Ultra-thin boundary layer for high-accuracy simulations of light propagation

GERWIN OSNABRUGGE,* MAAIKE BENEDICTUS, AND IVO M. VELLEKOOP

Biomedical Photonic Imaging Group, Technical Medical Centre, University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands
*g.osnabrugge@utwente.nl

Abstract: The modified Born series method is currently one of the most efficient methods available for simulating light scattering in large inhomogeneous media. However, to achieve high accuracy, the method requires thick gradually absorbing layers around the simulation domain. Here, we introduce new boundary conditions, combining a padding-free acyclic convolution with an ultra-thin boundary layer. Our new boundary conditions minimize the wrap-around and reflection artefacts originating from the edges of the simulation domain, while also greatly reducing the computational costs and the memory requirements of the method. Our GPU-accelerated Matlab implementation is available on GitHub.

© 2021 Optical Society of America under the terms of the OSA Open Access Publishing Agreement

1. Introduction

Understanding how light propagates in structured media is a fundamental problem in many fields of optics. However, the analytical solution to Maxwell’s equations can often only be found for simple systems, e.g. a Mie sphere [1]. For more complex problems, the solution has to be computed using numerical methods such as finite difference time domain [2–4], finite element [5] and pseudospectral time domain [6] methods. Although widely used in nanophotonics [7], these methods are too computationally demanding for large (three-dimensional) structures. In forward-scattering media, light propagation can be simulated using the beam propagation method [8,9], where a large scattering structure is modeled as a stack of infinitely thin phase plates. This method is commonly used to study light scattering in deep tissue microscopy [10–13]. While efficient and easy to implement, the beam propagation method is limited to forward scattering media, as paraxial light propagation is assumed and the backscattered light is neglected.

A completely different category of numerical methods involves computing the Born series [14–16]. Recently, we developed a new modified Born series (MBS) method to solve the scalar wave equation [17]. We proved that our modifications guarantee convergence to the analytical solution for any inhomogeneous structures, including both isotropically and forward scattering media. We demonstrated that the MBS method is orders of magnitude faster and more accurate than state-of-the-art numerical techniques. The MBS method was later generalized to vector fields [18] and birefringent media [19].

Ideally when simulating an open system, a light wave should vanish as it exits the simulation domain. To mimic such an open system, the MBS method, like many other numerical methods [20], requires padding of the domain with an absorbing boundary layer. For computational efficiency, this layer should be as thin as possible. However, if the layer does not absorb enough energy, waves can leak through the boundary layer and re-enter the simulation domain from the opposite side (see Fig. 1(a)). This wrap-around effect is caused by the fast convolutions at the heart of the MBS algorithm. On the other hand, if the absorbing layer cuts off the electric field too abruptly, reflection artefacts arise, as shown in Fig. 1(b). Until now, a thick gradually absorbing boundary layer was needed to achieve the maximum accuracy, increasing the computational costs and memory requirements of the algorithm. This problem becomes even more pressing when
simulating open three-dimensional systems, where such an absorbing layer has to be added in all three directions.

Fig. 1. Potential errors in light simulations introduced by the boundary layers. (a) A simulation using thin weakly absorbing boundaries, where the light leaks through the edges and re-enters through the opposite side. (b) A simulation using thin strongly absorbing boundaries, introducing strong reflection artefacts. (c) Simulation results obtained using the new boundary conditions without the wrap-around and reflection artefacts.

Here, we introduce a new class of boundary conditions for the MBS method, greatly reducing the computational costs and memory requirements of the method, while suppressing the wrap-around errors and reflection artefacts (see Fig. 1(c)). The key novelty of our approach is to combine the left- and right-circular convolutions (first introduced by Radhakrishnan et al. [21]), and alternate them in a pipelined fashion. This way, wrap-around artefacts are cancelled without affecting the computation time. This padding-free acyclic convolution method is combined with a thin anti-reflection boundary layer. The performance of our new boundary conditions is evaluated in a three-dimensional test simulation.

