will refer to (4.2) as method EK2 (Exponential Krylov 2nd-order). More background information on EK2 supporting its choice in the current investigation is given in the Appendix of this paper.

4.2 Krylov subspace iteration

Let e_1 be the first unit vector in $\mathbb{R}^{n+m}$ ($n+m$ is the dimension of the matrix J). Krylov subspace iteration for $\varphi(\tau J)b$ computes an approximation $d \approx \varphi(\tau J)b$ through

$$d = V_k\,\varphi(\tau H_k)\,e_1 \cdot \|b\|\,, \tag{4.5}$$

where $V_k = [v_1, \ldots, v_k]$ is the $(n+m) \times k$ matrix containing the Arnoldi (or Lanczos) basis of the k-th Krylov subspace with respect to τJ and b and H_k is an $k \times k$ upper Hessenberg matrix. So $\varphi(\tau H_k)$ replaces $\varphi(\tau J)$ which explains the success of this method as long as $k \ll n+m$, because then $\varphi(\tau H_k)$ is relatively cheap to compute, e.g. through the Schur decomposition. The costs of building d mainly consists of k matrix-vector multiplications with τJ within the Arnoldi process. Hereby it is noted that one such multiplication costs about the same as one single integration step with method CO2. So when comparing EK2 to CO2 with regard to CPU time, the latter can be applied with a k times smaller step size.

A practical drawback is that matrix V_k must be kept in storage before d can be formed. Hence if $n+m$ is large as is the case in large-scale 3D simulations, the storage requirement for k vectors of dimension $n+m$ can be substantial. For example, a worst-case estimate for skew-symmetric matrices with uniformly distributed eigenvalues says that k can get as large as $\|\tau J\|$ before the iteration error starts to decay [22]. It is obvious that this may require too much storage if we allow $\|\tau J\| \gg 1$ which after all is the main purpose of using an exponential integrator like EK2. Fortunately, in applications one often obtains convergence for k substantially smaller than $\|\tau J\|$. If not one can split the time interval in subintervals and use restarts, although at the expense of additional work (the software package from [43] does this automatically). For the theory behind Krylov subspace iteration for matrix functions we refer to the research monograph [48] and to the seminal papers [14, 15, 22, 28, 40, 47] and references therein.

4.3 Krylov implementation

Like for CG we need a proper stopping criterion for the Arnoldi process. Consider the step with (4.2) from t_n to t_{n+1} starting in w_n and write in short

$$w_{n+1} = w_n + \tau\Phi_1 + \tau\Phi_2\,. \tag{4.6}$$

We stop after k_1, k_2 iterations for Φ_1, Φ_2 approximating w_{n+1} by

$$\hat{w}_{n+1} = w_n + \tau\Phi_1^{(k_1)} + \tau\Phi_2^{(k_2)}. \tag{4.7}$$

Ideally, $\|w_{n+1} - \hat{w}_{n+1}\|$ is smaller than the unknown local truncation error for w_{n+1} which we represent by the quantity $\tau\|w_n\|\delta$ for a certain constant δ. So we require

$$\|w_{n+1} - \hat{w}_{n+1}\| \leqslant \tau\|w_n\|\,\delta\,, \tag{4.8}$$

which holds if

$$\|\Phi_i - \Phi_i^{(k_i)}\| \leqslant \frac{1}{2}\|w_n\|\,\delta\,, \qquad i = 1, 2\,. \tag{4.9}$$

The number of iterations k_i, $i = 1, 2$, is now chosen as follows. We assume for $i = 1, 2$ separately that (4.9) is satisfied if, in the L_2 norm, p_δ times in succession

$$\|\Phi_i^{(k_i)} - \Phi_i^{(k_i-1)}\|_2 \leqslant \frac{1}{2}\,\|w_n\|_2\,\delta\,, \tag{4.10}$$

where p_δ is an integer we can choose. Like for ITR we use constant τ and have not implemented a local error estimator. So also here we make an educated guess for δ and assume that (4.10) works properly. In our experiments this turned out to be the case, even with $p_\delta = 1$ which we have chosen henceforth. In our tests all occurring matrix functions $\varphi(\tau H_k)$ have been computed exactly using the exponential operator. Finally we note that $\Phi_2 = \mathcal{O}(\tau)$ because of the difference $f(t_{n+1}) - f(t_n)$. This means that normally this term will require less Krylov subspace iterations than the first one which is confirmed in the experiments.

4.4 Comparing EK2 and CO2

We have repeated the experiments of Section 3.5.2 with ITR replaced by EK2, again focusing on the comparison to method CO2 in terms of workload expressed in matvecs. For the chosen step size range the spatial error again dominates (so Figure 3.1 also applies to EK2) justifying our focus on workload without referring to the temporal errors. Workload is found in Table 4.1 for $\delta = 10^{-3}$ and $p_\delta = 1$, see (4.10). The $\sigma = 0$ test indicates that for problems without damping EK2 will be more costly in matvecs when compared to CO2, let alone the much larger memory demand. Lowering or increasing δ will not change much for the larger step sizes. For example, for $\sigma = 0$ and grid 4 we find for $\delta = 10^{-2}$ and $\delta = 10^{-4}$ the total matvec sequences $(1900, 1457, 1222, 1132, 1075)$ and $(2942, 2043, 1592, 1363, 1230)$.

