Computer Physics Communications 185 (2014) 1273–1281

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Implementation of the iterative finite-difference time-domain technique for simulation of periodic structures at oblique incidence

Ilya Valuev [a], Alexei Deinega [b,*], Sergey Belousov [c,d]

[a] Joint Institute for High Temperatures of RAS, Izhorskaya, 13, build. 2, Moscow, 125412, Russia
[b] Department of Chemistry, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208-3113, USA
[c] Kintech Lab Ltd., 1 Kurchatov Sq., Moscow, 123182, Russia
[d] National Research Centre "Kurchatov Institute", 1 Kurchatov Sq., Moscow, 123182, Russia

ARTICLE INFO

Article history:
Received 12 July 2012
Received in revised form
1 January 2014
Accepted 5 January 2014
Available online 11 January 2014

Keywords:
FDTD
Oblique incidence

ABSTRACT

In this paper we review a recently developed finite-difference time-domain (FDTD) iterative technique for the analysis of periodic structures at oblique incidence. We show how it can be implemented in FDTD code and estimate required computer memory and time resources. To illustrate performance of our technique we demonstrate the plasmon formation in a thin gold film placed at air/glass interface and calculate reflectance from silicon textured coating at oblique incidence.

1. Introduction

Finite-difference time-domain (FDTD) [1–3] is one of the most popular numerical methods in computational electrodynamics. It is widely used for the calculation of transmission and reflection for planar layers of scatterers, like photonic crystals or antenna arrays. Transmission and reflection can be obtained by simulating the propagation of a temporal electromagnetic impulse through the considered structure. A numerical experiment needed to obtain transmission and reflection properties of a layer may become quite complicated if we consider oblique wave incidence.

The usual simulation scheme for obtaining transmission and reflection from FDTD calculation consists of the following. An incident plane wave is generated in FDTD space as coming from outside the structure with the required direction. Usually it has a wide wavelength spectrum, so the transmission and reflection may be obtained for a range of frequencies from one simulation. To generate a plane wave in FDTD several techniques may be applied, the most efficient of them is the Total Field/Scattered Field method [1] which we discuss below.

In the case of periodicity in planar direction a single unit cell with periodic boundary conditions may be simulated (Fig. 1). At *normal incidence* of the incoming plane wave due to the symmetry of the system we have

$$\mathbf{F}(\mathbf{x}, t) = \mathbf{F}(\mathbf{x} \pm \mathbf{a}, t), \tag{1}$$

where $\mathbf{F}$ is the electric or magnetic field ($\mathbf{E}$ or $\mathbf{H}$), $\mathbf{a}$ is a lattice translation vector parallel to the structure surface, $\mathbf{x}$ and t are coordinates in space and time. Eq. (1) taken at boundary points $\mathbf{x}$ of the simulated unit cell may be used as boundary conditions, stating that the fields at opposite boundaries should be equal. These boundary conditions are implemented in FDTD by simply using the same grid points for both boundaries. Absorbing Perfectly Matched Layers (PMLs) [1] are usually used for non-periodical direction and absorb the reflected and transmitted waves modeling their withdrawal to the infinity.

In the course of simulation the numerically obtained fields at locations corresponding to transmitted and reflected waves are recorded. The transmitted fields are recorded behind the planar structure, and the reflected fields are recorded in front of it (taking into account the incident wave). Total exit of the radiation from the structure determines the simulation time. The recorded transmitted and reflected waves are transformed to the frequency domain and normalized to the incident spectrum to calculate transmittance (reflectance).

At *oblique incidence* periodic boundary conditions analogous to (1) contain a time shift. In 2D they take a form (generalization for 3D case is straightforward)

$$\mathbf{F}(\mathbf{x}, t) = \mathbf{F}(\mathbf{x} \pm \mathbf{a}, t \pm \delta t), \tag{2}$$

$$\delta t = a \sin\theta / c, \tag{3}$$

* Corresponding author. Tel.: +1 647 832 80 80.
E-mail address: deinega@northwestern.edu (A. Deinega).

0010-4655/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cpc.2014.01.001

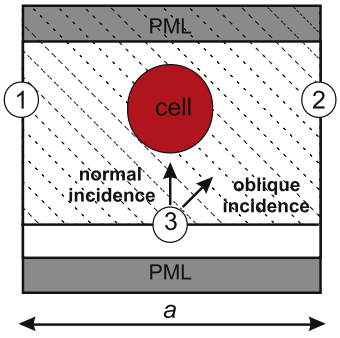

Fig. 1. FDTD geometry setup for single unit cell of photonic crystal slab, consisting of a square lattice of spheres. Unit cell is confined by periodic boundaries 1 and 2. The virtual total field/scattered field (TF/SF) surface 3 generates incident plane wave impulse impinging the cell. PMLs absorb reflected and transmitted waves simulating their withdraw to infinity.

where θ is the angle of incidence, c is the speed of light in the incident medium. The meaning of the expression (2) can be clarified by Fig. 1. Oblique wave comes to periodic border 1 earlier than to periodic border 2. Therefore the field values at these borders are shifted in time. Using of periodic boundary conditions requires knowledge of retarded field values at border 1 (for applying at border 2) and advanced field values at border 2 (for applying at border 1). The retarded fields can in principle be picked up from a recorded wave propagation. Obtaining the advanced fields constitutes a problem since they are unknown during numerical experiment.

Several methods were proposed to deal with this problem. These methods can be classified in three groups.

In the first group, a special field transformation is used to eliminate the time shift between the adjacent unit cells [4–10]. However, the transformed equations differ from the standard Maxwell's equations and their numerical solution becomes unstable when the angle of incidence approaches $\theta = 90°$. Besides this, additional modification of the method is required to handle dispersive [11–15], anisotropic [16] and nonlinear [17] structures.

In the second group of methods, which are referred to as "spectral", time shifted periodic condition (2) is replaced by the complex Bloch boundary condition $F(t + a\sin\theta/c) = F(t)\exp(i\alpha)$, formulated for the time domain. Using monochromatic initial incident wave $F(t) = F(0)\exp(i\omega t)$ with the frequency ω satisfying $\alpha = \omega a \sin\theta/c$ one can get results for a single frequency per simulation [18]. To obtain the results for a wider frequency range one can apply a non-monochromatic incident wave. In this case the time-domain solution is regarded as an intermediate result. At the final stage of the calculation one should transform into the frequency domain where results for a range of (ω, θ) pairs can be extracted [19–22]. The time-dependent solution for the fields is not available in spectral methods, which is a serious drawback for such applications, as molecular nanopolaritonics [23,24], where the charge transport between nanoparticles and molecules is studied and a coupled system of Maxwell/Schrödinger equations should be solved [24–26]. Direct time-dependent FDTD simulation is also necessary within novel FDTD approaches to solve Maxwell-Liouville equations for single quantum emitters (such as quantum dots or single molecules) [27].

In the third group of methods additional unit cells are introduced to simplify getting the time-advanced field values. In the multiple unit cells method [28,29] these cells are added along the direction from which the incident wave arrives. This sequence of cells is terminated by PMLs. The terminating cell is a source of error, the magnitude of which depends on the number of additional cells and the incidence angle. In the angled-update method [1] mesh points are updated non-simultaneously, which allows one to obtain the future field values from the time-advanced adjacent cells. The drawback of this method is a restriction to small angles (in 3D case the angle of incidence is limited to 35°).

In our previous work we introduced a new method for the analysis of periodic structures at oblique incidence [30]. This method cannot be classified into any of the presented above groups since it is based on a different principle. Our method implies performing several FDTD numerical experiments, which we call iterations later on. Field values at the periodic boundaries are recorded during each iteration. It gives the key to a solution of the problem with the advanced field values: even if they are unknown at the current iteration, they are known at the previous one since field history have been recorded. Time shifted field values from the previous iteration can be used at the current iteration as an approximation for the advanced fields. As we have shown in our previous work [30], the difference between the true advanced fields and the approximate ones decreases from iteration to iteration, so the iterative process converges. To manage this iterative process we use "soft" Total field/Scattered field (TF/SF) correction [1] instead of "strong" periodic conditions (2). This TF/SF correction acts like periodic conditions (2) after a number of iterations required for convergence.

In the previous work we have been focused on the basic principle of iterative method and its verification. In this paper we describe its numerical implementation.

