Computing 71, 265–292 (2003)
Digital Object Identifier (DOI) 10.1007/s00607-003-0026-2

Computing
Printed in Austria

# Non-Reflecting Boundary Conditions for Maxwell's Equations

**R. Hiptmair,** Zürich, and **A. Schädle**, Berlin

Received April 29, 2003; revised July 27, 2003
Published online: October 30, 2003

**Abstract**

A new discrete non-reflecting boundary condition for the time-dependent Maxwell equations describing the propagation of an electromagnetic wave in an infinite homogenous lossless rectangular waveguide with perfectly conducting walls is presented. It is derived from a virtual spatial finite difference discretization of the problem on the unbounded domain. Fourier transforms are used to decouple transversal modes. A judicious combination of edge based nodal values permits us to recover a simple structure in the Laplace domain. Using this, it is possible to approximate the convolution in time by a similar fast convolution algorithm as for the standard wave equation.

*AMS Subject Classification:* 78A50, 65N06, 65R99, 44A10, 44A35.

*Keywords:* Finite difference time domain methods, transparent boundary conditions, absorbing boundary conditions, fast convolution, waveguide.

## 1. Introduction

To compute the propagation of electromagnetic (micro-) waves in waveguides is a standard task in computational electromagnetism. Often the waveguides have a simple rectangular cross-section and are filled with non-conducting material of constant index of refraction. This is true, except for a small zone of inhomogeneity, corresponding, for instance, to a waveguide junction, a taper structure. As a model setting we consider two rectangular, homogeneous, lossless waveguides $\Omega_1$ and $\Omega_2$ with perfectly conducting walls that are linked by an inhomogeneous region $\Omega$ of potentially complicated geometry, see Figure 1.

The region $\Omega$ requires discretization by means of spatial finite element or finite volume schemes. Of course, this is not an option for the infinite waveguides. Their impact on the propagation of electromagnetic waves should be modeled by imposing non-reflecting boundary conditions that link the tangential components of electric and magnetic fields at the two ports $\bar{\Omega} \cap \bar{\Omega}_1$ and $\bar{\Omega} \cap \bar{\Omega}_2$.

Non-reflecting boundary conditions are a crucial numerical tool whenever the propagation of waves in unbounded domains has to be computed. There are basically three different approaches. First, one may use time-domain integral equations as in [1]. However, this is only an option when one wants to tackle the

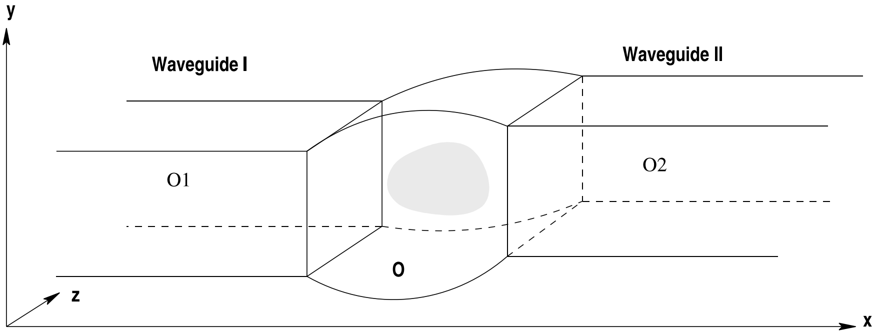

**Fig. 1.** Two connected rectangular waveguides

exterior of a bounded object. Second, absorbing layers have become very popular, starting with the pioneering work of Bérenger [2]. They are widely used in waveguide simulation. Third, one can opt for radiation boundary conditions based on Laplace transforms and temporal convolution, see the articles by Hagstrom [3, 4] for a survey.

The current paper describes an approach of the third category. It heavily relies on techniques introduced in [7, 10] for the wave equation. First the non-reflecting boundary condition is formulated using sine and cosine transforms on the boundary and the Laplace transform in time. The new challenge in the case of Maxwell's equations arises from the staggered location of the discrete unknowns, which makes it difficult to get a simple form of the Dirichlet-to-Neumann map for individual modes. We present a judicious recombination of discrete variables that renders the problem tractable. The boundary condition obtained is discrete non-reflecting, which means that no spurious reflections due to space discretization at the boundary enter the computational domain. Transforming back to time-domain gives a convolution in time that is evaluated using the fast convolution algorithm introduced in [7, 10] and described in Section 4.

## 2. Spatial Discretization

The time-domain electric wave equation in both waveguides, after suitable scaling, reads

$$-\frac{d^2\mathbf{u}}{dt^2} = \mathbf{curl}\,\mathbf{curl}\,\mathbf{u}. \tag{1}$$

It has to be supplemented by vanishing tangential components $\mathbf{u} \times \mathbf{n}$ at the outer walls of the waveguides.

We take a closer look at the waveguide occupying the region $\Omega_2 := ]0;\infty[\times]0;a[\times]0;b[,\ a,b > 0$. It will be equipped with an infinite regular tensor-product *virtual* grid $G_h$ of mesh-width $h > 0$. We assume $a = Jh$ and

$b = Kh$ for $J, K \in \mathbb{N}$. On this grid the electric wave equation is discretized by means of simple finite differences, known as the Yee scheme [11, 14], which can also be obtained from the finite integration technique [13]. The discrete field components can be viewed as being located on midpoints of edges. The resulting difference stencil for the double-**curl** operator and an edge in $x$-direction is depicted in Figure 2. The stencils for the double-**curl** operator and edges in $y$- and $z$-direction are obtained by rotation of the stencil shown in Figure 2. Basically, these stencils describe a discrete counterpart of the **curl curl**-operator.

To begin with we have to distinguish between edges pointing into different coordinate directions. Therefore we introduce the sub-grids

$$\begin{aligned}
G_h^x &:= \left(\left(i+\frac{1}{2}\right)h, jh, kh\right), \quad i \in \mathbb{N}_0,\ j \in \{1,\dots,J-1\},\ k \in \{1,\dots,K-1\},\\
G_h^y &:= \left(ih, \left(j+\frac{1}{2}\right)h, kh\right), \quad i \in \mathbb{N}_0,\ j \in \{0,\dots,J-1\},\ k \in \{1,\dots,K-1\},\\
G_h^z &:= \left(ih, jh, \left(k+\frac{1}{2}\right)h\right), \quad i \in \mathbb{N}_0,\ j \in \{1,\dots,J-1\},\ k \in \{0,\dots,K-1\},
\end{aligned}$$

where $G_h^d$, $d \in \{x, y, z\}$, contains the midpoints of edges in direction $d$. Let $F_h^d$ denote the space of real valued grid functions on grid $G_h^d$. Then the discrete electric fields in $\Omega_2$ can be described by grid functions in the space $\mathbf{F}_h := F_h^x \times F_h^y \times F_h^z$.

The discrete **curl curl**-operator in $\Omega_2$ subject to homogeneous Dirichlet boundary conditions represents a linear mapping $\mathscr{C}_h : \mathbf{F}_h \mapsto \mathbf{F}_h$. For a grid function $\mathbf{u}_h = (u_h^x, u_h^y, u_h^z) \in \mathbf{F}_h$ the components $v_h^x, v_h^y$ and $v_h^z$ of $\mathbf{v}_h := \mathscr{C}_h \mathbf{u}_h$ are given by

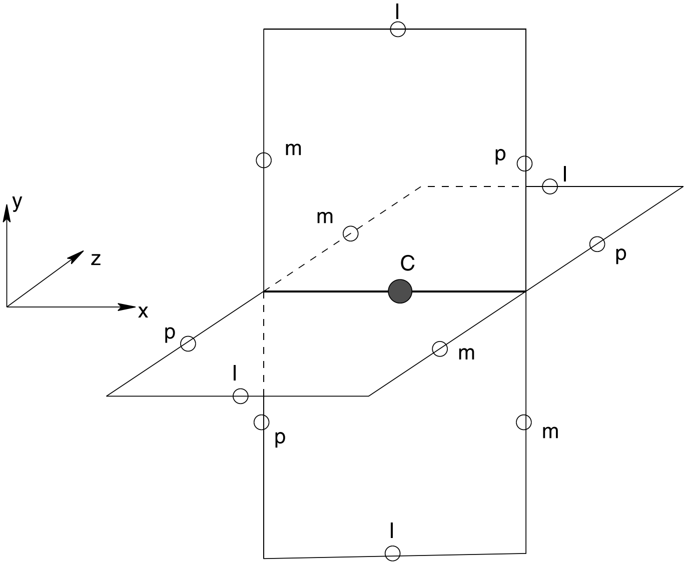

**Fig. 2.** Difference stencil for double-**curl** operator and edge in $x$-direction (scaled by $h^2$)

$$\begin{aligned} v_h^x(\mathbf{p}) &= \frac{1}{h^2}\Big((-\Delta_h^{(y,z)}u_h^x)(\mathbf{p}) + (R_h^{(y,x)}u_h^y)(\mathbf{p}) + (R_h^{(z,x)}u_h^z)(\mathbf{p})\Big), \quad \mathbf{p} \in G_h^x, \\ v_h^y(\mathbf{p}) &= \frac{1}{h^2}\Big((R_h^{(x,y)}u_h^x)(\mathbf{p}) + (-\Delta_h^{(x,z)}u_h^y)(\mathbf{p}) + (R_h^{(z,y)}u_h^z)(\mathbf{p})\Big), \quad \mathbf{p} \in G_h^y, \\ v_h^z(\mathbf{p}) &= \frac{1}{h^2}\Big((R_h^{(x,z)}u_h^x)(\mathbf{p}) + (R_h^{(y,z)}u_h^y)(\mathbf{p}) + (-\Delta_h^{(x,y)}u_h^z)(\mathbf{p})\Big), \quad \mathbf{p} \in G_h^z. \end{aligned} \tag{2}$$

Here, writing $\mathbf{e}_d$, $d \in \{x,y,z\}$, for the unit vector in direction $d$, the grid operators $\Delta_h^{(d,f)} : F_h^g \mapsto F_h^g$, $\{d,f,g\} = \{x,y,z\}$, are defined by

$$\begin{aligned} (-\Delta_h^{(d,f)}u_h^g)(\mathbf{p}) = 4u_h^g(\mathbf{p}) - u_h^g(\mathbf{p} - h\mathbf{e}_d) - u_h^g(\mathbf{p} + h\mathbf{e}_d) \\ - u_h^g(\mathbf{p} - h\mathbf{e}_f) - u_h^g(\mathbf{p} + h\mathbf{e}_f), \end{aligned} \tag{3}$$

for $\mathbf{p} \in G_h^g$. Similarly the grid operators $R_h^{(d,f)} : F_h^d \mapsto F_h^f$ are given by

$$\begin{aligned} (R_h^{(d,f)}u_h^g)(\mathbf{p}) = u_h^g\left(\mathbf{p} + \frac{h}{2}(\mathbf{e}_d + \mathbf{e}_f)\right) - u_h^g\left(\mathbf{p} + \frac{h}{2}(\mathbf{e}_d - \mathbf{e}_f)\right) \\ - u_h^g\left(\mathbf{p} + \frac{h}{2}(-\mathbf{e}_d + \mathbf{e}_f)\right) + u_h^g\left(\mathbf{p} + \frac{h}{2}(-\mathbf{e}_d - \mathbf{e}_f)\right) \end{aligned} \tag{4}$$

for $\mathbf{p} \in G_h^f$, $\{d,f,g\} = \{x,y,z\}$. Thus we have converted the stencils into formulas. Eventually the semi-discrete electric wave equation becomes

$$-\frac{d^2\mathbf{u}_h}{dt^2} = \mathscr{C}_h\mathbf{u}_h. \tag{5}$$

It has to be supplied with initial values at time $t = 0$. For the sake of simplicity we will always assume that in the beginning there are no fields in $\Omega_i$, $i = 1, 2$, that is $\mathbf{u}_h(0) = (\frac{d}{dt}\mathbf{u}_h)(0) = 0$. Then the Laplace transform with respect to $t$ gives

$$\mathscr{S}_h\hat{\mathbf{u}}_h := (\mathscr{C}_h + s^2)\hat{\mathbf{u}}_h = 0. \tag{6}$$

