Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 202 (2005) 236–261
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A fast, higher-order solver for scattering by penetrable bodies in three dimensions

E. McKay Hyde [a,*,1], Oscar P. Bruno [b,2]

[a] School of Mathematics, University of Minnesota, 206 Church Street SE, 127 Vincent Hall, Minneapolis, MN 55455, USA
[b] Applied and Computational Mathematics, Caltech, Pasadena, CA 91125, USA

Received 31 October 2003; received in revised form 22 June 2004; accepted 8 July 2004
Available online 26 August 2004

Abstract

In this paper, we introduce a new fast, higher-order solver for scattering by inhomogeneous media in three dimensions. As in previously existing methods, the low complexity of our integral equation method, $\mathcal{O}(N \log N)$ operations for an N point discretization, is obtained through extensive use of the fast Fourier transform (FFT) for the evaluation of convolutions. However, the present approach obtains significantly higher-order accuracy than these previous approaches, yielding, at worst, third-order far field accuracy (or substantially better for smooth scatterers), even for discontinuous and complex refractive index distributions (possibly containing severe geometric singularities such as corners and cusps). The increased order of convergence of our method results from (i) a partition of unity decomposition of the Green's function into a smooth part with unbounded support and a singular part with compact support, and (ii) replacement of the (possibly discontinuous) scatterer by an appropriate "Fourier smoothed" scatterer; the resulting convolutions can then be computed with higher-order accuracy by means of $\mathcal{O}(N \log N)$ FFTs. We present a parallel implementation of our approach, and demonstrate the method's efficiency and accuracy through a variety of computational examples. For a very large scatterer considered earlier in the literature (with a volume of $3648\lambda^3$, where λ is the wavelength), using the same number of points per wavelength and in computing times comparable to those required by the previous approach, the present algorithm produces far-field values whose errors are two orders of magnitude smaller than those reported previously.

* Corresponding author. Present address: Computational and Applied Mathematics, MS 134, Rice University, 6100 Main St., Houston, TX 77005-1892, USA. Tel.: +1 713 348 4094; fax: +1 713 348 5318.
E-mail address: hyde@caam.rice.edu (E. McKay Hyde).

[1] Supported through a DOE Computational Science Graduate Fellowship, an Achievement Rewards for College Scientists (ARCS) Fellowship and an NSF Mathematical Sciences Postdoctoral Research Fellowship.

[2] Supported by the AFOSR under grant numbers F49620-96-1-0008, F49629-99-1-0010 and F49620-02-10049, by the NSF through the NYI award DMS-9596152 and contracts No. DMS-9523292, DMS-9816802 and DMS-0104531 and by the Powell Research Foundation.

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.07.006

MSC: 35J05; 65D30; 65N12; 65R20; 65T50; 78A45; 78M25
Keywords: Helmholtz equation; Lippmann–Schwinger equation; Fourier series; Fast Fourier transform; High-order quadrature

1. Introduction

Scattering problems find application in a wide range of fields including communications, materials science, plasma physics, biology, medicine, radar, and remote sensing. The evaluation of useful numerical solutions for scattering problems remains a highly challenging problem requiring novel mathematical approaches and powerful computational tools: applications of interest make it necessary to evaluate highly oscillatory fields in large, complex three-dimensional geometries. Hence, without an extremely efficient and high-order accurate method, the computation of solutions to such problems is infeasible even with modern computing hardware. In this paper, we present a new fast, higher-order method for evaluation of scattering by inhomogeneous media in three dimensions, which in $\mathcal{O}(N \log N)$ operations (where N is the total number of discretization points) is able to produce highly accurate solutions for problems of unprecedented size. Before describing our method, we formulate the problem and present an overview of previous work in this area.

Given an incident field u^{i}, we denote by u the total field – which equals the sum of u^{i} and the resulting scattered field u^{s}:

$$u = u^{\mathrm{i}} + u^{\mathrm{s}}. \tag{1}$$

Calling λ the wavelength of the incident field and $\kappa = \frac{2\pi}{\lambda}$ the corresponding wavenumber, we require that the total field u satisfies [1, p. 2]

$$\Delta u + \kappa^2 n^2(x) u = 0, \quad x \in \mathbb{R}^3, \tag{2}$$

where the given incident field u^{i} is assumed to satisfy

$$\Delta u^{\mathrm{i}} + \kappa^2 u^{\mathrm{i}} = 0, \quad x \in \mathbb{R}^3. \tag{3}$$

Finally, to guarantee that the scattered wave is outgoing, u^{s} is required to satisfy the Sommerfeld radiation condition [1, p. 67]

$$\lim_{r\to\infty} r\left(\frac{\partial u^{\mathrm{s}}}{\partial r} - \mathrm{i}\kappa u^{\mathrm{s}}\right) = 0. \tag{4}$$

The algorithms available for computing solutions to this problem fall into two broad classes: (i) finite element and finite difference methods and (ii) integral equation methods. Use of finite element and finite difference methods can be advantageous in that, unlike other methods, they lead to sparse linear systems. Their primary disadvantage, on the other hand, lies in the fact that in order to satisfy the Sommerfeld radiation condition (4), a relatively large computational domain containing the scatterer must be used, together with appropriate absorbing boundary conditions on the boundary of the computational domain (see, for example, [2–7]). Thus, these procedures give rise to very large numbers of unknowns and, hence, to very large linear systems. In addition, accurate absorbing boundary conditions with efficient numerical implementations are quite difficult to construct: the error associated with such boundary conditions typically dominates the error in the computed solution.

A second class of algorithms is based on the use of integral equations. An appropriate integral formulation for our problem is given by the Lippmann–Schwinger equation [1, p. 214]

$$u(x) = u^{\mathrm{i}}(x) - \kappa^2 \int_\Omega g(x-y) m(y) u(y) \mathrm{d}y, \tag{5}$$

where $g(x) = \frac{e^{ik|x|}}{4\pi|x|}$ is the Green's function for the Helmholtz equation in three dimensions, and where m is the *compactly supported* function $m = 1 - n^2$, with $\mathrm{supp}(m) = \Omega$. Integral equation approaches are advantageous in a number of ways: they only require discretization of the equation *on the support Ω of the scatterer*, and the solutions they produce satisfy the radiation condition at infinity *automatically*. Direct use of integral equation methods is costly, however, since they lead to dense linear systems: a straightforward computation of the required convolution requires $\mathcal{O}(N^2)$ operations per iteration of an iterative linear solver. As mentioned above, however, the highly accurate integral method that we present in this paper, in which the computational complexity of the convolution evaluation is reduced to $\mathcal{O}(N \log N)$ operations per iteration, is highly competitive with finite element or finite difference approaches.

Fast solvers for (5), based on the fast Fourier transform (FFT), have been available for some time [8–10]; see also the more recent papers [11,12]. In these solvers, the convolution with the Green's function is computed via Fourier transforms, which, in turn, can be evaluated with low complexity by means of FFTs. These methods do give rise to a reduced complexity for a given discretization, but, unfortunately, they are only first-order accurate for discontinuous refractive indices – since they rely on use of discrete Fourier transforms of non-smooth and/or non-periodic functions. Our approach also uses FFTs to achieve a reduced complexity, but, by means of a reformulation of the problem, it also yields higher-order accuracy: at least third-order far field accuracy for discontinuous scatterers and much higher-order accuracies for smoother inhomogeneities. In fact, this increase in convergence order requires minimal additional computational effort and, thus, allows for the solution of large problems in roughly the same amount of time as the first-order methods cited above, but with significantly more accuracy. In our final computational example in Section 4, for example, we show that, for a very large scatterer considered in [12] ($3648\lambda^3$, where λ is the wavelength) the present approach produces, in computing times comparable to those used in that reference, far-field values whose errors are two orders of magnitude smaller than those reported for the previous approach.

