Optimal Parallel Algorithm of Fast Inverse Laplace Transform for Electromagnetic Analysis

Di Wu, Seiya Kishimoto, and Shinichiro Ohnuki, *Member, IEEE*

***Abstract*—Fast inverse Laplace transform (FILT) is a simple and concise technique for performing inverse Laplace transforms. This letter develops an efficient parallel algorithm for FILT for solving various electromagnetic problems. This algorithm can be easily combined with existing computational methods used in electromagnetics to realize efficient time- and frequency-domain analyses.**

***Index Terms*—Complex frequency domain, electromagnetics, inverse Laplace transform, parallel computation, time-domain analysis.**

I. Introduction

THE Laplace transform is a well-known and efficient mathematical tool for solving partial differential and ordinary equations in physics and engineering problems [1]–[4] in areas such as system modeling, analysis of electrical and electronic circuits, digital signal processing, nuclear physics, and process control [5]. It can be used to calculate the output of a linear time-invariant system by convolving the unit impulse response with the input signal. The Laplace transform essentially converts the convolution operation into a multiplication operation in the Laplace space, where algebraic manipulation is easier.

Generally, a forward Laplace transform is relatively easy to apply. However, an inverse Laplace transform proves difficult when it is used for solving practical problems, such as the design of electromagnetic (EM) devices. To perform an inverse Laplace transform, many numerical implementations have been proposed [6]. The fast inverse Laplace transform (FILT) is an easy and concise implementation [7] that has been successfully applied to practical EM analyses for plasmonic antennas [8], ground-penetrating radar [9], transmission lines [10], biological media [11], low-frequency issues, and problems related to dc components [12]–[14]. FILT can obtain the time-domain response independently at an arbitrary single time; therefore, it is convenient for performing parallel computations. With regard to computational speed, completely parallel computing through multiple computing nodes can significantly reduce the processing time.

We recently developed a novel technique for conducting both time- and frequency-domain analyses; specifically, we combined FILT with the finite-difference complex-frequency-domain (FDCFD) method [12] for the near-field analysis of plasmonic objects. We also developed a time-division algorithm for the finite-difference time-domain (FDTD) technique [15]. This algorithm combines the FDTD [16], FDCFD, and FILT methods. It performs FDTD computations by using the initial condition of an EM field that is obtained using the FDCFD and FILT methods [12]. By substituting the FDCFD result at time t_1 into the FDTD process, the time response in the period $[t_1\ t_2]$ can be computed without computing from $t = 0$ and sequentially updating the EM field. Other techniques, such as curve fitting, macro modeling, and fast Fourier transform (FFT)-based numerical inversion of Laplace transform [17]–[19], are available for evaluating the time-domain response from the complex-frequency domain (CFD) response obtained using FDCFD. The proposed FILT algorithm can be used to directly define the pole for the inverse Laplace transform without requiring a fitting function and to easily perform error analysis. Because the inverse Laplace transform is calculated by a series of image functions, a butterfly computation method like FFT is not required. Therefore, parallel computations can be performed easily. Furthermore, the proposed method can efficiently perform calculations using only a few poles.

This letter proposes a further improvement of the parallel algorithm to study long-period problems. We discuss an optimized parallel algorithm for FILT and use it to perform parallel computations for optimizing the processing time. This method enables parallel computations for both the time domain and the CFD. We combine FILT with a recently developed FDCFD method for 3-D analyses of EM scattering from metallic objects. The computational accuracy and speed of the proposed parallel algorithm are verified.

