Progress In Electromagnetics Research B, Vol. 102, 1–17, 2023

A Parallel 3D Spatial Spectral Volume Integral Equation Method for Electromagnetic Scattering from Finite Scatterers

Stefan Eijsvogel*, **Roeland J. Dilz, and Martijn C. van Beurden**

Abstract—Parallel computing for the three-dimensional spatial spectral volume integral equation method is presented for the computation of electromagnetic scattering by finite dielectric scatterers in a layered medium. The first part exploits the Gabor-frame expansion to compute the Gabor coefficients of scatterers in a parellel manner. The second part concerns the decomposition and restructuring of the matrix-vector product of this spatial spectral volume integral equation into (partially) independent components to enable parallel computing. Both capitalize on the hardware to reduce the computation time by shared-memory parallelism. Numerical experiments in the form of solving electrically large scattering problems, namely volumes up to 1300 cubic wavelengths, in combination with a large number of finite scatterers show a significant reduction in wall-clock time owing to parallel computing, while maintaining accuracy.

1. INTRODUCTION

Scatterometry, a non-destructive inspection technique [1] for assessing the quality of the manufacturing process of structures on a wafer, uses the electromagnetic reflectivity of metrology targets on a wafer to obtain shape-deformation details [2]. These details are obtained by comparing the measured electromagnetic fields scattered from these targets to those obtained from a computational model [3, 4], e.g., a Maxwell solver providing accurate electromagnetic scattering data. To fit the scatterometry setting, this Maxwell solver needs to provide an accurate and fast scattering analysis for three-dimensional (3D) finite dielectric objects in the presence of a planarly layered medium [5].

In [6, 7], a 3D volume integral equation (VIE) method is presented that computes the time-harmonic electromagnetic scattering from finite dielectric scatterers embedded in a planarly layered medium. This VIE is formulated in both the spatial and spectral domain simultaneously, to exploit the transverse translation symmetry present within the layered medium. The employed discretization is key for this formulation. Piecewise-linear functions in combination with Green-function recursions [8] ensure a linear scaling of the computational complexity with respect to the number of expansion functions in the direction perpendicular to the interfaces of the planarly layered medium. A Gabor-frame expansion is employed in the transverse plane, i.e., the plane parallel to these interfaces, since it enables fast, exact and aperiodic local Fourier transformations from the spatial to the spectral domain and vice versa. The combination of piecewise-linear functions and a Gabor-frame expansion ensures an overall computational complexity of $O(N \log N)$ for the matrix-vector product, where N is the number of unknowns in the scattering problem.

According to [9], the most time-consuming components of this Maxwell solver [6] are the computation of the Gabor-frame expansion coefficients of the scatterers and iteratively solving the VIE, which take 146 seconds (12.7%) and 962 seconds (84.1%) for a 3D simulation domain of 60 cubic wavelengths with 36 scatterers as in [10], respectively. The extension of this solver to scatterometry

Received 7 June 2023, Accepted 19 July 2023, Scheduled 1 August 2023

* Corresponding author: Stefan Eijsvogel (s.eijsvogel@tue.nl).
The authors are with The Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands.

applications is challenging: a simulation domain quickly becomes larger than $1000\lambda^3$, with λ the wavelength, since the illumination areas can range up to several micrometers [11, Ch. 12], and even more so for Extreme-Ultraviolet (EUV) illumination with a wavelength of tens of nanometers [12]. This is more than this Maxwell solver can handle within a reasonable time span on a single core. Additionally, increasing the number of scatterers in such a simulation domain increases the computational burden on this Maxwell solver. It becomes even more challenging in the case of an inverse scattering problem, since this requires iterative updates of the scatterers to find fitting shapes and permittivities [13, Ch. 6]. Thus, the Maxwell solver in [6, 7] would fit the scatterometry setting better if we can significantly reduce the wall-clock computation time for both the iterative solution process of the VIE and the computation of the Gabor-frame expansion of the scatterers by means of parallel-computing methods.

Exploiting the increasing capabilities of modern computational hardware in combination with computationally efficient algorithms has resulted in the possibility of simulating large-scale electromagnetic problems faster in terms of wall-clock time, e.g., [14–20]. Inspired by these techniques, we investigate converting the Maxwell solver of [6, 7] from a single-core algorithm to a multi-core algorithm, to achieve a significant reduction in wall-clock time. First, we exploit the localized property of the Gabor-frame formulation to split the simulation domain containing the scatterers into smaller subdomains. Hence, we can compute the Gabor-frame expansion coefficients of the scatterers in parallel. Second, we decompose and restructure parts of the matrix-vector product of the spatial spectral VIE method into independent components suitable for parallel computing. We test the parallelization of these two components of the spatial spectral VIE method with two simulation cases that involve computing the electromagnetic scattering from two gratings, i.e., a grating with 36 scatterers in a volume of $60\lambda^3$ for $\lambda = 425\,\text{nm}$ and a grating with 529 scatterers in a volume of $1300\lambda^3$ with $\lambda = 13.5\,\text{nm}$. We evaluate the performance of the parallelization methods as a function of the number of CPU cores, ranging from 1 up to 20, to measure the scaling in computation time and the corresponding parallel efficiency. Overall, we show, in case of the largest grating, that the wall-clock time is reduced by more than a factor of 9, which improves the suitability of the spatial spectral VIE method for analyzing the electric response of large-scale scattering problems filled with a large number of finite scatterers.

This paper is organized as follows. We introduce the key concepts and definitions of the 3D spatial spectral Maxwell solver in Section 2. In Section 3, we describe an algorithm for computing the Gabor-frame expansion coefficients for multiple scatterers in a parallel fashion. Section 4 describes a parallelization algorithm for the matrix-vector product of the 3D spatial spectral VIE. In Section 5, we give two simulation examples to evaluate the gain in performance owing to parallelization. Section 6 contains the conclusions.

2. PRELIMINARIES

2.1. Geometrical Description

The geometrical description for a forward scattering problem in this work is as follows. We consider a layered background medium, consisting of homogeneous isotropic dielectric layers, placed between two half-spaces. The layer interfaces are located at z_n and the relative permittivities are set as $\varepsilon_{rb,n}$ with $n = 0, \ldots, N_L$. We explicitly use index $n = 0$ and $n = N_L$ for the top and bottom half-space, respectively. Fig. 1 illustrates an example of such a layered medium with embedded scatterers. This example matches the geometrical description of the 3D spatial spectral Maxwell solver as in [7, p.27] for forward scattering problems. In Fig. 1, two scatterers are embedded in a single layer of the layered background medium, where the transparent box with black edges with dimensions $[-W_x, W_x] \times [-W_y, W_y] \times [z_{\min}, z_{\max}]$ is the complete simulation domain $\mathcal{D}$.

2.2. Spatial Spectral VIE

The total electric field $\mathbf{E}(\mathbf{x})$ within the simulation domain $\mathcal{D}$ is defined as

$$\mathbf{E}(\mathbf{x}) = \mathbf{E}^i(\mathbf{x}) + \mathbf{E}^s(\mathbf{x}), \tag{1}$$

where $\mathbf{x} = (x, y, z)$ denotes the Cartesian coordinates. The incident electric field $\mathbf{E}^i(\mathbf{x})$ originates from the upper half-space, e.g., a plane wave with an arbitrary angle of incidence and polarization.

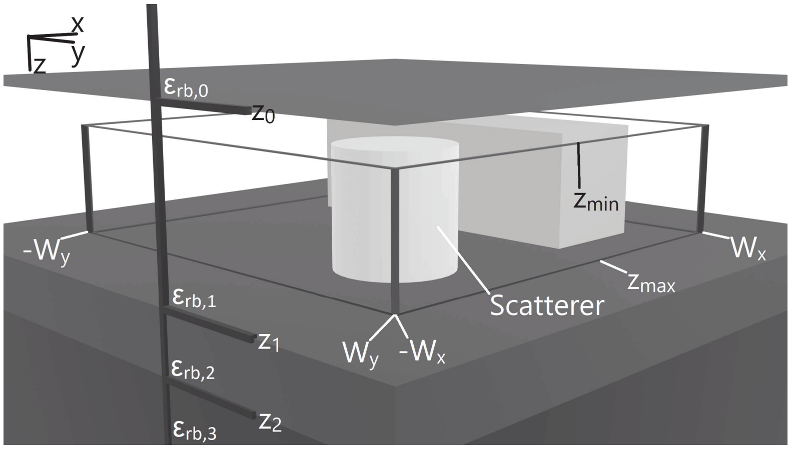


Figure 1. An example of a geometrical setup for evaluating the electromagnetic scattering from finite scatterers.