## 3. Derivation of Non-Reflecting Boundary Conditions

Next we carry out a modal decomposition of the tangential components of $\mathbf{u}_h$ in the plane of the port $\Gamma := \bar{\Omega} \cap \bar{\Omega}_2$, which lies in the $y$-$z$ coordinate plane. Let $\Gamma_h$ denote the grid restricted to $\Gamma$. Hence only nodes for $y$- and $z$-components are located on $\Gamma_h$, see Figure 3. More precisely they form the sub-grids

$$\begin{aligned} \Gamma_h^y &:= \left(0, \left(j + \frac{1}{2}\right)h, kh\right), \quad j \in \{0,\dots,J-1\},\ k \in \{1,\dots,K-1\}, \\ \Gamma_h^z &:= \left(0, jh, \left(k + \frac{1}{2}\right)h\right), \quad j \in \{1,\dots,J-1\},\ k \in \{0,\dots,K-1\}. \end{aligned}$$

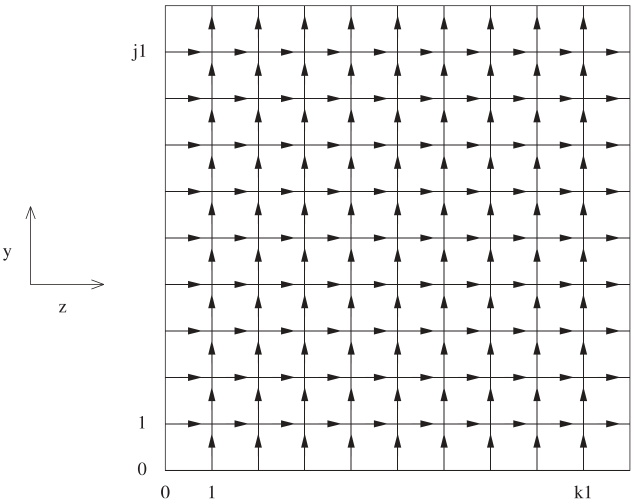

**Fig. 3.** Grid $\Gamma_h$

The associated spaces of grid functions are denoted by $P_h^y$ and $P_h^z$. They are restrictions of $F_h^y$ and $F_h^z$ to $\Gamma_h$, respectively. Write

$$Y^{(j,k)}(y,z) := \cos(\eta y) \cdot \sin(\zeta z),$$

$$\eta := j\pi/a, \quad j \in \{0,\dots,J-1\}, \qquad \zeta := k\pi/b, \quad k \in \{0,\dots,K-1\},$$

$$Z^{(j,k)}(y,z) := \sin(\eta y) \cdot \cos(\zeta z),$$

$$\eta := j\pi/a, \quad j \in \{0,\dots,J-1\}, \qquad \zeta := k\pi/b, \quad k \in \{0,\dots,K-1\},$$

and observe that

$$P_h^y = \operatorname{Span}\{\mathbf{p} \mapsto Y^{(j,k)}(\mathbf{p}), j \in \{0,\dots,J-1\}, k \in \{0,\dots,K-1\}, \mathbf{p} \in \Gamma_h^y\},$$

$$P_h^z = \operatorname{Span}\{\mathbf{p} \mapsto Z^{(j,k)}(\mathbf{p}), j \in \{0,\dots,J-1\}, k \in \{0,\dots,K-1\}, \mathbf{p} \in \Gamma_h^x\}.$$

For ease of notation we consider a zeroth sine coefficient, which is redundant. For a grid function $u_h^y \in P_h^y$ the cosine-sine transform coefficients $\hat{u}_h^y$ are given by the relation

$$u_h^y(0,y,z) = \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} \hat{u}_h^y(j,k) Y^{(j,k)}(y,z) \quad \text{for } (y,z) \in \Gamma_h^y. \tag{7}$$

This bijective cosine-sine transformation $u^y(0,\cdot,\cdot) \mapsto \hat{u}_h^y$ will be denoted by $\mathscr{Y}$. Similarly, for $u_h^z \in P_h^z$, we define the sine-cosine transform coefficients $\hat{u}_h^z$ by the relation

$$u_h^z(0,y,z) = \sum_{j=0}^{J-1}\sum_{k=0}^{K-1} \hat{u}_h^z(j,k) Z^{(j,k)}(y,z) \quad \text{for } (y,z) \in \Gamma_h^z. \tag{8}$$

The corresponding bijective mapping will be abbreviated by $\mathscr{Z}$.

Beside the grids $\Gamma_h^y$ and $\Gamma_h^z$ we need two layers of edges in $x$-direction adjacent to the port. They bear the grids

$$\begin{aligned}\Gamma_h^{-x} &:= (-h/2, jh, kh), \quad j \in \{1,\dots,J-1\},\ k \in \{1,\dots,K-1\},\\ \Gamma_h^{x} &:= (h/2, jh, kh), \quad j \in \{1,\dots,J-1\},\ k \in \{1,\dots,K-1\}.\end{aligned}$$

We denote the restriction of $F_h^x$ to $\Gamma_h^{\pm x}$ by $P_h^{\pm x}$ and set

$$X^{(j,k)}(y,z) := \sin(\eta y) \cdot \sin(\zeta z),$$
$$\eta = j\pi/a, \quad j \in \{0,\dots,J-1\}, \qquad \zeta = k\pi/b, \quad k \in \{0,\dots,K-1\}.$$

For $u_h^x \in P_h^x$ we define the sine-sine transform coefficients $\hat{u}_h^x$ by the relation

$$u_h^x(h/2,y,z) = \sum_{j=0}^{J-1}\sum_{k=0}^{K-1} \hat{u}_h^x(j,k) X^{(j,k)}(y,z) \quad \text{for } (y,z) \in \Gamma_h^x. \tag{9}$$

We adopt the notation $\mathscr{X}$ for this bijective transformation.

Now we study the action of the discrete differential operator $\mathscr{C}_h$ on a function $\mathbf{u}_h = (u_h^x, u_h^y, u_h^z) \in \mathbf{F}_h$ that is of the special form

$$\begin{aligned} u_h^x(x,y,z) &= X^{(j,k)}(y,z) \cdot w_h^x(x), \quad (x,y,z) \in G_h^x,\\ u_h^y(x,y,z) &= Y^{(j,k)}(y,z) \cdot w_h^y(x), \quad (x,y,z) \in G_h^y,\\ u_h^z(x,y,z) &= Z^{(j,k)}(y,z) \cdot w_h^z(x), \quad (x,y,z) \in G_h^z, \end{aligned} \tag{10}$$

where $w_h^y$, $w_h^z$: $\{ih\}_{i\in\mathbb{N}_0} \to \mathbb{R}$ and $w_h^x$: $\{(i+\frac{1}{2})h\}_{i\in\mathbb{N}_0} \to \mathbb{R}$ are grid functions on a one-dimensional equidistant grid, see Figure 4. In the sequel the spatial frequencies $\eta$ and $\zeta$ are fixed, since we focus on individual modes.

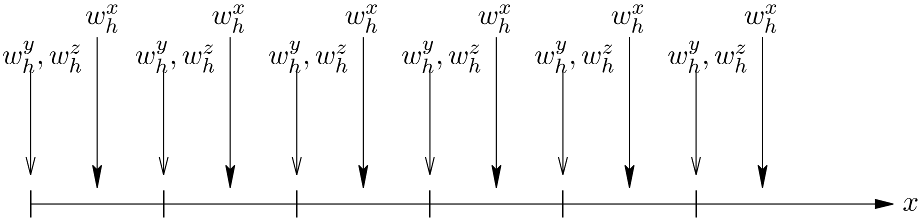

**Fig. 4.** Location of nodes for $w_h^x$, $w_h^y$, and $w_h^z$ on the $x$-axis

Using elementary manipulations we calculate for a single mode

$$\begin{aligned}
R_h^{(y,x)}u_h^y(\mathbf{p}) =& R_h^{(y,x)}(Y^{j,k}(y,z)w_h^y(x))\\
=& \cos(\eta(y+h/2))\sin(\zeta z)w_h^y(x+h/2)-\cos(\eta(y+h/2))\sin(\zeta z)w_h^y(x-h/2)\\
&-\cos(\eta(y-h/2))\sin(\zeta z)w_h^y(x+h/2)+\cos(\eta(y-h/2))\sin(\zeta z)w_h^y(x-h/2)\\
=&(\cos(\eta y)\cos(\eta h/2)-\sin(\eta y)\sin(\eta h/2))\sin(\zeta z)w_h^y(x+h/2)\\
&-(\cos(\eta y)\cos(\eta h/2)+\sin(\eta y)\sin(\eta h/2))\sin(\zeta z)w_h^y(x+h/2)\\
&+(\cos(\eta y)\cos(\eta h/2)+\sin(\eta y)\sin(\eta h/2))\sin(\zeta z)w_h^y(x-h/2)\\
&-(\cos(\eta y)\cos(\eta h/2)-\sin(\eta y)\sin(\eta h/2))\sin(\zeta z)w_h^y(x-h/2)\\
=&-2\sin(\eta h/2)X^{j,k}(y,z)(w_h^y(x+h/2)-w_h^y(x-h/2))
\end{aligned}$$

for $\mathbf{p} \in G_h^x$ and

$$\begin{aligned}
-\Delta_h^{(y,z)}u_h^x(\mathbf{p}) &= -\Delta_h^{(y,z)}(X^{j,k}(y,z)w_h^x(x))\\
&= (4\sin(\eta y)\sin(\zeta z)w_h^x(x)) - (\sin(\eta(y-h))\sin(\zeta z)w_h^x(x))\\
&\quad - (\sin(\eta y)\sin(\zeta(z+h))w_h^x(x)) - (\sin(\eta(y+h))\sin(\zeta z)w_h^x(x))\\
&\quad - (\sin(\eta y)\sin(\zeta(z-h))w_h^x(x))\\
&= (4\sin(\eta y)\sin(\zeta z)w_h^x(x))\\
&\quad - ((\sin(\eta y)\cos(\eta h) - \cos(\eta y)\sin(\eta h))\sin(\zeta z)w_h^x(x)\\
&\quad - ((\sin(\eta y)\cos(\eta h) + \cos(\eta y)\sin(\eta h))\sin(\zeta z)w_h^x(x)\\
&\quad - ((\sin(\zeta z)\cos(\zeta h) - \cos(\zeta z)\sin(\zeta h))\sin(\eta y)w_h^x(x)\\
&\quad - ((\sin(\zeta z)\cos(\zeta h) + \cos(\zeta z)\sin(\zeta h))\sin(\eta y)w_h^x(x)\\
&= ((2\sin(\eta h/2))^2 + (2\sin(\zeta h/2))^2)X^{j,k}(y,z)w_h^x(x)
\end{aligned}$$