II. Formulation

A. Original FILT Algorithm

We briefly explain the original FILT algorithm for performing inverse Laplace transforms [6]. In this algorithm, the exponential function in the integral of the inverse Laplace transform is replaced by a hyperbolic function [3], [7]. Upon usage of the residue theorem and application of the Euler transform for truncating the infinite series, the following approximated form is obtained:

$$f\ (t) = \lim_{y\to\infty} \frac{1}{2\pi j} \int_{x-jy}^{x+jy} F\,(s)\, e^{st} ds \approx f_{ec}\,(t, \alpha)$$

Manuscript received May 19, 2020; revised July 7, 2020 and August 11, 2020; accepted August 24, 2020. Date of publication August 31, 2020; date of current version December 22, 2020. This work was supported in part by the Japan Society for the Promotion of Science through the Grant-in-Aid for Scientific Research (C) under Grant 17K06401 and in part by the Nihon University College of Science and Technology Project for Research. *(Corresponding author: Shinichiro Ohnuki.)*

The authors are with the College of Science and Technology, Nihon University, Tokyo 101-8308, Japan (e-mail: cste18001@g.nihon-u.ac.jp; kishimoto.seiya@nihon-u.ac.jp; ohnuki.shinichiro@nihon-u.ac.jp).

Digital Object Identifier 10.1109/LAWP.2020.3020327

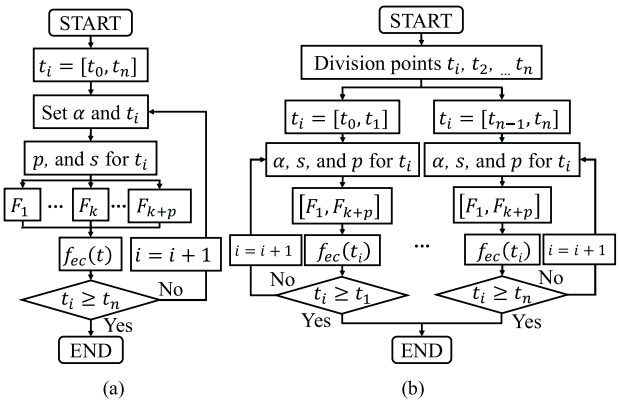

Fig. 1. Flowchart of proposed parallel FILT algorithm in (a) s domain and (b) time domain.

$$= \frac{e^{\alpha}}{t}\left(\sum_{n=1}^{k} F_n + \frac{1}{A_{p0}}\sum_{n_e=1}^{p} A_{pn_e} F_{k+n_e}\right) \tag{1}$$

where

$$F_n = (-1)^n \,\mathrm{Im}\left[F\left(s\right)\right], \quad s = \frac{\alpha + j\left(n-0.5\right)\pi}{t} \tag{2a}$$

$$A_{pp} = 1, \quad A_{p0} = 2^p, \quad A_{pn_e} = A_{pn_e-1} - \frac{(p+1)!}{n_e!\,(p+1-n_e)!} \tag{2b}$$

and k is the truncation number. The second summation is for performing the Euler transformation with p terms to expedite the convergence of the alternation series. Here, $F(s)$ represents the image function in the CFD of the original time-domain function $f(t)$. The accuracy of f_{ec} can be controlled to $10^{-\alpha}$ by using the approximation parameter α. The above equation is the original FILT equation described in [3]. It is advantageous in that the field distribution at an arbitrary time can be obtained via a simple summation of a few corresponding singularities on a complex plane.

B. Optimized Parallel Implementation

$f_{ec}(t, \alpha)$ in (1) can be obtained by summing up all the terms one by one. To obtain the time response at t_1, we can simply replace t by t_1. Each term in the summation of FILT can be computed independently; therefore, we can perform parallel computations for obtaining $f_{ec}(t, \alpha)$ in both the time domain and the CFD. Fig. 1 shows the parallel implementation of FILT for performing time response analysis. First, we set the observation time t, approximation parameter α, and s and k_1 values, following which we set the number of terms of the Euler transformation p. We can solve the image function in parallel and obtain F_n by using multiple computing nodes. Finally, summations are taken for obtaining the response at observation time t.

Fig. 1(a) shows the parallel implementation in CFD. To sum up the series in the algorithm, we can evaluate them independently in parallel. We can also parallelize the time-domain computation, as shown in Fig. 1(b). Because the FILT algorithm is used for computing the time response at each observation time, we perform the same process using multiple nodes for varying observation times.