2. Theory

2.1. Modified Born series

Here, we use the modified Born series method to solve Maxwell’s equations for a given source $S$ with a wavenumber $k_0$ in a medium with refractive index distribution $n(r)$. The standard Born series is known to diverge for large inhomogeneous structures [14]. Therefore, in previous work, we modified the Born series to guarantee convergence without affecting the solution [17]. The electric field $E$ can be obtained from the modified Born series, which is given by:

$$E = [1 + M + MM + MMM + \ldots ] \gamma GS,$$

with $M \equiv -i\epsilon \gamma G - \gamma + 1$. Here, $\epsilon$ is a damping factor, $\gamma \equiv i[n(r)^2 - 1]k_0^2/\epsilon$ is the modified scattering potential and the operator $G$ represents the convolution with the dyadic Green’s function [18].

In the MBS algorithm, we iteratively compute the terms in Eq. (1) until the desired accuracy is reached. Starting with $\hat{E}_1 = \gamma GS$, the difference field computed in $(j + 1)$th iteration is given by

$$\hat{E}_{j+1} = M\hat{E}_j.$$
The solution, $E$, is obtained from accumulation of all computed difference fields $\hat{E}$. With every iteration, the operator $M$ is applied to the difference field of the previous iteration. $M$ is composed of the operator $G$, which ‘propagates’ the field through the medium, and $\gamma$, which applies the scattering potential of the medium.

For performance reasons, the convolution with the Green’s function is implemented as a fast-convolution, with $G \equiv \mathcal{F}^{-1} \tilde{g}(k) \mathcal{F}$. Here, $\mathcal{F}$ and $\mathcal{F}^{-1}$ represent the forward and inverse fast Fourier transforms and $\tilde{g}(k)$ is the Green’s function in Fourier coordinates $k$. Since the simulation domain is finite, $G$ is a cyclic convolution, resulting in the wrap-around artefacts as can be seen in Fig. 1(a). The key point of our new efficient boundaries is to remove these wrap-around errors without increasing the computation time.

### 2.2. Padding-free acyclic convolution

Conventionally, the wrap-around artefacts introduced by the fast-convolution are removed through zero-padding. Our new method is instead based on a padding-free acyclic convolution that was first described by Radhakrishnan et al. [21]. In this section, we describe how this special type of convolution can be used to eliminate the wrap-around errors introduced by the operator $G$.

To simplify the notation, we start by considering a one dimensional scalar wave problem with spatial coordinates $x \in (-L/2, L/2)$.

First, the modified Fourier transforms are introduced:

$$\mathcal{F}E \equiv \int_{-L/2}^{L/2} E(x) e^{-i(k + \Delta k/4)x} dx,$$

and

$$\mathcal{F}E \equiv \int_{-L/2}^{L/2} E(x) e^{-i(k - \Delta k/4)x} dx,$$

where $\Delta k \equiv 2\pi/L$ is the grid spacing in $k$-space (Fourier space). Compared to a standard Fourier transform, the modified operators $\mathcal{F}$ and $\mathcal{F}$ modulate the field with a linear phase ramp from $-\pi/4$ to $\pi/4$, or vice versa, before transforming the field to Fourier coordinates. Applying these phase ramps in real-space is equivalent to a translation of the field in $k$-space of $\pm \Delta k/4$. These operations are reversed by the modified inverse Fourier transform operators $\mathcal{F}^{-1}$ and $\mathcal{F}^{-1}$, that first perform an inverse Fourier transform and then remove the phase ramp.

In Ref. [21], the modified Fourier transforms are used to perform the so called circular convolutions. Similarly, we now define a left- and right-circular version of the Green’s function operators, which are given by

$$\mathcal{G} \equiv \mathcal{F}^{-1}(k + \Delta k/4)\mathcal{F}$$

and

$$\mathcal{G} \equiv \mathcal{F}^{-1}(k - \Delta k/4)\mathcal{F}.$$
In Fig. 2, the acyclic convolution is demonstrated in a simple example, where the Green’s function is convolved with a point source. The results of $\vec{G}$ (see Fig. 2(a)) and $\vec{G}$ (see Fig. 2(b)) are averaged to obtain the acyclic convolution results. As can be seen in Fig. 2(c), the wrap-around errors are completely canceled by the ACC.