for $\mathbf{p} \in G_h^x$. For all the other terms in Equation (2), due to the symmetry of the stencil, we get, using the abbreviations

$$r_\eta := \frac{2}{h}\sin\left(\frac{1}{2}h\eta\right), \quad r_\zeta := \frac{2}{h}\sin\left(\frac{1}{2}h\zeta\right),$$

$$\begin{aligned}
R_h^{(z,x)}u_h^z(\mathbf{p}) &= X^{j,k}(y,z)hr_\zeta(w_h^z(x-h/2) - w_h^z(x+h/2)), \quad \mathbf{p} \in G_h^x,\\
R_h^{(x,y)}u_h^x(\mathbf{p}) &= Y^{j,k}(y,z)hr_\eta(w_h^x(x+h/2) - w_h^x(x-h/2)), \quad \mathbf{p} \in G_h^y,\\
R_h^{(z,y)}u_h^z(\mathbf{p}) &= -Y^{j,k}(y,z)h^2r_\zeta r_\eta w_h^z(x), \quad \mathbf{p} \in G_h^y,\\
R_h^{(x,z)}u_h^x(\mathbf{p}) &= Z^{j,k}(y,z)hr_\zeta(w_h^x(x+h/2) - w_h^x(x-h/2)), \quad \mathbf{p} \in G_h^z,\\
R_h^{(y,z)}u_h^y(\mathbf{p}) &= -Z^{j,k}(y,z)h^2r_\zeta r_\eta w_h^y(x), \quad \mathbf{p} \in G_h^z,\\
-\Delta_h^{(x,z)}u_h^y(\mathbf{p}) &= Y^{j,k}(y,z)((2+(hr_\zeta)^2)w_h^y(x) - w_h^y(x+h) - w_h^y(x-h)), \quad \mathbf{p} \in G_h^y,\\
-\Delta_h^{(x,y)}u_h^z(\mathbf{p}) &= Z^{j,k}(y,z)((2+(hr_\eta)^2)w_h^z(x) - w_h^z(x+h) - w_h^z(x-h)), \quad \mathbf{p} \in G_h^z.
\end{aligned}$$

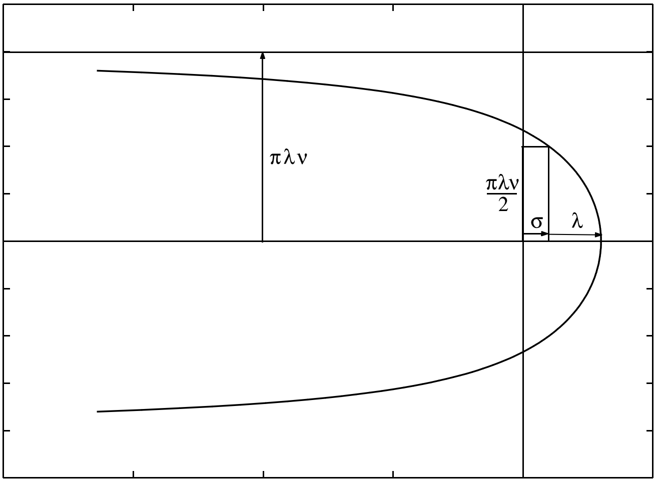

**Fig. 5.** Talbot contour

This yields

$$\mathscr{C}\begin{pmatrix} X^{(j,k)}w_h^x \\ Y^{(j,k)}w_h^y \\ Z^{(j,k)}w_h^z \end{pmatrix} = \begin{pmatrix} X^{(j,k)}q_h^x \\ Y^{(j,k)}q_h^y \\ Z^{(j,k)}q_h^z \end{pmatrix},$$

where the grid functions $q_h^x$, $q_h^y$ and $q_h^z$ are given by

$$\begin{aligned} q_h^x &= (r_\eta^2 + r_\zeta^2)w_h^x - r_\eta \delta_{h/2}^x w_h^y - r_\zeta \delta_{h/2}^x w_h^z, \\ q_h^y &= r_\eta \delta_{h/2}^x w_h^x + (-\Delta_h^x + r_\zeta^2)w_h^y - r_\eta r_\zeta w_h^z, \\ q_h^z &= r_\zeta \delta_{h/2}^x w_h^x - r_\zeta r_\eta w_h^y + (-\Delta_h^x + r_\eta^2)w_h^z. \end{aligned}$$

The difference operators are defined as

$$(\delta_{h/2}^x \tilde{f}_h)(x) := \frac{\tilde{f}_h(x + \frac{1}{2}h) - \tilde{f}_h(x - \frac{1}{2}h)}{h}, \quad x = \left(i + \frac{1}{2}\right)h,\ i \in \mathbb{N}_0, \tag{11}$$

$$(-\Delta_h^x \tilde{f}_h)(x) := \frac{2\tilde{f}_h(x) - \tilde{f}_h(x-h) - \tilde{f}_h(x+h)}{h^2}, \quad x = ih,\ i \in \mathbb{N}. \tag{12}$$

The crucial insight is that each pair of spatial frequencies $(\eta, \zeta)$ defines an invariant subspace for $\mathscr{C}_h$ containing functions of the form (10). Hence modal decomposition reduces the action of $\mathscr{S}$ to coupled one-dimensional difference equations. Equation (6), when considered in the invariant subspaces, reads

$$
\begin{aligned}
&(r_\eta^2 + r_\zeta^2 + s^2)w_h^x - r_\eta \delta_{h/2}^x w_h^y - r_\zeta \delta_{h/2}^x w_h^z = 0,\\
&r_\eta \delta_{h/2}^x w_h^x + (-\Delta_h^x + r_\zeta^2 + s^2)w_h^y - r_\eta r_\zeta w_h^z = 0, \qquad (13)\\
&r_\zeta \delta_{h/2}^x w_h^x - r_\zeta r_\eta w_h^y + (-\Delta_h^x + r_\eta^2 + s^2)w_h^z = 0.
\end{aligned}
$$

What bars us from a straightforward application of the ideas that have been successful in the case of the wave equation, cf. [7, 10], is the staggered location of the $w_h^x$ nodal values. A remedy is to introduce the difference of these nodal values as auxiliary grid function

$$
p_h^x(x) = w_h^x\left(x - \frac{1}{2}h\right) - w_h^x\left(x + \frac{1}{2}h\right), \qquad (14)
$$

which results in

$$
\begin{aligned}
&(r_\eta^2 + r_\zeta^2 + s^2)p_h^x + hr_\eta \Delta_h^x w_h^y + hr_\zeta \Delta_h^x w_h^z = 0,\\
&-r_\eta \frac{1}{h} p_h^x + (-\Delta_h^x + r_\zeta^2 + s^2)w_h^y - r_\eta r_\zeta w_h^z = 0, \qquad (15)\\
&-r_\zeta \frac{1}{h} p_h^x - r_\zeta r_\eta w_h^y + (-\Delta_h^x + r_\eta^2 + s^2)w_h^z = 0.
\end{aligned}
$$

Now we apply the *zeta*-transform to (15). Defining the formal series

$$
X(\xi) := \sum_{n=0}^{\infty} p_h^x(nh)\xi^n\ , \quad Y(\xi) := \sum_{n=0}^{\infty} w_h^y(nh)\xi^n \quad \text{and} \quad Z(\xi) := \sum_{n=0}^{\infty} w_h^z(nh)\xi^n,
$$

we can rewrite Equation (15):

$$
\begin{aligned}
&(r_\eta^2 + r_\zeta^2 + s^2)X(\xi) - \frac{r_\eta}{h}(\xi^{-1} - 2 + \xi)Y(\xi) - \frac{r_\zeta}{h}(\xi^{-1} - 2 + \xi)Z(\xi)\\
&\quad = -\frac{r_\eta}{h}(\xi^{-1}w_h^y(0) - w_h^y(-h)) - \frac{r_\zeta}{h}(\xi^{-1}w_h^z(0) - w_h^z(-h))\\
&r_\eta \frac{1}{h}X(\xi) - \left(\frac{1}{h^2}\xi^{-1} - \left(\frac{2}{h^2} + r_\zeta^2 + s^2\right) + \frac{1}{h^2}\xi\right)Y(\xi) - r_\eta r_\zeta Z(\xi)\\
&\quad = \frac{1}{h^2}(w_h^y(-h) - \xi^{-1}w_h^y(0))\\
&r_\zeta \frac{1}{h}X(\xi) - r_\eta r_\zeta Y(\xi) - \left(\frac{1}{h^2}\xi^{-1} - \left(\frac{2}{h^2} + r_\eta^2 + s^2\right) + \frac{1}{h^2}\xi\right)Z(\xi)\\
&\quad = \frac{1}{h^2}(w_h^z(-h) - \xi^{-1}w_h^z(0))
\end{aligned}
$$

Solving for $X(\xi)$, $Y(\xi)$ and $Z(\xi)$ we get

$$\begin{aligned} X(\xi) &= h\frac{r_\eta(w_h^y(-h)\xi - w_h^y(0)) + r_\zeta(w_h^z(-h)\xi - w_h^z(0))}{-1 + ((r_\eta^2 + r_\zeta^2 + s^2)h^2 + 2)\xi - \xi^2}, \\ Y(\xi) &= \frac{-w_h^y(0) + w_h^y(-h)\xi}{-1 + ((r_\eta^2 + r_\zeta^2 + s^2)h^2 + 2)\xi - \xi^2}, \\ Z(\xi) &= \frac{-w_h^z(0) + w_h^z(-h)\xi}{-1 + ((r_\eta^2 + r_\zeta^2 + s^2)h^2 + 2)\xi - \xi^2}. \end{aligned} \tag{16}$$

To begin with, since $p_h^x(0)$ does not occur we obtain the relationship

$$p_h^x(0) = w_h^x(-h/2) - w_h^x(h/2) = -h(r_\eta w_h^y(0) + r_\zeta w_h^z(0)). \tag{17}$$

The second and third equation of (16) are of the form

$$\sum_{n=0}^{\infty} \alpha_n \xi^n = \frac{1}{1 - ((r_\eta^2 + r_\zeta^2 + s^2)h^2 + 2)\xi + \xi^2} q(\xi),$$

where $q(\xi) = q_0 + q_1\xi$ is a polynomial of degree 1 in $\xi$, with coefficients depending on boundary values $w_h^z(0)$ and $w_h^y(0)$ and on auxiliary values $w_h^z(-h)$ and $w_h^y(-h)$. A fractional decomposition gives

$$\begin{aligned} \sum_{n=0}^{\infty} \alpha_n \xi^n &= c\left(\frac{1}{\xi_0^{-1} - \xi} - \frac{1}{\xi_0 - \xi}\right)(q_0 + q_1\xi) \\ &= c\left(\sum_{n=0}^{\infty} \xi_0^n \xi^n - \frac{1}{\xi_0^n}\xi^n\right)(q_0 + q_1\xi) \\ &= c\left(\sum_{n=0}^{\infty} q_0\left(\xi_0^n - \frac{1}{\xi_0^n}\right)\xi^n + \sum_{n=0}^{\infty} q_1\left(\xi_0^n - \frac{1}{\xi_0^n}\right)\xi^{n+1}\right) \\ &= c\left(\sum_{n=0}^{\infty}\left(\left(q_0 + q_1\frac{1}{\xi_0}\right)\xi_0^n - (q_0 + q_1\xi_0)\frac{1}{\xi_0^n}\right)\xi^n - q_1(\xi_0 - \xi_0^{-1})\right), \end{aligned}$$

where $c := (\xi_0 - \xi_0^{-1})^{-1}$ and $\xi_0$ is the root of $1 - ((r_\eta^2 + r_\zeta^2 + s^2)h^2 + 2)\xi + \xi^2$, such that $|\xi_0| < 1$ for $\Re(s) > 0$. Hence the sequence $\alpha_n$ will only be bounded for $n \to \infty$, if

$$q_0 + q_1\xi_0 = 0.$$

Since the sequences $w_h^y(nh)$ and $w_h^z(nh)$ have to be uniformly bounded, this condition gives

$$w_h^y(0) - w_h^y(-h)\xi_0 = 0, \quad w_h^z(0) - w_h^z(-h)\xi_0 = 0,$$

and equivalently

$$w_h^y(0) = \frac{w_h^y(-h) - w_h^y(0)}{\xi_0^{-1} - 1}, \tag{18}$$

$$w_h^z(0) = \frac{w_h^z(-h) - w_h^z(0)}{\xi_0^{-1} - 1}, \tag{19}$$

These yield non-reflecting boundary conditions for the time-dependent Maxwell equation in a waveguide. Equation (17) is redundant, as we will see in Section 5.