Despite the significant advantages exhibited by high-order methods over their low-order counterparts, only limited attempts have been made to develop high-order methods for the problem under consideration. A method is proposed in [13–15] on the basis of a locally corrected Nyström discretization. These locally corrected quadrature rules were originally proposed in [16–18], wherein the author proves that the accuracy obtained in an $\mathcal{O}(N \log N)$ implementation is "$\mathcal{O}(\varepsilon + h^k)$", where ε is fixed by the choice of the method parameters and h is the discretization spacing. In other words, the method exhibits kth order convergence (where k may be chosen arbitrarily large), *but only to an accuracy of ε*; if more accuracy is desired, the parameters must be adjusted, which, in turn, leads to a higher computational cost. Our method, on the other hand, has no such accuracy limitation. Furthermore, because of the simplicity of our implementation, our method requires much less computational time. For example, in Fig. 2 of [15], it is reported that a computation involving approximately 10^4 unknowns requires on the order of 10^4 s, whereas Table 1 in the current paper shows that our approach requires approximately 100 s for a computation involving over 64,000 unknowns. It should be noted that in [15] the authors solve the electric field integral equation and that their results were obtained on a 750 MHz HP PA-RISC processor. However, these factors can only account for a small fraction of the 100-fold difference in computational time.

An $\mathcal{O}(N \log N)$ FFT-based method that, in the case of *smooth* scatterers, achieves high-order convergence is presented by Vainikko in [19]. Briefly, this method follows from the following insight: as seen in (5), if $\mathrm{diam}(\Omega) \leqslant R$, then, for $x \in \Omega$, the argument $x - y$ of g satisfies, $|x - y| \leqslant R$; hence, changing $g(x)$ for $|x| > R$ has no effect on the solution *inside* Ω. Hence, $g(x)$ is set equal to zero (either discontinuously or smoothly) outside of the ball of radius R centered at the origin and periodically extended with a periodic cell given by a cube of side length $L \geqslant 2R$. This allows for efficient and accurate evaluation of the convolution operator for $x \in \Omega$ by means of the Fourier coefficients of the modified Green's function and the Fourier coefficients of $(\mathrm{mu})(x)$.

For Vainikko's method, with a grid spacing of $h \approx N^{-1/3}$, the L^2-norm of both the near and far field errors decays as $h^{-\mu}$ for m in the Sobolev space $W^{\mu,2}$. However, as with other FFT-based approaches, this method provides only first order convergence for discontinuous scatterers. (In the same paper, an alternative approach is proposed that can be applied to problems involving piecewise-smooth refractive indices that yields L^∞-errors of the order of $h^2(1 + |\log h|)$ in both the near and far fields. This method requires evaluation of the volume fraction of each discretization cell on each side of a discontinuity in $m = 1 - n^2$.)

The method we introduce in this paper is related to Vainikko's approach in that we periodically extend part of the Green's function so that the convolution may be evaluated accurately by means of FFTs. However, our method introduces the following two significant improvements. First, instead of setting the Green's function equal to zero for $|x| > R$, we decompose g by means of a smooth partition of unity into a smooth part with unbounded support, g_{smth}, and a singular part with compact support, g_{cmp}, i.e.,

$$g(x) = g_{\text{smth}}(x) + g_{\text{cmp}}(x).$$

The convolution of mu with each of g_{smth} and g_{cmp} can be computed efficiently and accurately by means of FFTs.

This approach has the advantage of reducing the number of unknowns quite significantly for many problems. More precisely, in Vainikko's approach, one is constrained to represent the modified Green's function on *cubic* periodic cells; for elongated scatterers (large aspect ratios), this leads to an unnecessarily large number of unknowns. In our approach, on the other hand, there is no such restriction, and we can use box-shaped periodic cells (not necessarily cubic) to avoid this problem. For example, if a given scatterer has a size of $1\lambda \times 1\lambda \times 5\lambda$, then for Vainikko's approach one requires a periodic cell of size $10.4\lambda \times 10.4\lambda \times 10.4\lambda$, whereas our method requires a much smaller periodic cell, $2\lambda \times 2\lambda \times 10\lambda$. Thus, in this case, to obtain a given accuracy, our method will require roughly a factor of 28 fewer unknowns with corresponding time and memory savings.

Our second significant improvement results by replacing, in the evaluation of the convolution, the function $m = 1 - n^2$ by an appropriate "Fourier smoothed" inhomogeneity $\widetilde{m}$; see Section 2.3 for details. (Of course, the number of Fourier modes in the approximation $\widetilde{m}$ is increased as the discretization is refined.) Although such a substitution for a discontinuous m leads to a poor pointwise approximation of m, it gives rise, rather counterintuitively, to a higher-order convergence rate in the evaluation of the convolution integrals, *even in the case of a discontinuous function m* (see also [20–22]); in fact, by this Fourier smoothing procedure, our algorithm substantially exceeds the convergence rate of Vainikko's approach for any given regularity of the inhomogeneity while still retaining the relative simplicity of the algorithm – numerical implementation requires little more than a couple FFTs and the iterative solution of the linear system. (For a more detailed comparison of convergence rates in these approaches, see our theoretical work on a related method in two dimensions [22]; we observe similar convergence rates in the current approach.)

In Sections 2 and 3, we describe, respectively, the method and its parallel numerical implementation in detail. Finally, in Section 4 we present several computational examples to illustrate the low computational complexity, the higher-order convergence rate, and the parallel performance of our method.

2. Numerical method

We turn now to a detailed description of the numerical method. The core of our approach is an efficient, higher-order scheme for computing the integral operator (see (5))

$$(Ku)(x_j) = -\kappa^2 \int_\Omega g(x_j - y)m(y)u(y)\mathrm{d}y$$

at certain discretization points x_j – whose definition we defer to Section 2.1. We thereby obtain the linear system

$$u(x_j) - (Ku)(x_j) = u^{\mathrm{i}}(x_j), \tag{6}$$

whose solution $u(x_j)$, obtained by means of an iterative solver, approximates the total field. (We discuss our choice of iterative solver in Section 3.)

As described briefly in Section 1, we decompose the Green's function $g(x)$ into a smooth part with unbounded support, $g_{\mathrm{smth}}(x)$, and a singular part with compact support, $g_{\mathrm{cmp}}(x)$, by means of a partition of unity. More precisely, we define $g_{\mathrm{smth}}(x)$ and $g_{\mathrm{cmp}}(x)$ by

$$g(x) = g(x)(1 - p(x)) + g(x)p(x) = g_{\mathrm{smth}}(x) + g_{\mathrm{cmp}}(x),$$

where $p(x) \in C^\infty$, $p(x) = 1$ near $x = 0$, and $p(x) = 0$ outside some neighborhood of $x = 0$. (Of course, there are many such functions, but we do not specify a particular choice at this time.) Thus, our approach requires the evaluation of the two convolutions

$$(K_{\mathrm{cmp}}u)(x) = -\kappa^2 \int_\Omega g_{\mathrm{cmp}}(x-y)m(y)u(y)\mathrm{d}y, \tag{7}$$

$$(K_{\mathrm{smth}}u)(x) = -\kappa^2 \int_\Omega g_{\mathrm{smth}}(x-y)m(y)u(y)\mathrm{d}y. \tag{8}$$

2.1. Convolution with the compactly-supported singular kernel g_{cmp}

To evaluate $K_{\mathrm{cmp}}u$ defined in (7) above, we make use of a highly accurate Fourier series approximation. It is well known that Fourier series provide high-order accurate approximations for smooth and periodic functions; this follows from the fact that the decay rate of the Fourier coefficients of a periodic function depends on the function's regularity [23, pp. 48, 71]. As explained below, the function $K_{\mathrm{cmp}}u$ can be viewed as a (relatively) smooth and periodic function, and it is therefore approximated with higher-order accuracy by a truncated Fourier series.