The scattered electric field is evaluated in the spatial and the spectral domain simultaneously, to incorporate the reflections and transmissions across the layer interfaces efficiently without computing tedious Sommerfeld integrals [6, 21]. Therefore, we describe the coordinate system as a spatial transverse plane $\mathbf{x}_t = (x, y)$ with the z-axis acting as a longitudinal axis. The Fourier transform $f(\mathbf{k}_t)$ of $f(\mathbf{x}_t)$ in the transverse plane is defined as

$$f(\mathbf{k}_t) = \mathcal{F}_{\mathbf{x}_t}[f(\mathbf{x}_t)](\mathbf{k}_t) = \iint\limits_{\mathbb{R}^2} f(\mathbf{x}_t)\exp(-j\mathbf{k}_t \cdot \mathbf{x}_t) d\mathbf{x}_t, \tag{2}$$

where $\mathbf{k}_t = (k_x, k_y)$ is used as the argument for functions in the spectral domain, whereas $\mathbf{x}_t$ is used for functions defined in the spatial domain. For scattering objects confined to a single layer, we compute the scattered field as

$$\mathbf{E}^s(\mathbf{k}_t, z) = \int_{z_{\min}}^{z_{\max}} \mathcal{G}(\mathbf{k}_t, z|z') \cdot \mathbf{J}(\mathbf{k}_t, z')dz', \tag{3}$$

with a Green function $\mathcal{G}(\mathbf{k}_t, z|z')$ [6] that contains the 3D effective reflection coefficients [22, 23] at the layer interfaces of the layered background medium, to incorporate its response. This integral is efficiently evaluated in the z-direction in a recursive manner via the algorithm in [8], for piecewise-linear functions. In addition, a complex-plane deformation is applied to the integration manifold of the two-dimensional (2D) transverse Fourier transformations [6]. This deformation circumvents the poles and branch cuts present in the spectral Green function. This deformation results in nine distinct regions in the spectral domain, see Fig. 2(a). We classify the workload of these regions into three types, due to differences in the number of function samples, namely type 1, with the largest computational workload, up to type 3, with the smallest workload. Further, the contrast current density $\mathbf{J}(\mathbf{k}_t, z)$ is defined as

$$\mathbf{J}(\mathbf{k}_t, z) = j\omega\varepsilon_0\varepsilon_{rb,n}\mathcal{F}_{\mathbf{x}_t}[\chi(\mathbf{x}_t, z)\mathbf{E}(\mathbf{x}_t, z)](\mathbf{k}_t, z), \tag{4}$$

with a contrast function $\chi(\mathbf{x})$ representing all scatterers in layer n of the layered background medium. In [6], each scatterer is assumed to be a compact, finite-sized, and locally homogeneous permittivity distribution with respect to its background medium per z-sample, owing to the integral formulation in Eq. (3). Therefore, we identify a scatterer, embedded in layer n, per z-sample by its contrast function χ defined as

$$\chi_s(\mathbf{x}_t, z) = \begin{cases} \dfrac{\varepsilon_r^s}{\varepsilon_{rb,n}} - 1 & (\mathbf{x}_t, z) \in \mathcal{D}_s \\ 0 & (\mathbf{x}_t, z) \in \mathbb{R}^3 \setminus \mathcal{D}_s. \end{cases} \tag{5}$$

Domain $\mathcal{D}_s$ is a subdomain of $\mathcal{D}$, which is characterized by its relative permittivity ε_r^s and the shape of its cross-section for a fixed value of z, in the form of an arbitrary simple 2D polygon. Accordingly, domain $\mathcal{D}$ is filled with scatterers represented by $\chi_s(\mathbf{x}_t, z) \neq 0$ with corresponding subdomains $\mathcal{D}_s$ with the index $s = 1, 2, \ldots, S$, where S represents the total number of scatterers. The shape of a cross-section is completely defined by the sequence of the coordinates of its vertices, namely $\mathbf{x}_i^s = (x_i^s, y_i^s)$ given $i = 1, 2, \ldots, L_s$ in a counter-clockwise direction. The contrast function in Eq. (5) is assumed to be deterministic. For completeness, we refer to [24] for an approach to include stochastic effects in the permittivity.

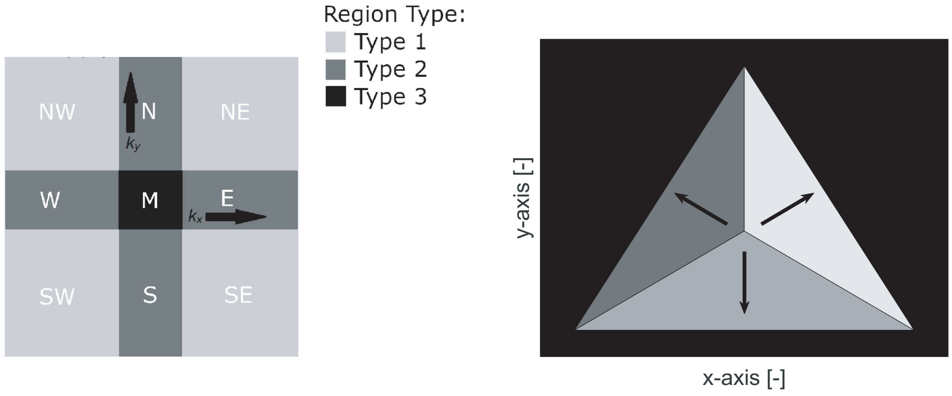
We also apply a local normal-vector field (NVF) formulation in the $\mathbf{x}_t$-plane [6, 25] to ensure an accurate field-material interaction in a spectral basis, since the form in Eq. (4) violates the Li factorization rules [26, 27]. Moreover, the NVF formulation also leads to an improved rate of convergence in spectral formulations for a continuous spectral expansion, such as a Gabor-frame expansion [6, 23]. We use a normal-vector field on the material interfaces of $\chi(\mathbf{x})$, to construct a normalized vector field over all the scatterers per z-sample. This is used to combine the continuous components of the electric flux density $\mathbf{D}$ and electric field $\mathbf{E}$ into a continuous auxiliary vector field $\mathbf{F}$. The contrast current density then becomes

$$\mathbf{J}(\mathbf{x}_t, z) = j\omega\varepsilon_0\varepsilon_{rb,n}\left[\chi C_\varepsilon\right](\mathbf{x}_t, z)\mathbf{F}(\mathbf{x}_t, z), \tag{6}$$

while the total electric field is obtained as

$$\mathbf{E}(\mathbf{x}_t, z) = \left[C_\varepsilon\right](\mathbf{x}_t, z)\mathbf{F}(\mathbf{x}_t, z). \tag{7}$$

The linear operators $[C_\varepsilon]$ and $[\chi C_\varepsilon]$ are defined per Cartesian component, i.e., $[C_\varepsilon]_{i,j}(\mathbf{x})$ where $i, j \in \{x, y, z\}$, and their definitions can be found in [6]. Note that the operator $[\chi C_\varepsilon]$ is spatially the product of the contrast function $\chi(\mathbf{x})$ and the operator $[C_\varepsilon]$. An effect of this local NVF formulation is that we treat a single scatterer as a collection of scatterers as shown in Fig. 2(b), where a triangle is subdivided into three smaller triangles to obtain only a single straight material interface per smaller triangle, which facilitates the definition of the normal-vector field.



(a) (b)

Figure 2. (a) The nine regions in the spectral domain owing to the complex-plane deformation of the integration manifold. The regions resemble the cardinal and ordinal directions of a compass, as indicated by the capital letters per region. The region types, indicated by the shading, relate to the associated workload per domain. (b) A local NVF formulation on the $\mathbf{x}_t$-plane definition for a triangular scatterer. The arrows denote the normal-vector directions of the outer boundary of the initial scatterer.

The combination of all mentioned aspects forms the spatial spectral VIE as

$$\left[\mathcal{C}_\varepsilon\right](\mathbf{x}_t, z)\mathbf{F}(\mathbf{x}_t, z) - \mathcal{F}^{-1}_{\mathbf{k}_t}\left[\int_{z_{\min}}^{z_{\max}} \mathcal{G}(\mathbf{k}_t, z|z')\mathcal{F}_{\mathbf{x}_t}\left[[\chi\mathcal{C}_\varepsilon](\mathbf{x}'_t, z')\mathbf{F}(\mathbf{x}'_t, z')\right](\mathbf{k}_t, z')dz'\right](\mathbf{x}_t, z') = \mathbf{E}^i(\mathbf{x}), \tag{8}$$

which we rewrite to the following decomposition of linear operators for the overall linear system

$$(C - F^{-1}GFM)\mathbf{F} = \mathbf{E}^i. \tag{9}$$

The matrices M and C represent $[\chi C_\varepsilon](\mathbf{x}_t, z')$ and $[C_\varepsilon](\mathbf{x}_t, z')$, respectively. We use F for the forward transverse Fourier transformation, while F^{-1} represents the inverse transverse Fourier transformation. The matrix G contains the point-wise multiplications of the Green function in the $\mathbf{k}_t$-plane in combination with the upward and downward recursions in the z-direction.

3. COMPUTING THE GABOR COEFFICIENTS OF SCATTERERS

3.1. Preprocessing

The spatial spectral VIE method starts with preprocessing. This concerns setting up the discretization, e.g., the piecewise-linear functions in the z-direction. It also concerns the computation of components such as the Green function $\mathcal{G}(\mathbf{k}_t, z|z')$ with its effective reflection coefficients, the incident field $\mathbf{E}^i(\mathbf{x})$, the contrast function pertaining to the scatterers $\chi(\mathbf{x})$, and the linear operators $[C_\varepsilon](\mathbf{x}_t, z)$ and $[\chi C_\varepsilon](\mathbf{x}_t, z')$. These specific components only have to be computed once and they are characterized by the geometrical description of the scattering problem. The required computation time for preprocessing is dominated by the computation of Gabor coefficients of the contrast functions $\chi(\mathbf{x})$ that represent the scatterers, e.g., more than 80% of the total preprocessing time in [9, Sec. 6]. This computationally expensive procedure directly relates to the construction of the linear operators $[C_\varepsilon](\mathbf{x}_t, z)$ and $[\chi C_\varepsilon](\mathbf{x}_t, z')$, since their definitions depend on the spectral expansion of the scatterers [6, 25], i.e., the Gabor expansion coefficients. Consequently, we focus on computing these Gabor coefficients of the scatterers by means of parallel computing.