The optimized selection for the parallel computation process for both the time domain and the CFD is discussed in Section III.

C. Double Euler Transformation

Because the proposed method is completely parallel, its efficiency is almost 100%. However, the computational cost becomes high when $k + p$ becomes very large. This often occurs in the late-time responses of EM waves for strongly resonant problems, such as designing plasmonic devices and solving cavity problems. The original FILT algorithm becomes expensive for late-time responses when t in (2a) becomes large.

In this letter, we propose an improved FILT algorithm to overcome this issue and realize rapid convergence. Upon usage of the improved algorithm, a similar computational cost can be realized even for the late-time response. The following equation shows the improved FILT algorithm using double Euler transformation:

$$f_{ec}\ (t,\alpha) = \frac{e^{\alpha}}{t}\left(\begin{array}{c}\sum\limits_{n=k_1}^{k_2} F_n + \frac{1}{A_{p0}}\sum\limits_{n_e=1}^{p} A_{pn_e} F_{k_2+n_e} \\ + \frac{1}{A_{q0}}\sum\limits_{n_e=1}^{q} A_{q(q-n_e)} F_{k_1-n_e}\end{array}\right) \tag{3}$$

where k_1 and k_2 are the lower and upper truncation number, respectively. The method for selecting these numbers is discussed in Section III. Furthermore, in the double Euler transformation, the second and third summations are applied for the upper and lower sides of the summation. This modification helps accelerate the convergence for a reduced truncation number from k_1 to k_2. Here, p and q are the numbers of terms in the Euler transformation to accelerate the convergence of f_{ec}.

D. Three-Dimensional FDCFD Formulation

In this letter, an FDCFD solver is used for calculating the EM field in the s domain. The following equation is a better alternative to Maxwell's equations because it eliminates unknown elements of the electric fields:

$$\nabla_{\mathrm{PML}} \times \left(\mu^{-1} \nabla_{\mathrm{PML}} \times \boldsymbol{E}\right) + s^2 \varepsilon \boldsymbol{E} = -s\boldsymbol{J} - \nabla_{\mathrm{PML}} \times \mu^{-1} \boldsymbol{M}$$

$$\nabla_{\mathrm{PML}} = \frac{1}{c_x}\frac{\partial}{\partial x}\boldsymbol{a_x} + \frac{1}{c_y}\frac{\partial}{\partial y}\boldsymbol{a_y} + \frac{1}{c_z}\frac{\partial}{\partial z}\boldsymbol{a_z} \tag{4}$$

where c_x, c_y, and c_z are the coordinate stretching variables for the purpose of applying a convolutional perfect matched layer (CPML) [20] as an absorbing boundary condition. In (4), the time dependence is assumed to be e^{st}, and $\boldsymbol{J}$ and $\boldsymbol{M}$, respectively, represent electric current and magnetic sources, and ε and μ, respectively, represent the permittivity and permeability of the scatterer. In this letter, (4) is used for calculating the $\boldsymbol{E}$-field to evaluate a surface plasmon on a metal surface. The permeability μ of materials for visible light can be assumed to be μ_0 because the incident wave is of visible light.

Discretizing (4) and collecting the whole element in the x-, y-, and z-directions gives

$$\left(\begin{bmatrix} D_y^2 + D_z^2 & D_x D_y & D_x D_z \\ D_x D_y & D_x^2 + D_z^2 & D_y D_z \\ D_x D_z & D_y D_z & D_x^2 + D_y^2 \end{bmatrix} - s^2 \varepsilon \mu_0 \boldsymbol{I}\right) \times \begin{bmatrix} E_x^s \\ E_y^s \\ E_z^s \end{bmatrix} = \begin{bmatrix} J_x' \\ J_y' \\ J_z' \end{bmatrix} \tag{5}$$

where $\boldsymbol{I}$ represents the unit matrix. The coefficients in the first term on the left-hand side, D_x, D_y, and D_z, are matrices for