The $\sigma = 60\pi$ test is much more favorable for EK2. We see that for step sizes far away from the critical CO2 limit method EK2 becomes competitive in terms of matvecs, similar to what we have observed for ITR. For EK2 however the gain is less substantial and given the large memory demand this method will likely not to be of great practical interest when it comes to truly large-scale computations. A positive point of EK2 is that for the range of step sizes used its temporal error behavior turned out to be very good. Of course, would the temporal error dominate, method CO2 will be hard to beat as it is optimally efficient (just one matvec per time step).

Table 4.1: Computational costs of CO2 (applied with maximal τ) versus the costs of EK2 (applied with different τ); stopping criterion parameters $\delta = 10^{-3}$, $p_\delta = 1$.

		τ	$\sigma = 0$ # matvecs per t.step	$\sigma = 0$ total # matvecs	$\sigma = 60\pi$ # matvecs per t.step	$\sigma = 60\pi$ total # matvecs
grid 4	CO2	0.025	1	400	1	400
	EK2	0.0625	14.93	2388	11.48	1836
	EK2	0.125	21.96	1757	13.7	1096
	EK2	0.25	35.45	1418	16.35	654
	EK2	0.5	62.2	1252	21.55	431
	EK2	1.0	116	1160	29.6	296
grid 5	CO2	0.0125	1	800	1	800
	EK2	0.25	61.88	2475	25.88	1035
	EK2	0.5	116.5	2330	37.10	742
	EK2	1.0	196.8	1968	53	530

5 Chebyshev series expansion

In the remainder of the paper we assume that in (1.2) no damping due to conduction is active and that there are no sources.[5)] This poses the task of approximating exponents of huge skew-symmetric matrices. An elegant approximation method for this task proposed in [45] is based on a Chebyshev series expansion. Unlike Krylov subspace iteration this method requires a fixed amount of storage (only four arrays) and hence can be a nice alternative to Krylov subspace iteration in skew-symmetric cases where storage becomes a genuine problem. In [45] the series expansion was proposed and analyzed as a spectral method in time for hyperbolic equations. Two application papers are [30] and [12] devoted to, respectively, the Schrödinger and Maxwell equations. In [24] the series expansion is reported to be less efficient than Krylov for a quantum-classical molecular dynamics computation. In this section we will present another comparison between the Krylov and Chebyshev method.

Chebyshev series expansion seems less known than Krylov subspace iteration. We therefore first review it following [45]. Consider system (3.3) and assume $S = 0$ and a zero source function g. Its skew-symmetric transformed counterpart (cf. (2.3)) then reads

$$\tilde{w}' = \tilde{A}\tilde{w}\,, \quad \tilde{A} = \begin{pmatrix} 0 & -\tilde{K} \\ \tilde{K}^T & 0 \end{pmatrix}. \tag{5.1}$$

We will review the Chebyshev method for this transformed system. The method rests on the Chebyshev series expansions 9.1.44 and 9.1.45 for the trigonometric functions given in [1]. For e^z these expansions combine for $z \in i\mathbb{R}$ to

$$e^z = J_0(R) + 2\sum_{k=1}^{\infty} J_k(R)\, Q_k(z/R)\,, \tag{5.2}$$

where $R > |z|$ is a constant, J_k is the Bessel function of order k and $Q_k(\tilde{z}) = i^k T_k(-i\tilde{z}), \tilde{z} = z/R$ with T_k the k-th degree Chebyshev polynomial of the first kind. From the Chebyshev recursion $T_{k+1}(x) = 2x\,T_k(x) - T_{k-1}(x)$, $T_0(x) = 1$, $T_1(x) = x$ follows

$$Q_{k+1}(\tilde{z}) = 2\tilde{z}\,Q_k(\tilde{z}) + Q_{k-1}(\tilde{z})\,, \qquad Q_0(\tilde{z}) = 1\,, \quad Q_1(\tilde{z}) = \tilde{z}\,, \tag{5.3}$$

showing that $Q_k(\tilde{z}) = Q_k(z/R)$ is a k-th degree polynomial with real coefficients. Truncating (5.2) to N terms yields the polynomial approximation

$$e^z \approx P_N(z) = J_0(R) + 2\sum_{k=1}^{N} J_k(R)\, Q_k(z/R)\,, \tag{5.4}$$

which defines the expansion method for approximating $\exp(t\tilde{A})$ (replace z by $t\tilde{A}$).