Unfortunately, this procedure to remove wrap-around artefacts is computationally very costly, since one needs to compute two convolutions instead of one. This problem is even worse when generalizing to higher-dimensions, where combinations of the left- and right-circular convolutions have to be considered in all dimensions [22], increasing the computation complexity by a factor of 4 and 8 for two and three-dimensional problems, respectively. Because of these extra computational costs, the applicability of the acyclic convolution has been limited and usually zero padding is used instead.

In the next section, however, we introduce a modification to this scheme. With our approach, in an iterative algorithm (such as the MBS algorithm) the different circular convolutions can be efficiently computed in a pipelined fashion without incurring the factor 2-8 increase in computational cost.

3. Pipelined MBS algorithm

We will now discuss how the acyclic convolution is efficiently incorporated into the MBS method. We start by replacing the operator $M$ (from Eq. (1)) with the left- and right-circular operators: $\vec{M} = -i\epsilon \vec{G} \gamma - \gamma + 1$ and $\vec{M} = -i\epsilon \vec{G} \gamma - \gamma + 1$. The direct implementation of the acyclic convolution into the MBS method would require us to apply both $\vec{M}$ and $\vec{M}$ in every iteration, which would double the computation time as explained above. Instead, in our approach we alternate between these two different operators

$$\hat{E}_{j+1} = \begin{cases} \vec{M} \hat{E}_j & \text{if } j \text{ is odd,} \\ \vec{M} \hat{E}_j & \text{if } j \text{ is even.} \end{cases}$$

The idea is now to add a source term both in the first and second iteration: $\hat{E}_1 = \gamma \vec{G} S / 2$, and $\hat{E}_2 = \vec{M} \hat{E}_1 + \gamma \vec{G} S / 2$. The algorithm keeps alternating between $\vec{M}$ and $\vec{M}$ until it converged to the final solution. This way we have effectively computed the following two sequences simultaneously

$$E = \frac{1}{2} \left( 1 + \vec{M} + \vec{M} \vec{M} + \vec{M} \vec{M} \vec{M} + \ldots \right) \gamma \vec{G} S +$$

$$\frac{1}{2} \left( 1 + \vec{M} + \vec{M} \vec{M} + \vec{M} \vec{M} \vec{M} + \ldots \right) \gamma \vec{G} S.$$ (7)
These two sequences differ only in that the left- and right-circular operators are swapped. To analyze how the wrap-around artefacts are affected, we split these operators in a part without artefacts $\tilde{M}$, representing the ideal situation, and an operator $\tilde{M}$ corresponding to the artefacts only. We now have $\tilde{M} = \tilde{M} - i\tilde{M}$ and $\tilde{M} + i\tilde{M}$. Consequently, the wrap-around artefact introduced by a term in the first sequence is exactly canceled by the same term in the second sequence. With the exception of these removed artefacts, the pipelined MBS algorithm finds the same solution as the conventional algorithm (see Eq. (1)).

Using this efficient ‘pipelining’ scheme, we can perform the acyclic convolutions without having to perform any extra evaluations of the operator $M$. However, from the addition of $\tilde{M} - i\tilde{M} + i\tilde{M} = \tilde{M} + i\tilde{M}$, we can see that the $\tilde{M}$ term remains, while the first-order wrap-around terms all exactly cancel. This means that a wave passing through the boundary layer twice (or any even number of times) will not be cancelled. In Appendix A, we show that these second-order artefacts can be mitigated when performing the acyclic convolution in three-dimensions. As becomes clear from the performance test results, the remaining wrap-around errors have a negligible contribution to the total error.

The new MBS algorithm suppresses the wrap-around artefacts that originate from the fast convolutions. As a result, sharp transitions in the field are introduced exactly at the boundaries causing reflections. Furthermore, to ensure the algorithm converges to the steady state solution, some absorbing layer at the boundary is still required. In Appendix B, we introduce a new boundary layer that is designed to minimize these reflection artefacts. In the performance test, we will show that the implementation of the acyclic convolution allows us to greatly reduce the size of the boundary layers.

4. Performance test

Next, we compare the performance of the original MBS method to our new pipelined MBS method with the acyclic convolution. These two convolution methods were both tested using the newly designed anti-reflection boundary layer (ARL, see Appendix B) and using the polynomial boundary layer (PBL) as implemented in our previous work [17]. This PBL was designed to minimize the reflection and wrap-around artefacts at normal incidence.