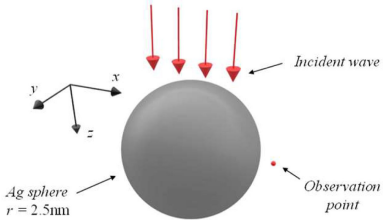

Fig. 2. Computational model of a metallic sphere for investigation of the computational accuracy of the proposed algorithm.

the curl operator as in (4). Specifically, they can be written as follows:

$$D_y^2\, E_x = \frac{1}{c_y|_{i,j,k}\Delta y^2} \times \left(\frac{E_x|_{i,j+1,k} - E_x|_{i,j,k}}{c_y|_{i,j+0.5,k}} - \frac{E_x|_{i,j,k} - E_x|_{i,j-1,k}}{c_y|_{i,j-0.5,k}} \right)$$

$$D_z^2\, E_x = \frac{1}{c_z|_{i,j,k}\Delta z^2} \times \left(\frac{E_x|_{i,j,k+1} - E_x|_{i,j,k}}{c_z|_{i,j,k+0.5}} - \frac{E_x|_{i,j,k} - E_x|_{i,j,k-1}}{c_z|_{i,j,k-0.5}} \right)$$

$$D_x D_y\, E_x = \frac{E_x|_{i,j+1,k} + E_x|_{i-1,j,k} - E_x|_{i-1,j+1,k} - E_x|_{i,j,k}}{c_x|_{i,j,k} c_y|_{i,j+0.5,k}\Delta x\Delta y}$$

$$D_x D_z\, E_x = \frac{E_x|_{i,j,k+1} + E_x|_{i-1,j,k} - E_x|_{i-1,j,k+1} - E_x|_{i,j,k}}{c_x|_{i,j,k} c_z|_{i,j,k+0.5}\Delta x\Delta z}.$$

Upon the same procedure being performed for the entire computational space, a system of linear equations is obtained as

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{6}$$

where $\boldsymbol{A}$ is an operator; $\boldsymbol{x}$, the unknown EM components we solve for; and $\boldsymbol{b}$, a column vector determined by the given current source. The coefficient matrix $\boldsymbol{A}$ is typically very large but extremely sparse because it is constructed via the finite-difference scheme. The unknown vector $\boldsymbol{x}$ in the CFD can be solved by using a direct or iterative solver [21] similar to that in conventional finite-difference methods. In this letter, an iterative method is used to solve the linear equations. The iteration is performed until the norm of the residual becomes 10^{-7}. Upon application of (4), the number of iterations can be drastically reduced, and the computational speed can be made much faster than that of the linear system obtained using Maxwell's equation.

III. Computational Results

Fig. 2 shows the computational model of a metallic sphere with a radius of $r = 2.5$ nm. A plane wave is incident on the sphere. The electric field is observed at a distance of 0.5 nm from the sphere's surface.

To consider the dispersion of permittivity in a metal, the relative permittivity $\varepsilon_r\ (:= \varepsilon/\varepsilon_0)$ is calculated using the Lorentz–Drude model [22] as

$$\varepsilon_r\ (s) = 1 + \frac{f_0\omega_p^2}{s^2 + s\Gamma_0} + \sum_{j=1}^{N} \frac{f_j\omega_p^2}{s^2 + s\Gamma_j + \omega_j^2} \tag{7}$$

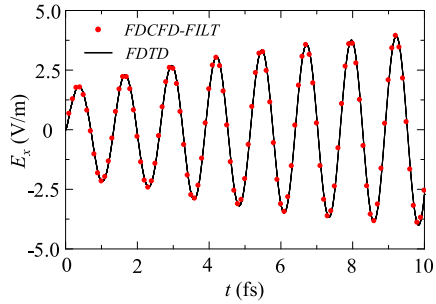

Fig. 3. Time-domain response of a metallic sphere with sine-step pulse by using FDCFD-FILT.