For a selected degree N the vector

$$\tilde{w}_N(t) = P_N(t\tilde{A})\tilde{w}(0) \tag{5.5}$$

defines the sought approximation to $\tilde{w}(t)$ (no intermediate results are produced within the time interval $[0, t]$). Apart from vector additions and Bessel function computations, computationally this approximation amounts to N matrix-vector multiplications carried out in the matrix-valued counterpart of recursion (5.3). Denote $\tilde{w}_k = Q_k(\frac{t}{R}\tilde{A})\tilde{w}(0)$. Then it follows directly that

$$\tilde{w}_N(t) = J_0(R)\,\tilde{w}(0) + 2\sum_{k=1}^{N} J_k(R)\,\tilde{w}_k\,, \tag{5.6}$$

where

$$\tilde{w}_{k+1} = \frac{2t}{R}\tilde{A}\,\tilde{w}_k + \tilde{w}_{k-1}\,, \qquad \tilde{w}_0 = \tilde{w}(0)\,, \quad \tilde{w}_1 = \frac{t}{R}\tilde{A}\,\tilde{w}_0\,. \tag{5.7}$$

[5)] Time-independent sources can be included, see Remark 5.2. Unless noted otherwise we assume a zero source.

With only four storage vectors $\tilde{w}_k$ needed to form $\tilde{w}_N(t)$, the method has very low storage requirements as compared to Krylov subspace iteration.

R must be taken greater than or equal to the spectral radius $\sigma(t\tilde{A})$ of $t\tilde{A}$. Then all imaginary eigenvalues z of $t\tilde{A}$ lie within the interval $[-iR, iR]$ assuring that $|Q_k(z/R)| \leqslant 1$ for all eigenvalues z while all solution modes are captured. Would one take $R < \sigma(t\tilde{A})$, huge errors will occur because then $|Q_k(z/R)| \gg 1, k \gg 1$ for eigenvalues which are not captured. The condition $R \geqslant \sigma(t\tilde{A})$ can therefore be seen as a stability property. The expansion converges due to the fact that $J_k(R)$ converges to zero exponentially fast for $k \geqslant N$ if $N > R$ is sufficiently large, see formula 9.3.1 in [1]. So the working condition for the Chebyshev method reads

$$N > R \geqslant \sigma(t\tilde{A})\,. \tag{5.8}$$

With this condition the spectral norm accuracy of the truncation can be made arbitrarily small at any time t by choosing N sufficiently large. If $\sigma(t\tilde{A})$ is small, say near one, this can also be achieved with the Taylor expansion. However, computing the Taylor expansion for $\sigma(t\tilde{A}) \gg 1$ is not stable and will not work.

In actual application N can be automatically adapted to the desired truncation accuracy as follows. Suppose the aim is

$$\|\tilde{w}(t) - \tilde{w}_N(t)\| \leqslant tol\,, \tag{5.9}$$

where tol is a given tolerance. Then estimate $R \geqslant \sigma(t\tilde{A})$ and execute (5.6) - (5.7) for $N_R = \lceil R \rceil$. Next continue the expansion for $k = N_R + 1, N_R + 2, \ldots$ until for a prescribed number p_{tol} in succession $|J_k(R)| \leqslant tol$.

Example 5.1 To illustrate this simple adaptive algorithm we present a numerical example using $p_{tol} = 3$. The example is academic but instructive for our purpose. It is based on the periodic advection problem $u_t + u_x = 0$ on the unit interval for the peaked initial function $u(x, 0) = (\sin(\pi x))^{100}$ and second-order symmetric differences on a uniform grid with $N_g = 500$ nodes for spatial discretization. We then approximate the resulting semi-discrete solution $\tilde{w}(t) = \exp(t\tilde{A})\tilde{w}(0)$ at time $t = 1$. So we advect the grid function $u_h(x, 0)$ to the right over one time unit. The spatial discretization error $\|u_h(x, t) - \tilde{w}(t)\|_2$ at $t = 1$ is approximately equal to $1.5\,10^{-1}$ ($2.9\,10^{-2}$ in the infinity norm).

We put R equal to the spectral radius $\sigma(t\tilde{A}) = 500$ and choose as tolerances $tol = 10^{-3}, \ldots, 10^{-8}$. Table 5.1 contains the values of N computed by the adaptive algorithm along with the error norms $\|\tilde{w}(t) - \tilde{w}_N(t)\|_2$ which according to (5.9) are anticipated to be close to tol. The increase of N and resulting decrease of the errors with tol can be seen to be almost perfect and in line with the exponential decay property of the Bessel function. Because of the exponential decay a substantial gain in temporal accuracy is obtained for just little more work (though redundant in view of the spatial error). The good performance of the Chebyshev method is in line with experiences reported in [12, 45].

tol	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
N *errors*	524 $2.7\,10^{-3}$	530 $3.6\,10^{-4}$	536 $3.6\,10^{-5}$	542 $3.1\,10^{-6}$	548 $2.2\,10^{-7}$	554 $1.3\,10^{-8}$

Table 5.1: Example 5.1: numbers of Chebyshev expansion terms and errors $\|\tilde{w}(1) - \tilde{w}_N(1)\|_2$.