To assess the performance of these different boundary conditions, we simulated the propagation of light in a sphere with a diameter of 12 µm. We use refractive indices of 1.2 and 1 for the sphere and the background medium, respectively. The simulation grid spanned over a volume of $24^3$ µm$^3$, excluding boundaries, with a grid spacing of 0.20 µm. As the source, we use a linearly-polarized apodized plane wave with $\lambda = 1$ µm. The source is placed at $z = -12$ µm, and is polarized in the $x$-direction. To perform the ACC in all three dimensions, we used the convolution sequences as described in Appendix A. All MBS simulations were run for 140 iterations.

The scattered vector fields, $E_s$, obtained from the simulations, are compared to the results $E_a$ from Mie theory (generated with code from Ref. [23]). Boundary layers with a width of $W$ are added to all sides of the simulation volume. During the simulations, $W$ is varied from 1 to 12 µm. The accuracy of the simulation results is quantified by the relative error, given by: $\langle |E_s - E_a|^2 \rangle / \langle |E_a|^2 \rangle$. Here, $\langle \cdot \rangle$ denotes the averaging over all vector components and all grid points in the medium, excluding the boundary layers.

In Fig. 3, the results of the Mie sphere simulations are shown. As an example, cross sections of the scattered vector field amplitude are shown in Fig. 3(a)-(f). These figures contain the $x$- and $z$-component fields obtained from Mie theoretic (Fig. 3(a) and (d)), a simulation using the PBL without ACC (Fig. 3(b) and (e)), and a simulation using the ARL with ACC (Fig. 3(c) and (f)). Finally, in Fig. 3(g), the relative error of the simulation results are plotted against $W$. The blue and orange circles denote the simulation results obtained using the PBL and the ARL,
respectively. The results obtained using the acyclic convolution are denoted by the yellow and the purple squares for the PBL and ARL.

In the PBL simulation results, the plane wave source is not completely absorbed at the boundaries and wraps around the simulation domain. This leaked wave then interferes with the wave that scatters off the Mie sphere, resulting in a fringe pattern in the x-component of the field, as can be seen in Fig. 3(b). In the z-component of the field (Fig. 3(e)), a wrap-around artefact can also be seen. In this case, there are no interference fringes because the source was not polarized in that direction. In the ARL simulation results with the acyclic convolution method, in Fig. 3(c) and (f), the wrap-around artefacts have been successfully removed. In Fig. 3(g), we can observe a decrease in the relative error with the increase of the boundary width for all boundary conditions, until the maximum accuracy is reached (a relative error of 0.014). In this simulation, the maximum accuracy is limited by the discretization of the Mie sphere, and can be further improved by performing the simulation on a finer grid. For thin boundary layers (W<12 µm), however, the dominant error is a combination of wrap-around and reflection artefacts at the boundaries. The introduction of the ARL and the ACC method greatly reduce these errors. The best performance is achieved with the combination of the ARL with ACC.

Running one iteration of the pipelined MBS algorithm takes practically the same amount of computation time as the conventional MBS algorithm. However, as can be seen in Fig. 3(g), the new algorithm allows us to reach maximum accuracy while reducing the boundary width from 12 µm to 2 µm. We performed these simulations on both a central processing unit (CPU, Intel Core i7-3770 CPU @ 3.40GHz) and a graphics processing unit (GPU, Nvidia GeForce GTX 1080 Ti). In Table 1, we provide an overview of the CPU/GPU time and the required memory for the performance test. We compared the performance of the PBL and the ARL (with ACC) at maximum accuracy.

| Boundaries      | W (µm) | Rel. error | Total size (µm³) | GPU time (s) | CPU time (s) |
|-----------------|--------|------------|------------------|--------------|--------------|
| ARL with ACC    | 2      | 0.014      | 21952            | 3            | 150          |
| PBL             | 12     | 0.014      | 110592           | 15           | 890          |
For this three-dimensional test simulation, our method achieves a reduction of the total required simulation size from $(48 \, \mu m)^3$ to $(28 \, \mu m)^3$, which amounts to a reduction of a factor of 5 in required memory. Furthermore, the reduction of the total number of grid points also led to a reduction of the computation time of a factor of 5. Note that the fast Fourier transform at the core of our algorithm is an easily parallelizable operation, which allows us to run the simulations up to 60 times faster on the GPU compared to the CPU.