The paper is organized as follows. In Section 2 we present the main idea of the method. In Section 3 we discuss how to implement the method in an FDTD code. In Section 4 we illustrate the work of the method for some physical examples. In Sections 6 and 5 we discuss the performance and convergence issues. In Section 7 we summarize our results.

2. Method

In the following we will refer to the FDTD contour path approach and the Total field/Scattered field (TF/SF) technique [1], so we need to review these methods here.

FDTD discretization of Maxwell's equations can be derived using the contour path approach. This approach is helpful for formulation of TF/SF technique which is a part of our iterative method. FDTD discretization proposed by Yee [3] does not necessarily need to be formulated within this approach, however, we will still use it to describe our iterative technique.

Contour path approach deals with the integral formulation of Maxwell's equations:

$$\frac{\partial}{\partial t}\int_S \mathbf{E}d\mathbf{S} = \frac{1}{\varepsilon}\oint_l \mathbf{H}d\mathbf{l}, \qquad \frac{\partial}{\partial t}\int_S \mathbf{H}d\mathbf{S} = -\frac{1}{\mu}\oint_l \mathbf{E}d\mathbf{l}, \tag{4}$$

where the line and surface integrations are performed over an arbitrary flat contour l in space and its internal enclosed area S correspondingly. Here for simplicity we consider linear, isotropic, nondispersive materials. Eqs. (4) may be rewritten in a discretized central-difference form:

$$\frac{\mathbf{E}_c\left(t+\frac{1}{2}\Delta t\right) - \mathbf{E}_c\left(t-\frac{1}{2}\Delta t\right)}{\Delta t}\mathbf{S} = \frac{1}{\varepsilon}\sum_i \mathbf{H}_i(t)\mathbf{l}_i \tag{5}$$

$$\frac{\mathbf{H}_c(t+\Delta t) - \mathbf{H}_c(t)}{\Delta t}\mathbf{S} = -\frac{1}{\mu}\sum_i \mathbf{E}_i\left(t+\frac{1}{2}\Delta t\right)\mathbf{l}_i, \tag{6}$$

where Δt is the time step, the subscript c denotes the field measured at the center of the contour (output point) and the subscript i denotes the field observed at the edge centers along the contour (input points) (see Fig. 2). Eqs. (5), (6) are used to express the fields at the next time step via the fields at the previous step:

$$E_c\left(t+\frac{1}{2}\Delta t\right) = E_c\left(t-\frac{1}{2}\Delta t\right) + \frac{l_i\Delta t}{\varepsilon S}\sum_i H_i(t), \tag{7}$$

$$H_c(t+\Delta t) = H_c(t) + \frac{l_i\Delta t}{\mu S}\sum_i E_i\left(t+\frac{1}{2}\Delta t\right). \tag{8}$$

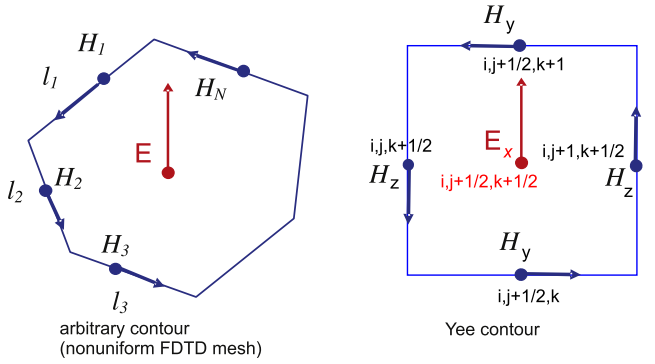

Fig. 2. **H**-contour used to update **E** field in its center. Direction of the electric field **E** is assumed to be perpendicular to the contour plane.

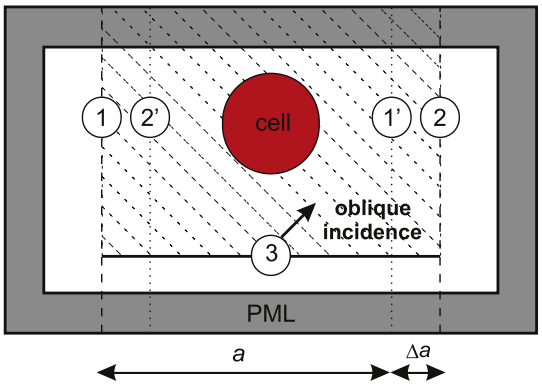

Fig. 3. Iterative method geometry. 1, 2—generating (TF/SF) borders where transferred fields are taken as a source; 1′, 2′—corresponding image locations; 3—TF/SF border with analytic oblique source.

Eqs. (7), (8) are written for the projection of the field to the contour normal (left-hand-side and first term of the right-hand-side) and to the contour edges (last term of the right-hand-side).

In the most common FDTD formulation given by Yee [3], **E**- and **H**-contours are aligned with a uniform rectangular mesh and shifted at the half of the mesh step. In this case the relations (7), (8) can be simplified to the standard Yee's discretization [1,3]. For example, discretization of the Ampere's law (7) for **H**-contour parallel to the *yz*-plane takes the form

$$E_x|^{n+1/2}_{i,j+1/2,k+1/2} = E_x|^{n-1/2}_{i,j+1/2,k+1/2} + \Delta t \frac{H_z|^n_{i,j+1,k+1/2} - H_z|^n_{i,j,k+1/2} + H_y|^n_{i,j+1/2,k} - H_y|^n_{i,j+1/2,k+1}}{\varepsilon_{i,j+1/2,k+1/2}\Delta},$$

where Δ is the mesh space step, upper index corresponds to the time step ($t = n\Delta t$) and bottom indices i, j, k correspond to the field component location at the Yee mesh. Formally, one does not need to refer to contours when working with the Yee mesh, however we still use the contour notation since it is useful in understanding such FDTD concepts as TF/SF or boundary conditions.

Relations (7), (8) constitute the basic explicit FDTD update procedure (Yee discretization scheme for the special case of Yee mesh), where the fields at the output point of the contour are updated using the fields at the input points of the contour. We use the terms "input" and "output" for the locations of the contour to underline their role in the update procedure.

One of the ways to generate an incident electromagnetic wave in FDTD is applying the Total field/Scattered field (TF/SF) technique [1]. Within the TF/SF framework, one divides the computational volume into the total field (TF) and scattered field (SF) regions (at Fig. 1 the TF region is marked by dashed lines). The update equations in the SF zone are solved for the scattered field which is a difference between the total field and the known unscattered incident field. The TF zone contains scattering objects and the update equations for the total field are solved there. The update equations have exactly the same form (7), (8) in both zones, except for the places where the update contour passes the TF/SF boundary. The output point and the input points of such a contour may be classified to be TF or SF points according to their positions. If the update involves input points of the type different from the type of the output point, then a corresponding correction (incident field at the input point) is added or subtracted:

$$E\left(t+\frac{1}{2}\Delta t\right) = E_{\text{bulk}}\left(t+\frac{1}{2}\Delta t\right) \pm \frac{l_i \Delta t}{\varepsilon S}\sum_{\text{diff. zone}} H^i_{\text{inc}}(t), \quad (9)$$

$$H\left(t+\frac{1}{2}\Delta t\right) = H_{\text{bulk}}\left(t+\frac{1}{2}\Delta t\right) \pm \frac{l_i \Delta t}{\mu S}\sum_{\text{diff. zone}} E^i_{\text{inc}}(t), \quad (10)$$

where $E_{\text{bulk}}, H_{\text{bulk}}$ is the result of basic update procedure at the right hand of (7), (8) (Yee discretization scheme for the case of the Yee mesh), $E_{\text{inc}}, H_{\text{inc}}$ is known incident field, "+" sign is used when the output point belongs to the TF zone and "−" otherwise.

The TF/SF technique is commonly used to simulate periodic structures for the normal incidence case. Typical scheme assumes generation of a plane wave impulse at the TF/SF border and applying periodic boundary conditions (1) (see Fig. 1). In this scheme the incident wave $H^i_{\text{inc}}(t)$, $E^j_{\text{inc}}(t)$ is analytically known in both time and space. As we discussed before, direct application of the periodic boundary conditions (2) for the oblique incidence case is a problem since advanced fields are unknown.