Equations (18), (19) and (17) are spatially discrete non-reflecting boundary conditions in the Laplace domain expressed with respect to the transversal modes, because they provide equations for the values $w_h^y(0)$, $w_h^z(0)$, and $w_h^x(h/2)$ as functions of nodal values inside $\Omega$. It is important to note that one more layer of the regular grid is required on the $\Omega$-side of the port, cf. Section 5.

Transforming back to time and grid domain we get the following relation between boundary values and differences of boundary and auxiliary values for $\mathbf{u}_h$

$$\begin{aligned} u_h^x(h/2,y,z,t) &= u_h^x(-h/2,y,z,t) \\ &\quad + \mathscr{X}^{-1}(h(r_\eta \mathscr{Y}(u_h^y(0,\cdot,\cdot,t)(j,k)) + r_\zeta \mathscr{Z}(u_h^z(0,\cdot,\cdot,t)(j,k))))(y,z), \\ u_h^y(0,y,z,t) &= \mathscr{Y}^{-1}\left(\int_0^t f_{j,k}(t-\tau)\mathscr{Y}(u_h^y(-h,\cdot,\cdot,\tau) - u_h^y(0,\cdot,\cdot,\tau))(j,k)\ d\tau\right)(y,z), \\ u_h^z(0,y,z,t) &= \mathscr{Z}^{-1}\left(\int_0^t f_{j,k}(t-\tau)\mathscr{Z}(u_h^z(-h,\cdot,\cdot,\tau) - u_h^z(0,\cdot,\cdot,\tau))(j,k)\ d\tau\right)(y,z), \end{aligned} \tag{20}$$

where $\mathscr{X}$, $\mathscr{Y}$ and $\mathscr{Z}$ denote sine-sine, cosine-sine, and sine-cosine transforms, respectively.

The Laplace transform $F_{j,k}(s)$ of $f_{j,k}(t)$ is given by

$$\begin{aligned} F_{j,k}(s) &:= \frac{1}{\xi_0^{-1} - 1} \\ &= \frac{2}{h^2\left((r_\eta^2 + r_\zeta^2 + s^2) + \left(r_\eta^2 + r_\zeta^2 + s^2\right)^{1/2}\left(r_\eta^2 + r_\zeta^2 + s^2 + \frac{4}{h^2}\right)^{1/2}\right)}. \end{aligned} \tag{21}$$

Recall that

$$r_\eta = \frac{2}{h}\sin\left(\frac{1}{2}hj\frac{\pi}{a}\right), \quad r_\zeta = \frac{2}{h}\sin\left(\frac{1}{2}hk\frac{\pi}{b}\right)$$

were defined previously.

Thus, we are left with evaluating the temporal convolution for each transform coefficient. Its efficient evaluation will be the focus of the next section.

## 4. Convolution Algorithm

In this section we describe the algorithm for computing temporal convolutions. This algorithm was first presented in [7], where it was applied to non-reflecting boundary conditions for Schrödinger and wave equations. Here we give a brief description of the algorithm and provide the outline of an implementation in MATLAB [8].

From (21) it is clear that the function $s \mapsto F_{\eta,\zeta}(s)$ has singularities in

$$\pm\alpha\,, \quad \alpha := i\sqrt{r_\eta^2 + r_\zeta^2} \quad \text{and in} \quad \pm\gamma\,, \quad \gamma := i\sqrt{r_\eta^2 + r_\zeta^2 + 4/h^2}.$$

For convenience we drop the subscripts of $F_{j,k}$ and $f_{j,k}$ for the remainder of this section.

Consider the convolution

$$\int_0^t f(t-\tau)g(\tau)\,dt$$

which is to be computed on the grid $t = 0, \Delta t, 2\Delta t, \ldots, T = N_t\Delta t$ with step size $\Delta t$. Here $f$ and $g$ play different roles. As shown above we are interested in situations where the evaluation of $g(\tau)$ at $\tau = n\Delta t$ requires knowledge of the values of the convolution up to $(n-1)\Delta t$, so that the required values of $g(\tau)$ cannot be computed in advance. It is the Laplace transform $F(s)$ of the convolution kernel $f(t)$, rather than the kernel itself, which is known *a priori* and can be evaluated easily. Therefore the algorithm should use evaluations only of $F(s)$.

The algorithm presented below will only require $\mathcal{O}(N_t \log N_t)$ operations and $\mathcal{O}(\log N_t)$ memory. It approximates the kernel $f(t)$ by sums of exponentials *locally* on a sequence of fast-growing intervals $I_\ell$ covering $[\Delta t, T]$:

$$I_\ell = [B^{\ell-1}\Delta t, (2B^\ell - 1)\Delta t], \tag{22}$$

where $B > 1$ is an integer and $\ell = 1, \ldots, K$ such that $(2B^K - 1) \geq N_t$. The approximation of $f(t)$ on $I_\ell$ results from applying the trapezoidal rule to a parameterization of the contour integral for the inverse Laplace transform,

$$f(t) = \frac{1}{2\pi i}\int_{\Gamma_\ell} F(\lambda)\, e^{t\lambda}\, d\lambda \approx \sum_{j=-N}^{N} w_j^{(\ell)} F(\lambda_j^{(\ell)})\, e^{t\lambda_j^{(\ell)}}, \quad t \in I_\ell, \tag{23}$$

with a suitably chosen complex contour $\Gamma_\ell$ to be described in detail below. The number of quadrature points on $\Gamma_\ell$ is chosen independent of $\ell$, but may depend on the kernel. It is much smaller than what would be required for a uniform approximation of the contour integral on $[0, T]$.

The numerical integration in (23) is done by applying the trapezoidal rule with equidistant steps to a parameterization of a *Talbot contour* [9, 12], which is given by

$$(-\pi, \pi) \to \Gamma \quad \theta \mapsto \sigma + \mu(\theta \cot(\theta) + i\nu\theta) \tag{24}$$

where the parameters $\mu$, $\nu$ and $\sigma$ are such that the singularities of $F(s)$, $\pm\alpha$ and $\pm\gamma$, lie to the left of the contour. We will use up to four shifted Talbot contours to enclose all the singularities of $F$, see Figures 6 to 9.

We set $\sigma_0 = 0$, $\mu_0 = 8$, $\mu = \mu_0/((2B^{\ell} - 1)\Delta t)$, $\nu_0 = 0.6$ and $\beta = \pi\mu\nu_0/2$, i.e. $\beta$ is the imaginary part of the intersection of the Talbot contour given by $\nu_0$ and $\mu$ with $\sigma = 0$. The parameter $\mu$ depends only on $\ell$, whereas the parameters $\nu$ and $\sigma$ depend on the singularities of the kernel, $\pm\alpha$ and $\pm\gamma$. The parameters $\mu_0$ and $\nu_0$ were obtained by minimizing the error in approximation (23). For a more detailed discussion on the error for general $F$ see the references [9, 12] and for a discussion on the error for the Laplace transform $F$ of the convolution kernel under consideration here, which is the same as for the wave equation, we refer to [7, 10].

We choose $\Gamma$ to be one contour $\Gamma_0$ enclosing all four singularities if $\alpha < \beta$ and $\beta < (\gamma - \alpha)/2$, setting $\nu = \nu_0(1 + \alpha/\beta)$ and $\sigma = \sigma_0$.

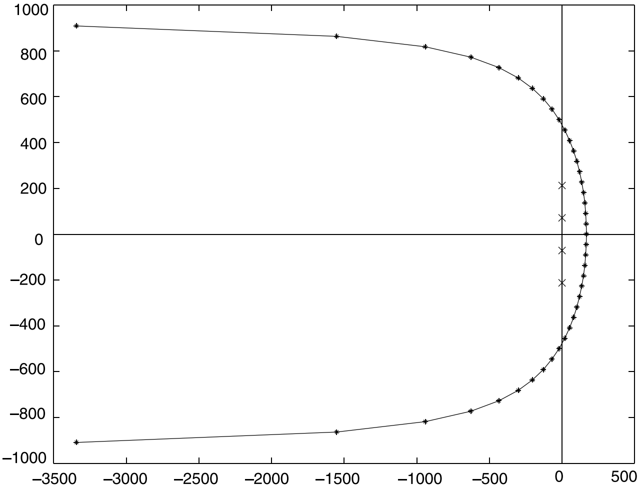

**Fig. 6.** One contour enclosing all singularities

We choose two contours $\Gamma_0$ and $\widehat{\Gamma}_0$ if $\alpha \geq \beta$ and $\beta > (\gamma - \alpha)/2$ where $\Gamma_0$ is given by $\nu = \nu_0(1 + (\gamma - \alpha)/(2\beta))$ and $\sigma = (\gamma + \alpha)/2$ and $\widehat{\Gamma}_0$ is given by $\nu = \nu_0(1 + (\gamma - \alpha)/(2\beta))$ and $\sigma = -(\gamma + \alpha)/2$.

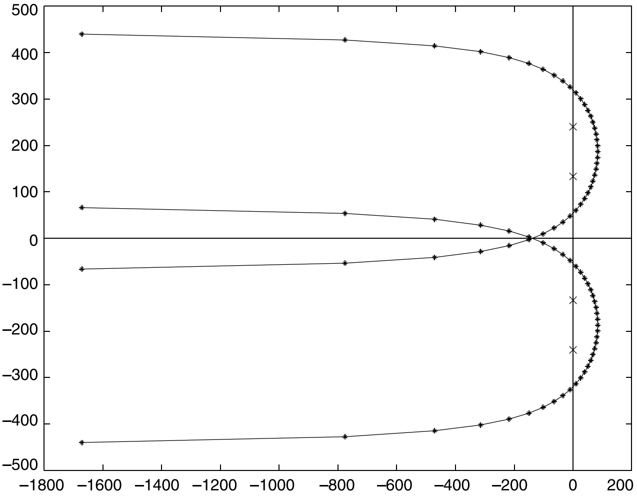

**Fig. 7.** Two contours enclosing all singularities

This choice of $\Gamma_\ell$ can be written as the pseudo code given in Figure 10. There we use the abbreviation $\lambda(\mu, v, \sigma, N)$ for a vector of length $2N-1$ whose entries are points on a Talbot contour $\Gamma$ given by the parameters $\mu$, $v$ and $\sigma$ and $\mathtt{w}(\mu, v, N)$ for a vector consisting of the quadrature weights $w_i$ corresponding to $\Gamma$, given in Equation (23). `exp` denotes the exponential function and $.*$ the element by element product of two vectors.