3.2. Gabor-Frame Expansion

We use a Gabor-frame expansion in the $\mathbf{x}_t$-plane, since it facilitates efficient and analytic transverse Fourier transformations from the spatial domain to the spectral domain and vice versa [6]. For the Gabor frames, we use a Gaussian window function [28] of the following form

$$g(\mathbf{x}_t) = \sqrt{2}\exp\left[-\pi\left(\frac{x^2}{X^2} + \frac{y^2}{Y^2}\right)\right]. \tag{10}$$

The parameters X and Y are used to control the widths of the window function and these are linked to the quantities $K_x = \frac{2\pi}{X}$ and $K_y = \frac{2\pi}{Y}$. This sets the spatial Gabor frames as

$$g_{\mathbf{mn}}(\mathbf{x}_t) = g(x - m_x\alpha_x X, y - m_y\alpha_y Y)\exp\left(j\beta_x n_x K_x x + j\beta_y n_y K_y y\right), \tag{11}$$

in combination with the oversampling parameters $\alpha_x\beta_x < 1$ and $\alpha_y\beta_y < 1$. We use a rational oversampling of $\alpha_x\beta_x = \alpha_y\beta_y = q/p = 2/3$ in this work. The indices $\mathbf{m} = (m_x, m_y)$ describe the shifts, while indices $\mathbf{n} = (n_x, n_y)$ express the modulations. The so-called spatial dual window $\eta(\mathbf{x})$, which is currently synthesized from the Gaussian window function via the Moore-Penrose inverse [28], forms a dual Gabor frame that exhibits exponential decay and it is denoted as

$$\eta_{\mathbf{mn}}(\mathbf{x}_t) = \eta(x - m_x\alpha_x X, y - m_y\alpha_y Y)\exp\left(j\beta_x n_x K_x x + j\beta_y n_y K_y y\right), \tag{12}$$

such that a square-integrable function $f(\mathbf{x}_t)$ can be represented as

$$f(\mathbf{x}_t) = \sum_{\mathbf{m}}\sum_{\mathbf{n}} c_{\mathbf{mn}} g_{\mathbf{mn}}(\mathbf{x}_t), \tag{13}$$

where the Gabor coefficients $c_{\mathbf{mn}}$ are computed as

$$c_{\mathbf{mn}} = \int_{\mathbb{R}^2} f(\mathbf{x}_t)\eta^*_{\mathbf{mn}}(\mathbf{x}_t)d\mathbf{x}_t. \tag{14}$$

The integral in Eq. (14) is finite, since the function $f(\mathbf{x}_t)$ is square-integrable by assumption and the dual window $\eta(\mathbf{x}_t)$ is also square-integrable and exhibits exponential decay towards infinity [29]. It is possible to use other window functions within a Gabor frame and this impacts the shape and decay behavior of the dual window [29–31]. We use the Fourier transform of Eq. (10) to form the spectral Gabor frames with exponentially decaying dual frames. Additionally, it links the spatial Gabor coefficients $c_{\mathbf{mn}}$ and the spectral Gabor coefficients $\hat{c}_{\mathbf{mn}}$ by

$$c_{\mathbf{mn}} = \hat{c}_{\mathbf{nm}}\exp\left[j2\pi(\alpha_x\beta_x m_x n_x + \alpha_y\beta_y m_y n_y)\right]. \tag{15}$$

3.3. Sequential Algorithm

We compute the spatial Gabor coefficients of the scatterers via the spectral transformation method as described in [9]. This method starts by computing the spectral Gabor coefficients of an analytical and continuous description of Eq. (5) in the spectral domain via a spectral version of a Discrete Gabor Transform (DGT), i.e., an integration scheme based on the use of several Fast Fourier Transforms (FFT) [28]. This leads to an error that decays exponentially with the number of sample points N [9]. The spatial Gabor coefficients are obtained via the analytic Fourier transformation of the spectral Gabor coefficients in Eq. (15). The spectral description of a (non-self-intersecting) cross-section $\chi_s(\mathbf{x}_t, z) \neq 0$ is equal to

$$\hat{\chi}_s(\mathbf{k}_t, z) = -\left(\frac{\varepsilon_r^s}{\varepsilon_{rb,n}} - 1\right) \sum_{i=1}^{L_s} j(\mathbf{k}_t \cdot \boldsymbol{\nu}_i^{i+1,s}) |\mathbf{x}_{i+1}^s - \mathbf{x}_i^s| \exp\left(j\mathbf{k}_t \cdot \mathbf{x}_i^s\right) \frac{\exp\left[j\mathbf{k}_t \cdot (\mathbf{x}_{i+1}^s - \mathbf{x}_i^s)\right] - 1}{j(\mathbf{k}_t \cdot \mathbf{k}_t)\left[\mathbf{k}_t \cdot (\mathbf{x}_{i+1}^s - \mathbf{x}_i^s)\right]}, \qquad (16)$$

where the vertices $\mathbf{x}_{i+1}^s$ and $\mathbf{x}_i^s$ are connected by a line segment with an outward-pointing unit normal vector $\boldsymbol{\nu}_i^{i+1,s}$. The variable $\mathbf{x}_{i+1}^s$ for $i = L_s$ is set as $\mathbf{x}_1^s$, to close the boundary of the cross-section. We note the following two additional details. First, we assume that each scatterer has a cross-section that is uniform along its height. This means that a scatterer with a non-uniform cross-section along its height is decomposed into a collection of scatterers, where each scatterer has an uniform cross-section along its height. Consequently, S represents the number of scatterers in simulation domain D with an uniform cross-section along its height in this work. As a result, a scatterer in the form of a sphere is decomposed into an unique scatterer per z-sample, since the cross-section changes per z-sample. Further, we highlight that an extension to an inhomogeneous permittivity distribution per z-sample is achieved by describing the permittivity distribution as the multiplication of the support function, namely Eq. (16) without the scaling by $\varepsilon_r^s/\varepsilon_{rb,n} - 1$, and a smooth but more rapidly varying permittivity function, while expanded in Gabor frames.

This method can also be used for computing the Gabor coefficients of multiple scatterers by performing a superposition of Eq. (16), i.e., a summation of the spectral equivalent of the cross-sections representing all scatterers at each z-sample. Unfortunately, this is computationally inefficient, as we will now explain. Consider a support function of a cross-section, $f(\mathbf{x}_t)$, centered at the origin and the same support function of a cross-section shifted away from the origin, $f_s(\mathbf{x}_t)$, such that the Fourier transforms of these support functions are linked as $\hat{f}_s(\mathbf{k}_t) = \hat{f}(\mathbf{k}_t)\exp(-j2\pi\mathbf{k}_t \cdot \mathbf{x}_0)$ [32]. Thus, the minimum Nyquist sample rate is dictated by the cross-section placed furthest away from the origin of the $\mathbf{x}_t$-plane. Consequently, cross-sections closer to the origin are unnecessarily over-sampled. Additionally, it ignores the possibility to exploit the locality of the window functions within the Gabor-frame expansion to compute expansion coefficients of cross-sections with a reduced number of Gabor coefficients to reduce the workload.

3.4. Parallel Algorithm

We want to compute the Gabor coefficients for all scatterers individually on a smaller subset of Gabor coefficients, to improve efficiency and to enable parallel computation. Therefore, we exploit several Gabor-frame expansion properties [33] to make this possible. Gabor coefficients are effectively local, owing to the exponentially decaying dual window functions, see Eq. (14). So, only a limited number of coefficients are relevant for accurately representing a scatterer [9, Eq. (A9)]. Additionally, a Gabor frame is discretely translation invariant on its underlying discrete lattice defined by the shift indices $\mathbf{m}$ and modulation indices $\mathbf{n}$. The combination of these two properties allows us to change the position of a scatterer, while expanded in Gabor coefficients, by an integer number of discrete shifts of size $(\Delta x, \Delta y) = (\alpha_x X, \alpha_y Y)$. As an example, we can change the transverse position of scatterer with its cross-section centered at $\mathbf{x}_t = (\alpha_x m_{sx} X, \alpha_y m_{sy} Y)$ to $\mathbf{x}_t = (0, 0)$ with the following phase factor and change of the m_x and m_y indices per Gabor coefficient, i.e.,

$$c_{m_x - m_{sx}, m_y - m_{sy}, \mathbf{n}} = c_{\mathbf{mn}} \exp\Big(j2\pi(\alpha_x \beta_x n_x m_{sx} + \alpha_y \beta_x n_y m_{sy})\Big), \qquad (17)$$

with the integers m_{sx} and m_{sy}. Another key Gabor-frame property is that the superposition principle holds. Hence, we can compute the Gabor coefficients of each cross-section individually, while minimizing the oscillations in the spectral domain by translating cross-sections as close as possible to the origin of the $\mathbf{x}_t$-plane with the discrete shifts as in Eq. (17).