In our method, instead of using the periodic boundary conditions, we use a TF/SF correction (9), (10) at the border of the unit cell. The corresponding geometry is shown in Fig. 3. For simplicity we assume that the plane of incidence is parallel to one of the lattice translation vectors, so we can keep the regular periodic boundary conditions in this direction (which is perpendicular to the figure plane). Border 3 represents the standard TF/SF boundary that generates the known obliquely incident plane wave. We also apply TF/SF-like wave generation at the side borders $\vec{x}_1$ and $\vec{x}_2$ using the time-shifted field evolution obtained from the image points at opposite borders $\vec{x}_{1'} = \vec{x}_1 + \vec{a}$ and $\vec{x}_{2'} = \vec{x}_2 - \vec{a}$. As it is already mentioned, our method implies performing several FDTD experiments (iterations). For the negative time shift (field value from the past), fields from the current iteration q may be used. For the time-advanced fields we use the recorded evolution from the previous iteration:

$$\mathbf{F}_q(\mathbf{x}_2, t) = \mathbf{F}_q(\mathbf{x}_{2'}, t - \delta t), \quad (11)$$

$$\mathbf{F}_q(\mathbf{x}_1, t) = \mathbf{F}_{q-1}(\mathbf{x}_{1'}, t + \delta t), \quad (12)$$

where δt is defined in (3), and subscript q is used for the iteration number. Note that the distance between borders 1 and 2 is taken greater than the period a by some span Δa of several mesh steps Δ to separate image points and the TF/SF borders. The purpose of this is to have a clear and independent definition of all TF/SF zones for all kinds of the applied field corrections. The borders 1, 2 and 3 differ only by field $\mathbf{F}_{\text{inc}}$ used to define the correction term for (9), (10). For border 3, $\mathbf{F}_{\text{inc}}$ is the obliquely incident plane wave (known analytically), for borders 1 and 2 $\mathbf{F}_{\text{inc}}$ is the field recorded at the corresponding image points during the current (if the time shift is negative) or previous (if the time shift is positive) iteration.

The iterative process is illustrated in Fig. 4. One can see that as soon as the final self-consistent solution is obtained at some iteration q, the scattered fields both to the left of TF/SF border 1 and to the right of TF/SF border 2 vanish. Therefore these borders act as periodic boundary conditions (2) with the proper time shifts. As a result, field evolution does not change at the each next iteration $\mathbf{E}_{q+i}(\mathbf{r}, t) = \mathbf{E}_q(\mathbf{r}, t)$, $\mathbf{H}_{q+i}(\mathbf{r}, t) = \mathbf{H}_q(\mathbf{r}, t)$, $i > 0$, which means that simulation can be stopped.

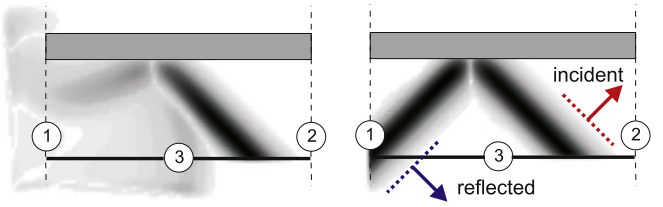

Fig. 4. The first and the fifth iteration of the experiment with oblique incidence on a metallic plate, $\theta = 45°$. The incident wave is a planar pulse of a finite duration. We use Gaussian function: $E_{\text{inc}}(\mathbf{r}, t) = G(t - \frac{\mathbf{k}\cdot\mathbf{r}}{c}), G(t) = \exp\left[-\left(\frac{t-t_0}{t_w}\right)^2\right]$, where $\mathbf{r}$ and t are coordinates in space and time, $\mathbf{k}$ is direction of the incident pulse propagation, c is speed of light, t_0 and t_w are some parameters. The incident and reflected waves are clearly seen on the fifth iteration: the incoming incident wave propagates from the bottom left to the top right, the reflected wave propagates from the top left to the bottom right. Arrows correspond to wave propagation, dotted lines are parallel to the wavefront. Incident wave is generated by TF/SF border 3 (it obviously does not propagate below this border). Wave reflected by metal plate propagates below TF/SF border 3 to the Scattered Field region. While solution is found, our technique acts as periodic boundary conditions (with time delay) applied at the borders 1, 2. Therefore, both waves (incident and reflected) do not propagate outside these borders.

The video illustration of our technique is available online [31] as a movie showing a sequence of iterations in simulation of a photonic crystal slab at oblique incidence.

3. Implementation

Typically, the FDTD simulation consists of two stages: initialization stage, where coefficients for the update equations are specified according to the calculated geometry (structure, sources, boundary conditions), and calculation stage. Below we describe the way our method can be implemented at these two stages.

We distinguish between the basic FDTD contour update procedure, which we call the "bulk" update and the special updates (contours inside PMLs, contours passing TF/SF boundaries, etc.). The special updates may be formulated as corrections or *fixes*, applied *additionally* to the basic update. The reason to this distinction is that the number of contours (mesh points) participating in fixes is usually small compared to the total number of contours, so it is numerically more efficient to group similar calculations together into fix-specific loops. Also, because of the linearity of Maxwell's equations, most of the fixes turn out to be independent of each other and may be placed in the structurally independent parts of code. An example of such are the TF/SF and "oblique" fixes discussed below. This "fix" principle has been followed by in the parallel C++ Electromagnetic Template Library (EMTL) code [32,33] where the iterative oblique technique was first implemented. In this work our aim is to describe the implementation of the method in a code-independent way. In this context the term "fix" should be understood as a procedure, applied to a selected set of mesh points before (or after) the normal time step is done. This way any existing FDTD code may be modified to implement the iterative oblique method, provided the programmer has direct access to the mesh and knows how the memory indices relate to the spatial locations of the field components stored in the mesh memory.

3.1. Initialization

At the geometry setup stage special contours requiring correction (9)–(12) are found and a selected set of points for a "fix" is formed. This set holds recorded data needed for efficiently applying corrections at every FDTD time step. For the TF/SF and oblique fixes this can be done by the following procedure. Points (output or input) forming each contour are marked with two flags:

(a) "TF/SF" flag which is turned on if a point is located in the TF zone (in Fig. 3 it lies above the generating TS/SF border 3),

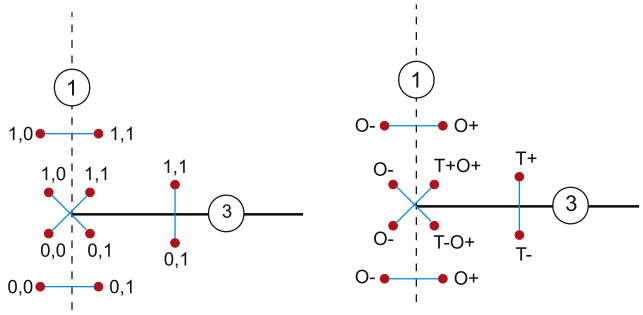

Fig. 5. Zoomed part of bottom-left side of Fig. 3. Pairs of points that may form a contour (filled circles) are connected by lines. *Left*, each point is labeled by two numbers. The first number corresponds to the TF/SF flag, the second number corresponds to the oblique flag (1—on, 0—off). *Right*, each point is labeled by sequence of symbols that specify the correction which is applied if this point is an output point, and adjacent point in the pair is an input point. T corresponds to TF/SF correction and O corresponds to oblique correction. Plus or minus sign $\pm$ is the one used in (9), (10). Both corrections are applied to output points in periodic cell which are connected to adjacent input points by diagonal line at this picture. However, in the case of the Yee mesh, TF/SF borders are aligned with mesh directions, and there are no "diagonal" lines in the contours.

(b) "oblique" flag which is turned on if a point lies inside the periodic cell (between borders 1 and 2 in Fig. 3).

If an output and an input points differ by the TF/SF flag and the oblique flag of the output point is turned on, then the TF/SF correction is applied. If an output and an input point differ by the oblique flag, then the oblique correction is applied, irrespectively of applying or not applying the TF/SF correction. The procedure used to specify which correction should be applied, is illustrated in Fig. 5. The conventional TF/SF correction is applied in its usual way and we are not considering its "fix" procedure in details here.

In the case of oblique correction one should record the following data for each input point having different "oblique" flag with respect to the output point of the contour:

(i) the index in the mesh array where the output point field value is stored,

(ii) coefficients $\frac{l_i \Delta t}{\varepsilon S}$ in (9), (10),

(iii) the image point position.