5. Conclusion

We have introduced a new class of boundary conditions for the modified Born series method, increasing the accuracy, speed and the memory efficiency of the method. First, we replaced the fast convolution operation in the algorithm with a padding-free acyclic convolution. The additional steps in this convolution method were efficiently pipelined, such that the wrap-around artefacts can be removed without any extra computational costs. Secondly, we designed a new ultra-thin boundary layer which minimizes the reflection artefacts introduced at the edge of the simulation domain. Compared to the previously implemented absorbing boundary layers, our new boundary conditions allowed us to reduce the total simulation size by 80% in the three-dimensional test simulation.

With each iteration, the modified Born series method alternates between two fixed linear operators, one applied in the spatial domain (the medium scattering potential) and the other in the spatial frequency domain (the Green’s function). Similar computational schemes can be found in other numerical techniques, such as pseudo-spectral methods [24]. We believe that our pipelined ACC scheme may be used to reduce wrap-around artefacts and improve the accuracy and efficiency of these methods as well.

Appendix A: Pipelined acyclic convolution algorithm in 3D

**Sequences of circular convolutions**

In the main text, we described the acyclic convolution method for a one-dimensional example. Here, we extend the idea of the pipelined modified Born series to three-dimensional (3D) problems. In a 3D simulation of an open system, waves can leak through the boundary in either of the three dimensions. To guarantee that the wrap-around artefacts are suppressed in all directions ($x$, $y$, $z$ and all diagonals), 8 combinations of left- and right-circular convolutions need to be considered.

For 3D problems, we iterate over the convolution sequences shown in Table 2, where ← and → represent the left- and right-circular convolutions in that particular dimension. Similar to the sequences in Eq. (7), these eight sequences can be computed in a pipelined fashion. This is achieved by alternating between eight different operators, corresponding to different combinations of shifts in the $k_x$, $k_y$, and $k_z$ coordinates, as shown in Table 2. It is worth noting that these operators are evaluated ‘on-the-fly’ on the GPU, so there is no memory penalty for using eight instead of one operator.

| step | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| $x$  | →     | ←     | →     | ←     | →     | ←     | →     | ←     |
| $y$  | →     | ←     | ←     | →     | ←     | →     | ←     | ←     |
| $z$  | →     | ←     | ←     | →     | ←     | ←     | ←     | ←     |

In the first eight iterations of the algorithm, we now add 1/8th of the source term, (as opposed to the standard scheme where the full source term is added at the first iteration only). The algorithm...
cycles through all eight operators until the desired accuracy is reached. Note that, inside the simulation domain, the shifted operators $\hat{M}$, $\hat{M}$, etc. are identical to the original operator $M$, so the convergence behavior is not affected by this approach. As a final result, we will now have effectively computed eight different modified Born series at once.

**Suppression of the second-order wrap-around artefacts**

Next, we will analyze how the sequences in Table 2 lead to a suppression of the second-order wrap-around artefacts, i.e. the waves that passed through the boundary layers twice. As discussed in the main text, the single wrap-around errors $\hat{M}$ are always nullified since the sequences have an equal number of $\leftarrow$ and $\rightarrow$ steps. With this, the dominant source of wrap-around artefacts is removed completely already.

The second-order artefacts take the form of $\tilde{M} M^n \tilde{M}$, where $n$ is the number of iterations in between the two wrap-around events. In our pipelined approach some, but not all, second-order terms will cancel. Exactly what terms cancel depends on $n$, the directions in which the wrap-around events occurred, and on the sequence in which the eight operators are alternated. In a brute-force search (see Ref. [25]) through all the permutations of the possible sequences, we found that the best results were obtained with the sequences in Table 2. These sequences cancel as many as 75% of the second-order wrap-around artefacts in addition to all first-order wrap-around artefacts. Compared to a standard sequence, we found that this optimized convolution sequence leads to 1.4% improvement in the accuracy for ultra-thin boundaries ($\leq$ 2 wavelengths in width).