Algorithm 1 shows a pseudocode for computing the Gabor-frame expansion of the scatterers in a concurrent manner; it makes use of the discrete shifts in Eq. (17). Its inputs are a list with the vertex coordinates of all cross-sections within the entire simulation domain and the number of CPU cores for parallel computing N_{cores}. The output is a five-dimensional (5D) array $\chi_{\mathbf{mn}}^{domain}$ containing the Gabor coefficients of the cross-sections for each z-sample. The algorithm consists of three parts: the initialization of a 5D array per CPU core $\chi_{\mathbf{mn}}^{core}$ for storing the Gabor coefficients of cross-sections at all z-samples and two parallel do-loops. Note that the arrays $\chi_{\mathbf{mn}}^{domain}$ and $\chi_{\mathbf{mn}}^{core}$ have the same size. Additionally, each version of $\chi_{\mathbf{mn}}^{core}$ is assigned to only a single CPU core in the first parallel do-loop to prevent problematic read/write interactions, i.e., race-conditions [34].

Algorithm 1: Pseudocode parallel Gabor-frame expansion of S scatterers for all N_z samples in the z-direction

```
Input: The vertex coordinates from all cross-sections and $N_{cores}$
Output: $\chi_{\mathbf{mn}}^{domain}$
Initialize $N_{cores}$ versions of $\chi_{\mathbf{mn}}^{core}$
do parallel $k = 1, N_z$
    Task 1: analyze cross-section
    Compute surface area $A$ by Eq. (18)
    Compute centroid coordinates $(c_x, c_y)$ by Eq. (19)
    Task 2: translate cross-section
    Update vertices of a cross-section, i.e., $\mathbf{x}_i^s$ with $i = 1, 2, \ldots, L_s$, by Eqs. (20)–(21)
    Task 3: compute Gabor coefficients
    Compute Gabor coefficients $c_{\mathbf{mn}}^{shift}$ by the spectral transformation method
    Task 4: translate cross-section to original position
    Shift Gabor coefficients of cross-section to its original position by Eq. (17): $c_{\mathbf{mn}} \leftarrow c_{\mathbf{mn}}^{shift}$
    Task 5: load Gabor coefficients in array
    $\chi_{\mathbf{mn}}^{core} \leftarrow \chi_{\mathbf{mn}}^{core} + c_{\mathbf{mn}}$
do parallel $k = 1, N_z$
    Task 6: combine all $\chi_{\mathbf{mn}}^{core}$ arrays into one domain array
    do i $= 1, N_{cores}$
        $\chi_{\mathbf{mn}}^{domain} \leftarrow \chi_{\mathbf{mn}}^{domain} + \chi_{\mathbf{mn}}^{core}$
```

The first parallel do-loop in Algorithm 1 divides the cross-sections within the entire simulation domain over all N_{cores} cores. The workload per parallel do-loop iteration for each CPU core consists of five tasks. The first task is analyzing a provided cross-section by computing its area A and centroid coordinates (c_x, c_y) [35] as

$$A = 0.5 \sum_{i=1}^{L_s} (x_i^s y_{i+1}^s - x_{i+1}^s y_i^s), \tag{18}$$

$$(c_x, c_y) = \frac{1}{6A} \left(\sum_{i=1}^{L_s} (x_i^s + x_{i+1}^s)(x_i^s y_{i+1}^s - x_{i+1}^s y_i^s), \sum_{i=1}^{L_s} (y_i^s + y_{i+1}^s)(x_i^s y_{i+1}^s - x_{i+1}^s y_i^s) \right), \tag{19}$$

with $x_{i+1}^s = x_1^s$ and $y_{i+1}^s = y_1^s$ for $i = L_s$. The second task is to translate the cross-section as close as possible to the origin of the spatial coordinate system given its centroid coordinates (c_x, c_y) and the Gabor-frame window widths X and Y. This shift per vertex of a cross-section is achieved by

$$(x_i^s, x_i^s) = (x_i^s - \alpha_x X m_{sx}, y_i^s - \alpha_y Y m_{sy}) \tag{20}$$

given $i = 1, 2, \ldots, L_s$ and defining the integers (m_{sx}, m_{sy}) as

$$(m_{sx}, m_{sy}) = (\text{round}[c_x/(\alpha_x X)], \text{round}[c_y/(\alpha_y Y)]), \tag{21}$$

where we round to the nearest integer. The Gabor coefficients of the shifted cross-section $c_{\mathbf{mn}}^{shift}$ are computed via the spectral transformation method, which is the third and most time-consuming task. Note that cross-sections require different numbers of coefficients depending on size and position with respect to the Gabor-frame expansion. The fourth task is to shift the cross-section, expanded in Gabor coefficients, back to its original position by Eq. (17) applied to its Gabor coefficients. The final task within this parallel do-loop is the transfer of the Gabor coefficient of a cross-section to the array $\chi_{\mathbf{mn}}^{core}$. The fourth and fifth task exploit the locality of Gabor-frame expansion by only performing the operations on the Gabor coefficients of a cross-section instead of the Gabor coefficients for the entire simulation domain. The last task is combining all $\chi_{\mathbf{mn}}^{core}$ arrays into $\chi_{\mathbf{mn}}^{domain}$ such that it holds the Gabor coefficients of all cross-sections for each z-sample. This task can also be executed in a parallel manner if one makes use of a parallel do-loop focused on the z-samples. This prevents problematic read-and-write interactions, since the cross-sections are independent from each other in the z-direction.

We note that the computational workload in the first parallel do-loop is heavily reduced if the cross-section of a scatterer does not change in the z-direction, namely its cross-section is uniform along its height. In that case, only a single cross-section of a scatterer needs to be computed. We also remark that a key part of a parallel algorithm is proper load balancing. Computing the Gabor coefficients comprises the majority of the computational workload in the first parallel do-loop. This workload is defined by the number of (relevant) Gabor coefficients. For instance, a larger cross-section results in a larger workload. It is possible to sort all cross-sections in terms of their area to ensure a relatively even distribution of the total workload per CPU core. However, this bookkeeping is cumbersome. Therefore, a subtle way to deal with the load balancing is to make use of the dynamic scheduling capabilities of OpenMP [36].

4. THE SPATIAL SPECTRAL VIE AND ITS MATRIX-VECTOR PRODUCT

We continue by focusing on the operations applied to the auxiliary vector field $\mathbf{F}(\mathbf{x})$, see the left-hand side of Eq. (9). These operations can be viewed as a matrix-vector product in every iteration of a Krylov subspace method, e.g., BiCGstab(ℓ) [37], for solving the linear system in Eq. (9). Hereafter, a superscript r expresses the r^{th} BiCGstab(ℓ) iteration for each field or current quantity, such that we write the sequential order of these operations acting on $\mathbf{F}^r$ as follows:

(i) $\mathbf{F}^r$ is multiplied by matrix M to obtain $\mathbf{J}^r(\mathbf{x}_t, z')$ given the local NVF formulation, see Eq. (6).

(ii) The forward transverse Fourier transformation F is applied to $\mathbf{J}^r(\mathbf{x}_t, z')$ to provide $\mathbf{J}^r(\mathbf{k}_t, z')$.

(iii) Matrix G applies the spectral Green function operations on $\mathbf{J}^r(\mathbf{k}_t, z')$ to obtain $\mathbf{E}^{s,r}(\mathbf{k}_t, z)$, see Eq. (3).

(iv) The inverse transverse Fourier transformation F^{-1} is applied to $\mathbf{E}^{s,r}(\mathbf{k}_t, z)$ to retrieve $\mathbf{E}^{s,r}(\mathbf{x}_t, z)$.

(v) Matrix C is applied on $\mathbf{F}^r$ to get $\mathbf{E}^r(\mathbf{x}_t, z')$ given the local NVF formulation, see Eq. (7).

(vi) $\mathbf{E}^{s,r}(\mathbf{x})$ is subtracted from $\mathbf{E}^r(\mathbf{x})$ to obtain $\mathbf{E}^{i,r}(\mathbf{x})$.

The iterative Krylov subspace method stops if $||(C - F^H GFM)\mathbf{F}^r - \mathbf{E}^i|| < \epsilon_c$, where ϵ_c is a pre-chosen threshold value, i.e., the solver tolerance. Note that step (v) can be also be executed before step (i), but the sequential flow through steps (ii) up to (iv) is fixed. More details can be found in [6].