We choose $\Gamma$ to be three contours $\Gamma_0$ enclosing $\pm\alpha$, $\Gamma_1$ enclosing $\gamma$ and $\widehat{\Gamma}_1$ enclosing $-\gamma$ if $\alpha < \beta$ and $\beta \geq (\gamma - \alpha)/2$ where $\Gamma_0$ is given by $v = v_0(1 + \gamma/\beta)$ and $\sigma = \sigma_0$ and $\Gamma_1$ is given by $v = v_0$ and $\sigma = \gamma$ and $\widehat{\Gamma}_1$ is given by $v = v_0$ and $\sigma = -\gamma$

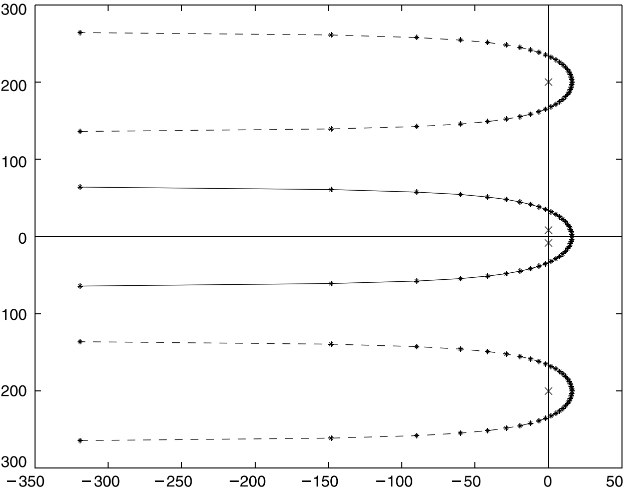

**Fig. 8.** Three contours enclosing all singularities

We choose four contours $\Gamma_0$, $\widehat{\Gamma}_0$, $\Gamma_1$ and $\widehat{\Gamma}_1$ if $\alpha \geq \beta$ and $\beta \geq (\gamma - \alpha)/2$ where $\Gamma_0$ is given by $v = v_0$ and $\sigma = \alpha$ and $\widehat{\Gamma}_0$ is given by $v = v_0$ and $\sigma = -\alpha$ and $\Gamma_1$ is given by $v = v_0$ and $\sigma = \gamma$ and $\widehat{\Gamma}_1$ is given by $v = v_0$ and $\sigma = -\gamma$.

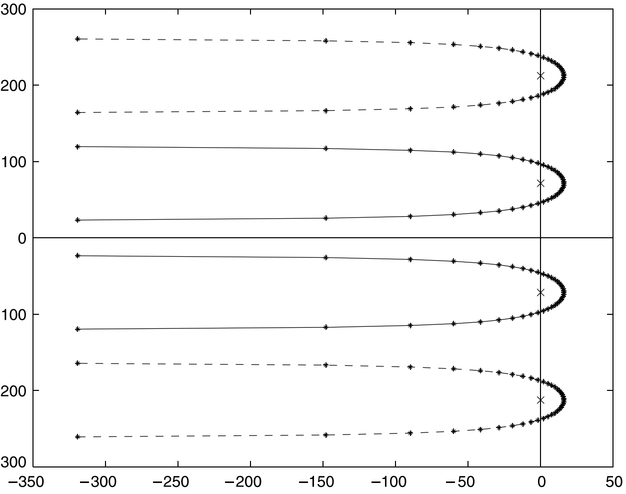

**Fig. 9.** Four contours enclosing all singularities

For general boundary points $a < b$ in the integral we have

$$\int_a^b f(t-\tau)g(\tau)d\tau = \int_a^b \frac{1}{2\pi i}\int_\Gamma F(\lambda)e^{(t-\tau)\lambda}\,d\lambda g(\tau)d\tau$$
$$= \frac{1}{2\pi i}\int_\Gamma F(\lambda)e^{(t-b)\lambda}\underbrace{\int_a^b e^{(b-\tau)\lambda}g(\tau)d\tau}_{(*)}\,d\lambda$$

where the inner integral $(*)$, henceforth denoted by $y(b,a,\lambda)$, is recognized as the solution at time $b$ of the scalar linear initial value problem

$$y' = \lambda y + g, \qquad y(a) = 0. \tag{25}$$

If $[t-b, t-a] \subset I_\ell$, then the contour integral over the Talbot contour $\Gamma = \Gamma_\ell$ is replaced with its trapezoidal rule approximation (23), which gives (omitting the superscripts $\ell$ for notational simplicity)

$$\int_a^b f(t-\tau)g(\tau)d\tau \approx \int_a^b \sum_{j=-N}^{N} w_j F(\lambda_j)e^{(t-\tau)\lambda_j}g(\tau)d\tau$$
$$= \sum_{j=-N}^{N} w_j F(\lambda_j)e^{(t-b)\lambda_j}y(b,a,\lambda_j). \tag{26}$$

```
$(\Lambda, E\Lambda F\Lambda W) = \texttt{talbotcontour}(c, \Delta t, dx, K, \alpha, \gamma, B, N, \mu_0, \nu_0)$

for $\ell = 1, \ldots, K$
$\mu = \mu_0/(\Delta t * (2 * B^{\ell} - 1)$ ; $\kappa = \pi * \mu * \nu_0/2$
  if $\alpha < \kappa$
    if $\kappa < (\gamma - \alpha)/2$
      $\Lambda_\ell = \lambda(\mu, \nu_0 * (1 + \alpha/\kappa), 0, N)$
      $W = \texttt{w}(\mu, \nu_0 * (1 + \alpha/\kappa), N)$
      $E\Lambda F\Lambda W_\ell = \texttt{exp}(\Lambda_\ell * dt * B^{\ell-1})$ .* $\texttt{F}(\Lambda_\ell, c, \alpha, \gamma, dx)$ ...
          .* $W * 1/(2 * i * N)$
    else
      $\Lambda_\ell = [\,\lambda(\mu, \nu_0 * (1 + \gamma/\kappa), 0, N)\,,\ \lambda(\mu, \nu_0, \gamma, N)\,,\ \lambda(\mu, \nu_0, -\gamma, N)\,]$
      $W = [\,\texttt{w}(\mu, \nu_0 * (1 + \gamma/\kappa), N)\,,\ \texttt{w}(\mu, \nu_0, N)\,,\ \texttt{w}(\mu, \nu_0, N)\,]$
      $E\Lambda F\Lambda W_\ell = \texttt{exp}(\Lambda_\ell * dt * B^{\ell-1})$ .* $\texttt{F}(\Lambda_\ell, d, \alpha, \gamma, dx)$ .*  ...
        $W * 1/(2 * i * N)$
    end
  else
    if $\kappa > (\gamma - \alpha)/2$
      $\Lambda_\ell = [\,\lambda(\mu, \nu_0 * (1 + (\gamma - \alpha)/2), (\gamma + \alpha)/2, N)$ , ...
        $\lambda(\mu, \nu_0 * (1 + (\gamma - \alpha)/2), -(\gamma + \alpha)/2, N)\,]$
      $W = [\,\texttt{w}(\mu, \nu_0 * (1 + (\gamma - \alpha)/2), N)\,,\ \lambda(\mu, nu_0 * (1 + (\gamma - \alpha)/2), N)\,]$
      $E\Lambda F\Lambda W_\ell = \texttt{exp}(\Lambda_\ell * dt * B^{\ell-1})$ .* $\texttt{F}(\Lambda_\ell, c, \alpha, \gamma, dx)$ .*  ...
        $W * 1/(2 * i * N)$
    else
      $\Lambda_\ell = [\,\lambda(\mu, \nu_0, \alpha, N)\,,\ \lambda(\mu, \nu_0, -\alpha, N)\,,\ \lambda(\mu, \nu_0, \gamma, N)$ ,...
        $\lambda(\mu, \nu_0, -\gamma, N)\,]$
      $W = [\,\texttt{w}(\mu, \nu_0, N)\,,\ \texttt{w}(\mu, \nu_0, N)\,,\ \texttt{w}(\mu, \nu_0, N)\,,\ \texttt{w}(\mu, \nu_0, N)\,]$
      $E\Lambda F\Lambda W_\ell = \texttt{exp}(\Lambda_l * dt * B^{\ell-1})$ .* $\texttt{F}(\Lambda_\ell, c, \alpha, \gamma, dx)$ ...
          .* $W * 1/(2 * i * N)$
    end
  end
end
```

**Fig. 10.** Algorithm for the choice of Talbot contours

The $2N + 1$ differential equations (25) with $\lambda = \lambda_j$ are solved approximately by replacing the function $g$ with its piecewise linear approximation and then solving exactly. Setting $g_n = g(a + n\Delta t)$, we get approximations $y_n \approx y(a + n\Delta t)$ recursively via

$$\begin{aligned} y_{n+1} &= e^{\Delta t\lambda} y_n + \Delta t \int_0^1 e^{(1-\theta)\Delta t\lambda}(\theta g_{n+1} + (1-\theta) g_n) d\theta \\ &= y_n + \frac{e^{\Delta t\lambda} - 1}{\Delta t\lambda}\left(\Delta t\lambda y_n + \Delta t g_n + \Delta t \frac{g_{n+1} - g_n}{\Delta t\lambda}\right) - \Delta t \frac{g_{n+1} - g_n}{\Delta t\lambda} \\ &= y_n + \Phi(\Delta t\lambda)\left(\Delta t\lambda y_n + \Delta t g_n + \Delta t \frac{g_{n+1} - g_n}{\Delta t\lambda}\right) - \frac{g_{n+1} - g_n}{\lambda}, \end{aligned} \tag{27}$$

where $\Phi(s) = (e^s - 1)/s$. The values of the vector $\varphi$, $\Phi$ evaluated in $\Lambda$, used by the function `expint`, Figure 11, are calculated once in the beginning, see the `initialize` function given in Figure 14. There `phis` denotes the function $s \mapsto \Phi(s)$ implemented in the package EXP4 described in [5].

To estimate the error, note that in total we approximate

$$\int_a^b f(t-\tau) g(\tau) d\tau \approx \int_a^b f(t-\tau) g(\tau) d\tau,$$

where $f$ is the quadrature approximation to $f$ constructed using the Laplace inversion due to Talbot, whose error is well under control, and $g$ is the piecewise linear interpolant of $g$.

The approximations of the inverse Laplace transform and the solution of the initial value problems (25) can be combined into a fast convolution algorithm that requires $O(N_t \log_B N_t)$ arithmetical operations and $O(\log_B N_t)$ memory. For a detailed step by step explanation of the algorithm we strongly advise the reader to have a look at one of the references [7, 10].