Next, to elucidate the storage issue, we apply Krylov subspace iteration with a fixed, prescribed number of iterations tabulated with N in Table 5.2. It is clear that for reaching an acceptable accuracy, a substantial number of iterations is required. If storage would be limited, this limitation can be overcome with a restart procedure as for example used in the solver EXPV [43]. But then the total number of iterations can grow substantially and even considerably larger than $\sigma(t\tilde{A})$ (confirmed by EXPV). In such a situation the Chebyshev method can readily offer a practical and even faster alternative.

N	50	100	150	200	250	300
errors	5.9	7.5	8.8	$8.0\,10^{-4}$	$1.2\,10^{-13}$	$2.2\,10^{-13}$

Table 5.2: Example 5.1: numbers of Krylov matvecs and errors $\|\tilde{w}(1) - \tilde{w}_N(1)\|_2$.

Finally we note that, due to the structure of the matrix $\tilde{A}$ and the chosen initial condition $u(x, 0)$ in this example, at $t = 1$ non-restarted Krylov subspace methods will require approximately $N_g/2$ iterations (N_g being the number of grid points) to get a reasonably well approximation to $\tilde{w}(1)$. In this sense, the current test presents a worst-case scenario for the Krylov subspace methods. A possible solution would be to apply preconditioning as proposed in [16]. With this technique we get mesh independent convergence within 100 iterations. ◇

Remark 5.2 In actual application we use the variable w of the original system (3.3) (with zero S and g). This requires to back-transform all stages of the computation resulting in

$$w_N(t) = P_N(Bt)w(0) = J_0(R)\,w(0) + 2\sum_{k=1}^{N} J_k(R)\,w_k\,, \tag{5.10}$$

where

$$w_{k+1} = \frac{2t}{R}Bw_k + w_{k-1}\,, \qquad w_0 = w(0)\,, \quad w_1 = \frac{t}{R}B\,w_0\,. \tag{5.11}$$

where $B = M^{-1}A$.

The Chebyshev method can be modified for approximating $\varphi(z) = (e^z - 1)/z$ which enables us to include constant in time sources. Consider the system $w' = Bw + f$ with f constant. The counterpart of (5.10) defining $w_N(t) \approx w(t) = w(0) + t\varphi(tB)\,w'(0)$ reads

$$w_N(t) = w(0) + 2t\sum_{k=1}^{N} J_k(R)\,w_k\,, \tag{5.12}$$

where

$$w_{k+1} = \frac{2t}{R}Bw_k + w_{k-1} + \frac{2\delta_k}{R}w'(0)\,, \qquad w_0 = 0\,, \quad w_1 = \frac{1}{R}\,w'(0)\,. \tag{5.13}$$

Herein $\delta_k = 1$ for k even and $\delta_k = 0$ for k odd. ◇

6 Concluding remarks

Maxwell's equations (1.1) provide a prime example of a damped wave equation system. After spatial discretization such systems are commonly integrated in time by implicit-explicit methods such as method CO2 defined by (3.1) which is prototypical for Maxwell's equations. CO2 is symmetric and thus of second order and requires just one derivative evaluation per time step which makes it very economical. However, the step size is limited by stability which may turn out restrictive, for example when the spatial error dominates for step sizes larger than the incurred step size limit. In such cases implicit time stepping, for which no such limit exists, may come into sight.

In the setting of the generic semi-discrete system (1.2) we have examined the feasibility of implicit time stepping for three different techniques:

(i) The conventional integrator ITR (Implicit Trapezoidal Rule, see Section 3) combined with preconditioned CG (Conjugate Gradient) iteration. Experiments with the 3D problem posed in Section 3.5.1 indicate that in the absence of conduction (no damping) our ITR-CG implementation based on either Schur-complement mass-matrix or incomplete-Cholesky preconditioning falls short. To truly become competitive with CO2 for problems without conduction more effective preconditioners are needed. Whether these exist for the linear systems we are facing, is an open

question. On the other hand, for our test problem with conduction the experiments were no doubt promising for the ITR-CG implementation. This warrants further investigation to the effectiveness of implicit time stepping for problems with conduction.
(ii) The exponential integrator EK2 (Exponential Krylov 2nd order, see Section 4) combined with Arnoldi-based Krylov subspace iteration to deal with the φ functions. For this combination we have reached similar conclusions as for ITR-CG. For conduction free problems CO2 remains the method of choice, whereas with conduction EK2 can become competitive, but most likely not more efficient than a well-designed ITR-CG implementation. Given, in addition, the substantial memory demand, we consider this method not promising for truly large-scale Maxwell computations.
(iii) The Chebyshev expansion method for computing exponents of huge skew-symmetric matrices, see Section 5. This method is meant for conduction-free problems with constant source terms when output is required just for a few single time points covering a long time span. The method of choice for this task is Krylov-Arnoldi, except when memory restrictions occur due to too many iterations. In such cases the Chebyshev method offers a practical alternative.

Acknowledgement M.A. Botchev acknowledges financial support from the BSIK ICT project BRICKS through the subproject MSV1 Scientific Computing.