Figure 3 and Algorithm 2 display the parallelized spatial spectral VIE algorithm, which both depict it as the matrix-vector product of a linear system of the following form

$$(C - F^{-1}P^T GPFM)\mathbf{F} = \mathbf{E}^i. \tag{22}$$

For exploiting the independence of $\chi(\mathbf{x}_t, z)$ along the z-direction and of $\mathcal{G}$ in the $\mathbf{k}_t$-plane, we introduce permutation operations in form of the matrices P and P^T, where ^{T} represents the transpose. These permutation operations are essentially reshapes of the data, such that C, M, and G become block-diagonal matrices. The algorithm starts by performing the local NVF formulation operations coupled to matrices C and M in parallel, owing to their independence with respect to each other. In addition, the local NVF formulation is only independent in the z-direction, since it is an operation applied to all cross-sections per z-sample. Therefore, we execute these operations only in parallel over the z-direction by dividing the z-samples over the CPU cores. The workload per z-sample is identical, since the

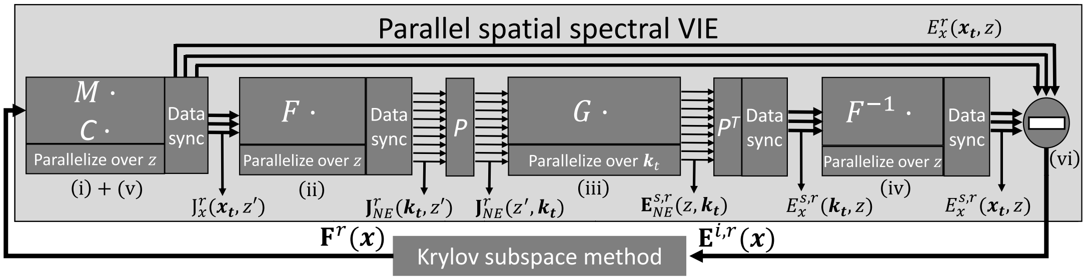

Parallel spatial spectral VIE
$E_x^r(\boldsymbol{x_t}, z)$
$M \cdot$
$C \cdot$
Parallelize over z
Data sync
(i) + (v)
$\mathbf{J}_x^r(\boldsymbol{x_t}, z')$
$F \cdot$
Parallelize over z
Data sync
(ii)
$\mathbf{J}_{NE}^r(\boldsymbol{k_t}, z')$
P
$\mathbf{J}_{NE}^r(z', \boldsymbol{k_t})$
$G \cdot$
Parallelize over $\boldsymbol{k}_t$
(iii)
$\mathbf{E}_{NE}^{s,r}(z, \boldsymbol{k_t})$
P^T
Data sync
$E_x^{s,r}(\boldsymbol{k_t}, z)$
$F^{-1} \cdot$
Parallelize over z
Data sync
(iv)
$E_x^{s,r}(\boldsymbol{x_t}, z)$
(vi)
$\mathbf{F}^r(\boldsymbol{x})$
Krylov subspace method
$\mathbf{E}^{i,r}(\boldsymbol{x})$

Figure 3. An overview of the parallel spatial spectral VIE operations given the r^{th} step of a Krylov subspace method. The Roman numerals refer to the original and sequential flow of the spatial spectral VIE operations.

number of Gabor coefficients per z-sample and per Cartesian coordinate in $\mathbf{F}^r$ is the same. Therefore, we expect improved load balancing if the number of z-samples is a multiple of the number of CPU cores. The next operation concerns data synchronization, which can be a time-consuming operation within an algorithm [38]. The array $\mathbf{J}^r(\mathbf{x}_t, z')$ is passed from this data synchronization point to the forward transverse Fourier transformation F. These transformations, each with an identical workload, are performed independently per Cartesian component and z-sample. Ideally, the number of CPU cores is a multiple of three times the number of z-samples, to ensure load balance. The next part concerns the spectral Green-function operations, which requires the knowledge of all z-samples of $\mathbf{J}^r(\mathbf{k}_t, z')$ per independent complex-plane region shown in Fig. 2(a). As mentioned in Section 2, the workload per region differs, which results in load-balance issues if we compute the spectral Green-operations G of all regions in parallel. Consequently, we focus on the parallel execution of the Green-function operations per region. The structure of Eq. (3) makes it possible to compute the recursions in the z-direction per $\mathbf{k}_t$-sample of $\mathcal{G}(\mathbf{k}_t, z|z') \cdot \mathbf{J}^r(\mathbf{k}_t, z')$ independently. However, it changes the order of accessing the elements of $\mathbf{J}^r(\mathbf{k}_t, z')$ from row-major to column-major or vice versa, depending on the chosen programming language. This typically increases the overall memory-access time. Therefore, we use the second and third synchronization point to pass $\mathbf{J}^r(\mathbf{k}_t, z')$ per complex-plane region to the array permutations P and P^T to facilitate memory-contiguous alignment during the spectral Green-function operations G, in view of the parallel evaluation over $\mathbf{k}_t$. The inverse Fourier transformation F^{-1} follows the same strategy of the first Fourier transformation, namely a parallelization over the three Cartesian components and over the samples in the z-direction. The last data synchronization point is used to gather $\mathbf{E}^{s,r}(\mathbf{x}_t, z)$ such that we continue by combining $\mathbf{E}^r(\mathbf{x}_t, z)$ and $\mathbf{E}^{s,k}(\mathbf{x}_t, z)$ element-wise in a parallel manner. In the Krylov subspace method, we check if $||(C - F^{-1}P^TGPFM)\mathbf{F}^r - \mathbf{E}^i|| < \epsilon_c$. Overall, the spatial spectral VIE operations do provide opportunities for parallel computing, but it follows from the above that most operations are either limited to a parallel execution in the longitudinal direction or the transverse direction.

5. GRATING SIMULATIONS

5.1. Implementation and Computation Details

The computations were performed on an Intel Xeon Gold 6148 processor, which has 20 CPU cores. The computer is also equipped with 755 GiB DDR4-2666 MHz RAM. The entire 3D spatial spectral VIE method is developed in FORTRAN95 and it is compiled with the Intel FORTRAN compiler. We make use of the OpenMP API for parallel computing and for measuring the wall-clock via the routine `omp_get_wtime()`.

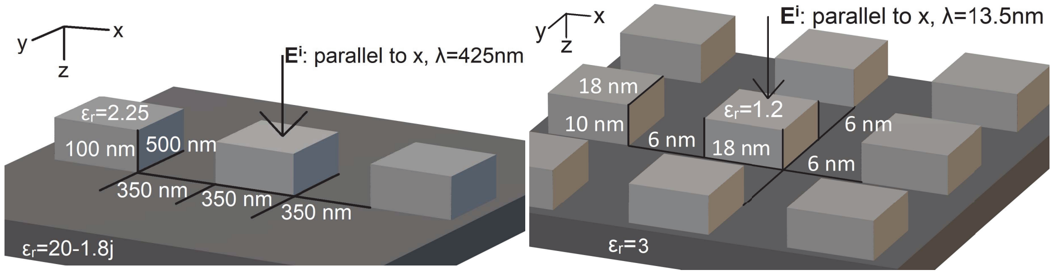
5.2. Description of the Two Grating Cases

We evaluate the performance of the parallel Gabor-frame expansion of scatterers and the parallel spatial spectral VIE with two different demonstrations. The first demonstration concerns a grating

Algorithm 2: Pseudocode parallel spatial spectral VIE operations given the r^{th} step of a Krylov subspace method.

```
Input: The $r^{th}$ iteration of auxiliary vector field $\mathbf{F}^r$, the number of samples in the $x$-, $y$-, and $z$-direction as $N_x$, $N_y$, $N_z$
Output: The $r^{th}$ iteration of $\mathbf{E}^r$
do parallel $z = 1, N_z$
    do $y = 1, N_y$
        do $x = 1, N_x$
            Task 1: apply normal vector field operator
            Compute $\mathbf{E}^r$ by the element-wise evaluation $\mathbf{E}^r(x, y, z) = C \cdot \mathbf{F}^r(x, y, z)$
            Compute $\mathbf{J}^r$ by the element-wise evaluation $\mathbf{J}^r(x, y, z) = M \cdot \mathbf{F}^r(x, y, z)$
----------------
  Data sync
----------------
do parallel $z = 1, N_z$
    Task 2: apply transverse Fourier operator
    Compute $\mathbf{J}^r$ by the transverse Fourier transformation $\mathbf{J}^r(k_x, k_y, z) = F \cdot \mathbf{J}^r(x, y, z)$
----------------
  Data sync
----------------
do $z = 1, N_z$
    do $kx = 1, N_x$
        do $ky = 1, N_y$
            Task 3: apply permutation operator
            Restructure $\mathbf{J}^r$ by permutation $\mathbf{J}^r(z, k_x, k_y) = P \cdot \mathbf{J}^r(k_x, k_y, z)$
do parallel $\mathbf{k_t} = 1, N_x N_y$
    do $z = 1, N_z$
        Task 4: apply Green operator
        Compute $\mathbf{E}^{s,r}$ by $\mathbf{E}^{s,r}(z, k_x, k_y) = G \cdot \mathbf{J}^r(z, k_x, k_y)$
do $z = 1, N_z$
    do $kx = 1, N_x$
        do $ky = 1, N_y$
            Task 5: apply inverse permutation operator
            Restructure $\mathbf{E}^{s,r}$ by inverse permutation $\mathbf{E}^{s,r}(k_x, k_y, z) = P^{-1} \cdot \mathbf{E}^{s,r}(z, k_x, k_y)$
do parallel $z = 1, N_z$
    Task 6: apply inverse transverse Fourier operator
    Compute $\mathbf{E}^{s,r}$ by the inverse Fourier transformation $\mathbf{E}^{s,r}(x, y, z) = F^{-1} \cdot \mathbf{E}^{s,r}(k_x, k_y, z)$
----------------
  Data sync
----------------
do parallel $\mathbf{x} = 1, N_x N_y N_z$
    Task 7: Substract scattered electric field from total electric field
    Compute $\mathbf{E}^{i,r}$ by the inverse Fourier transformation $\mathbf{E}^{i,r}(x, y, z) = \mathbf{E}^r(x, y, z) - \mathbf{E}^{s,r}(x, y, z)$
----------------
  Data sync
----------------
```

consisting of 36 repeating dielectric scatterers, as partly shown in Fig. 4(a). This case is also treated in [6] and [9]. All scatterers are placed on top of a dielectric half-space with a relative permittivity of $\varepsilon_r = 20.21 - 1.8j$, while the other (top) half-space is free space with $\varepsilon_r = 1$. Each scatterer has a relative permittivity $\varepsilon_r = 2.25$. The incident field is a plane wave, polarized in the x-direction, originating from the top half-space with a wavelength of 425 nm in free space. Its propagation direction is along the z-axis. The volume of the simulation domain is $60\lambda \times 6\lambda \times 1/6\lambda$. The discretization in the z-direction consists of 21 piecewise-linear functions with a stepsize of 5 nm. The Gabor frames in the $\mathbf{x}_t$-plane are characterized by $X = Y = 500\,\text{nm}$, in combination with all oversampling parameters set to $\sqrt{2/3}$. We only consider Gabor coefficients $m_x \in \{-40, \ldots, 40\}$, $m_y \in \{-4, \ldots, 4\}$, and $n_x, n_y \in \{-7, \ldots, 7\}$ within the simulation domain. This provides a resolution of approximately $\lambda/10.4$ in both the x- and y-