To see that $K_{\mathrm{cmp}}u$ can be extended as a smooth and periodic function, note first that, since g_{cmp} and m are both compactly supported, $(K_{\mathrm{cmp}}u)(x)$ vanishes for points x sufficiently far from $\mathrm{supp}(m)$. More precisely, assume that $\mathrm{supp}(m) \subset \Omega_{[a,b]}$, where $\Omega_{[a,b]}$ denotes the box with corners $a, b \in \mathbb{R}^3$, i.e., $\Omega_{[a,b]} = \{x : a_q \leqslant x_q \leqslant b_q,\ q = 1, 2, 3\}$. Thus, given that $\mathrm{supp}(g_{\mathrm{cmp}}) \subset \Omega_{[-\sigma,\,\sigma]}$ for some $\sigma \in \mathbb{R}^3$ with $\sigma_q > 0$, we have that $(K_{\mathrm{cmp}}u)(x) = 0$ for $x \notin \Omega_{[a-\sigma,b+\sigma]}$.

Furthermore, in view of the known regularizing properties of the convolution operator, $(K_{\mathrm{cmp}}u)$ is a smooth function, with its regularity determined by the regularity of the inhomogeneity m: if m is piecewise-smooth, then $K_{\mathrm{cmp}}u$ is $C^{1,\alpha}$ and piecewise-smooth; if m is $C^{k,\alpha}$ and piecewise-smooth, then $K_{\mathrm{cmp}}u$ is $C^{k+2,\alpha}$ and piecewise-smooth; see [24, p. 223, 25, pp. 78–80, 26, pp. 53, 56]. Therefore, as claimed, $(K_{\mathrm{cmp}}u)$ can be extended as a smooth (at least C^1 and piecewise-smooth for the class of inhomogeneities that we consider) and periodic function with a periodic cell $\Omega_{[A,B]}$, for values of A and B such that $\Omega_{[a-\sigma,b+\sigma]} \subset \Omega_{[A,B]}$. (The actual choice of A and B is discussed later in this section.)

It follows that, for $x \in \Omega_{[A,B]}$, $K_{\mathrm{cmp}}u$ is represented with higher-order accuracy by the truncated Fourier series

$$(K_{\mathrm{cmp}}u)(x) \approx \sum_{\ell_1=-M_1}^{M_1} \sum_{\ell_2=-M_2}^{M_2} \sum_{\ell_3=-M_3}^{M_3} (K_{\mathrm{cmp}}u)_\ell \, \mathrm{e}^{2\pi \mathrm{i} d_\ell \cdot (x - x_0)}, \tag{9}$$

where the components $(d_\ell)_q = \frac{\ell_q}{B_q - A_q}$ for $q = 1, 2, 3$. The choice of the truncation parameter $M = (M_1, M_2, M_3)$ and the value x_0 is discussed below. To simplify the notation, we will denote the triple sum in (9) by $\sum_{\ell=-M}^{M}$.

We must now compute the Fourier coefficients $(K_{\text{cmp}}u)_\ell$. Defining $\Pi(z) = z_1 z_2 z_3$ for any $z \in \mathbb{R}^3$, we have

$$\begin{aligned}(K_{\text{cmp}}u)_\ell &= -\frac{\kappa^2}{\Pi(B-A)} \int_{\Omega_{[A,B]}} (K_{\text{cmp}}u)(x) \mathrm{e}^{-2\pi \mathrm{i} d_\ell \cdot (x-x_0)} \, \mathrm{d}x \\ &= -\frac{\kappa^2}{\Pi(B-A)} \int_{\Omega} m(y)u(y) \mathrm{e}^{-2\pi \mathrm{i} d_\ell \cdot (y-x_0)} \, \mathrm{d}y \cdot \int_{\Omega_{[-\sigma,\sigma]}} g_{\text{cmp}}(z) \mathrm{e}^{-2\pi \mathrm{i} d_\ell \cdot z} \, \mathrm{d}z = -\kappa^2 (g_{\text{cmp}})_\ell (\text{mu})_\ell,\end{aligned}$$

where

$$(g_{\text{cmp}})_\ell = \int_{\Omega_{[-\sigma,\sigma]}} g_{\text{cmp}}(z) \mathrm{e}^{-2\pi \mathrm{i} d_\ell \cdot z} \, \mathrm{d}z \tag{10}$$

and

$$(\text{mu})_\ell = \frac{1}{\Pi(B-A)} \int_{\Omega} m(y)u(y) \mathrm{e}^{-2\pi \mathrm{i} d_\ell \cdot (y-x_0)} \, \mathrm{d}y. \tag{11}$$

Because $g_{\text{cmp}}(x)$ is known analytically, we compute its Fourier coefficients $(g_{\text{cmp}})_\ell$ only once, at the beginning of each run. Note that the evaluation of the coefficients $(g_{\text{cmp}})_\ell$ is the *only step* of our approach that requires the explicit integration over the singularity of the Green's function. As described in Section 2.4, we make use of a spherical coordinate change of variables that precisely *cancels* this singularity, thus allowing the efficient and high-order accurate evaluation of $(g_{\text{cmp}})_\ell$.

To compute the Fourier coefficients $(\text{mu})_\ell$, on the other hand, we use the trapezoidal rule. However, since the inhomogeneity m is, in general, a piecewise-smooth function, *straightforward* application of the trapezoidal rule would yield only first-order accuracy. We obtain higher-order values for (11) by first replacing m by an appropriate Fourier-smoothed inhomogeneity $\widetilde{m}$, and then integrating by means of the trapezoidal rule. The somewhat surprising fact that this simple procedure leads to higher-order accuracy for such integrands – the product of the known piecewise-smooth inhomogeneity m and a C^1, piecewise-smooth function – is the trademark of our approach. We discuss this key element of our method in detail in Section 2.3.

Given that $\text{supp}(\widetilde{m}) \subset \Omega_{[\tilde{a},\tilde{b}]}$ (to accommodate the Fourier smoothing, we require that $\text{supp}(\widetilde{m})$ properly contain $\text{supp}(m)$), and given the number of discretization points $\overline{N} = (\overline{N}_1, \overline{N}_2, \overline{N}_3)$, we define the discretization points x_j such that the components $(x_j)_q = \tilde{a}_q + j_q h_q$, where $j_q = 0, 1, \ldots, \overline{N}_q$ and $h_q = (\tilde{b}_q - \tilde{a}_q)/\overline{N}_q$ for $q = 1, 2, 3$. Letting $x_0 = \tilde{a}$ and replacing m by $\widetilde{m}$ in (11), the trapezoidal rule gives that

$$(\text{mu})_\ell \approx \frac{1}{\Pi(B-A)} \int_{\Omega_{[\tilde{a},\tilde{b}]}} \widetilde{m}(y)u(y) \mathrm{e}^{-2\pi \mathrm{i} d_\ell \cdot (y-\tilde{a})} \, \mathrm{d}y \approx \frac{\Pi(h)}{\Pi(B-A)} \sum_{j=0}^{\overline{N}-1} \widetilde{m}_j u_j \, \mathrm{e}^{-2\pi \mathrm{i} d_\ell \cdot (j_1 h_1, j_2 h_2, j_3 h_3)}, \tag{12}$$

where $\widetilde{m}_j = \widetilde{m}(x_j)$, $u_j = u(x_j)$.