A Appendix on the exponential integrator EK2

A.1 Connection with the Adams-Moulton method

EK2, that is method (4.2), can also be seen to belong to the class of $(k+1)$-st order multistep methods

$$w_{n+1} = R(\tau_n J_n) w_n + \sum_{l=0}^{k} \tau_n \beta_l(\tau_n J_n) \left[w'_{n+1-l} - J_n w_{n+1-l} \right], \tag{A.1}$$

where F may be nonlinear in w, J_n is an arbitrary matrix, $R(z) = e^z + \mathcal{O}(z^{k+2}), z \to 0$ and

$$\begin{array}{ll} \sum_{l=0}^{k} q_{l-1}^{j-1} \beta_l(z) = \varphi_j(z)\,, & j = 1, \ldots, k+1\,, \\ \varphi_1(z) = (R(z)-1)/z\,, \;\; \varphi_{j+1}(z) = (j\varphi_j(z) - 1)/z\,, & j = 1, \ldots, k\,, \\ q_l = (t_{n-l} - t_n)/\tau_n\,, \; \tau_n = t_{n+1} - t_n\,, & l = -1, 0, \ldots, k\,. \end{array} \tag{A.2}$$

Putting $k = 1, R(z) = e^z, \tau_n = \tau$ and $J_n = J$, a simple calculation leads us to EK2. Method (A.1) is a generalization of the classical, variable step size, Adams-Moulton method. The precise formulation (A.1)-(A.2) stems from [26, 46]. An earlier closely related Adams-Bashforth paper is [38]. As a further example we give the fixed-step fourth-order method from class (A.1) which for system (4.1) can be written as

$$w_{n+1} = w_n + \tau\varphi_1(\tau J) F(t_n, w_n) + \tau\varphi_2(\tau J)\, D_{n,2} + \tau\varphi_3(\tau J)\, D_{n,3} + \tau\varphi_4(\tau J)\, D_{n,4}\,. \tag{A.3}$$

Evaluating derivatives of f at $t = t_n$, the $D_{n,j}$ satisfy

$$\begin{array}{lclcl} D_{n,2} & = & \frac{1}{3} f_{n+1} + \frac{1}{2} f_n - f_{n-1} + \frac{1}{6} f_{n-2} & = & \tau f^{(1)} + \frac{1}{12}\tau^4 f^{(4)} + \mathcal{O}(\tau^5)\,, \\ D_{n,3} & = & \frac{1}{2} f_{n+1} - f_n + \frac{1}{2} f_{n-1} & = & \frac{1}{2}\tau^2 f^{(2)} + \frac{1}{24}\tau^4 f^{(4)} + \mathcal{O}(\tau^6)\,, \\ D_{n,4} & = & \frac{1}{6} f_{n+1} - \frac{1}{2} f_n + \frac{1}{2} f_{n-1} - \frac{1}{6} f_{n-2} & = & \frac{1}{6}\tau^3 f^{(3)} - \frac{1}{12}\tau^4 f^{(4)} + \mathcal{O}(\tau^5)\,. \end{array} \tag{A.4}$$

So the $D_{n,j}$ act as correction terms of decreasing size $\mathcal{O}(\tau^{j-1})$ which can be exploited in computing the vectors $\varphi_j(\tau J)\, D_{n,j}$ by means of the Krylov method.

A.2 Stiff source terms

The source function $f(t)$ of (4.1) may grow without bound if the spatial grid is refined due to contributions from time-dependent boundary functions (stiff source term). For Maxwell's equations these contributions are proportional to h^{-1} for $h \to 0$ where h is the spatial grid size. Stiff source terms may cause order reduction, that is, the actual order observed under simultaneous space-time grid refinement can be smaller than the ODE order observed on a fixed space grid. Assuming sufficient differentiability of the exact solution $w(t)$ we will prove that method EK2 is free from order reduction for any $f(t)$ and any stable J with its spectrum in $\mathbb{C}^-$ (so not necessarily emanating from Maxwell's equations).

First we expand the right-hand side of EK2 at $t = t_n$ for $w_n = w(t_n)$. By eliminating $f(t_n)$ and $f(t_{n+1})$ through the relation $f(t) = w'(t) - Jw(t)$ this yields

$$\hat{w}_{n+1} = w + \tau\varphi_1 w' + \tau\varphi_2 \sum_{j=1} \frac{1}{j!}\,\tau^j \left(w^{(j+1)} - Jw^{(j)}\right)\,, \tag{A.5}$$

where $w = w(t_n)$, etc., and $\varphi_k = \varphi_k(\tau J)$. Using the definition of φ_2 we next eliminate the Jacobian J from this expansion and reorder some terms. This yields

$$\hat{w}_{n+1} = w + \tau w' + (\frac{1}{2} + \psi)\tau^2 w'' + S\,, \tag{A.6}$$

where $\psi = \varphi_2 - \frac{1}{2}\varphi_1$ and

$$S = \sum_{j=3} \left(\frac{1}{j!}\,(I - \varphi_1) + \frac{1}{(j-1)!}\,\varphi_2\right) \tau^j w^{(j)}\,. \tag{A.7}$$