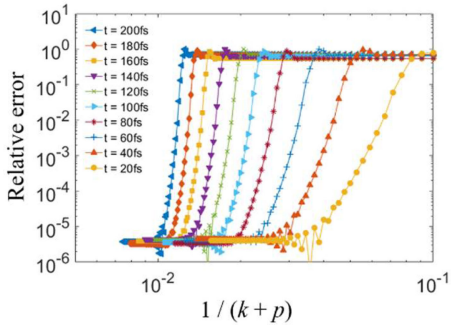

Fig. 4. Convergence for $k + p$ for various observation times t when using the original FILT algorithm.

where ω_p is the plasma frequency, and N is the number of oscillators with frequency ω_j, strength f_j, and damping coefficient Γ_j.

Fig. 3 shows the time-domain response of the electric field E_x at the observation point. The solid line and dots, respectively, indicate the computational results of FDTD and FDCFD-FILT (i.e., combination of FDCFD with FILT). These results are observed to be in very good agreement. Because the incident wave is set as the resonant wavelength $\lambda = 350$ nm, the electric field is enhanced by the surface plasmon resonance.

Fig. 4 shows the convergence history of the electric field for different truncation numbers of FILT in (1). Here, the approximation parameter is selected as $\alpha = 6$. The lines indicate the convergence history for various observation times t. To obtain six-digit accuracy, a larger truncation number is needed for a later observation time. For $t = 200$ fs, almost 300 truncation numbers are required.

A similar convergence test is performed to investigate the performance of the improved FILT algorithm with the double Euler transformation and reduced truncation number. Fig. 5 shows the convergence history of this algorithm for various lower and upper truncation numbers (k_1 and k_2, respectively) in (3) when $t = 200$ fs. $k_1\ (\geq 1)$ and k_2 are selected as follows:

$$k_1 = k_c - n, \quad k_2 = k_c + n, \quad (n = 1, 2, 3, \ldots) \tag{8a}$$

$$k_c = \text{int}\,(2f_c t) \tag{8b}$$

where f_c represents the center frequency of the incident wave. Here, the approximation parameter is $\alpha = 6$ and the number of upper side Euler transforms is $p = 5$. The lines indicate the convergence history for various numbers of lower side Euler transforms q. In this convergence test, the horizontal axis indicates the inverse number of poles defined by $k_2 - k_1 + p + q$.

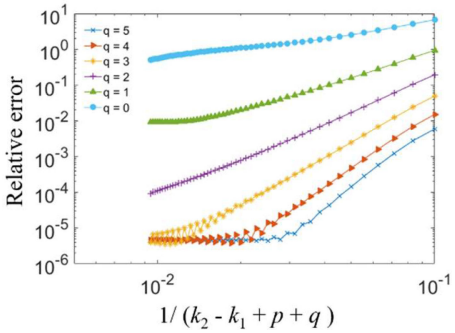

Fig. 5. Convergence for different numbers of poles when using the improved FILT algorithm with the double Euler transformation.

TABLE I
COMPUTATIONAL TIMES FOR VARIOUS NUMBERS OF COMPUTING NODES

Number of time-divisions	Number of s-divisions	Computational time (a.u.) for conventional Euler transformation	Computational time (a.u.) for double Euler transformation
1	1	6.620	1.000
1	10	0.663	0.103
1	30	0.222	0.034
1	100	0.066	0.034
10	1	0.663	0.102
30	1	0.223	0.033
100	1	0.223	0.033
10	10	0.066	0.011

Clearly, the convergence is accelerated upon application of the Euler transform for both the upper and the lower sides. The error converges to $\sim 10^{-6}$ when the truncation number is $k_2 - k_1 = 50$ with $p = q = 5$.

FILT is very suitable for parallel computation using clusters. Each term in (3) can be calculated independently. There are two strategies for parallel computation—parallel computation for the CFD with the summation of FILT and parallel computation for the time domain.