So that (11) can be evaluated by means of an FFT, we use values of A and B such that $(B_q - A_q)/h_q \in \mathbb{N}$ for $q = 1, 2, 3$ – note that this implies that the domain $\Omega_{[A,B]}$ is exactly an integer number of cells larger than the smaller domain $\Omega_{[\tilde{a},\tilde{b}]}$ in each dimension. Thus, defining $\widetilde{N} \in \mathbb{N}^3$ by

$$\widetilde{N}_q = (B_q - A_q)/h_q \tag{13}$$

for $q = 1, 2, 3$, we obtain

$$(\text{mu})_\ell \approx \frac{1}{\Pi(\widetilde{N})} \sum_{j=0}^{\widetilde{N}-1} \widetilde{m}_j u_j \, \mathrm{e}^{-2\pi \mathrm{i} \ell \cdot (j_1/\widetilde{N}_1, j_2/\widetilde{N}_2, j_3/\widetilde{N}_3)}, \tag{14}$$

where $\widetilde{m}_j u_j = 0$ if $j_q > \overline{N}_q$ for any $q = 1, 2, 3$, and where $|\ell_q| \leqslant M_q$ with $M_q = \widetilde{N}_q/2 - 1$ (see (9)). This discrete Fourier transform is evaluated by means of an FFT in $\mathcal{O}(\Pi(\widetilde{N})\log\Pi(\widetilde{N}))$ operations. Finally, given this higher-order approximation of $(\mathrm{mu})_\ell$ and the pre-computed coefficients $(g_{\mathrm{cmp}})_\ell$, the Fourier series (9) is also evaluated by means of an FFT. In this manner, the desired higher-order approximation to $K_{\mathrm{cmp}}u$ is obtained – at the expense of a total of $\mathcal{O}(\Pi(\widetilde{N})\log\Pi(\widetilde{N}))$ operations.

2.2. Convolution with the smooth kernel g_{smth}

For each discretization point x_j, $j_q = 0, 1, \ldots, \overline{N}_q$ for $q = 1, 2, 3$, $g_{\mathrm{smth}}(x_j - y) \in C^\infty$ as a function of y, and thus the integrand of (8) is, as in (11), a product of the known piecewise-smooth inhomogeneity and a C^1, piecewise-smooth function. Hence, as with the evaluation of (11), we again use the Fourier smoothing technique, described in detail in Section 2.3, for the higher-order evaluation of $(K_{\mathrm{smth}}u)(x_j)$ by means of the trapezoidal rule. We thus obtain that

$$(K_{\mathrm{smth}}u)(x_j) \approx \Pi(h)\sum_{k_1=0}^{\overline{N}_1-1}\sum_{k_2=0}^{\overline{N}_2-1}\sum_{k_3=0}^{\overline{N}_3-1} g_{\mathrm{smth}}(x_j - x_k)\widetilde{m}(x_k)u(x_k) = \Pi(h)\sum_{k=0}^{\overline{N}-1}(g_{\mathrm{smth}})_{j-k}\widetilde{m}_k u_k, \tag{15}$$

where $(g_{\mathrm{smth}})_k = g_{\mathrm{smth}}((k_1h_1, k_2h_2, k_3h_3))$, $\widetilde{m}_k = \widetilde{m}(x_k)$, and $u_k = u(x_k)$, and where we have denoted the triple sum as $\sum_{k=0}^{\overline{N}-1}$.

Since (15) is a discrete convolution, $(K_{\mathrm{smth}}u)(x_j)$ is computed for all discretization points x_j in $\mathcal{O}(\Pi(\overline{N})\log\Pi(\overline{N}))$ operations using FFTs [27, pp. 531–536]. We thus obtain an efficient and higher-order accurate method for computing $K_{\mathrm{smth}}u$.

2.3. Fourier-smoothed scatterers

As described in Sections 2.1 and 2.2, the higher-order accuracy of our method depends fundamentally on the higher-order accurate evaluation of $(\mathrm{mu})_\ell$ and $(K_{\mathrm{smth}}u)(x_j)$, defined in (11) and (8), respectively. These quantities both involve the integral of the product of the known piecewise-smooth inhomogeneity m and a C^1, piecewise-smooth function. As mentioned above, *straightforward* application of the trapezoidal rule for the evaluation of these integrals yields only first-order accuracy. Somewhat surprisingly, however, first replacing m by an appropriate Fourier-smoothed inhomogeneity $\widetilde{m}$ as defined below, and then integrating by means of the trapezoidal rule produces higher-order accurate values of $(\mathrm{mu})_\ell$ and $(K_{\mathrm{smth}}u)(x_j)$. We emphasize that this simple procedure yields higher-order accuracy for all piecewise-smooth scattering inhomogeneities m, including those with discontinuities and geometric singularities such as corners and cusps, *in spite of* the low-order accurate truncation of the Fourier series of m and the associated Gibbs errors.

In detail, we require the higher-order accurate evaluation of integrals of the form

$$\int_\Omega m(y)w(y)\mathrm{d}y, \tag{16}$$

where m is the piecewise-smooth inhomogeneity with $\mathrm{supp}(m) \subset \Omega$ and w is a C^1, piecewise-smooth function; compare (11) and (8). (As discussed in Section 2.1, the regularity of w depends on the regularity of m – if m is piecewise-smooth, then w is $C^{1,\alpha}$ and piecewise-smooth; if m is $C^{k,\alpha}$ and piecewise-smooth, then w is $C^{k+2,\alpha}$ and piecewise-smooth.) It is well known that the trapezoidal rule delivers high-order accuracy for smooth and periodic integrands [28, p. 288]. To obtain a smooth and periodic integrand, we first replace $m(y)$ by $p_m(y)m(y)$, where $p_m \in C^\infty$ with $p_m(y) = 1$ for $y \in \mathrm{supp}(m)$ and $\mathrm{supp}(p_m) \subset \Omega_{[\tilde{a},\tilde{b}]}$ for some points $\tilde{a}, \tilde{b} \in \mathbb{R}^3$ – of course, this change has absolutely no effect on the value of the integral (16). We then replace m by its truncated Fourier expansion with periodic cell $\Omega_{[\tilde{a},\tilde{b}]}$:

$$m^{\mathrm{F}}(x) = \sum_{\ell=-F}^{\mathrm{F}} m_\ell \, \mathrm{e}^{2\pi \mathrm{i} c_\ell \cdot x}, \tag{17}$$

where $\sum_{\ell=-F}^{\mathrm{F}}$ denotes a triple sum with $F = (F_1,F_2,F_3)$, and where the components $(c_\ell)_q = \ell_q/(\tilde{b}_q - \tilde{a}_q)$ for $q = 1, 2, 3$. Thus, replacing the piecewise-smooth inhomogeneity m in (16) by

$$\widetilde{m}(x) = m^{\mathrm{F}}(x) p_m(x), \tag{18}$$

we obtain

$$\int_{\Omega_{[\tilde{a},\tilde{b}]}} \widetilde{m}(y) w(y) \mathrm{d}y. \tag{19}$$