The general base-$B$ algorithm approximates the convolution as follows: in the $n$th step ($n = 1, \ldots, N_t$), let $t = n\Delta t$ approximate $g$ on $[t - \Delta t; t]$ linearly. So we get

```
$Y$ = expint($Y, \Lambda, \varphi, gn, gn1, \Delta t, K$)
% input
%     $Y, \Lambda, \varphi$  vectors of length $K$ of complex vectors
%        (i.e. $\Lambda_\ell$ is a complex vector and $1 \leq \ell \leq K$)
%     $\Delta t$ real ; $gn, gn1$ complex ; $K$ integer
% output
%     $Y$ vector of length $K$ of complex vectors

$dg = gn1 - gn$ ; $dgDdt = dg/\Delta t$
for $\ell = 1, \ldots, K$
   $Y_\ell = Y_\ell + \varphi_\ell .* \left(\Lambda_\ell .* Y_\ell + gn + dgDdt./\Lambda_\ell\right) - dgDdt./\Lambda_\ell$
end
```

**Fig. 11.** Pseudocode for the integration of differential equation (25)

$$\int_{t-\Delta t}^{t} f(t-\tau)g(\tau)d\tau \approx \int_{t-\Delta t}^{t} f(t-\tau)d\tau g(t-\Delta t) + \int_{t-\Delta t}^{t} f(t-\tau)\tau\, d\tau \frac{g(t)-g(t-\Delta t)}{\Delta t}. \tag{28}$$

The integrals are approximated as the inverse Laplace transforms of $F(s)/s$ and $F(s)/s^2$, respectively:

$$\begin{aligned} \phi_1 &:= \int_0^{\Delta t} f(\Delta t-\tau)d\tau \approx \sum_{j=-N}^{N} w_j F(\lambda_j)/\lambda_j\, e^{\Delta t \lambda_j}, \\ \phi_2 &:= \int_0^{\Delta t} f(\Delta t-\tau)\tau\, d\tau \approx \sum_{j=-N}^{N} w_j F(\lambda_j)/\lambda_j^2\, e^{\Delta t \lambda_j}, \end{aligned} \tag{29}$$

where the weights $w_j$ and nodes $\lambda_j$ correspond to a Talbot contour with the parameters chosen for $t = \Delta t$. $\phi_1$ and $\phi_2$ are calculated once and are reused in every step, so it is favorable to calculate them with high accuracy, i.e using more points on $\Gamma$ than usual.

To approximate

$$\int_0^{t-\Delta t} f(t-\tau)g(\tau)d\tau$$

let $L$ be the smallest integer for which $t < 2B^L\Delta t$. For $\ell = 1, 2, \ldots, L-1$ determine integers $B \geq q_\ell \geq 1$ such that

$$\tau_\ell = q_\ell B^\ell \Delta t \quad \text{satisfies} \quad t - \tau_\ell \in [B^{\ell-1}\Delta t, (2B^\ell - 1)\Delta t].$$

Note that $q_\ell$ is increased by 1 every $B^\ell$ steps, and $t - \Delta t > \tau_1 > \cdots > \tau_{L-1} > 0$. Set $\tau_0 = t - \Delta t$ and $\tau_L = 0$. Then we split and approximate

$$\begin{aligned} \int_0^{t-\Delta t} f(t-\tau)g(\tau)d\tau &= \sum_{\ell=1}^{L} \int_{\tau_l}^{\tau_{l-1}} f(t-\tau)g(\tau)d\tau & (30) \\ &\approx \sum_{\ell=1}^{L} \sum_{j=-N}^{N} w_j^{(\ell)} F(\lambda_j^{(\ell)}) e^{(t-\tau_{\ell-1})\lambda_j^{(\ell)}} y(\tau_{\ell-1}, \tau_\ell, \lambda_j^{(\ell)}), & (31) \end{aligned}$$

where $w_j^{(\ell)}$ and $\lambda_j^{(\ell)}$ are the weights and quadrature points for the Talbot contour $\Gamma_\ell$ that corresponds to the base-$B$ approximation interval $I_\ell = [B^{\ell-1}\Delta t, (2B^\ell - 1)\Delta t]$ of (22). Note that $[t - \tau_{\ell-1}, t - \tau_\ell] \subset I_\ell$ for all $\ell$. The differential equations determining $y(t, \tau, \lambda)$ are advanced by one step of (27) for all values $\lambda$ on all Talbot contours for every time step $t \to t + \Delta t$. The operation counts and memory requirements are proportional to $N_t K_t N$ and $K_t N$, respectively, where the number of integration contours is bounded by $K_t \leq \log_B N_t$.

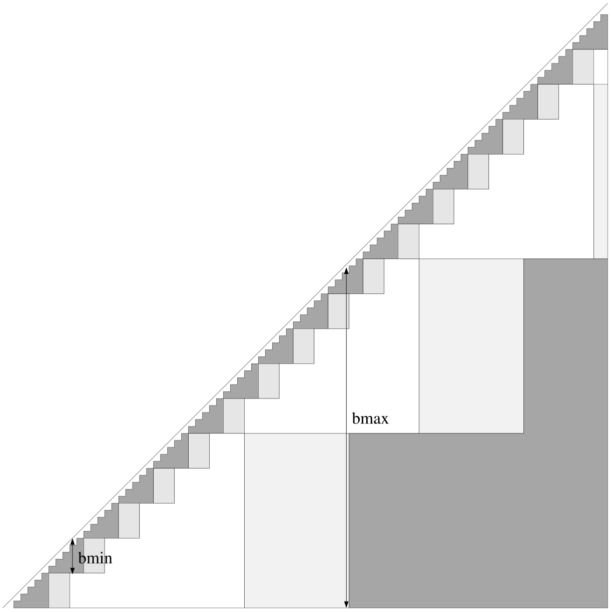

**Fig. 12.** Tessellation of the $t - \tau$ plane for $B = 5$. For $\ell = 2$ the maximal and minimal value of $t - \tau_\ell$ are indicated by arrows

The approximation (31) is calculated by the algorithm `convolutionint` given in Figure 13. Note that $E\Lambda F\Lambda W$ and $E\Lambda$ are vectors calculated once in the beginning by the `initialize` function given in Figure 14.

The algorithm that stores and organizes the solutions of all the differential Equations (25) according to the splitting (30) is given in Figure 15. There

- $Y$ is the solution of the differential equation, which is updated in every step by `expint` and has to be restarted with starting values 0 if $n$ is a multiple of $B^\ell$,
- $YM$ stores the solution of the differential equation if it is completed, i.e it corresponds to the stair-shaped regions in Figure 12,
- $YT$ contains the solution of (25) that are required by `convolutionint` and
- $YA$ corresponds to the rectangular intermediate light gray regions in Figure 12, where the tessellation of the $t - \tau$ plane is given for $B = 5$.

Further explanations are given in [7], Sections 2.4 and 2.5.

## 5. Complete Discretization

Many options are available for the spatial discretization of the electric wave equation inside $\Omega$ (taking into account the transition layer adjacent to the port, where Yee's scheme still has to be used). We assume that the mass matrix related

```
$(b, YT) = \texttt{convolutionint}(E\Lambda F\Lambda W, YT, E\Lambda, L)$
$b = \texttt{sum}(E\Lambda F\Lambda W_1 \;.\!*\; YT_1)$
for $k = 2 \dots L$
    $b = b + \texttt{sum}(E\Lambda F\Lambda W_k \;.\!*\; YT_k)$
    $YT_k = E\Lambda_k \;.\!*\; YT_k$
end
```

**Fig. 13.** Pseudocode for the evaluation of the convolution integral

```
$(\Lambda, E\Lambda F\Lambda W, Y, YT, YM, YMt, YA, YAt, \varphi, E\Lambda, E\Lambda J, q, K, L)$
  $= \texttt{initialize}(c, dx, \Delta t, nt, B, N, \mu_0, \nu_0, \alpha, \gamma)$
%  input
%     $c, dx, \Delta t, \mu_0, \nu_0$ real ; $nt, B, N$ integer ; $\alpha, \gamma$ complex
%  output
%     $S, E\Lambda F\Lambda W, Y, YT, YM, YMt, YA, YAt, \varphi, E\Lambda, E\Lambda J$
%        vectors of complex vectors
%     $q$ real vector of variable size ; $K, L$ integer

$K = \texttt{ceil}(\texttt{log}((nt + 1)/2)/\texttt{log}(B))$
$[\Lambda, E\Lambda F\Lambda W] = \texttt{talbotcontour}(c, \Delta t, dx, K, \alpha, \gamma, B, N, \mu_0, \nu_0);$
for $j = 1 : K$
    $YA_j = YM_j = YT_j = Y_j = 0 * \Lambda_j$ ; $YMt_j = [0, 0]$ ; $YAt_j = [0, 0]$
    $\varphi_j = \Delta t * \texttt{phis}(\Delta t * \Lambda_j)$
    $E\Lambda_j = \texttt{exp}(\Lambda_j * \Delta t)$
    $E\Lambda J_j = \texttt{exp}(\Lambda_j * \Delta t * B^{j-1})$
end
$q = [0]$ ; $L = 0$
```

**Fig. 14.** Overview of required values necessary for the convolution algorithm, which are calculated in advance

to the dielectric coefficient $\epsilon$ will still be constant. Then the semi-discrete equation inside $\Omega$ reads

$$\frac{d^2 \mathbf{u}_h}{dt^2} + \mathscr{T}_h \mathbf{u}_h = 0, \tag{32}$$

where $\mathscr{T}_h$ is a discrete version of $\mathbf{curl}\,\mu^{-1}\mathbf{curl}$ subject to Dirichlet boundary condition on the ports and scaled with the inverse $\epsilon$ mass matrix. Next, we use the explicit leap-frog scheme for time-stepping

```
$(Y, YA, YM, YT, YAt, YMt, L, q) =$
  yupdate$(Y, YA, YM, YT, YAt, YMt, L, q, E\Lambda J, n, \Delta t, K, B)$
$\tau = [\,]$
$n1 = n + 1$
$\ell = 1\ ;\ m = 1$
if $2B^L = n1 + 1\ ;\ L = L + 1;$ end
while mod$(n1 + 1, B^\ell) = 0\ \&\ \ell < L$
  if $l >$length$(q)\ ;\ q = [\ q\ ,\ 0\ ]$ end
  $q_\ell = q_\ell + 1\ ;\ \ell = \ell + 1$
end
$\tau = dt * [\ n1\ ,\ q\ .*\ (B\ .\hat{}\,[\ 1\ :\ $length$(q)\ ])\ ,\ 0\ ]$
while mod$(n1, B^{(m-1)}) == 0\ \&\ m < K$
  if mod$(n1, B^m) == 0$
    $YA_m = Y_m\ ;\ YAt_m = [\ n1 * \Delta t\ ,\ YMt_m(2)\ ]$
    $YM_m = Y_m\ ;\ YMt_m = [\ n1 * \Delta t\ ,\ YAt_m(2)\ ]$
    $Y_m = 0 * Y_m$
  else
    $YM_m = Y_m\ ;\ YMt_m = [\ n1 * \Delta t\ ,\ YAt_m(1)\ ]$
  end
  $m = m + 1$
end
for $k = 1 \ldots$ length$(\tau) - 1$
  if $\tau_k = YMt_k(1)$
    $YT_k = YM_k$
    if $\tau_{k+1} \neq YMt_k(2)$
      $YA_k = E\Lambda J\ .*\ YA_k\ ;\ YT_k = YT_k + YA_k$
    end
  end
end
```