Note that the image point position may be recorded in form of a reference to an auxiliary memory buffer (discussed below) related to the image point of interest.

3.2. Calculation

The key difference between the conventional TF/SF and the oblique TF/SF correction is the usage of the time-shifted boundary conditions in the form (11), (12) at the TF/SF oblique borders before applying the correction (9), (10) itself. For this purpose it is necessary to have access to the time-shifted fields obtained at image borders 1′ and 2′ during the calculation. This can be implemented via auxiliary memory buffers for each type of fields ($\mathbf{E}$ and $\mathbf{H}$), responsible for storing the retarded fields at image border 2′ from the *current* oblique iteration (11) and the advanced fields at image border 1′ from the *previous* iteration (12). We will refer to these buffers as *c-buffer* and *p-buffer* respectively. The buffers collect field values at the image points, and subsequently are used to apply the oblique fix (9)–(12). For each type of field ($\mathbf{E}$ or $\mathbf{H}$) and time shift $\pm\delta t = \pm a \sin\theta / c$, see (3), separate buffers are allocated.

The memory for the buffers is allocated before the calculation stage starts. Each buffer stores a certain part of the signal history for its respective image border. Therefore it is convenient to organize the buffers as a time-ordered sequence of space slices. The size of each space slice is equal to $N = N_y N_z$, where N_y and N_z are number of the mesh points along the y and z direction respectively. The total number n_c of space slices in the c-buffer should only be

large enough to store the fields evolution from image border $2'$ in a relatively short time interval between $t - \delta t$ and t, whereas the total number n_p of space slices in the p-buffer should allow for the storage of the fields evolution from image border $1'$ throughout most of the calculation at a given oblique iteration. Thus, the p-buffer requires considerably more memory than the c-buffer.

Let us now determine the numbers n_c and n_p. First of all, we should define how the time shifts are represented on the time grid. Let k be the number of the current time step, corresponding to the current time moment $t_k = k\Delta t$. At each oblique iteration, the calculation consists of $n + 1$ time steps, enumerated from $k = 0$ to $k = n$, where $n\Delta t = T$ is the duration of the calculation. We define an integer number $m \geq 0$, such that $\delta t = (m + \tau)\Delta t$, where $0 \leq \tau < 1$. Then the field value at $t \pm \delta t$ can be taken as a linear interpolation between $t_{k\pm m}$ and $t_{k\pm(m+1)}$ at the time mesh. Then (11), (12) can be rewritten in the following way:

$$\mathbf{F}_q(\mathbf{x}_2, t_k) = (1 - \tau)\mathbf{F}_q(\mathbf{x}_{2'}, t_{k-m}) + \tau\mathbf{F}_q(\mathbf{x}_{2'}, t_{k-m-1}), \quad (13)$$

$$\mathbf{F}_q(\mathbf{x}_1, t_k) = (1 - \tau)\mathbf{F}_{q-1}(\mathbf{x}_{1'}, t_{k+m}) + \tau\mathbf{F}_{q-1}(\mathbf{x}_{1'}, t_{k+m+1}). \quad (14)$$

We should note here, that the fields at the image points are assumed to be zero in the case $t < 0$ or $t > T$. Thus, the number of space slices in the c-buffer $n_c = \tilde{m} + 1$, where $\tilde{m} = m$ ($\tilde{m} = m + 1$) for $\tau = 0$ ($0 < \tau < 1$). The number of space slices in the p-buffer is chosen equal to the total number of time steps $n_p = n + 1$. The reason for this choice will be clarified later. The space slices in the c- (p-)buffer are enumerated from 0 to $n_c - 1$ ($n_p - 1$).

The access to the buffers is organized in such a way, that the field values that have been used, are overwritten with the new values from the current iteration, which will be used later.

The c-buffer is only concerned with the current oblique iteration (see (13)) and should always be reset to zero before each next iteration starts. Let $i_0 = 0$ be the starting index in the buffer. Let $j = i_0 + k$ indicate the current space slice in the buffer, where the field values at border $2'$ from the current kth time step are written. It is convenient to represent the buffer as a circle (Fig. 6), with $(j - l)$th space slice corresponding to the time moment t_{k-l}, where $l = 0, 1, \ldots, \tilde{m}$, and $(j - l)$ is taken modulo n_c. Thus, we have

$$\begin{aligned}(j - \tilde{m}) \bmod n_c &= (j - \tilde{m}) \bmod (\tilde{m} + 1) \\ &= (j + 1) \bmod n_c, \end{aligned} \quad (15)$$

$$\begin{aligned}\big(j - (\tilde{m} - 1)\big) \bmod n_c &= \big(j - (\tilde{m} - 1)\big) \bmod (\tilde{m} + 1) \\ &= (j + 2) \bmod n_c. \end{aligned} \quad (16)$$

So the time-shifted fields at $t_{k-\tilde{m}}$ and $t_{k-(\tilde{m}-1)}$ appear in $(j + 1) \bmod n_c$ and $(j + 2) \bmod n_c$ space slices respectively. Now the algorithm for the c-buffer at the kth time step can be written in the following way:

(a) the current fields $\mathbf{F}_q(\mathbf{x}_{2'}, t_k)$ are written to the jth space slice in the c-buffer (where $j = k \bmod n_c$ can be visualized as rotating around the circle);

(b) the basic update procedure (7), (8) is performed;

(c) the retarded fields $\mathbf{F}_q(\mathbf{x}_{2'}, t_{k-\tilde{m}})$ and $\mathbf{F}_q(\mathbf{x}_{2'}, t_{k-(\tilde{m}-1)})$ are read from $(j + 1) \bmod n_c$ and $(j + 2) \bmod n_c$ space slices of the c-puffer and used to found interpolated value of $\mathbf{F}_q(\mathbf{x}_2, t_k)$ according to (13) (note that $k - \tilde{m} = k - m$ if $\tau = 0$ and $k - \tilde{m} = k - m - 1$ if $0 < \tau < 1$);

(d) the oblique correction (9), (10) is applied.

Note, that in this algorithm, once the retarded field $\mathbf{F}_q(\mathbf{x}_{2'}, t_{k-\tilde{m}})$ has been read from the buffer, it is overwritten with the current field $\mathbf{F}_q(\mathbf{x}_{2'}, t_{k+1})$ at the next time step. Also, the condition, that the field values at $t < 0$ be zero, is automatically fulfilled, as the buffer is reset to zero before each next oblique iteration. In case of the normal incidence the algorithm is reduced to applying conventional periodic boundary conditions (the time shift $\delta t = 0$ in (3) and $n_c = 1$).

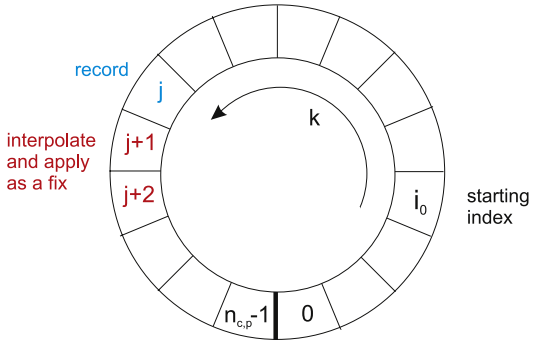

Fig. 6. The buffer as a sequence of space slices, $n_{c,p}$—their number, i_0 is the starting index. At each time step k, field is recorded to j slice, $j = (i_0 + k) \bmod n_{c,p}$. Slices $j + 1$ and $j + 2$ are used to interpolate field value at time $t_k \pm \delta t$ according to (13), (14). Interpolated field is applied in oblique fix (9), (10).