We claim that (19) provides a higher-order approximation to (16). Initially, this claim may appear somewhat dubious considering that the truncated Fourier series of the piecewise-smooth m converges quite slowly. Of course, this intuition is confirmed when one seeks Fourier series approximations for the *function values* $m(y)w(y)$ for which one obtains only first-order accuracy; when approximating the *integral* of this product, however, one obtains significantly higher-order accuracy. The presence of the smooth and periodic factor $p_m w$ makes the difference: indeed, in terms of the Fourier coefficients m_ℓ and $(p_m w)_\ell$ of m and $p_m w$, respectively, the error in this approximation is given by

$$\begin{aligned} \int_\Omega m(y) w(y) \mathrm{d}y - \int_{\Omega_{[\tilde{a},\tilde{b}]}} \widetilde{m}(y) w(y) \mathrm{d}y &= \int_{\Omega_{[\tilde{a},\tilde{b}]}} (m - m^{\mathrm{F}})(y)(p_m w)(y) \mathrm{d}y \\ &= \Pi(\tilde{b} - \tilde{a}) \sum_{|\ell_1|>F_1} \sum_{|\ell_2|>F_2} \sum_{|\ell_3|>F_3} m_{-\ell} (p_m w)_\ell. \end{aligned} \tag{20}$$

Since $p_m w$ is C^1, piecewise-smooth and periodic, its Fourier coefficients $(p_m w)_\ell$ decay rapidly with increasing ℓ; thus the series in (20) decays rapidly with increasing F despite the relatively slow decay of the coefficients $m_{-\ell}$, yielding higher-order convergence of (19) to (16). Finally, since the new integrand $\widetilde{m}w$ is smooth and periodic, the integral (19) is evaluated with higher-order accuracy by means of the trapezoidal rule:

$$\int_{\Omega_{[\tilde{a},\tilde{b}]}} \widetilde{m}(y) w(y) \mathrm{d}y \approx \Pi(h) \sum_{j=0}^{\overline{N}-1} \widetilde{m}(x_j) w(x_j), \tag{21}$$

where we fix $\overline{N} = sF$ for an integer $s \geqslant 2$ to assure that there are enough points $\overline{N}$ to resolve the Fourier modes of the integrand. The increased convergence order obtained by this procedure is demonstrated in Section 4 in the case of scattering by a layered sphere with piecewise-constant refractive index; the convergence results with and without Fourier smoothing are presented in Tables 1 and 2, respectively. Furthermore, in Appendix A, we give a complete proof of the convergence order and a numerical example in the one-dimensional case.

Although we do not present a complete theoretical analysis of the method here, we expect the convergence rates of our method to be similar to those that were established rigorously [22] for a related two-dimensional algorithm. In that method, for example, for piecewise-smooth functions $m(x)$, the method yields second- and third-order convergence on the interior and exterior of the scatterer, respectively, with significantly increased convergence rates for more regular inhomogeneities.

In defining the Fourier smoothed function $\widetilde{m}$ (see (17) and (18)), the user is relatively free to choose (i) the number of Fourier modes F in the truncated Fourier series of m and (ii) the size of the enlarged computational domain $\Omega_{[\tilde{a},\tilde{b}]}$. In making these choices one should consider several competing factors. First, smaller sizes of $\Omega_{[\tilde{a},\tilde{b}]}$ yield a smaller integration domains, but at the same time, give rise to sharper increases in the function p_m and thus require finer discretizations (i.e., smaller values of h) to obtain a given accuracy.

There are similar competing factors in the choice of F, or equivalently s, where $\overline{N} = sF$ for some fixed integer $s \geqslant 2$. Indeed, although any choice of $s \geqslant 2$ leads to the same asymptotic convergence (see Appendix A), some values of s may be better than others in practice. In the numerical examples of Section 4, $s = 2$ provides the best overall performance.

Remark 1. Although gains in the asymptotic rate of convergence are always expected when substituting m by $\widetilde{m}$, real practical gains are most significant for scatterers with a low degree of regularity. For this reason, one need not typically perform this substitution if $m(x)$ is already sufficiently smooth; see, for example, Fig. 2 in Section 4.

2.4. Computation of the Fourier coefficients of g_{cmp}

As defined in Section 2.1, $g_{\text{cmp}}(x) = g(x)p(x)$ where $p(x)$ has support in $\Omega_{[-\sigma,\sigma]}$ for some $\sigma \in \mathbb{R}^3$ with $\sigma_q > 0$, $q = 1, 2, 3$. In our construction of p we make use of the following C^∞ function of one variable [29]

$$\phi(t) = \begin{cases} 1, & \text{for } |t| \leqslant r \\ \exp\left(\frac{2e^{-1/x}}{x-1}\right), & \text{for } r < |t| < R, \quad \text{where } x = \frac{|t|-r}{R-r} \\ 0, & \text{for } |t| \geqslant R, \end{cases} \tag{22}$$

where $R \leqslant \min_q \sigma_q$. Thus, we define $p(x) = \phi(|x|)$. Our choice of the parameters r and R is subject to considerations similar to those presented in Section 2.3 with regards to the size of $\Omega_{[\tilde{a},\tilde{b}]}$; see Remark 2 in Section 3.1 for details.

Using spherical coordinates, we have

$$\begin{aligned} (g_{\text{cmp}})_\ell &= \int_{\Omega_{[-\sigma,\sigma]}} g_{\text{cmp}}(z) e^{-2\pi i d_\ell \cdot z}\, dz = \int_0^R \int_{S^1} \frac{e^{i\kappa\rho}}{4\pi\rho} p(\rho) e^{-2\pi i \rho d_\ell \cdot \hat{z}} \rho^2\, d\rho\, d\sigma(\hat{z}) \\ &= \int_0^R g_{\text{cmp}}(\rho) j_0(2\pi|d_\ell|\rho)\rho\, d\rho = \frac{1}{2\pi|d_\ell|} \int_0^R p(\rho) e^{i\kappa\rho} \sin(2\pi|d_\ell|\rho) d\rho, \end{aligned} \tag{23}$$

where $\int_{S^1} d\sigma(\hat{z})$ denotes integration over the unit sphere, and where the second to last equality follows from [1, p. 32]. Note that the spherical coordinate transformation leads to two significant simplifications, namely, (i) the associated Jacobian cancels the ρ^{-1} singularity in Green's function, and (ii) the three-dimensional integral defining $(g_{\text{cmp}})_\ell$ is reduced to a one-dimensional integral, which needs to be evaluated for various values of the one-dimensional parameter $|d_\ell|$.

The integrals (23) depends on the two parameters κ and $|d_\ell|$; we can reduce this dependence to a single parameter as follows:

$$\begin{aligned} (g_{\text{cmp}})_\ell &= \frac{1}{\alpha} \int_0^R p(\rho) e^{i\kappa\rho} \sin(\alpha\rho) d\rho = \frac{1}{2i\alpha} \left[\int_0^R p(\rho) e^{i(\kappa+\alpha)\rho}\, d\rho - \int_0^R p(\rho) e^{i(\kappa-\alpha)\rho}\, d\rho \right] \\ &= \frac{1}{2i\alpha} \{ \mathscr{H}[p](\kappa+\alpha) - \mathscr{H}[p](\kappa-\alpha) \}, \end{aligned} \tag{24}$$

where $\alpha = 2\pi|d_\ell|$ and

$$\mathscr{H}[p](\omega) = \int_0^R p(\rho) e^{i\omega\rho}\, d\rho. \tag{25}$$

It is important to note that we can only use (24) to evaluate $(g_{\text{cmp}})_\ell$ when $|\ell| \neq 0$. For $|\ell| = 0$, on the other hand, we have

$$(g_{\mathrm{cmp}})_0 = \int_0^R \rho p(\rho) \mathrm{e}^{\mathrm{i}\kappa\rho}\, \mathrm{d}\rho = \mathscr{H}[\rho p(\rho)](\kappa).$$

Therefore, to compute $(g_{\mathrm{cmp}})_\ell$, we need an accurate and efficient method for the evaluation of $\mathscr{H}[f](\omega)$, where either $f(\rho) = p(\rho)$ or $f(\rho) = \rho p(\rho)$. This problem is not trivial since the values of ω that need to be considered can be quite large, thus producing highly oscillatory integrands. Furthermore, straightforward integration by means of the trapezoidal rule will give only first-order accuracy since $p(\rho)$ and $\rho p(\rho)$ cannot be extended as smooth and periodic functions. In Appendix B, we present an accurate and efficient method for the evaluation of these integrals.