(a)

(b)

Figure 4. (a) A part of the total simulation domain of the $60\lambda^3$ grating. (b) A part of the total simulation domain of the $1300\lambda^3$ grating.

direction. A local NVF formulation is applied in combination with the Gabor-frame discretization, which means that each rectangular scatterer cross-section is a combination of four triangular cross-sections pieced together [25]. So, we compute the Gabor coefficients for 144 triangular cross-sections, instead of the 36 rectangular cross-sections. Overall, we solve an electromagnetic scattering problem with a volume of $60\lambda^3$ consisting of 36 scattering objects placed on top of a dielectric half-space represented by 10.3 million unknowns with a peak memory usage of 16 GiB of RAM.

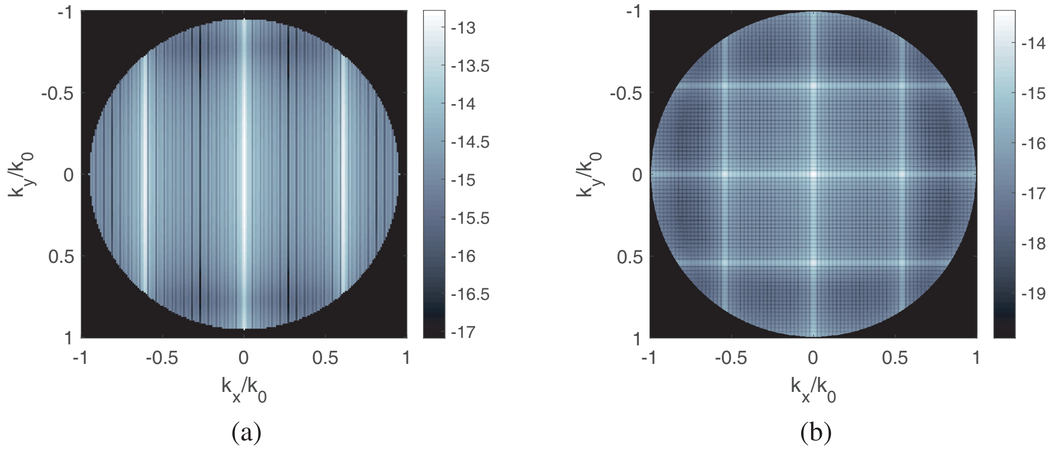
Figure 4(b) shows a part of the second demonstration setup, which is a grating consisting of $23 \times 23 = 529$ dielectric scatterers. The scatterers are placed on top of a dielectric half-space with a relative permittivity of $\varepsilon_r = 3$. Each scattering object has a volume of $18 \times 18 \times 10\,\text{nm}^3$. The spacing between each scattering object is 6 nm in both the x- and y-direction. Every scatterer has a relative permittivity $\varepsilon_r = 1.2$. The top half-space is free space. The incident field is a plane wave characterized by its wavelength of $\lambda = 13.5\,\text{nm}$, propagation direction along the z-axis, and polarization in the x-direction. The volume of this simulation domain is $42\lambda \times 42\lambda \times 0.74\lambda$. We use 21 piecewise-linear functions in the z-direction with $\Delta_z = 0.5\,\text{nm}$. The Gabor frames are set with $X = Y = 16\,\text{nm}$ for the window functions and an oversampling of $\sqrt{2/3}$. The $\mathbf{x}_t$-planes of the simulation domain are spanned by the Gabor coefficients $m_x \in \{-25, \dots, 25\}$, $m_y \in \{-25, \dots, 25\}$, and $n_x, n_y \in \{-13, \dots, 13\}$. This provides a resolution of $\lambda/18.6$ in each direction of the $\mathbf{x}_t$-plane. The application of a local NVF formulation means that we actually work with the Gabor coefficients of 2116 triangular cross sections instead of 529 rectangular ones. So, we solve an electromagnetic scattering problem with 119.5 million unknowns consisting of 529 scatterers in a volume of $1300\lambda^3$, while having a peak memory usage of 90 GiB of RAM. We identify the two grating cases by the volumes of their respective computational domains, i.e., $60\lambda^3$ and $1300\lambda^3$.

5.3. Accuracy

Figure 5 contains the magnitude of the computed spectral-domain electric field, i.e., $|\mathbf{E}(k_x, k_y)|$, for both the $60\lambda^3$ grating and the $1300\lambda^3$ grating. These two fields are located in the xy-plane right on top of the scatterers. The spectral-domain electric fields depict a planar plane-wave decomposition of the far-field response, where each plane wave is defined by the combination of k_x and k_y coordinates. We only show the part of the propagating plane waves within the Ewald circle for an NA smaller than 1, i.e., 0.95 in Fig. 5(a) and 0.99 in Fig. 5(b).

The spectral-domain electric field of the $60\lambda^3$ grating is compared to a simulation by the sequential version of the spatial spectral VIE discussed in [9]. The spectral-domain electric fields for the sequential and the parallel version are obtained after 32 iterations of BiCGstab(ℓ) with the pre-chosen threshold value $\epsilon_c = 10^{-5}$. A difference by a relative ℓ^2 norm of approximately 10^{-15} was observed between the spectral-domain electric field obtained via sequential and parallel spatial spectral VIE method.

We obtained the spectral-domain electric field of the $1300\lambda^3$ grating after 16 iterations of BiCGstab(ℓ). We have used the RETICOLO software [39] for periodic-grating analysis based on the rigorous coupled wave analysis (RCWA) to evaluate the precision obtained with the spatial spectral solver. Table 1 displays the amplitude of the spectral-domain electric fields of RETICOLO and the



(a)

(b)

Figure 5. The spectral far-field magnitude $|\mathbf{E}(k_x, k_y)|$ in log10 scale in case of (a) 60 λ^3 grating (b) $1300\lambda^3$ grating.

Table 1. The propagating mode amplitudes within the planar plane-wave decomposition given the $1300\lambda^3$ grating and the RCWA framework. The term *error* refers to a relative error as defined in (24).

$k_x/k_0[-]$	± 0.54	± 0.54	0	0
$k_y/k_0[-]$	± 0.54	0	± 0.54	0
$\mathbf{E}_{reticolo}$ [V/m]	$0.0008 + 0.0091i$	$0.0087 - 0.0343i$	$0.0023 - 0.0436i$	$0.0715 - 0.1120i$
$\mathbf{E}_{spatspec}$ [V/m]	$0.0007 + 0.0090i$	$0.0087 - 0.0340i$	$0.0023 - 0.0433i$	$0.0713 - 0.1117i$
$error[-]$	0.0010	0.0018	0.0021	0.0028

spatial spectral solver as

$$|\mathbf{E}(k_x, k_y)| = \sqrt{E_x^2(k_x, k_y) + E_y^2(k_x, k_y) + E_z^2(k_x, k_y)}, \tag{23}$$

at (k_x, k_y)-locations of the propagating modes within the RCWA framework. Note that the amplitudes originating from the spatial spectral VIE as in Fig. 5(b) are re-normalized in accordance with the RETICOLO approach, to accurately evaluate the differences between the two solvers. This normalization concerns the scaling by the flux of the incident Poynting vector within a 2D unit cell [39]. We evaluate the precision per $\mathbf{k}_t$ sample in Table 1 by means of the definition

$$error = \frac{|\mathbf{E}_{spatspec}(k_x, k_y) - \mathbf{E}_{reticolo}(k_x, k_y)|}{|\mathbf{E}_{reticolo}(0, 0)|}, \tag{24}$$