The p-buffer is concerned with the previous oblique iteration and should only be set to zero before the start of the first iteration. Similarly to the c-buffer, we use the circle geometry for the p-buffer. At the current time step k the current fields at the image border $1'$ are written to a $j = i_0 + k$ space slice, while the fields recorded during the previous oblique iteration are read from $(j + 1) \bmod n_p$ and $(j+2) \bmod n_p$ space slices. The difference consists in shifting the starting index i_0 by $m - 1$ positions counter-clockwise after each oblique iteration, as shown in Fig. 6. Such shift ensures that $(j + 1) \bmod n_p$ and $(j + 2) \bmod n_p$ space slices correspond to the advanced fields at t_{k+m} and t_{k+m+1} from the previous oblique iteration, as $j+1 = j-(m-1)+m$ and $j+2 = j-(m-1)+m+1$. Thus, the algorithm for the p-buffer at each oblique iteration can be formulated as described below:

The starting index $i_0 = 0$ before the first oblique iteration $q = 1$;

At qth oblique iteration, at kth time step:

(a) the current fields $\mathbf{F}_q(\mathbf{x}_{1'}, t_k)$ are written to the jth space slice in the p-buffer (where $j = (i_0 + k) \bmod n_p$);

(b) the basic update procedure (7), (8) is performed;

(c) if $k \leq n - m$, the advanced fields $\mathbf{F}_{q-1}(\mathbf{x}_{1'}, t_{k+m})$ and $\mathbf{F}_{q-1}(\mathbf{x}_{1'}, t_{k+m+1})$ are read from $(j + 1) \bmod n_p$ and $(j + 2) \bmod n_p$ space slices of the p-buffer and used to found interpolated value of $\mathbf{F}_q(\mathbf{x}_1, t_k)$ according to (14), otherwise they are assumed to be zero;

(d) the oblique correction (9), (10) is applied.

At the end of each oblique iteration starting index is shifted $i_0 = (m - 1)(q - 1) \bmod n_p$.

4. Tests

We implemented the iterative method as a part of the parallel C++ Electromagnetic Template Library (EMTL) [32,33]. To test our method we calculate reflectance from different periodic structures. To reduce the error caused by the staircasing effects, the subpixel smoothing technique is applied in all cases considered [34]. We use Convolution PML formulation [1] since it allows for submerging a dispersive medium such as metal. To reduce undesired numerical reflection from the PMLs we use additional back absorbing layers technique [35]. We use incident plane wave impulse in a form $(t - t_0) \exp\left[-\left(\frac{t-t_0}{t_w}\right)^2\right]$. Parameters t_0 and t_w are chosen to cover considered spectral range by incident impulse.

4.1. Plasmon formation in a thin gold film

In the first test we consider a thin gold film (width $d = 50$ nm) between air and glass ($n = 1.5$) semi-infinite interfaces. We

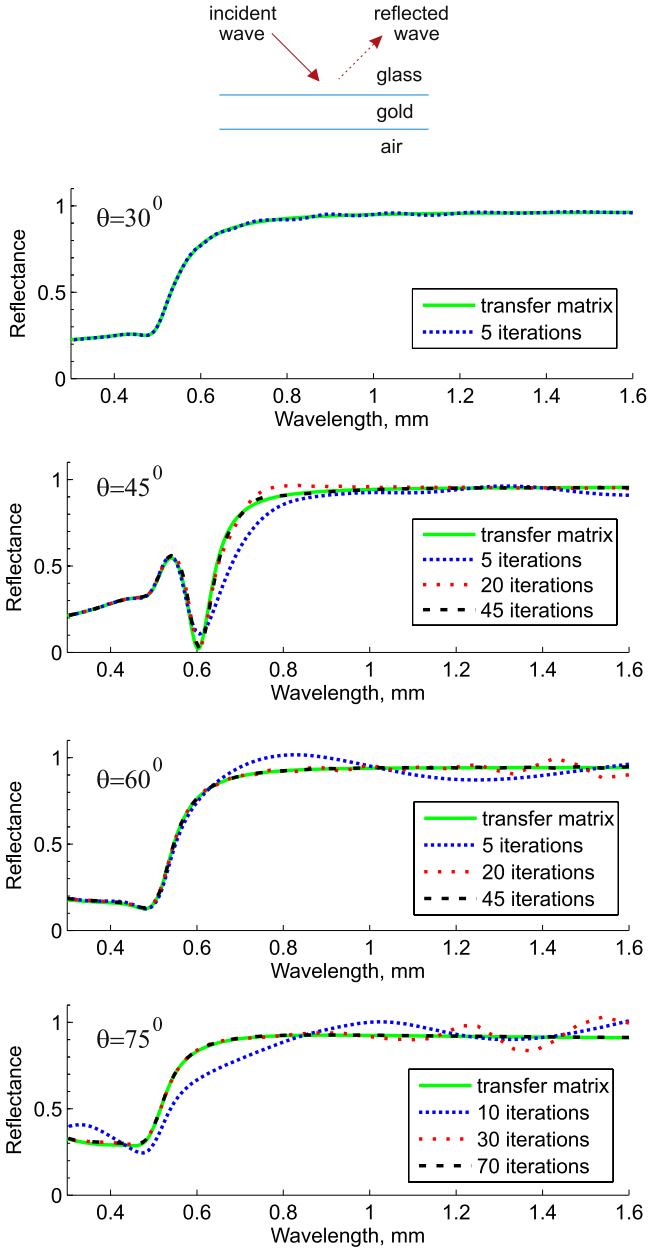

Fig. 7. Reflectance from a thin gold film (width $d = 50$ nm) between air and glass ($n = 1.5$) semi-infinite interfaces. Different angles of incidence θ are considered, p-polarization. We model gold film as a periodic structure with period $a = 2$ μm (periodicity direction is formed by intersection of gold surface and incidence plane). Subpixel smoothing method [34] is used to improve accuracy of calculation.

model gold film as a periodic structure with period $a = 0.4$ μm (periodicity direction is formed by intersection of gold surface and incidence plane). Experimental data on the gold dielectric permittivity $\varepsilon(\omega)$ is taken from [36,37]. The frequency dependence of $\varepsilon(\omega)$ is assigned in FDTD by considering a modified Lorentz approximation where the dielectric polarization depends both on the electric field and its first time derivative [38]:

$$\varepsilon(\omega) = \varepsilon_\infty - \frac{\omega_D^2}{\omega^2 + i\omega\gamma_D} + \sum_{p=1}^{2} \frac{\Delta\varepsilon_p(\omega_p^2 - i\gamma_p'\omega)}{\omega_p^2 - 2i\omega\gamma_p - \omega^2} \tag{17}$$

with (ω_D, γ_D, ω_p, γ_p and γ_p' are in $1/\mu$m, and the speed of light is unity): $\varepsilon_\infty = 1.14$, $\omega_D = 7$, $\gamma_D = 0.057$, $\Delta\varepsilon_1 = 0.23$, $\Delta\varepsilon_2 = 4.48$, $\omega_1 = 2.07$, $\omega_2 = 2.54$, $\gamma_1 = 0.237$, $\gamma_2 = 1.25$, $\gamma_1' = 4.5$, $\gamma_2' = 2.71$. Modified Lorentz approximation is implemented in FDTD using auxiliary differential equation (ADE) technique [38].

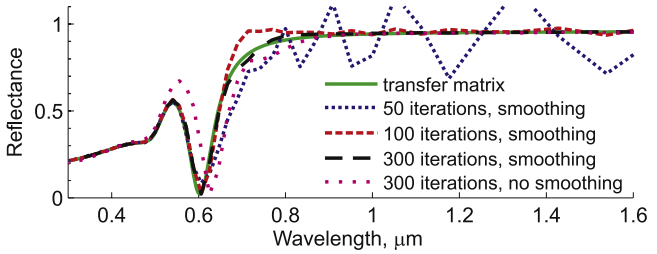

Fig. 8. The same as Fig. 8, but period $a = 0.4$ μm. Angle of incidence $\theta = 45°$, p-polarization. Subpixel smoothing method [34] is used to improve accuracy of calculation.

The considered structure represents a standard Kretschmann configuration used for the surface plasmon excitation [39,40]. In Fig. 7, we present comparison between reflectance calculated by our iterative technique and the classical transfer matrix formalism [41]. Note that our technique remains stable for arbitrary angles (Fig. 7), and results for lower angles converge faster. Surface plasmon resonance manifests itself as a dip in the reflectance spectrum when the p-polarized light is incident from optically thick interface (glass) at a particular angle ($\theta = 45°$).

As discussed in Section 5, convergence can be improved by using larger period a. At the same time, the usage of smaller period a can reduce memory requirements but slows down convergence. For example, simulating period $a = 0.4$ μm for $\theta = 45°$ requires more iterations for convergence (compare Figs. 7 and 8). Note that in our test case, the period a for the simulation can be chosen arbitrarily, since the gold film is homogeneous in lateral direction. If there is a real periodicity (e.g. photonic crystal slab), a could be chosen as a single period of the structure, or as a multiple number of periods.