3. Implementation details

The main components in an implementation of our algorithm are (i) the evaluation of the discrete convolution (15) and the discrete Fourier transforms (9) and (14), and (ii) the iterative solution of the associated linear algebra problem. These elements of our algorithm are discussed in Sections 3.1 and 3.2, respectively. A parallel implementation of our algorithm, finally, is described briefly in Section 3.3.

3.1. Evaluation of discrete convolutions and Fourier transforms

As described previously, the convolution with the smooth part of the Green's function is approximated by the discrete convolution

$$(K_{\mathrm{smth}}u)(x_j) \approx \Pi(h) \sum_{k=0}^{\overline{N}-1} (g_{\mathrm{smth}})_{j-k} \widetilde{m}_k u_k, \tag{26}$$

where $j_q = 0, \ldots, \overline{N}_q$ and $h_q = (\tilde{b}_q - \tilde{a}_q)/\overline{N}_q$ for $q = 1, 2, 3$ (see Section 2.2). Note that this operation involves the values of $(g_{\mathrm{smth}})_{jk}$ for $0 \leqslant j_q \leqslant \overline{N}_q$ and $0 \leqslant k_q \leqslant \overline{N}_q - 1$. Therefore, since $-\overline{N}_q + 1 \leqslant j_q - k_q \leqslant \overline{N}_q$, the FFT-evaluation of this convolution requires use of three-dimensional arrays of size $2\overline{N}$. (Note that the array $\widetilde{m}_k u_k$ needs to be zero padded; see [27, pp. 531–537].)

In detail, to compute the convolution (26), we first evaluate the discrete Fourier transforms

$$(\hat{g}_{\mathrm{smth}})_\ell = \sum_{j=0}^{2\overline{N}-1} (g_{\mathrm{smth}})_j \, \mathrm{e}^{-2\pi\mathrm{i}\ell\cdot(j_1/2\overline{N}_1, j_2/2\overline{N}_2, j_3/2\overline{N}_3)}$$

and

$$\widehat{\mathrm{mu}}_\ell = \sum_{j=0}^{2\overline{N}-1} \widetilde{m}_j u_j \, \mathrm{e}^{-2\pi\mathrm{i}\ell\cdot(j_1/2\overline{N}_1, j_2/2\overline{N}_2, j_3/2\overline{N}_3)}, \tag{27}$$

(where for $j_q > \overline{N}_q$, we set $\widetilde{m}_j u_j = 0$ and define $(g_{\mathrm{smth}})_j$ by periodic extension). We then use these quantities to obtain

$$\sum_{k=0}^{\overline{N}-1} (g_{\mathrm{smth}})_{j-k} m_k u_k = \sum_{\ell=0}^{2\overline{N}-1} (\hat{g}_{\mathrm{smth}})_\ell \widehat{\mathrm{mu}}_\ell \, \mathrm{e}^{2\pi\mathrm{i}\ell\cdot(j_1/2\overline{N}_1, j_2/2\overline{N}_2, j_3/2\overline{N}_3)} \tag{28}$$

for $j_q = 0, \ldots, \overline{N}_q$.

(Note that this straightforward approach for evaluation of discrete convolutions requires a factor of $2^3 = 8$ more memory than is otherwise necessary for storage of the unknowns u_j themselves. If memory usage becomes a limiting factor, it is possible to break the $\widetilde{m}_j u_j$ array into pieces and to compute the convolution with each piece separately. This saves memory, but substantially increases CPU-time.)

On the other hand, the approximation of the convolution with the singular part of the Green's function requires computation of the sums

$$(K_{\mathrm{cmp}}u)(x_j) \approx \sum_{\ell=-M}^{M} (g_{\mathrm{cmp}})_\ell \, (\mathrm{mu})_\ell \, \mathrm{e}^{2\pi \mathrm{i}\ell\cdot(j_1/\widetilde{N}_1, j_2/\widetilde{N}_2, j_3/\widetilde{N}_3)}, \tag{29}$$

where $M_q = \widetilde{N}_q/2 - 1, j_q = 0, \ldots, \overline{N}_q - 1$ and

$$(\mathrm{mu})_\ell \approx \eta_\ell \equiv \frac{1}{\Pi(\widetilde{N})} \sum_{j=0}^{\widetilde{N}-1} \widetilde{m}_j u_j \, \mathrm{e}^{-2\pi \mathrm{i}\ell\cdot(j_1/\widetilde{N}_1, j_2/\widetilde{N}_2, j_3/\widetilde{N}_3)}. \tag{30}$$

These sums may also be computed using three-dimensional FFTs, in this case, of size $\widetilde{N}$.

The parameters $\overline{N}$ and $\widetilde{N}$ characterize the sizes of the sums in (27) and (30), both of which are discrete Fourier transforms of the array $(\widetilde{m}_j u_j)$ – with appropriate zero-padding. These parameters need not be related, in principle, but time and memory savings result if the relation $\widetilde{N} = 2\overline{N}$ is enforced. Indeed, if $\widetilde{N} \neq 2\overline{N}$, it is necessary to compute and store the arrays $\widehat{\mathrm{mu}}_\ell$ and η_ℓ separately, with similar duplication in the storage of $(g_{\mathrm{cmp}})_\ell$ and $(\hat{g}_{\mathrm{smth}})_\ell$ as well as in the computation of (28) and (29). With the choice $\widetilde{N} = 2\overline{N}$, on the other hand,

$$\eta_\ell = \widehat{\mathrm{mu}}_\ell, \tag{31}$$

(see (27) and (31)), and defining

$$\hat{g}_\ell = (g_{\mathrm{cmp}})_\ell + (\hat{g}_{\mathrm{smth}})_\ell, \tag{32}$$

we need only compute and store $\widehat{\mathrm{mu}}_\ell$, store $\hat{g}_\ell$ and evaluate the FFT of $\hat{g}_\ell \widehat{\mathrm{mu}}_\ell$. Thus, the selection $\widetilde{N} = 2\overline{N}$ leads a simpler algorithm, with reduced time and memory requirements. A further consequence of this choice is that, in a parallel implementation of our algorithm, only a single array $\eta_\ell = \widehat{\mathrm{mu}}_\ell$ must be scattered and gathered at each iteration, thus significantly reducing communication costs.

In summary, by choosing $\widetilde{N} = 2\overline{N}$, an implementation of our algorithm for computing the integral operator requires the following five steps: (i) copy the values of mu (stored in an array of size $\overline{N}$) into a zero-padded array of size $2\overline{N}$; (ii) compute an (inverse) FFT of this array (see (27) and (30)); (iii) multiply the result by the $\hat{g}_\ell$ as defined in (32) (these values of $\hat{g}_\ell$ are evaluated once at the beginning of the computation); (iv) compute an FFT of the result of this multiplication; and (v) copy these values back into the array of values of mu. These steps require a total of $\mathcal{O}(\Pi(\overline{N}) \log \Pi(\overline{N}))$ floating-point operations. Note that in a parallel implementation the steps (i) and (v) are generally quite substantial, involving parallel scatters and gathers.