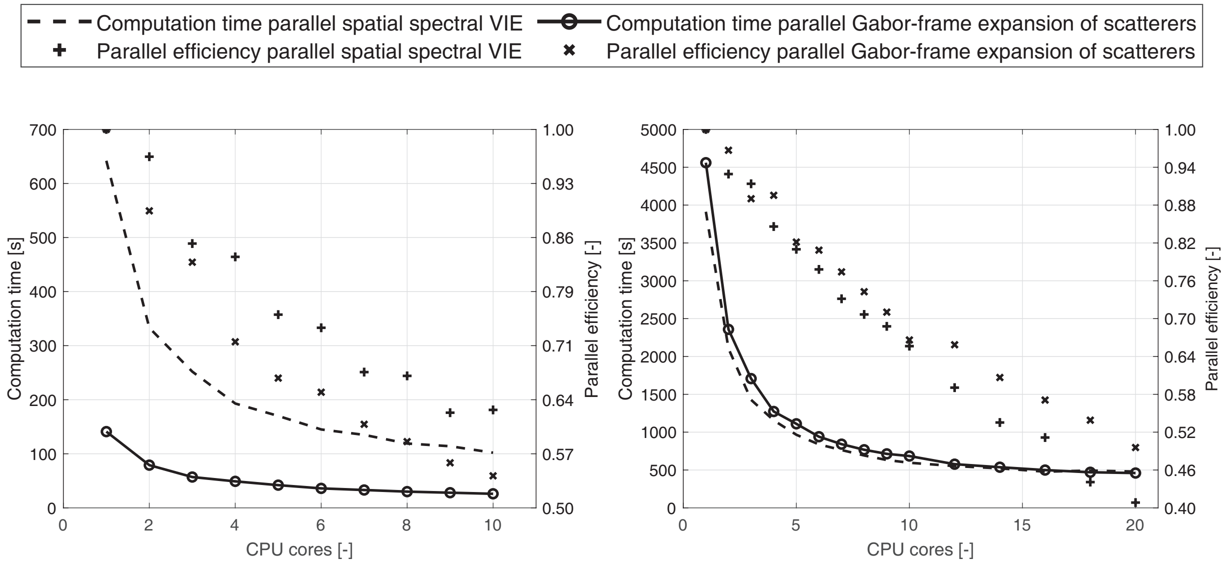
since the peak at $\mathbf{k}_t = \mathbf{0}$ concerns the clear majority of the far-field energy. Overall, results produced by the Reticolo software and the spatial spectral VIE are quite similar.

As a side note, we mention that the results in Fig. 5 are displayed in Volt per meter. Real experimental measurements of the far-field response have to cope with quantization effects of photons [40]. The application of far-field measurement improvements, such as [41, 42], may then yield improved accuracy between experimental data and simulations.

5.4. Performance

The computation times and parallel efficiencies of the parallel Gabor-frame expansion of the scatterers and the parallel spatial spectral VIE for both grating cases are displayed in Fig. 6(a) and Fig. 6(b), as functions of the number of used CPU cores N_{core}. The computation time $T_{N_{core}}$ is measured by the OpenMP routine `omp_get_wtime()`, which uses the elapsed wall-clock time. We measure the parallel efficiency as $T_1/(T_{N_{core}} \times N_{core})$. Note that T_1 is the computation time via the parallel computing extensions of both algorithms with only a single CPU core. This provides a more objective insight into the scalability of the parallel algorithms, since the sequential versions of both algorithms are different in structure. We do not provide the complete preprocessing time of the spatial spectral VIE method, since the computation of the Gabor expansion coefficients of the scatterers is by far the most dominant component, as mentioned in Section 3.1.

The results displayed in Fig. 6(a) show that the combined computation time decreases from 783 seconds to 128 seconds when we increase the number of CPU cores from 1 to 10, for the $60\lambda^3$ case. The parallel spatial spectral VIE is the main contributor to the decrease of the combined computation time in the absolute sense. The parallel spatial spectral VIE outperforms the parallel Gabor-frame expansion of scatterers in terms of parallel efficiency. We do note that the decrease in computation time stagnates, which means that both parallel algorithms exhibit sub-linear scaling, as shown by the parallel efficiency. There are two main contributors to this stagnation for the parallel spatial spectral VIE: the Fourier transformations F and F^{-1} and the spatial spectral Green-function operations P^TGP in memory-contiguous order. The Fourier transformations initially take up to 302 seconds with a single core, while its 10-core version requires 44 seconds. This is approximately only 6.9 times faster while using 10 cores. The memory requirement for each Fourier transformation surpasses 24 MB, since the involved number of double-precision complex numbers is equal to the number of Gabor coefficients multiplied by the oversampling parameter $p^2 = 9$, owing to $\alpha_x\beta_x = q/p = 2/3$. Since the L3 cache of the CPU is fitted with 27.5 MB, we expect that the Fourier transformations require high data movement and, therefore, it is a possible bottleneck. It is not uncommon that the memory size of (large) FFTs is challenging for effective parallel computing [43]. The spatial spectral Green-function operations scale from 280 seconds to 44 seconds when going from a single core to 10 cores, which is only approximately 6.4 times faster. The most time-consuming component of the spatial spectral Green-function operations are the permutations P and P^T, to work in a column-major manner to reduce the cache misses during the operations coupled to the operator $\mathcal{G}$. It requires 32 seconds for the 10 cores. This subpar parallel efficiency for the permutation operations are expected, since these operations are typically slower due to the large number of data element accesses [44]. A part of the stagnation in the parallel Gabor-frame expansion of scatterers originates from not being able to properly divide the 144 scatterers over the



(a) (b)

Figure 6. The computation time and efficiency of the parallel Gabor-frame expansion of scatterers and the parallel spatial spectral VIE as a function of the number of used CPU cores in case of (a) $60\lambda^3$ grating, (b) $1300\lambda^3$ grating.

CPU cores equally. Another reason is that the number of Gabor coefficients that we need to compute for the cross-sections is not always the same for each cross-section, because the spacing between the scatterers is not aligned with the Gabor-frame discretization, see Eq. (21). We observed that the difference between the fastest and slowest computation of the Gabor coefficients for a cross-section was more than 0.5 seconds, which is severe given that we require approximately 26 seconds for 144 scatterers while using 10 cores. Overall, we obtain the scattered electric fields in a simulation domain of $60\lambda^3$ more than six times faster when using 10 CPU cores versus a single CPU core, but both algorithms clearly show sub-linear parallel scaling.

Figure 6(b) provides the insight that the combined computation time for the $1300\lambda^3$ grating decreases from 8472 seconds with a single core to 939 seconds with 20 cores, which is more than 9 times faster. Interestingly, these results show that the parallel Gabor-frame expansion of scatterers is the main contributor to this time reduction. It outperforms the parallel spatial spectral VIE in terms of parallel efficiency for all numbers of used CPU cores. This does not align with the timing and parallel efficiency results of the $60\lambda^3$ grating. We expect that the $60\lambda^3$ grating did not provide enough scatterers to properly evaluate the parallel scaling capabilities of the parallel Gabor-frame expansion of scatterers in comparison to the $1300\lambda^3$ grating. To gain more insight into the robustness of the parallel efficiency, we created another additional synthetic test case, which reuses the description of the $60\lambda^3$ grating. Now, we copied each of the 36 scatterers three times and placed these three copies per scatterer on the exact positions of the original scatterer such that we obtain a (synthetic) grating with 144 rectangular scatterers. The use of the local NVF formulation leads then to a total of 576 triangular scatterers. This specific procedure makes it possible to maintain the exact same discretization as originally for the $60\lambda^3$ grating, while testing whether the parallel Gabor-frame expansion of scatterers has an improved parallel efficiency in case of a larger number of scatterers. The parallel Gabor-frame expansion of the 576 scatterers with 1 core and 10 cores required 543 seconds and 83 seconds, respectively. Thus, we note an increase in computation time, while fixing the discretization and increasing the number of scatterers. Further, we observe a parallel efficiency of 0.65, which is higher than the parallel efficiency for the $60\lambda^3$ grating with the original 36 rectangular scatterers and it is also more similar to the parallel efficiency for the $1300\lambda^3$ grating with 10 cores. Nevertheless, the parallelization algorithms still exhibit sub-linear scaling for the $1300\lambda^3$ grating. The Fourier transformations are the key contributors to the stagnation for the parallel spatial spectral VIE. The computation time for these transformations decreases from 2785 seconds to 348 seconds when the number of CPU cores increases from 1 to 20. This time, the permutations P and P^T decreased from 652 seconds to 57 seconds, while the time related to G decreased from 288 seconds to 17 seconds. Overall, we obtain accurate electric fields scattered by 2116 scattering objects in a simulation domain larger than $1300\lambda^3$ within 939 seconds when using 20 CPU cores.

6. CONCLUSION

We presented parallel computing solutions for the three-dimensional spatial spectral volume integral equation method for the computation of electromagnetic scattering from finite dielectric scatterers. The first part focused on exploiting the locality of the window functions within the Gabor-frame expansion to compute Gabor coefficients of cross-sections with a polygonal shape simultaneously in combination with the efficient spectral transformation method of [9]. The second part concerned the decomposition and restructuring of the spatial spectral volume integral equation into (partially) independent components to enable parallel computing for its matrix-vector product. Two grating cases were used as numerical demonstrations to evaluate the computation time and parallel efficiency of the spatial spectral volume integral equation with both parallel extensions, as a function of the number of employed CPU cores. We reduced the computation time by more than a factor of 6, while using 10 CPU cores for the grating with a volume of $60\lambda^3$ for $\lambda = 425\,\text{nm}$. The reduction in computation time was more than a factor of 9, while using 20 CPU cores, for the second grating with a volume of $1300\lambda^3$ with $\lambda = 13.5\,\text{nm}$. Overall, we reduced the wall-clock computation time for computing the electromagnetic scattering behavior of the gratings significantly by means of parallel computing. Nonetheless, the numerical experiments showed stagnation in the parallel efficiency due to the Fourier transformations, the permutation operations, and the load imbalance of the size-dependent Gabor-frame expansion per cross-section.