In what follows remainder terms $\mathcal{O}(\tau^p)$ are assumed independent of J and f implying proportionality to only τ^p for $\tau \to 0$ and $\|J\|, \|f\| \to \infty$. The local truncation error $\delta_n = w(t_{n+1}) - \hat{w}_{n+1}$ thus can be expressed as

$$\delta_n = -\psi\tau^2 w'' - S + \mathcal{O}(\tau^3)\,, \tag{A.8}$$

where the term $\mathcal{O}(\tau^3)$ is fully determined by solution derivatives. Further, because J is stable, the matrix functions φ_k featuring in S are bounded. This means that $S = \mathcal{O}(\tau^3)$ so that

$$\delta_n = -\psi\tau^2 w'' + \mathcal{O}(\tau^3)\,. \tag{A.9}$$

The matrix function ψ is also bounded implying $\delta_n = \mathcal{O}(\tau^2)$. Consequently, when adding up all preceding local errors towards the global error $\varepsilon_{n+1} = w(t_{n+1}) - \hat{w}_{n+1}$ in the standard way through the recursion

$$\varepsilon_{n+1} = e^{\tau J}\varepsilon_n + \delta_n\,, \tag{A.10}$$

we will loose one power of τ and predict first-order instead of second-order convergence. We can come around this non-optimal result by adopting the approach of Lemma II.2.3 from [27]. Write

$$\delta_n = (I - e^{\tau J})\xi_n + \mathcal{O}(\tau^3)\,, \qquad \xi_n = -(I - e^{\tau J})^{-1}\psi(\tau J)\tau^2 w''(t_n)\,. \tag{A.11}$$

Introducing $\hat{\varepsilon}_n = \varepsilon_n - \xi_n$ we can write

$$\hat{\varepsilon}_{n+1} = e^{\tau J}\hat{\varepsilon}_n + \hat{\delta}_n\,, \qquad \hat{\delta}_n = -(\xi_{n+1} - \xi_n) + \mathcal{O}(\tau^3)\,. \tag{A.12}$$

Since J is a stable Jacobian, the matrix function featuring in ξ_n is bounded which implies that $\xi_n = \mathcal{O}(\tau^2)$ and $\xi_{n+1} - \xi_n = \mathcal{O}(\tau^3)$ giving $\hat{\delta}_n = \mathcal{O}(\tau^3)$. Now we can add up all preceding local errors in the standard way to conclude second-order convergence for method EK2. We here tacitly assumed that $\varepsilon_0 = 0$ so that $\hat{\varepsilon}_0 = -\xi_0 = \mathcal{O}(\tau^2)$. This convergence result holds for any stable Jacobian J and any source function $f(t)$ eliminating the possibility of order reduction due to contributions from time-dependent boundaries. With a slight change the proof is also valid for the alternative method (4.4).

Example. We will illustrate the above convergence result for EK2 with a simple yet instructive numerical example. By way of contrast so as to emphasize that when it occurs order reduction may work out badly, we will also apply the method

$$w_{n+1} = e^{\tau J}\left(w_n + \frac{1}{2}\tau f(t_n)\right) + \frac{1}{2}\tau f(t_{n+1})\,. \tag{A.13}$$

This exponential integration method is obtained from the variation of constants formula (4.3) by directly approximating the integral term with the quadrature trapezoidal rule, rather than first interpolating and integrating the obtained terms analytically. The method can also be obtained through time splitting. As an ODE method it is second-order consistent and even symmetric. However, it suffers from order reduction. In fact, for $\tau \to 0$ and $\|J\|, \|f\| \to \infty$ it is not even convergent which we will illustrate numerically. Also, unlike EK2, the method is not exact for constant f.

We have integrated the 2×2-system (Prothero-Robinson type model from stiff ODEs [11])

$$w' = \begin{pmatrix} 0 & -s \\ s & 0 \end{pmatrix} w + f(t)\,, \qquad f(t) = g'(t) - \begin{pmatrix} 0 & -s \\ s & 0 \end{pmatrix} g(t)\,, \quad g(t) = e^t \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{A.14}$$

Putting $w(0) = [1,1]^T$ yields for any J the solution $w(t) = [e^t, e^t]^T, t \geqslant 0$. So we can take s as large as we wish to illustrate the order reduction phenomenon. Figure A.1 shows convergence results for $s = 10, \tau = \frac{1}{5}2^{-j}$ and $s = 5 \cdot 2^j, \tau = \frac{1}{5}2^{-j}$ where $j = 1, \ldots, 10$. So in the first case $\|\tau J\| \to 0$ and $\|\tau f\| \to 0$ whereas in the second case $\|\tau J\|$ and $\|\tau f\|$ are fixed and thus $\|J\|$ and $\|f\|$ are increasing. With the first case we test normal ODE convergence and with the second case order reduction. We plot maximum absolute errors at $t = 1$ versus τ for EK2 (left plot) and (A.13) (right plot). The dashed line is the exact order two line, +-marks refer to s fixed and o-marks to increasing s. EK2 is shown to converge in the right manner for both cases whereas in both cases (A.13) is much less accurate and in particular suffers from severe order reduction in the second case even resulting in lack of convergence.