**Appendix B: Anti-reflection boundary layer**

In this appendix, we describe the design of an ultra-thin anti-reflection layer to minimize the reflection artefacts at the boundaries.

We want the boundary layer to have the maximum amount of absorption that can stably be simulated, which is achieved when the modified scattering potential $\gamma = 0$ exactly. For simplicity, we implement the anti-reflection layer as a window function $\beta$ with which we multiply $\gamma$. The choice of $\beta$ is limited by the following restrictions: first, $\beta$ should be unity inside the simulation domain, as we do not want to affect the simulated structure. Second, the window function $\beta$ should be real-valued with values between 0 and 1. This second constraint ensures that convergence is still guaranteed after multiplication with the window function (also see $|\gamma|$ [17]).

We design $\beta$ for three-dimensional simulations of open systems. In all directions, the scattering potential $\gamma$ is first padded with a constant boundary layer with a width of $N_{\beta}$ grid points, and then it is multiplied by the window function to achieve $\gamma = 0$ at the very edges. Since the boundaries are identical in every direction, we will only consider the window function along the $z$-axis.

To find the optimal window function that minimizes the reflection artefacts, we define the cost function

$$C = \int_{-k_0}^{k_0} w(k_z)|\tilde{\beta}(2k_z)|^2 dk_z,$$

where $w(k_z)$ is a weighting function and $\tilde{\beta}$ is the Fourier transform of $\beta$.

In each iteration of the MBS algorithm, the electric field is multiplied with the scattering potential. This multiplication is equivalent to a convolution in $k$-space. For an incident plane wave $E \propto \exp(ik \cdot r)$, the reflection coefficient is determined by how strongly this convolution couples the incident ($k_z$) and reflected ($-k_z$) fields. For this reason, the reflection coefficient at normal incidence is proportional to $|\tilde{\beta}(k_z = \pm 2k_0)|^2$. The values of $|\tilde{\beta}(k_z < 2k_0)|^2$ determine the reflection coefficients of the incident waves with an oblique angle. Therefore, the cost function $C$ can be interpreted as a weighted average reflection coefficient of the boundary layer.

The optimal choice of the weighting function $w(k_z)$ in Eq. (8) depends on the exact geometry that is being simulated. However, since it is not practical to design a new boundary for each
geometry, we used a simple approach to compute a generic window function. First, we realize that waves at grazing incidence transport less energy through the surface due to Lambert’s cosine law. Additionally, in most simulated geometries, very little scattered light will reach the boundaries at grazing incidence, especially when the scattering structure is placed in the center of the simulation window. For these reasons, we choose a weighting function of \( w(kz) = k_z^2 \). This choice of weighting function has the additional advantage that the optimal window (minimizing \( C \)) is given by the simple function

\[
\beta(z_{\beta}) \approx \frac{|z_{\beta}| - 0.21}{N_{\beta} + 0.66},
\]

as determined numerically (see [25] for the filter optimization code). Here, \( z_{\beta} \) are the grid indices at the boundary layer ranging from \(-N_{\beta}\) to \(-1\) and from \(1\) to \(N_{\beta}\) at the bottom and top boundaries, respectively. The same linear window function is used along the boundary layers in the \(x\) and \(y\) axes. In the main text, we refer to this new boundary type as the anti-reflection boundary layer (ARL).

**Funding.** European Research Council (ERC-2016-StG-678919).

**Disclosures.** The authors declare no conflicts of interest.

**Data availability.** The Matlab code of our modified Born series method is available on GitHub under a BSD license [25]. The method can be used for scalar and vector wave simulations. Our new boundary conditions are implemented for both these cases.