REFERENCES

1. Rendon-Barraza, C., E. A. Chan, G. Yuan, G. Adamo, T. Pu, and N. I. Zheludev, "Deeply subwavelength non-contact optical metrology of sub-wavelength objects," *APL Photonics*, Vol. 6, 2021.
2. Dasari, P., J. Li, J. Hu, Z. Liu, O. Kritsun, and C. Volkman, "Scatterometry metrology challenges of EUV," *Proc. of SPIE*, Vol. 8324, 83240M, 2012.
3. Ansuinelli, P., W. M. J. Coene, and H. P. Urbach, "Automatic feature selection in EUV scatterometry," *Applied Optics*, Vol. 58, No. 22, 5916–5923, 2019.
4. Kumar, N., P. Petrik, G. K. P. Ramanandan, O. El Gawhary, S. Roy, S. F. Pereira, W. Coene, and H. Urbach, "Reconstruction of sub-wavelength features and nano-positioning of gratings using coherent Fourier scatterometry," *Optics Express*, Vol. 22, No. 20, 24 678–24 688, 2014.
5. Gross, H., A. Rathsfeld, F. Scholze, and M. Bär, "Profile reconstruction in extreme ultraviolet (EUV) scatterometry: Modeling and uncertainty estimates," *Meas. Sci. Technol.*, Vol. 20, No. 10, 105102, 2009.
6. Dilz, R. J., M. G. M. M. van Kraaij, and M. C. van Beurden, "A 3D spatial spectral integral equation method for electromagnetic scattering from finite objects in a layered medium," *Opt. Quant. Electron.*, Vol. 50, No. 206, 2018.
7. Dilz, R. J., "A spatial spectral domain integral equation solver for electromagnetic scattering dielectric layered media," Ph.D. Dissertation, Eindhoven University of Technology, 2017.
8. Gohberg, I. and I. Koltracht, "Numerical solution of integral equations, fast algorithms and Krein-Sobolev equation," *Numerical Mathematics*, Vol. 47, No. 2, 237–288, 1985.
9. Eijsvogel, S., L. Sun, F. Sepehripour, R. J. Dilz, and M. C. van Beurden, "Describing discontinuous finite 3D scattering objects in Gabor coefficients: Fast and accurate methods," *J. Opt. Soc. Am. A*, Vol. 39, No. 1, 86–96, 2022.
10. Dilz, R. J. and M. C. van Beurden, "Computational aspects of a spatial-spectral domain integral equation for scattering by objects of large longitudinal extent," *2017 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, 637–640, 2017.
11. Levinson, H. J., *Principles of Lithography*, 4th Edition, SPIE Press, 2019.
12. Bakshi, V., *EUV Lithography*, 2nd Edition, SPIE Press, 2018.
13. Van den Berg, P. M., *Forward and Inverse Scattering Algorithms Based on Contrast Source Integral Equations*, 1st Edition, Wiley, 2021.
14. Solis, D., F. Obelleiro, and J. Taboada, "Surface integral equation-domain decomposition scheme for solving multiscale nanoparticle assemblies with repetitions," *IEEE Photonics Journal*, Vol. 8, No. 5, 1–14, 2016.
15. Chanaud, M., L. Giraud, D. Goudin, J. J. Pesqué, and J. Roman, "A parallel full geometric multigrid solver for time harmonic Maxwell problems," *SIAM J. Sci. Comput.*, Vol. 36, No. 2, C119–C138, 2014.
16. Ruiz-Cabello, M., M. Abelenkovs, L. M. Diaz Angulo, C. Cobos Sanchez, F. Moglie, and S. G. Salvadore, "Performance of parallel FDTD method for shared- and distributed-memory architectures: Application to bioelectromagnetics," *PLOS ONE*, Vol. 15, No. 9, e0238115, 2020.
17. Vaccari, A., A. Lesina, L. Cristoforetti, and R. Pontalti, "Parallel implementation of a 3D subgridding FDTD algorithm for large simulations," *Progress In Electromagnetics Research*, Vol. 120, 263–292, 2011.
18. Pan, X. M., W. C. Pi, and X. Q. Sheng, "On OpenMP parallelization of the multilevel fast multipole algorithm," *Progress In Electromagnetics Research*, Vol. 112, 199–213, 2011.
19. Yang, M. L., H. W. Gao, X. M. Sun, and X. Q. Sheng, "Fast domain decomposition methods of FE-BI-MLFMA for 3D scattering/radiation problems (invited paper)," *Progress In Electromagnetics Research*, Vol. 155, 39–52, 2016.

20. MacKie-Mason, B., A. Greenwood, and Z. Peng, "Adaptive and parallel surface integral equation solvers for very large-scale electromagnetic modeling and simulation (invited paper)," *Progress In Electromagnetics Research*, Vol. 154, 143–162, 2015.
21. Wait, J. R., *Electromagnetic Waves in Stratified Media*, Pergamon Press, 1970.
22. Dilz, R. J. and M. C. van Beurden, "A domain integral equation approach for simulating two dimensional transverse electric scattering in a layered medium with a Gabor frame discretization," *Journal of Computational Physics*, Vol. 345, 528–542, 2017.
23. Dilz, R. J., M. G. M. M. van Kraaij, and M. C. van Beurden, "The 2D TM scattering problem for finite objects in a dielectric stratified medium employing Gabor frames in a domain integral equation," *Journal of the Optical Society of America A*, Vol. 8, No. 34, 1315–1321, 2017.
24. Barzegar, E., S. J. L. Eijndhoven, and M. C. van Beurden, "Scattered field in random dielectric inhomogeneous media: A random resolvent approach," *Progress In Electromagnetics Research B*, Vol. 62, 29–47, 2015.
25. Van Beurden, M. C. and I. D. Setija, "Local normal vector field formulation for periodic scattering problems formulated in the spectral domain," *J. Opt. Soc. Am. A*, Vol. 34, No. 2, 224–234, 2014.
26. Li, L., "Use of fourier series in the analysis of discontinuous periodic structures," *J. Opt. Soc. Am. A*, Vol. 13, 1019–1023, 1996.
27. Popov, E. and M. Neviére, "Maxwell equations in fourier space: fast-converging formulation for diffraction by arbitrary shaped, periodic, anisotropic media," *J. Opt. Soc. Am. A*, Vol. 18, 1019–1023, 2001.
28. Bastiaans, M., "Gabor's expansion and the Zak transform for continuous-time and discrete-time signals: Critical sampling and rational oversampling," EUT Report. E, Vol. 95-E-295 Fac. of Electrical Engineering, Eindhoven University of Technology, 1995.
29. Strohmer, T., "Approximation of dual gabor frames, window decay, and wireless communications," *Applied and Computational Harmonic Analysis*, Vol. 11, No. 2, 243–262, 2001.
30. Janssen, A. and P. Sondergaard, "Iterative algorithms to approximate canonical gabor windows: Computational aspects," *Journal of Fourier Analysis and Applications*, Vol. 31, No. 1, 211–241, 2007.
31. Janssen, A., "Some Weyl-Heisenberg frame bound calculations," *Indagationes Math*, Vol. 7, No. 7, 165–183, 1996.
32. Jenkins, W., *Digital Signal Processing Handbook*, CRC, 2010.
33. Christensen, O., *Frames and Bases: An introductory Course*, Birkhäuser, 2008.
34. Von Praun, C. and D. Padua, eds., *Encyclopedia of Parallel Computing*, Springer, 2011.
35. Yang, C., *Introduction to GIS Programming and Fundamentals with Python and ArcGIS*, CRC, 2017.
36. Hermanns, M., "Parallel programming in Fortran 95 using OpenMP," 2002, https://www.openmp.org/wp-content/uploads/F95_OpenMPv1 v2.pdf.
37. Sleijpen, G. L. G. and D. R. Fokkema, "BiCGstab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum," *Electron. Trans. Numer. Anal.*, Vol. 1, No. 1, 11–32, 1993.
38. Chandra, R., R. Menon, L. Dagum, D. Kohr, D. Maydan, and J. Mcdonald, *Parallel Programming in OpenMP*, Morgan Kaufmann, 2001.
39. Hugonin, J. P. and P. Lalanne, "RETICOLO software for grating analysis," 2022, https://arxiv.org/abs/2101.00901.
40. EMVA, "Standard for characterization of image sensors and cameras," 2016, https://www.emva.org/wp-content/uploads/EMVA1288-3.0.pdf.
41. Yu, C.-Y., C.-Y. Lin, S.-C. Yang, and H.-Y Lin, "Eight-scale image contrast enhancement based on adaptive inverse hyperbolic tangent algorithm," *MDPI Journal of Algorithms*, Vol. 7, 597–607, 2014.
42. Versaci, M., F. C. Morabito, and G. Angiulli, "Adaptive image contrast enhancement by computing distances into a 4-dimensional fuzzy unit hypercube," *IEEE Access*, Vol. 5, 26922–26931, 2017.

43. Kulkarni, A., F. Franchetti, and J. Kovacevic, "Algorithm design for large scale FFT-based simulations on cpu-gpu platforms," *47th International Conference on Parallel Processing*, 2018.

44. Hanounik, B. and X. Hu, "Linear-time matrix transpose algorithms using vector register file with diagonal registers," *Proceedings 15th International Symposium on Parallel and Distributed Processing*, 8, 2001.