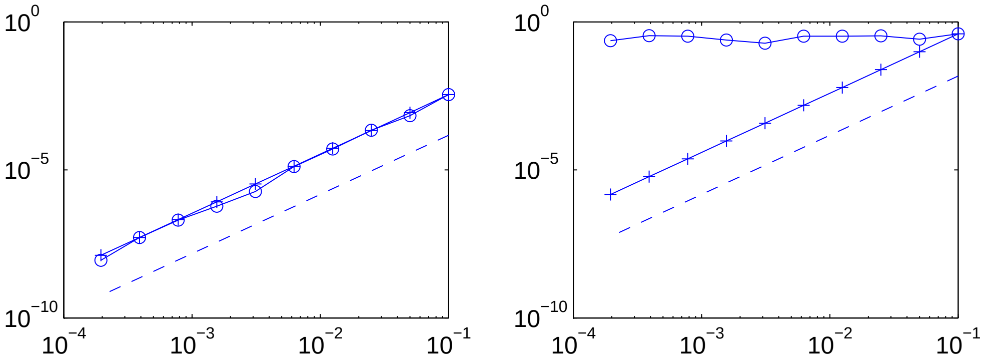

Figure A.1: Maximum absolute errors at $t = 1$ for EK2 (left plot) and (A.13) (right plot). The dashed line is the exact order two line, +-marks refer to s fixed and o-marks to increasing s.

References

[1] M. Abramowitz, I.A. Stegun (1968), *Handbook of Mathematical Functions.* Fifth edition, Dover Publications, Inc., New York.

[2] S. Benhassine, W.P. Carpes Jr., L. Pichon (2000), *Comparison of mass lumping techniques for solving the 3D Maxwell's equations in the time domain.* IEEE Transactions on Magnetics 36, pp. 1548–1552.

[3] H. Berland, B. Skaflestad, W. Wright (2007), *EXPINT–A Matlab package for exponential integrators.* ACM Transactions on Mathematical Software 33, No. 1, Article 4.

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

[5] M.A. Botchev, H.A. van der Vorst (2001), *A parallel nearly implicit scheme.* J. Comp. Appl. Math. 137, pp. 229–243.

[6] M.A. Botchev, J.G. Verwer (2008), *Numerical integration of damped Maxwell equations.* CWI Preprint MAS-E0804, http://ftp.cwi.nl/CWIreports/MAS/MAS-E0804.pdf (to appear in SIAM J. Sci. Comput.)

[7] M. Caliari, A. Ostermann (2008), *Implementation of exponential Rosenbrock-type integrators.* Appl. Numer. Math., to appear.

[8] E. Celledoni, D. Cohen, B. Owren (2008), *Symmetric exponential integrators with an application to the cubic Schrödinger equation.* Found. Comput. Math. 8, pp. 303–317.

[9] J. Certaine (1960), *The solution of ordinary differential equations with large time constants.* In: Mathematical Methods for Digital Computers, eds. A. Ralston, H.S. Wilf, K. Einstein, pp. 128–132, Wiley, New York.

[10] S.M. Cox, P.C. Matthews (2002), *Exponential time differencing for stiff systems.* J. Comput, Phys. 176, pp. 430–455.

[11] K. Dekker, J.G. Verwer (1984), *Stability of Runge-Kutta Methods for Stiff Nonlinear Differential Equations.* North-Holland, Amsterdam.

[12] H. De Raedt, K. Michielsen, J.S. Kole, M.T. Figge (2003), *Solving the Maxwell equations by the Chebyshev method: a one-step finite-difference time-domain algorithm.* IEEE Transactions on Antennas and Propagation 51, pp. 3155–3160.

[13] H. De Raedt (2005), *Advances in unconditionally stable techniques.* Ch. 18 in 'Computational Electrodynamics, The Finite-Difference Time-Domain Method', eds. A. Taflove, S.C. Hagness, Artech House, Boston and London.

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

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

[16] J. van den Eshof, M. Hochbruck (2006), *Preconditioning Lanczos approximations to the matrix exponential.* SIAM J. Sci. Comput. 27, pp. 1438–1457.

[17] I. Faragó, R. Horváth, W.H.A. Schilders (2005), *Investigation of numerical time-integrations of Maxwell's equations using the staggered grid spatial discretization.* Int. J. Numer. Model Electron Network Dev. 18, pp. 149–169.

[18] P.F. Fischer (1996), *Projection techniques for iterative solution of $Ax = b$ with successive right-hand sides.* Comput. Methods Appl. Mech. Engrg. 163, pp. 193–204.