Remark 2. Note that the choice of $\widetilde{N} = 2\overline{N}$ implies that $B - A = 2(\tilde{b} - \tilde{a})$. Hence, since we require that $\Omega_{[\tilde{a}-\sigma,\tilde{b}+\sigma]} \subset \Omega_{[A,B]}$, where $\mathrm{supp}(p) \subset \Omega_{[-\sigma,\sigma]}$, we have that $R \leqslant \sigma_q \leqslant \frac{1}{2}(\tilde{b}_q - \tilde{a}_q)$. In other words, the support of p must fit entirely inside the support of $\widetilde{m}$ (when translated appropriately). Choosing a value of R that is smaller than this maximum size has few, if any, real advantages: a smaller value of R may require a finer discretization to resolve the variations in p, and only minimal savings result from a smaller value of σ in the *pre-computation step* of evaluating the coefficients $(g_{\mathrm{cmp}})_\ell$. Having chosen R, we then choose r, which also involves a trade-off: a small value of r results in a more gradual transition in p from 0 to 1, but it also brings the support of g_{smth} closer to the singularity in the Green's function.

3.2. Iterative linear solvers

The algorithm described above for the evaluation of the matrix-vector products can be combined with any suitable iterative linear solver to obtain a fast, higher-order method for the solution of scattering problems. For our non-Hermitian linear system, one may, in principle, use solvers such as GMRES, CGS, BiCGSTAB, and QMR [30]. We have found, however, that only GMRES and BiCGSTAB performed consistently well for our problem. In fact, for each of the other solvers mentioned above, we found an example in which either rapid divergence or stagnation occurred.

Depending on the characteristics of a particular problem under consideration as well as the computing hardware available, use of one of the solvers GMRES or BiCGSTAB might be clearly advantageous over use of the other. As is known, GMRES always requires fewer matrix-vector products than BiCGSTAB to converge to a given residual error [30, p. 49]. However, at each iteration GMRES stores a new Krylov subspace basis vector whereas BiCGSTAB does not. Hence, in problems which require many iterations, GMRES may exhaust the system memory. Of course, in such cases, one may restart GMRES after a given number of iterations, thereby limiting the memory used, but also slowing the convergence rate. For these reasons and on the basis of numerical experiments, in problems requiring many iterations, BiCGSTAB has become our method of choice. On the other hand, since GMRES requires fewer matrix-vector products than BiCGSTAB, in problems for which memory is not a limiting factor, GMRES is clearly preferable.

3.3. Parallel implementation

It is not difficult to produce an efficient parallel implementation of our algorithm: as follows from Sections 3.1 and 3.2, all that is required is an efficient parallel FFT package and an effective parallel iterative solver for the linear system. Our implementation uses the parallel FFT package *fftw* [31,32] and the parallel iterative solvers in the software package PETSc [33–35].

4. Numerical results

In this section we demonstrate the properties of the algorithm introduced in this paper through a variety of numerical examples. In each case we present results for both the near field $u = u^{\mathrm{i}} + u^{\mathrm{s}}$ and the far field u_∞ – which is given by the expression [1, p. 223]

$$u_\infty(\hat{x}) = -\frac{\kappa^2}{4\pi}\int_\Omega \mathrm{e}^{-\mathrm{i}\kappa\hat{x}\cdot y} m(y)u(y)\mathrm{d}y, \tag{33}$$

where $\hat{x}$ is a point on the unit sphere. In our tests the integral in (33) was evaluated with higher-order accuracy by means of the trapezoidal rule after replacing m with $\widetilde{m}$ and Ω with $\Omega_{[\tilde{a},\tilde{b}]}$; see Section 2.3 for a discussion and justification of this procedure. In all cases the incident field used was the plane wave $u_i(x,y,z) = \mathrm{e}^{\mathrm{i}\kappa x}$. Many of the results presented here were obtained from runs on a a single processor: a 1.7 GHz Pentium Xeon with 2 GB of RAM. We also demonstrate the parallel capabilities of our codes in a number of cases, using a number between $P = 2$ and $P = 32$ of Pentium Xeon 1.7 GHz processors arranged in pairs, each pair sharing 1 GB of RAM. In all cases the processors were connected via a Myrinet interconnect.

The tables presented in this section demonstrate the higher-order accuracy and $\mathcal{O}(\Pi(\overline{N})\log\Pi(\overline{N}))$ complexity of our method: they report the number of discretization points $\overline{N}$ (in the form $\overline{N}_1 \times \overline{N}_2 \times \overline{N}_3$), the number of processors P used in the computation, the wall-clock time T_{setup} required prior to the iterative solution of the linear system (which is dominated by the time required to compute the coefficients $(g_{\mathrm{cmp}})_\ell$), the number of iterations N_{iter} of the linear algebra solver (either GMRES or BiCGSTAB), and the (average) wall-clock time per iteration T_{iter}. The tables also list the maximum errors in the near field $(\epsilon_u^{\mathrm{nf}})$ and the far

Table 1
Convergence for the two-layer sphere *with* Fourier smoothing

$\overline{N}$	P	T_{setup} (s)	N_{iter}	T_{iter} (s)	ϵ_u^{nf}	ϵ_u^{ff}
$10\times10\times10$	1	3.38	15	0.06	0.245	0.145
$20\times20\times20$	1	5.65	20	0.46	2.27e−2	4.77e−3
$40\times40\times40$	1	24.38	20	3.79	5.69e−3	9.46e−4
$80\times80\times80$	1	181.86	20	30.11	1.48e−3	5.25e−5
$160\times160\times160$	32	49.25	20	8.80	2.38e−4	6.68e−6

Table 2
Convergence for the two-layer sphere *without* Fourier smoothing: clearly use of Fourier smoothing (see Table 1) gives rise to significant accuracy improvements

$\overline{N}$	P	T_{setup} (s)	N_{iter}	T_{iter} (s)	ϵ_u^{nf}	ϵ_u^{ff}
$10\times10\times10$	1	2.27	15	0.06	3.04	1.66
$20\times20\times20$	1	4.29	20	0.46	0.781	0.401
$40\times40\times40$	1	20.86	20	3.71	0.187	9.44e−2
$80\times80\times80$	1	158.98	20	29.08	7.07e−2	2.97e−2
$160\times160\times160$	32	49.00	20	9.02	2.99e−2	4.29e−3

field (ϵ_u^{ff}), computed as the maxima of differences between the computed solution and a reference solution at relatively fine discretizations of the scattering body and the unit sphere, respectively. The reference solutions used for the layered-sphere tests were obtained from the exact Mie expression; in all other cases the reference solution was obtained as the numerical solution resulting from a fine discretization. Throughout this section the notation "e−n" stands for 10^{-n} so that, e.g., 6.68e−6 = 6.68×10^{-6}.

4.1. Moderately-sized layered sphere

Results obtained for a moderately-sized piecewise-constant two-layer sphere are presented in Fig. 1 and Table 1. The refractive indices, non-dimensional radii and incident wavenumber were chosen to be $n_1=\sqrt{2}$, $n_2=\sqrt{3}$, $a_1=0.5$ and $a_2=1.0$ and $\kappa=4$; thus, denoting by λ_{int} the wavelength corresponding to the outer layer, this scatterer has a diameter of $2.21\lambda_{\text{int}}$. For the algorithm parameters we used the values $\tilde{a}=(-1.25,-1.25,-1.25)$, $\tilde{b}=(1.25,1.25,1.25)$, $r=0.5$, and $R=1.0$. The near and far field reference solutions were computed analytically. The analytical near field values were obtained on a $16\times16\times16$ mesh with corners at (−1, −1, −1) and (1, 1, 1): a subset of the computed solution coincides with this mesh, and we computed the maximum near field error on these coincident meshes. The analytical far field values were evaluated on a 32×16 mesh covering the unit sphere; we computed the far field error on this mesh.