**Fig. 15.** Organization of the solution of differential equations

$$\mathbf{u}_h(\mathbf{p}, (n+1)\Delta t) = (2 + \Delta t^2 \mathscr{T}_h)\mathbf{u}_h(\mathbf{p}, n\Delta t) - \mathbf{u}_h(\mathbf{p}, (n-1)\Delta t).$$

How can we incorporate the new non-reflecting boundary conditions in the context of leap-frog time-stepping? The non-reflecting boundary conditions as given in Equations (20) are implicit in time and are used to update boundary values of $\mathbf{u}_h$ at the ports. Yet, the leap-frog time-stepping is explicit and it is highly desirable to have explicit non-reflecting boundary condition, too. To obtain them we have to modify the direct step in the convolution algorithm. Instead of interpolating differences of boundary and auxiliary values linearly as in Equation (28), we will use linear extrapolation at this stage. More precisely, we approximate, with $\phi_1$ and $\phi_2$ from Equation (29),

$$\int_t^{t+\Delta t} f(t-\tau)g(\tau)d\tau \approx \phi_1 g(t) + \phi_2 \frac{g(t) - g(t-\Delta t)}{\Delta t}. \tag{33}$$

The transition layer at a part of the port is shown in Figure 16. The solid edges are located in the interior of $\Omega$ and their nodal values (represented by white arrows on solid lines) are to be updated in the course of leap-frog time-stepping. To do this, the leap-frog scheme needs auxiliary values from edges located in the port plane $\Gamma$ and from $x$-edges in the transition layer (black arrows on dashed lines in Figure 16). These are computed using the convolution algorithm, i.e

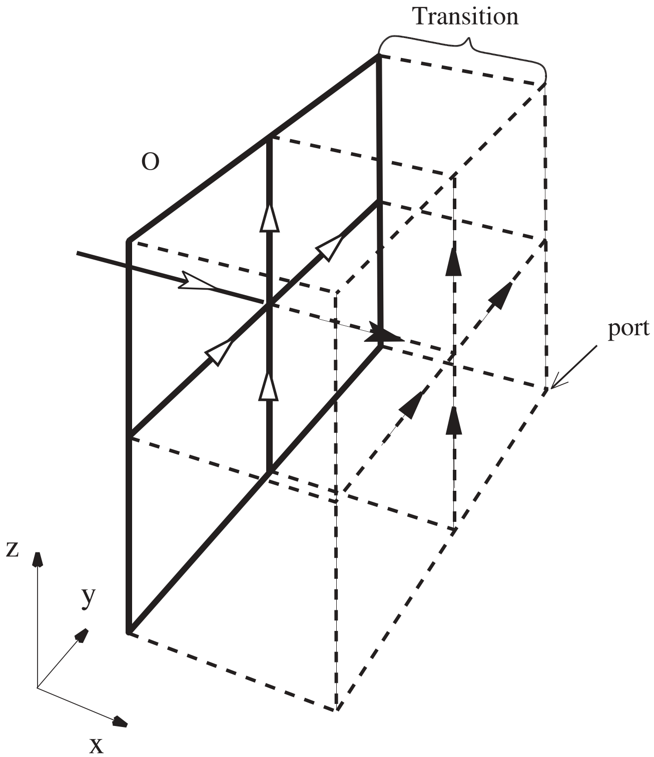

**Fig. 16.** Interface layer with auxiliary boundary values

$$
\begin{aligned}
u_h^y(0,y,z,t+\Delta t) = \mathscr{Y}^{-1}\Bigg[ & \Phi_1(j,k)\mathscr{Y}(u_h^y(-h,\cdot,\cdot,t) - u_h^y(0,\cdot,\cdot,t))(j,k) \\
& + \frac{\Phi_2(j,k)}{\Delta t}(\mathscr{Y}(u_h^y(-h,\cdot,\cdot,t) - u_h^y(0,\cdot,\cdot,t))(j,k) \\
& - \mathscr{Y}(u_h^y(-h,\cdot,\cdot,t-\Delta t) - u_h^y(0,\cdot,\cdot,t-\Delta t))(j,k)) \\
& + \underbrace{\int_0^t f_{j,k}(t+\Delta t-\tau)\mathscr{Y}(u_h^y(-h,\cdot,\cdot,t) - u_h^y(0,\cdot,\cdot,t))(j,k)d\tau}_{\text{calculated using the fast convolution algorithm}}\Bigg](y,z),
\end{aligned}
$$

$$
\begin{aligned}
u_h^z(0,y,z,t+\Delta t) = \mathscr{Z}^{-1}\Bigg[ & \Phi_1(j,k)\mathscr{Z}(u_h^z(-h,\cdot,\cdot,t) - u_h^z(0,\cdot,\cdot,t))(j,k) \\
& + \frac{\Phi_2(j,k)}{\Delta t}(\mathscr{Z}(u_h^z(-h,\cdot,\cdot,t) - u_h^z(0,\cdot,\cdot,t))(j,k) \\
& - \mathscr{Z}(u_h^z(-h,\cdot,\cdot,t-\Delta t) - u_h^z(0,\cdot,\cdot,t-\Delta t))(j,k)) \\
& + \underbrace{\int_0^t f_{j,k}(t+\Delta t-\tau)\mathscr{Z}(u_h^z(-h,\cdot,\cdot,t) - u_h^z(0,\cdot,\cdot,t))(j,k)d\tau}_{\text{calculated using the fast convolution algorithm}}\Bigg](y,z).
\end{aligned}
$$

where $\Phi_1$ and $\Phi_2$ are given by

$$
\Phi_1(j,k) = \sum_{l=-N}^{N} w_l \frac{F_{j,k}(\lambda_l)}{\lambda_l} e^{\Delta t \lambda_l} \quad \text{and} \quad \Phi_2(j,k) = \sum_{l=-N}^{N} w_l \frac{F_{j,k}(\lambda_l)}{\lambda_l^2} e^{\Delta t \lambda_l}, \quad \mathit{cf.}\ (29)
$$

The components of the field in the boundary layer orthogonal to the boundary (black sharp arrows on dashed lines pointing in $x$-direction in Figure 16) are updated using the leapfrog scheme in the transition layer, which requires boundary and auxiliary values only. Thus, as stated in Section 3, Equation (17) is redundant.

Using the definition of $\delta_h^x$ from (11), we can rewrite the boundary condition.

$$
\begin{aligned}
u_h^y(0,y,z,t+\Delta t) = \mathscr{Y}^{-1}\Bigg[ & \Phi_1(j,k)\mathscr{Y}(-h\delta_{h/2}^x u_h^y(-h/2,\cdot,\cdot,t))(j,k) \\
& + \frac{\Phi_2(j,k)}{\Delta t}\Big(\mathscr{Y}(-h\delta_{h/2}^x u_h^y(-h/2,\cdot,\cdot,t))(j,k) \\
& - \mathscr{Y}(-h\delta_{h/2}^x u_h^y(-h/2,\cdot,\cdot,t-\Delta t))(j,k)\Big) \\
& + \int_0^t f_{j,k}(t+\Delta t-\tau)\mathscr{Y}(-h\delta_{h/2}^x u_h^y(-h/2,\cdot,\cdot,t))(j,k)d\tau\Bigg](y,z)
\end{aligned}
$$

$$
\begin{aligned}
u_h^z(0,y,z,t+\Delta t) = \mathscr{Z}^{-1}\Bigg[&\Phi_1(j,k)\mathscr{Z}(-h\delta^x_{h/2}u_h^z(-h/2,\cdot,\cdot))(j,k) \\
&+\frac{\Phi_2(j,k)}{\Delta t}\Big(\mathscr{Z}(-h\delta^x_{h/2}u_h^z(-h/2,\cdot,\cdot,t))(j,k) \\
&-\mathscr{Z}(-h\delta^x_{h/2}u_h^z(-h/2,\cdot,\cdot,t-\Delta t))(j,k)\Big) \\
&+\int_0^t f_{j,k}(t+\Delta t-\tau)\mathscr{Z}(-h\delta^x_{h/2}u_h^z(-h/2,\cdot,\cdot,t))(j,k)d\tau\Bigg](y,z)
\end{aligned}
$$

**Remark:** Sine and cosine transform are implemented using fast transform algorithms. The fast sine transform can be implemented straightforward using the standard fast Fourier transform that comes with MATLAB. The usual cosine transform as given for example in Section 5.6 by Jain [6], uses a scaling of the first coefficient different from the one we use in Equations (7) and (8), so it has to be modified. The reordering given in [6] remains valid, though.

Figure 17 gives an overview of the algorithm for the time integration of Maxwell's equations.

## 6. Numerical Experiments

To validate our non-reflecting boundary conditions we calculate the evolution of an electric wave in a homogeneous lossless waveguide $\tilde{\Omega} = [-\infty;\infty] \times [0;\pi]\times[0;\pi]$ with perfectly conducting walls. The computational domain is given by $\Omega = [0;\pi] \times[0;\pi] \times [0;\pi]$. Thus we are using our non-reflecting boundary conditions at $x = 0$ and $x = \pi$. We compute an approximate solution $\mathbf{u}$ for the electric wave equation (1) in $\Omega \cup \Omega_1 \cup \Omega_2$ subject to the initial condition

$$\mathbf{u}(0) = \mathbf{u}_0 \quad \text{and} \quad \frac{d}{dt}\mathbf{u}(0) = 0,$$

where the divergence-free initial value $\mathbf{u}_0$ at $t = 0$ is given by

$$
\begin{aligned}
u_0^x(x,y,z,0) &= \alpha\exp(-((x-\pi/2)^2+(y-\pi/2)^2+(z-\pi/2)^2)/w_0)(2(y-\pi/2)/w_0) \\
u_0^y(x,y,z,0) &= \alpha\exp(-((x-\pi/2)^2+(y-\pi/2)^2+(z-\pi/2)^2)/w_0)(2(x-\pi/2)/w_0 \\
&\qquad +2(z-\pi/2)/w_0) \\
u_0^z(x,y,z,0) &= \alpha\exp(-((x-\pi/2)^2+(y-\pi/2)^2+(z-\pi/2)^2)/w_0)(2y-\pi/2)/w_0
\end{aligned}
$$