Using subpixel smoothing technique [34] greatly improves the accuracy of our calculations compared to the staircase model (we use mesh step $\Delta = 5$ nm). Results obtained without subpixel smoothing (staircase model) produce the wrong reflection minimum position. This is connected with the fact that the incident p-polarized electric field is not aligned with the gold interface (Fig. 8). To improve the accuracy subpixel smoothing should be used which takes the exact location of the interface boundaries for both magnetic and electric fields into account.

4.2. Silicon textured antireflective coating

In the second test we calculate reflectance from a silicon textured antireflective coating [42–44] at oblique incidence (angle of incidence $\theta = 45°$, s-polarization). Parameters for the coating are taken from [42]. In this work antireflective properties of textured coatings were investigated for a wide range of size-to-wavelengths ratios including long- and short-wave limits (where effective medium and geometric optics approximations are used along with FDTD) to find the optimal texture size. Simulated structure consists of an infinite silicon slab with silicon cones on the top of it. The cones of the radius $R = 0.15$ μm are arranged in a close-packed triangular lattice with the period $\Lambda = 0.3$ μm. The height of each cone is $d = 0.5$ μm. Triangular lattice translation vector is parallel to the incidence plane. The mesh step used in the calculation is $\Delta = 10$ nm. A scheme of the FDTD simulation geometry is shown in Fig. 9. Note that we extend the substrate inside the left and right PMLs which is different from Fig. 4 where the plate is limited by TF/SF oblique borders. As we discuss in Section 5, extension of the structure beyond oblique TF/SF borders does not affect the final results but may improve convergence speed.

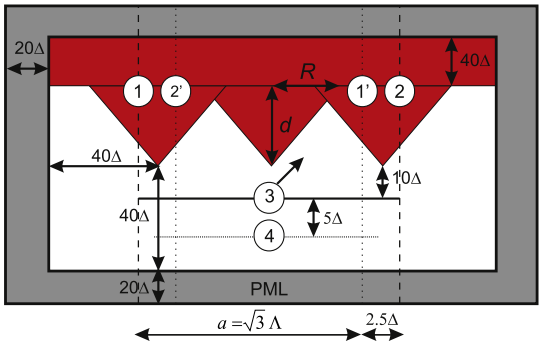

Fig. 9. Iterative method geometry for simulation of silicon textured antireflective coating at oblique incidence (relative sizes are not preserved). The cones of the radius $R = 0.15$ μm are arranged in a close-packed triangular lattice with the period $\Lambda = 0.3$ μm. The height of each cone is $d = 0.5$ μm. Triangular lattice translation vector is parallel to the incidence plane. The mesh step used in the calculation $\Delta = 10$ nm. 1, 2—generating (TF/SF) borders where transferred fields are taken as a source; 1′, 2′—corresponding image locations; 3—TF/SF border with analytic oblique source; 4 – Position where reflected field is recorded.

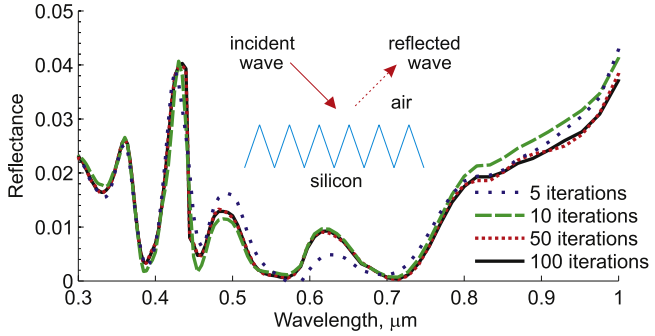

Fig. 10. Reflectance from antireflective textured coating formed by silicon cones of the radius $R = 0.15$ μm and height $d = 0.5$ μm on the top of the silicon substrate. Cones are arranged in a close-packed triangular lattice with the period $\Lambda = 0.3$ μm. Angle of incidence $\theta = 45°$, s-polarization.

Experimental data on the silicon dielectric permittivity $\varepsilon(\omega)$ is taken from [45]. Fitting of the gold dielectric function is taken from [38] in the form

$$\varepsilon(\omega) = \varepsilon_\infty + \sum_{p=1}^{2} \frac{\Delta\varepsilon_p(\omega_p^2 - i\gamma_p'\omega)}{\omega_p^2 - 2i\omega\gamma_p - \omega^2} \tag{18}$$

with (ω_p, γ_p and γ_p' are in 1/μm, and the speed of light is unity): $\varepsilon_\infty = 1$, $\Delta\varepsilon_1 = 8.93$, $\Delta\varepsilon_2 = 1.855$, $\omega_1 = 3.42$ (corresponding to wavelength $\lambda_1 \approx 0.292$ μm), $\omega_2 = 2.72$ ($\lambda_2 \approx 0.368$ μm), $\gamma_1 = 0.425$, $\gamma_2 = 0.123$, $\gamma_1' = 0.087$, $\gamma_2' = 2.678$. This model provides an accurate fit to response of bulk crystalline silicon to sunlight over the wavelength range from 300 to 1000 nm, while conventional Debye, Drude and Lorentz approximations fail. Fitting of silicon dielectric function is found with the help of the MatLab program [46] provided for open access.

Calculated reflectance from the textured coating is shown in Fig. 10. One can see that results converge after 10–50 iterations, and the reflectance for smaller wavelengths converges faster.

5. Convergence

Numerical examples from Section 4 demonstrate that calculated spectrum converges to the solution after some number of iterations. Final solution does not change at the each next iteration. This infers a stopping criterion for our technique: simulation can be stopped, if the field does not change at the each next iteration (difference between fields at the current and next iterations is negligible, within machine accuracy). Obviously, there is no need to apply this criterion for field at the each mesh point. One can observe change of the field evolution at some chosen point (Fig. 11) and stop the simulation while it does not change at the each next iteration. Note that when convergence is obtained, field does not change at the each next iteration.

We can represent the field at each qth iteration as a sum $\mathbf{F}_q = \mathbf{F}_{\text{conv}} + \delta\mathbf{F}_q$, where $\mathbf{F}_{\text{conv}}(t)$ is converged solution, $\delta\mathbf{F}_q$ is some error signal. In our previous paper [30] we showed that the error signal $\delta\mathbf{F}_q$ at each next iteration shifts to the future by time interval proportional to a/c (see Fig. 3 and corresponding comments at Ref. [30]):

$$\delta\mathbf{F}_{q+1}(t) \approx \delta\mathbf{F}_q(t - \tau), \tag{19}$$

where $\tau \sim a/c$ and a is the period of the considered unit cell. We do not use equality sign in (19), since this is just estimation, and shape of the function $\delta\mathbf{F}_q(t)$ changes as well. Obviously,

$$\delta\mathbf{F}_q(t) = 0, \quad t < 0. \tag{20}$$

Therefore, nonzero part of the function $\delta\mathbf{F}_q(t)$ if shifted to the future at the each time iteration. This is illustrated in Fig. 11: one can see no difference between 30 and 40 iteration, thus, $\delta\mathbf{F}_q(t) = 0$ for $q > 30$ within the given time range.

Let us estimate iterations number n_{iter}, necessary for the solution and the error signal to be completely separated within time of the FDTD experiment $t < T$:

$$\delta\mathbf{F}_q(t) = 0, \qquad \mathbf{F}_q(t) = \mathbf{F}_{\text{conv}}(t), \quad q \geq n_{\text{iter}}. \tag{21}$$

As follows from (19), (20), n_{iter} is proportional to cT/a. Field does not change at the each next iteration such as $q \geq n_{\text{iter}}$, and converged solution is found $\mathbf{F}_q(t) = \mathbf{F}_{\text{conv}}(t)$ (within time $t < T$).

From our experience, n_{iter} is around several times of the value of cT/a. For example, in our first numerical test from Section 4.1, $cT = 10$ μm, $a = 2$ μm, $cT/a = 5$, and n_{iter} varies from 5 to 70 (approximately) for angles $30° \leq \theta \leq 75°$ (see Fig. 7). In our second numerical test from Section 4.2, $cT = 20$ μm, $a = \sqrt{3} \cdot 0.3$ μm, $cT/a \approx 40$, and $n_{\text{iter}} \approx 100$ for $\theta = 45°$ (see Fig. 10).