**References**

1. C. F. Bohren and D. R. Huffman, *Absorption and scattering of light by small particles* (Wiley-VCH, 1998).
2. K. Yee, “Numerical solution of initial boundary value problems involving Maxwell’s equations in isotropic media,” IEEE Trans. Antennas Propag. 14(3), 302–307 (1966).
3. A. Taflove and S. C. Hagness, *Computational electrodynamics: The finite-difference time-domain method* (Artech House, 1995).
4. A. F. Oskooi, D. Roundy, M. Ibanescu, P. Bermel, J. D. Joannopoulos, and S. G. Johnson, “Meep: A flexible free-software package for electromagnetic simulations by the FDTD method,” Comput. Phys. Commun. 181(3), 687–702 (2010).
5. P. Monk, *Finite Element Methods for Maxwell’s Equations* (Clarendon Press, 2003).
6. Q. H. Liu, “The FSTD Algorithm: A Time-Domain Method Requiring Only Two Cells Per Wavelength,” Microw. Opt. Technol. Lett. 15(3), 158–165 (1997).
7. B. Gallinet, J. Butet, and O. J. F. Martin, “Numerical methods for nanophotonics: standard problems and future challenges,” Laser Photonics Rev. 9(6), 577–603 (2015).
8. J. W. Goodman, *Introduction to Fourier Optics* (W. H. Freeman, 2005).
9. J. Van Roey, J. Van Der Donk, and P. E. Lagasse, “Beam-propagation method: analysis and assessment,” J. Opt. Soc. Am. 71(7), 803–810 (1981).
10. A. K. Glaser, Y. Chen, and J. T. C. Liu, “Fractal propagation method enables realistic optical microscopy simulations in biological tissues,” Optica 3(8), 861–869 (2016).
11. M. Weigert, K. Subramanian, S. T. Bundschuh, E. W. Myers, and M. Kreyling, “Biobeam-multiplexed wave-optical simulations of light-sheet microscopy,” PLoS Comput. Biol. 14(4), e1006079 (2018).
12. J. Yang, J. Li, S. He, and L. V. Wang, “Angular-spectrum modeling of focusing light inside scattering media by optical phase conjugation,” Optica 6(3), 250–256 (2019).
13. X. Cheng, Y. Li, J. Mertz, S. Sakadžić, A. Devor, D. A. Boas, and L. Tian, “Development of a beam propagation method to simulate the point spread function degradation in scattering media,” Opt. Lett. 44(20), 4989–4992 (2019).
14. R. E. Kleinman, G. F. Roach, and P. M. Van Den Berg, “Convergent Born series for large refractive indices,” J. Opt. Soc. Am. A 7(5), 890–897 (1990).
15. O. Pankratov, D. Avdeev, and A. Kuvshinov, “Electromagnetic field scattering in a homogeneous Earth: a solution to the forward problem,” Phys. Solis. Earth 31(3), 201–209 (1995).
16. M. S. Zhdanov and S. Fang, “Quasi-linear series in 3-D electromagnetic modeling,” Radio Sci. 32(6), 2167–2188 (1997).
17. G. Osnabrugge, S. Leedumrongwathanakun, and I. M. Vellekoop, “A convergent Born series for solving the inhomogeneous Helmholtz equation in arbitrarily large media,” J. Comput. Phys. 322, 113–124 (2016).
18. B. Krüger, T. Brenner, and A. Kienle, “Solution of the inhomogeneous Maxwell’s equations using a Born series,” Opt. Express 25(21), 25165–25182 (2017).
19. T. Vettenburg, S. A. R. Horsley, and J. Bertolotti, “Calculating coherent light-wave propagation in large heterogeneous media,” Opt. Express 27(9), 11946–11967 (2019).
20. J. P. Berenger, “A perfectly matched layer for the absorption of electromagnetic waves,” *J. Comput. Phys.* **114**(2), 185–200 (1994).

21. C. Radhakrishnan and W. K. Jenkins, “Modified discrete fourier transforms for fast convolution and adaptive filtering,” in *IEEE International Symposium on Circuits and Systems*, (IEEE, 2010), pp. 1611–1614.

22. C. Radhakrishnan and W. K. Jenkins, “The 2-D Modulated Discrete Fourier Transform for 2-D fast convolution and digital filtering,” in *IEEE International Symposium of Circuits and Systems*, (IEEE, 2011), pp. 1508–1511.

23. J.-P. Schäfer, *MatScat* (2012). http://www.mathworks.com/matlabcentral/fileexchange/36831-matscat.

24. J. P. Boyd, *Chebyshev and Fourier spectral methods* (Springer, 1989).

25. G. Osnabrugge and I. M. Vellekoop, *WaveSim* (2020). https://github.com/ivovellekoop/wavesim.