Table I shows the computation time for varying numbers of nodes. The time responses for $t = 200$–203 fs for EM scattering from a silver sphere are simulated for investigating the efficiency of the parallel computation. Our proposed method is particularly effective for late-time analysis regardless of the size of objects. The number of required poles is defined as in (8a) and (8b). The time increment Δt is set to 0.1 fs, which is approximately one-tenth of one period of the incident wave. The computational load is divided in the time domain and s-domain and distributed equally across various nodes. In this numerical verification, the total number of sampling points for the time response is 30, and the number of poles on the complex-frequency plane for FILT is also set to 30 by using the double Euler transformation. Under this condition, the required computational time when using 30 computing nodes for time division or s division in parallel is reduced to 1/30 of the original time. However, it cannot be further reduced by increasing the number of nodes because the proposed algorithm does not allow for division of the computational load across more than 30 nodes in this case. The case of s-divisions has the same restriction. Both time division and s-division in parallel are ideal solutions for improving the computational efficiency in terms of load distribution. The last line in Table I shows an example of using 100 nodes. First, the total computational load is divided into ten parts in the time domain, where each part includes three sampling points for the time response. One sampling point requires 30 computations of poles on the complex-frequency plane. Thus, the number of required poles becomes 90 for three sampling points. The computation of 90 poles can be further divided into ten equal parts. Here, nine poles are distributed to one computing node. Therefore, the total computational load can be equally distributed to 100 nodes via the parallelization of the time domain and CFD. Finally, the required computational time is reduced to approximately 1/100 of the original time. The maximum speed-up in the ideal case can be estimated as

$$\text{Speed-up} = N_s \times N_t \tag{9}$$

where N_s and N_t represent the number of poles and sampling points, respectively.

We also performed a comparison between our method and the conventional Euler transformation regarding the computational time. Because the double Euler transformation reduces N_s, its computational time is shorter than that of the conventional Euler transformation in all cases.

TABLE II
IFFT AND FILT COMPUTATIONAL TIMES FOR VARIOUS NUMBERS OF NODES

Number of nodes	Computational time (s) of IFFT	Speed-up of IFFT	Computational time (s) of FILT	Speed-up of FILT
1	4.498	1.000	72.26	1.000
4	1.129	3.985	18.23	3.965
16	0.284	15.84	4.553	15.87
32	0.142	31.69	2.460	29.38
64	0.138	32.63	1.158	62.41
128	0.143	31.46	0.575	125.79
256	0.154	29.20	0.310	233.13
512	0.166	27.08	0.141	510.97

Table II compares the computational times of parallel inverse fast Fourier transform (IFFT) [23] and FILT for varying numbers of nodes. The frequency-domain or CFD data were transformed into time domain, and the data length was 2^{15}. It can be observed that for a single or few nodes, IFFT is faster than FILT. However, when the number of nodes increases, the speed-up of IFFT becomes saturated as parallel IFFT requires data communication. However, FILT allows embarrassingly parallel computation due to which the speed-up can be held. Consequently, FILT becomes faster when 512 nodes are used.

IV. CONCLUSION

An optimized parallel algorithm is proposed to reduce the computational cost of FILT and applied to an EM scattering problem. This algorithm greatly reduces the truncation number of FILT and, in turn, greatly reduces the computational time in comparison with those of the conventional method.

This method is very suitable for performing parallel computations. Both time-domain and CFD parallelization are applied for distributing the computational load optimally. The computational time can be reduced in proportion to the number of computing nodes. The maximum speed-up in the ideal case is given as the product of the number of poles on the complex plane and the number of sampling points of the time response.

REFERENCES

[1] D. V. Widder, "The Laplace transform," in *Princeton Mathematical Series*. Princeton, NJ, USA: Princeton Univ. Press, 1941.

[2] R. D. Strum and J. R. Ward, *Laplace Transform Solution of Differential Equations*. Hoboken, NJ, USA: Pearson Education, 1968.

[3] F. Oberhettinger and L. Badii, *Tables of Laplace Transform*. Berlin, Germany: Springer, 1973.

[4] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Table*. New York, NY, USA: Dover, 1964.

[5] J. L. Schiff, *The Laplace Transform: Theory and Applications*. Berlin, Germany: Springer, 1999.