Here $\alpha = 5$ and $w_0 = 0.05$ are parameters set to fit $\mathbf{u}_0$ to the computational domain. In the discrete scheme, $\mathbf{u}_0$ is interpolated onto the edges of the grid. Then another $L^2$-orthogonalization is carried out, in order to ensure that the discrete initial value $\mathbf{u}_h(0)$ is approximately orthogonal to the kernel of the discrete **curl**. This is essential for obtaining meaningful solutions of Maxwell's equations.

```
$B, N, \mu_0, \nu_0$ parameters for the convolution algorithm
$\Delta t, nt$ step size and number of steps for time discretization
$dx, dy, dz, sx, sy, sz$ mesh width and geometry size
$\alpha, \gamma$ singularities of the convolution kernel for "sin cos" coefficients and
  "cos sin" coefficients
$(S, E\Lambda F\Lambda W, Y, YT, YM, YMt, YA, YAt, \varphi, E\Lambda, E\Lambda J, q, K, L)$
  = initialize($c, dx, \Delta t, nt, B, N, \mu_0, \nu_0, \alpha, \gamma$)
$\Phi_1, \Phi_2$ matrices calculated from Equation (29)
initialize discretization in the interior
$g_0$, $g_1$ calculated from $\mathcal{Z}$, $\mathcal{Y}$ transforms of differences in transition layer
$Y$ = expint($Y, S, \varphi, g0, g1, \Delta t$)
$(Y, YA, YM, YT, YAt, YMt, L, q) =$
  yupdate$(Y, YA, YM, YT, YAt, YMt, L, q, E\Lambda J, 0, \Delta t, K, B)$
for $n = 1 \ldots N_t$
  advance leap-frog scheme in computational domain $\Omega$ from $n\Delta t$ to $(n+1)\Delta t$ using values in auxiliary boundary points at time $n\Delta t$
  $(b, YT) =$ convolutionint$E\Lambda, E\Lambda F\Lambda W, YT, L)$
  update values in auxiliary boundary points to $(n+1)\Delta t$
  calculate $g_{n+1}$ from $\mathcal{Z}$ $\mathcal{Y}$ transforms of differences in transition layer
  $Y$ = expint($Y, S, \varphi, g_n, g_{n+1}, dt$)
  $(Y, YA, YM, YT, YAt, YMt, L, q) =$
    yupdate$(Y, YA, YM, YT, YAt, YMt, L, q, E\Lambda J, n, \Delta t, K, B)$
end
```

**Fig. 17.** Complete time-stepping for Maxwell's equations in $\Omega$ using convolution based non-reflecting boundary conditions

The second initial condition is incorporated by formally setting

$$\mathbf{u}_h(-\Delta t) = \mathbf{u}_h(0) - \frac{1}{2}\Delta t^2 \mathscr{C}_h \mathbf{u}_h(0).$$

We refer to Section 2 for the definition of $\mathscr{C}_h$.

We measure the error introduced by the non-reflecting boundary conditions by comparing the discrete solution with a reference solution calculated on the larger domain $\Omega_{ref} = [-\pi; 2\pi] \times [0; \pi] \times [0; \pi]$. Hence, we can be certain that no spurious reflections pollute the reference solution before $t = 2\pi$. Any deviation of both solutions must be due to the approximation error of the inverse Laplace transform and the error introduced by the different ways of time-stepping used inside the computational domain and on the boundary.

Calculating on a uniform grid with 32 and 64 grid points in each direction (i.e. $h = \pi/32$ and $h = \pi/64$) and setting the parameters of the convolution algorithm $B = 5$, $N = 15$ the evolution of the $l_2$ norm of the error is shown in Figure 18 for different time step sizes, $\Delta t = h/2$, $h/4$ and $h/8$.

We see that the error only depends on the time step size $\Delta t$. This example further illustrates that our algorithm is of second order in time as can be expected. For this choice of parameters the dominating error is that of the time integration. For a discussion of the error introduced by the approximation of the inverse Laplace transform we refer to [7, 10]. We point out that the deviation is zero before the waves reach the ports.

The energy of the electric wave is given as the sum of magnetic energy and electric energy.

$$E(n\Delta t) := E_{\mathrm{mag}}(n\Delta t) + E_{\mathrm{el}}(n\Delta t),$$

where

$$\begin{aligned} E_{\mathrm{mag}}(n\Delta t) := h^3 \Bigg( \sum_{j,k,l} &(u^x(jh, kh, lh, n\Delta t + \Delta t) - u^x(jh, kh, lh, n\Delta t - \Delta t))^2/(4\Delta t^2) \\ &+ (u^y(jh, kh, lh, n\Delta t + \Delta t) - u^y(jh, kh, lh, n\Delta t - \Delta t))^2/(4\Delta t^2) \\ &+ (u^z(jh, kh, lh, n\Delta t + \Delta t) - u^z(jh, kh, lh, n\Delta t - \Delta t))^2/(4\Delta t^2) \Bigg), \end{aligned} \tag{34}$$

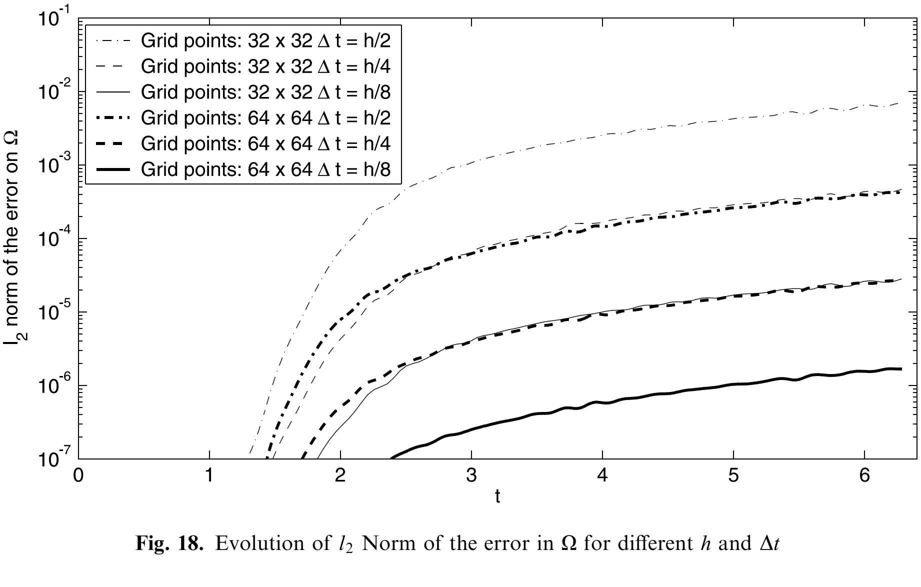

**Fig. 18.** Evolution of $l_2$ Norm of the error in $\Omega$ for different $h$ and $\Delta t$

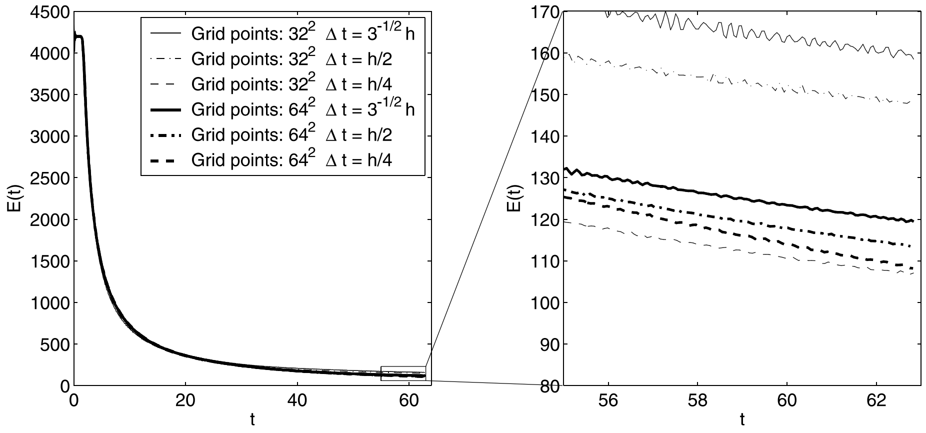

**Fig. 19.** Energy versus time for different $h$ and $\Delta t$. The right plot is a zoom of the left one as indicated

$$
\begin{aligned}
E_{\mathrm{el}}(n\Delta t) := h^3 \Bigg( \sum_{j,k,l} & (\mathbf{curl}\,\mathbf{curl}\,\mathbf{u})_x (jh, kh, lh, n\Delta t) u^x (jh, kh, lh, n\Delta t) \\
& + (\mathbf{curl}\,\mathbf{curl}\,\mathbf{u})_y (jh, kh, lh, n\Delta t) u^y (jh, kh, lh, n\Delta t) \\
& + (\mathbf{curl}\,\mathbf{curl}\,\mathbf{u})_z (jh, kh, lh, n\Delta t) u^z (jh, kh, lh, n\Delta t) \Bigg).
\end{aligned}
\tag{35}
$$

Figure 19 displays the evolution of the energy over a long time interval $[0, 20\pi]$ for $h = \pi/32$ and $h = \pi/64$ and $\Delta t = h/\sqrt{3}$, $\Delta t = h/2$ and $\Delta t = h/2$. We point out that we do not see any additional restrictions by the boundary condition on the stability limit for the time step size $\Delta t$ which is $\Delta t = h/\sqrt{3}$ for the leapfrog scheme in these special cases.

## Acknowledgement

We thank Christian Lubich for many helpful comments. It was his idea to introduce the auxiliary grid function given in (14). The work of both authors was supported by the DFG as part of SFB 382, Tübingen, Stuttgart. The second author thanks the Zuse Institute Berlin and the DFG research center (FZT 86), Berlin for providing the possibility to complete this work.

## References

[1] Bachelot, A., Bounhoure, L., Pujols, A.: Couplage éléments finis-potentiels retardés pour la diffraction electromagnétique par un obstacle hétérogene. Numer. Math. *89*, 257–306 (2001).

[2] Bérenger, J.-P.: A perfectly matched layer for the absorption of electromagnetic waves. J. Comput. Phys. *114*(2), 185–200 (1994).

[3] Hagstrom, T.: Radiation boundary conditions for numerical simulation of waves. Acta Numerica *8*, 47–106 (1999).

[4] Hagstrom, T.: New results on absorbing layers and radiation boundary conditions. In: Ainsworth, M. (ed.): Computational methods in wave propagation, pp. 1–42, New York: Springer 2003. In print.

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

[6] Jain, A. K.: Fundamentals of digital image processing. Prentice-Hall 1989.

[7] Lubich, Ch., Schädle, A.: Fast convolution for non-reflecting boundary conditions. SIAM J. Sci. Comput. *24*, 161–182 (2002).

[8] The MathWorks. Matlab the language of technical computing. User guide 2002.

[9] Rizzardi, M.: A modification of Talbot's method for the simultaneous approximation of several values of the inverse Laplace transform. ACM Trans. Math. Software *21*(4), 347–371 (1995).

[10] Schädle, A.: Ein schneller Faltungsalgorithmus für nichtreflektierende Randbedingungen. PhD thesis, Eberhard-Karls-Universität Tübingen 2002.

[11] Taflove, A.: The finite difference in time domain method. Boston London: Artech House 1995.

[12] Talbot, A.: The accurate numerical inversion of Laplace transforms. J. Inst. Math. Appl. *23*, 97–120 (1979).

[13] Weiland, T.: Time domain electromagnetic field computation with finite difference methods. Int. J. Numer. Modelling *9*, 295–319 (1996).

[14] Yee, K.: Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Trans. Antennas Propagation *16*, 302–307 (1966).

Ralf Hiptmair
SAM, ETH Zürich
CH-8092 Zürich
Switzerland
e-mail: hiptmair@math.ethz.ch

Achim Schädle
Konrad Zuse Zentrum
für Informationstechnik Berlin
Takustr. 7
D-14195 Berlin
Germany
e-mail: schaedle@zib.de

Verleger: Springer-Verlag KG, Sachsenplatz 4–6, A-1201 Wien. – Herausgeber: Prof. Dr. Wolfgang Hackbusch, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Inselstraße 22–26, D-04103 Leipzig – Satz und Umbruch: Scientific Publishing Services (P) Ltd., Madras; Offsetdruck: Manz Crossmedia. 1051 Wien. – Verlagsort: Wien. – Herstellungsort: Wien. – Printed in Austria.