[19] A. Fisher, R.N. Rieben, G.H. Rodrigue, D.A. White (2005), *A generalized mass lumping technique for vector finite-element solutions of the time-dependent Maxwell equations.* IEEE Transactions on Antennas and Propagation 53, pp. 2900-2910.

[20] B. Fornberg, J. Zuev, J. Lee (2007), *Stability and accuracy of time-extrapolated ADI-FDTD methods for solving wave equations.* J. Comp. Appl. Math. 200, pp. 178–192.

[21] A. Friedli (1978), *Verallgemeinerte Runge-Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme.* In: R. Bulirsch, R.D. Grigorieff, J. Schröder (Eds.), Numerical Treatment of Differential Equations, Lecture Notes in Mathematics 631, Springer, Berlin.

[22] M. Hochbruck, Ch. Lubich (1997), *On Krylov subspace approximations to the matrix exponential operator operator.* SIAM J. Numer. Anal. 34, pp. 1911–1925.

[23] M. Hochbruck, Ch. Lubich, H. Selhofer (1998), *Exponential integrators for large systems of differential equations.* SIAM J. Sci. Comput. 19, pp. 1552–1574.

[24] M. Hochbruck, Ch. Lubich (1999), *Exponential integrators for quantum-classical molecular dynamics.* BIT 39, pp. 620–645.

[25] M. Hochbruck, A. Ostermann, J. Schweitzer (2008), *Exponential Rosenbrock-type methods*, submitted.

[26] P.J. van der Houwen, J.G. Verwer (1974), *Generalized linear multistep methods I: Development of algorithms with non-zero parasitic roots.* Report NW 10/74, Mathematisch Centrum, Amsterdam, http://repository.cwi.nl:8888/cwi_repository/docs/I/09/9059A.pdf

[27] W. Hundsdorfer, J.G. Verwer (2003), *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations.* Springer Series in Computational Mathematics, Vol. 33, Springer, Berlin.

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

[29] J.D. Lawson (1967), *Generalized Runge-Kutta processes for stable systems with large Lipschitz constants.* SIAM J. Numer. Anal. 4, pp. 372-380.

[30] C. Leforestier, R. H. Bisseling, C. Cerjan, M. D. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H.-D. Meyer, N. Lipkin, O. Roncero, R. Kosloff (1991), *A comparison of different propagation schemes for the time dependent Schrödinger equation.* J. Comput. Phys. 94, pp. 59–80.

[31] J. Legras (1966), *Résolution numérique des grands systèmes différentiels linéaires.* Numer. Math. 8, pp. 14–28.

[32] J.A. Meijerink, H.A. van der Vorst (1977), *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix.* Math. Comput. 31, pp. 148–162.

[33] B. Minchev, W. Wright (2005), *A review of exponential integrators for first order semi-linear problems.* Techn. Report 2/05, Dept. of Mathematics, NTNU, Trondheim.

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

[35] T. Namiki (2000), *3D ADI-FDTD method–Unconditionally stable time-domain algorithm for solving full vector Maxwell's equations.* IEEE Trans. Microwave Theory Tech. 48, pp. 1743–1748.

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

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

[38] S.P. Nørsett (1969), *An A-stable modification of the Adams-Bashforth methods.* In: Lecture Notes in Math. 109, A. Dold, B. Echman (eds.), pp. 214–219, Springer.

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

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

[41] Y. Saad (1994), *ILUT: a dual threshold incomplete LU factorization.* Numer. Linear Algebra Appl. 1, pp. 387–402.

[42] Y. Saad (2000), *Iterative methods for sparse linear systems.* Available at http://www-users.cs.umn.edu/∼saad/books.html

[43] R.B. Sidje (1998), *Software Package for Computing Matrix Exponentials.* ACM - Transactions On Mathematical Software 24, pp. 130–156.

[44] A. van der Sluis, H.A. van der Vorst (1986), *The rate of convergence of conjugate gradients.* Numer. Math. 48, pp. 543–560.

[45] H. Tal-Ezer (1986), *Spectral methods in time for hyperbolic equations.* SIAM J. Numer. Anal. 23, pp. 11–26.

[46] J.G. Verwer (1977), *On generalized linear multistep methods with zero-parasitic roots and an adaptive principal root.* Numer. Math. 27, pp. 143–155.

[47] H.A. van der Vorst (1987), *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, pp. 249–263.

[48] H.A. van der Vorst (2003), *Iterative Krylov Methods for Large Linear Systems.* Cambridge University Press.

[49] K.S. Yee (1966), *Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media.* IEEE Trans. Antennas Propag. 14, pp. 302–307.

[50] F. Zheng, Z. Chen, J. Zhang (1999), *A finite-difference time-domain method without the Courant stability condition.* IEEE Microwave Guided Wave Lett. 9, pp. 441–443.

[51] F. Zheng, Z. Chen, J. Zhang (2000), *Toward the development of a three-dimensional unconditionally stabe finite-difference time-domain method.* IEEE Microwave Theory Tech. 48, pp. 1550–1558.