If the incident light is efficiently absorbed in the structure, the value n_{iter} can be small enough (5–10) since the error signal amplitude decays at each next iteration [30]. Note that the spectrum of the error signal $\delta\mathbf{F}_q$ usually covers longer wavelengths [30], therefore convergence for these wavelengths is slower (see Figs. 7 and 10).

In general, the time duration of the FDTD experiment T should be chosen to be sufficient for scattered waves to leave the structure and be absorbed in PMLs. If the structure is not absorptive and its size in longitudinal direction is large compared to the characteristic incident wavelength, this time T and therefore the value of n_{iter} are large. However, using several periods of the structure as one FDTD cell can improve the convergence, since n_{iter} is inversely proportional to the FDTD cell lateral size a (see corresponding discussion for the first numerical test from Section 4.1 and Figs. 7 and 8).

Extension of the structure beyond oblique TF/SF borders towards PML (as in Fig. 9) does not affect the results but may improve convergence speed. When convergence is reached, the structure beyond oblique TF/SF borders does not play any role, since these borders act as the (time–space) periodic boundaries, and we are ultimately interested in the fields inside these borders (simulated region). The main role of the structure outside the oblique TF/SF borders (scattered field zone) is minimizing spurious reflection in the direction towards the simulated region (total field zone). In some cases, convergence of the oblique iterations can be improved when the scattered field zone contains the nearest periodic image of the simulated cell (terminated by

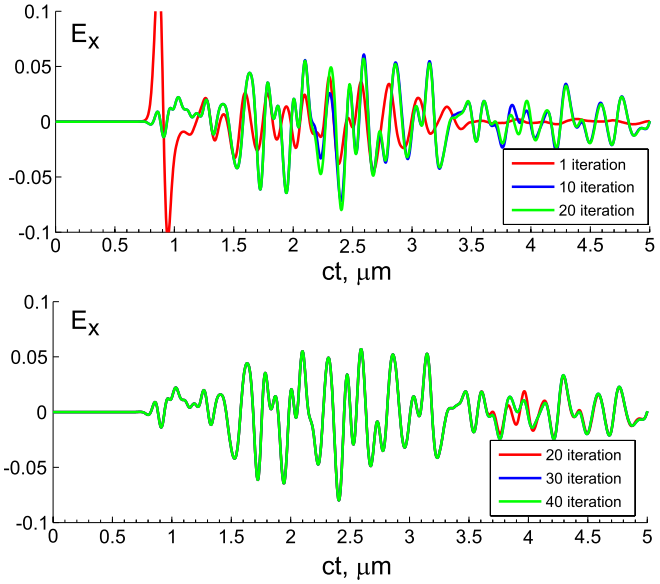

Fig. 11. Evolution of reflected electric field (component parallel to incident polarization) for the second numerical test from Section 4.2, see Figs. 9 and 10. Field is recorded at the to middle of the line 4, Fig. 9. One can see that field evolution does not change after 30 iterations within presented time range.

PMLs) since then the reflection from the nearest image acts as a good approximation for the fields on the TF/SF oblique borders. Generally convergence can be achieved independently on the structure beyond oblique TF/SF borders, even if PMLs in the lateral direction are absent. However, in the latter case the reflecting walls should be placed far enough from the oblique TF/SF borders and convergence is rather slow, so the usage of lateral PMLs is preferred for our technique.

We observed that in some cases the error signal amplitude may increase at each next iteration. This problem could be related to the long time instability in PML layers, arising in some FDTD setups with mixed periodic/absorbing boundary conditions [35]. It can be remedied to some extent by using additional back absorbing layers technique [35] or applying larger distance between the structure and PMLs. For example, this distance is taken to be 40 mesh steps for our second numerical test (Fig. 9), while using smaller distance could lead to instability during simulation.

Lastly, according to our experience, in some cases using the retarded fields from the current iteration may result in instability of the calculation. If this is the case, the c-buffer cannot be used and the p-buffer should be used instead to take the retarded time shift from the previous iteration as well. This is the reason for storing the full signal history in the p-buffer ($n_p = n + 1$), so that one can take retarded time shift ($\delta t \leq 0$) from the previous iteration using the same algorithm, as for the advanced fields. In this case i_0 decreases after each iteration (since $m - 1 < 0$) and circles clockwise around the buffer (Fig. 6).

6. Performance

To estimate the computer memory resources required for our technique, we compared it with the normal incidence calculation (see Table 1). We simulated the antireflective textured coating of the same geometry as in Section 4.2 (Fig. 9). The mesh step $\Delta = 10$ nm, the total number of time steps $n = 8000$. The mesh for the oblique incidence case includes CPML and additional space at the light propagation direction (compare Figs. 1 and 3 or Fig. 9). One time step for the oblique incidence case is two times slower than that for the normal incidence case, mostly because of the time used to update extended mesh in oblique incidence setup (p-buffers and c-buffers consume only 5% of the CPU time). The total mesh memory size for the normal incidence cases is 20 Mb, while that for the oblique incidence case is 45 Mb (including memory for CPML and ADE technique for dispersive media). The p-buffers and c-buffers (for **E** and **H** fields) require 800 Mb and 8 Mb correspondingly. We see that, that p-buffer size is much larger than the mesh memory size. However, since the data flow per FDTD step from the buffer is not very intensive, disk storage may be efficiently utilized for buffering.

Table 1
CPU and memory recourses for the second numerical test from Section 4.2, normal and oblique incidence cases. We assume that solution is found after 30 iterations (see Fig. 10). x is CPU time per one iteration for the case of normal incidence. This time depends on the computer architecture.

	Normal incidence	Oblique incidence
Mesh size	20 Mb	45 Mb
p- and c-buffers size	–	(800 + 8) Mb
CPU time per one iteration	x	$\approx 2x$
Number of iterations	1	≈ 30
Total calculation CPU time	x	$\approx 60x$

7. Conclusion

In this work we report an efficient implementation of the iterative FDTD technique for simulation of periodic structures at oblique incidence. This technique is free from the major drawbacks of other methods, such as stability problems in case of sharp incidence angles, lack of a direct time domain evolution, and need for large meshes. However it requires performing several numerical experiments instead of one and additional memory storage for the memory buffers.

Compared to a single numerical FDTD experiment the proposed iterative technique requires up to 50 times more computational resources (CPU and memory) to obtain a time domain solution for a single angle of incidence (for example, see Table 1). The memory requirement is not very restricting because of the limited data flow per time step from the buffer memory. In spite of these constraints, the proposed technique is useful as a method with (a) convergence at all incidence angles, (b) ability to treat dispersive and anisotropic materials, (c) direct time-dependent field solution. It may be recommended if one needs to obtain the properties of a layered structure for a small number of incidence angles or requires a time domain form for the electromagnetic field inside the structure. Note that the latter is the case when coupling FDTD to the other time domain models like in case of molecular nanopolaritonics [23,24], or plasma particle in cell [47]. The presented technique can be used together with the hybrid FDTD transfer matrix method [48] to calculate photonic band structure for periodic structures. Examples of application of our iterative technique can be found at Electromagnetic Template Library (EMTL) webpage [32].

Acknowledgments

This work is partially supported by the Programs of Fundamental Research of the Presidium of RAS Nos. 2, 13, 14 and by The Ministry of Science and Education of Russia under contract No. 16.523.11.3004.

References

[1] A. Taflove, S.H. Hagness, Computational Electrodynamics: The Finite Difference Time-Domain Method, Artech House, Boston, 2005.
[2] S.D. Gedney, Introduction to the FDTD Method for Electromagnetics, Morgan & Claypool, 2011.