[6] B. Davies and B. Martin, "Numerical inversion of the Laplace transform: A survey and comparison of methods," *J. Comput. Phys.*, vol. 33, no. 1, pp. 1–32, 1979.

[7] T. Hosono, "Numerical inversion of Laplace transform and some applications to wave optics," *Radio Sci.*, vol. 16, no. 6, pp. 1015–1019, 1981.

[8] S. Kishimoto, T. Okada, S. Ohnuki, Y. Ashizawa, and K. Nakagawa, "Efficient analysis of electromagnetic fields for designing nanoscale antennas by using a boundary integral equation method with fast inverse Laplace transform," *Prog. Electromagn. Res.*, vol. 146, pp. 155–165, 2014.

[9] R. Ozaki and T. Yamasaki, "Analysis of pulse responses from conducting strips with dispersion medium sandwiched air layer," *IEICE Electron. Express*, vol. 15, no. 6, pp. 1–6, 2018.

[10] M. S. Mamiş and M. Köksal, "Solution of eigenproblems for state space transient analysis of transmission lines," *Elect. Power Syst. Res.*, vol. 55, no. 1, pp. 7–14, 2000.

[11] J. Chakarothai, "Novel FDTD scheme for analysis of frequency-dependent medium using fast inverse Laplace transform and Prony's method," *IEEE Trans. Antennas Propag.*, vol. 67, no. 9, pp. 6076–6089, Sep. 2019.

[12] D. Wu, R. Ohnishi, R. Uemura, T. Yamaguchi, and S. Ohnuki, "Finite-difference complex-frequency-domain method for optical and plasmonic analyses," *IEEE Photon. Technol. Lett.*, vol. 30, no. 11, pp. 1024–1027, Jun. 2018.

[13] S. Masuda, S. Kishimoto, and S. Ohnuki, "Reference solutions for time domain electromagnetic solvers," *IEEE Access*, vol. 8, pp. 44318–44324, 2020.

[14] S. Ohnuki and T. Hinata, "Transient scattering from parallel plate waveguide cavities," *IEICE Trans. Electron.*, vol. E88-C, no. 1, pp. 112–118, Jan. 2005.

[15] S. Ohnuki, R. Ohnishi, D. Wu, and T. Yamaguchi, "Time-division parallel FDTD algorithm," *IEEE Photon. Technol. Lett.*, vol. 30, no. 24, pp. 2143–2146, Dec. 2018.

[16] A. Taflove and S. C. Hagness, *Computational Electrodynamics*, 2nd ed. Norwood, MA, USA: Artech House, 1995.

[17] V. I. Okhmatovski and A. C. Cangellaris, "Evaluation of layered media Green's functions via rational function fitting," *IEEE Microw. Wireless Compon. Lett.*, vol. 14, no. 1, pp. 22–24, Jan. 2004.

[18] S. Grivet-Talocia, "Package macromodeling via time-domain vector fitting," *IEEE Microw. Wireless Compon. Lett.*, vol. 13, no. 11, pp. 472–474, Nov. 2003.

[19] A. Yonemoto, T. Hisakado, and K. Okumura, "Accuracy improvement of the FFT-based numerical inversion of Laplace transforms," *IEE Proc. Circuits Devices Syst.*, vol. 150, no. 5, pp. 399–404, 2003.

[20] J. A. Roden and S. D. Gedney, "Convolution PML (CPML): An efficient FDTD implementation of the CFS-PML for arbitrary media," *Microw. Opt. Technol. Lett.*, vol. 27, no. 5, pp. 334–339, 2000.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 1992.

[22] A. D. Rakic, A. B. Djurisic, J. M. Elazar, and M. L. Majewski, "Optical properties of metallic films for vertical-cavity optoelectronic devices," *Appl. Opt.*, vol. 37, no. 22, pp. 5271–5283, 1998.

[23] D. Takahashi, *Fast Fourier Transform Algorithms for Parallel Computers*. Berlin, Germany: Springer, 2019.