[3] K.S. Yee K S, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE Trans. Antennas Propag. 14 (1966) 302.
[4] M. Veysoglu, R. Shin, J. Kong, A finite-difference time-domain analysis of wave scattering from periodic surfaces: oblique incidence case, J. Electromagn. Waves Appl. 7 (1993) 1595.
[5] Y.C.A. Kao, R.G. Atkins, A finite-difference time-domain approach for frequency selective surfaces at oblique incidence, in: Proc. 1996 IEEE Antennas and Propagation Society International Symposium, 1996, pp. 1432–1436.
[6] J.A. Roden, S.D. Gedney, M.P. Kesler, J.G. Maloney, P.H. Harms, Time-domain analysis of periodic structures at oblique incidence: orthogonal and nonorthogonal FDTD implementations, Microw. Theory Tech. 46 (1998) 420–427.
[7] G. Zheng, A.A. Kishk, A.W. Glisson, A.B. Yakovlev, A novel implementation of modified Maxwell's equations in the periodic finite-difference time-domain method, Prog. Electromagn. Res. 59 (2006) 85–100.
[8] S.M. Amjadi, M. Soleimani, Design of band-pass waveguide filter using frequency selective surfaces loaded with surface mount capacitors based on split-field update FDTD method, Progress Electromagn. Res. B 43 (2008) 271–281.
[9] J. Wang, B. Zhou, L. Shi, C. Gao, B. Chen, Efficiency-improved LOD-FDTD method for solving periodic structures at oblique incidence, IEEE Microw. Wireless Comp. Lett. 23 (2013) 521–523.
[10] Y. Wakabayashi, J. Shibayama, J. Yamauchi, H. Nakano, A locally one-dimensional finite difference time domain method for the analysis of a periodic structure at oblique incidence, Radio Sci. 46 (2011) 1–9.
[11] F.I. Baida, A. Belkhir, Split-field FDTD method for oblique incidence study of periodic dispersive metallic structures, Opt. Lett. 34 (2009) 2453–2455.
[12] A. Belkhir, O. Arar, S.S. Benabbes, O. Lamrous, F.I. Baida, Implementation of dispersion models in the split-field finite-difference-time-domain algorithm for the study of metallic periodic structures at oblique incidence, Phys. Rev. E 81 (2010) 046705.
[13] M. Hamidi, F.I. Baida, A. Belkhir, O. Lamrous, Implementation of the critical points model in a SFM-FDTD code working in oblique incidence, J. Phys. D: Appl. Phys. 44 (2011) 245101.
[14] L. Zhang, T. Seideman, Rigorous formulation of oblique incidence scattering from dispersive media, Phys. Rev. B 82 (2010) 155117.
[15] A. Shahmansouri, B. Rashidian, Comprehensive three-dimensional split-field finite-difference time-domain method for analysis of periodic plasmonic nanostructures: near- and far-field formulation, JOSA B 28 (2011) 2690–2700.
[16] C. Oh, M.J. Escuti, Time-domain analysis of periodic anisotropic media at oblique incidence: an efficient FDTD implementation, Opt. Express 14 (2006) 11870–11884.
[17] J. Frances, J. Tervo, A. Marquez, Tensorial split-field finite-difference time-domain approach for second- and third-order nonlinear materials, JOSA B 30 (2013) 1711–1719.
[18] P. Harms, R. Mittra, W. Ko, Implementation of the periodic boundary condition in the finite-difference time-domain algorithm for FSS structures, IEEE Trans. Antennas Propag. 42 (1994) 1317–1324.
[19] A. Aminian, Y. Rahmat-Samii, Spectral FDTD: a novel technique for the analysis of oblique incident plane wave on periodic structures, IEEE Trans. Antennas Propag. 54 (2006) 1818–1825.
[20] F. Yang, J. Chen, R. Qiang, A. Elsherbeni, A simple and ecient FDTD/PBC algorithm for periodic structure analysis, Radio Sci. 42 (2007) RS4004.
[21] Y.J. Zhou, X.Y. Zhou, T.J. Cui, Efficient simulations of periodic structures with oblique incidence using direct spectral FDTD method, PIER M 17 (2011) 101.
[22] Y.F. Mao, B. Chen, H.Q. Liu, J.L. Xia, J.Z. Tang, A hybrid implicit–explicit spectral FDTD scheme for oblique incidence problems on periodic structures, Prog. Electromagn. Res. 128 (2012) 153–170.
[23] E. Ozbay, Plasmonics: merging photonics and electronics at nanoscale dimensions, Science 311 (2006) 189–193.
[24] K. Lopata, D. Neuhauser, Multiscale Maxwell-Schrödinger modeling: a split field finite-difference time-domain approach to molecular nanopolaritonics, J. Chem. Phys. 130 (2009) 104707.
[25] A. Fratalocchi, C. Conti, G. Ruocco, Mode competitions and dynamical frequency pulling in Mie nanolasers: 3D ab-initio Maxwell–Bloch computations, Optics Express 16 (2008) 8342.
[26] S. Hellstrom, Y. Fu, Dynamic optical response of an excitonic quantum dot studied by solving the self-consistent Maxwell–Schrodinger equations nonperturbatively, Phys. Rev. B 82 (2010) 245305.
[27] A. Deinega, T. Seideman, New, self-interaction-free approaches for self-consistent solution of the Maxwell–Liouville equations, Phys. Rev. A (in press).
[28] J. Ren, O.P. Gandhi, L.R. Walker, J. Fraschilla, C.R. Boerman, Floquet-based FDTD analysis of two-dimensional phased array antennas, IEEE Microw. Guided Wave Lett. 4 (1994) 109–112.
[29] F. Le, P. Nordlander, Optical properties of nanoparticle arrays for oblique excitation using the multiple unit cell method, J. Comput. Theor. Nanosci. 6 (2009) 2031.
[30] I. Valuev, A. Deinega, S. Belousov, Iterative technique for analysis of periodic structures at oblique incidence in the finite-difference time-domain method, Opt. Lett. 33 (2008) 1491–1493.
[31] http://fdtd.kintechlab.com/en/tutorial?&#oblique_incidence.
[32] Electromagnetic Template Library, Kintech Lab Ltd, http://fdtd.kintechlab.com.
[33] I. Valuev, A. Deinega, A. Knizhnik, B. Potapkin, Creating numerically efficient FDTD simulations using generic C++ programming, Lecture Notes in Comput. Sci. 4707 (2007) 213.
[34] A. Deinega, I. Valuev, Subpixel smoothing for conductive and dispersive media in the FDTD method, Opt. Lett. 32 (2007) 3429–3431.
[35] A. Deinega, I. Valuev, Long-time behavior of PML absorbing boundaries for layered periodic structures, Comp. Phys. Comm. 182 (2011) 149–151.
[36] P. Johnson, R. Christy, Optical constants of the noble metals, Phys. Rev. B 6 (1972) 4370.
[37] A. Vial, T. Laroche, M. Dridi, L. Le Cunff, A new model of dispersion for metals leading to a more accurate modeling of plasmonic structures using the FDTD method, Appl. Phys. A 103 (2011) 849.
[38] A. Deinega, S. John, Effective optical response of silicon to sunlight in the finite-difference time-domain method, Opt. Lett. 37 (2012) 112–114.
[39] J. Homola, S.S. Yeea, Gunter Gauglitz, surface plasmon resonance sensors: review, Sensors Actuators B 54 (1999) 3–15.
[40] R.B.M. Schasfoort, A.J. Tudos, Handbook of Surface Plasmon Resonance, Roayl Society of Chemistry, Cambridge, 2008.
[41] M. Born, E. Wolf, Principles of Optics, Prgamon, London, 1980.
[42] A. Deinega, I. Valuev, B. Potapkin, Yu. Lozovik, Minimizing light reflection from dielectric textured surfaces, JOSA A 28 (2011) 770–777.
[43] D. Poxson, M. Schubert, F. Mont, E. Schubert, J.K. Kim, Broadband omnidirectional antireflection coatings optimized by genetic algorithm, Opt. Lett. 34 (2009) 728–730.
[44] B. Paivanranta, T. Saastamoinen, M. Kuittinen, A wide-angle antireflection surface for the visible spectrum, Nanotechnology 20 (2009) 375301.
[45] M.A. Green, M. Keevers, Optical properties of intrinsic silicon at 300 K, Prog. Photovolt. 3 (1995) 189–192.
[46] http://fdtd.kintechlab.com/en/fitting.
[47] D. Tskhakaya, K. Matyash, R. Schneider, F. Taccogna, The Particle-In-Cell Method, Contrib. Plasma Phys. 47 (2007) 563–594.
[48] A. Deinega, S. Belousov, I. Valuev, Transfer matrix approach for finite-difference time-domain simulation of periodic structures, Phys. Rev. E 88 (2013) 053305.