The last row of results in Table 1 was obtained from parallel runs in $P=32$ processors; we note that the parallel speedup is nearly perfect. Indeed, an increase by a factor of eight in the discretization from that of the fourth row of this table, which should lead to an increase by about a factor of eight in the computational time of our $\mathcal{O}(\Pi(\overline{N})\log\Pi(\overline{N}))$ algorithm, combined with a decrease by a factor of 32 that could optimally result from use of 32 processors, would optimally result in a decrease by a factor of 4 in the overall computing times – as indeed observed approximately in both the setup time and the time per iteration in the fourth and fifth rows of Table 1. A more detailed discussion of the parallel performance of our method is presented in the following paragraphs.

Per the discussion of Section 2.3, to obtain higher-order convergence m was replaced by $\widetilde{m}$: we see in Table 1 that the resulting near field solution converges roughly as h^2 while the corresponding far field solution converges as h^3. These convergence rates agree with those established for our related two-dimensional approach

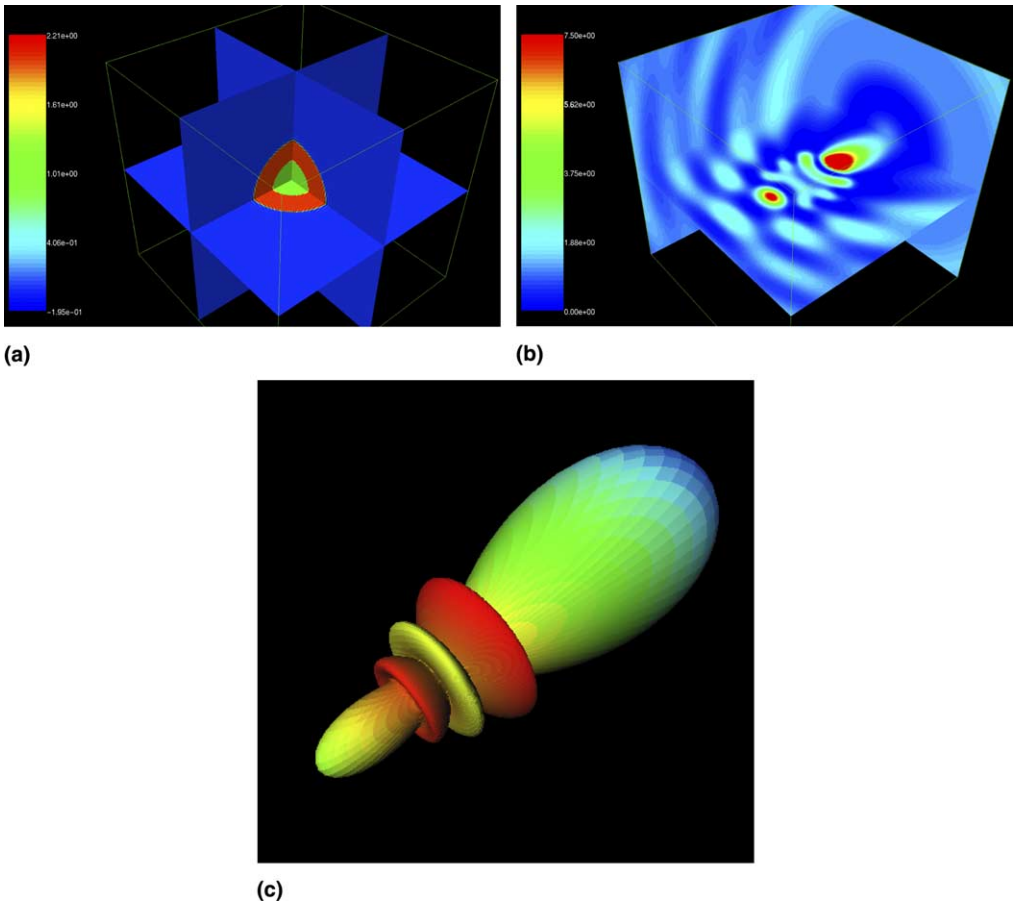

Fig. 1. Visualizations for the two-layer sphere. (a) Scatterer ($q = n^2 - 1$); (b) near field intensity ($|u|^2$); (c) far field ($|u_\infty|$).

[22]. For comparison we present, in Table 2, the results that are obtained when m is *not* replaced by $\widetilde{m}$. In this case, the convergence rates are significantly lower in both the near and the far fields, as expected.

4.2. Parallel performance, choice of iterative solver, memory usage

As mentioned in Section 3.2, memory vs. speed trade-offs result as the GMRES iterative linear algebra solver is substituted by another solver such BiCGSTAB. The impact of the iterative solver in the perform-

Table 3
Two-layer sphere with Fourier smoothing $\overline{N} = 160 \times 160 \times 160$

P	Solver	T_{setup} (s)	N_{iter}	N_{mvm}	T_{mvm} (s)	ϵ_u^{nf}	ϵ_u^{ff}
32	GMRES	49.25	20	20	8.80	2.38e−4	6.68e−6
32	BiCGSTAB	48.83	19	38	9.25	2.38e−4	7.03e−6
8	GMRES	191.56	20	20	35.27	2.38e−4	6.68e−6
8	BiCGSTAB	186.20	19	38	35.81	2.38e−4	7.04e−6
2	GMRES	725.07	20	20	130.22	2.38e−4	6.69e−6
2	BiCGSTAB	718.23	19	38	131.03	2.38e−4	7.07e−6

ance of our algorithm in a distributed-memory parallel environment is demonstrated in Table 3 and discussed in what follows.

To gain an understanding of the impact of the memory-speed trade-offs mentioned above it is important to have detailed information about the memory usage of the various portions of our algorithm, as detailed in what follows. Excluding the memory required by the iterations in the linear solver, the present implementation of our method requires almost exactly 31 times as much memory as it takes to store the unknowns: for example, for a $80 \times 80 \times 80$ mesh the code uses an amount of memory equal to that needed to store 31×81^3 unknowns. Of this factor of 31, 24 units are required for the large FFT and Fourier coefficient arrays, 1 unit is for the incident field, 1 unit is for the refractive index, and 1 unit is for the unknowns, for a total of about 27 units. The remaining 4 units are taken up almost entirely by the overhead required by the PETSc linear solvers, including work arrays, etc.

In addition to these 31 units of memory, GMRES requires 1 additional unit for every iteration, while BiCGSTAB requires 2 units for an arbitrary number of iterations. Thus, two iterations of GMRES requires as much memory as an unlimited number of iterations of BiCGSTAB (33 units), while 20 iterations of GMRES and BiCGSTAB require 51 and 33 memory units, respectively. The trade-off in this significant memory savings afforded by BiCGSTAB is computing time: while, in our context, GMRES and BiCGSTAB yield similar residuals and errors for a given number of iterations, each iteration of BiCGSTAB requires two matrix-vector multiplies while GMRES requires only one.

(This analysis does not scale perfectly as the number of processors is increased since there is a fixed memory overhead associated with use a parallel infrastructure; e.g., each processor requires additional memory to manage the parallel communication.)

The parallel performances of the GMRES and BiCGSTAB versions of our algorithm are demonstrated in Table 3. In addition to the standard notations of this section, in this table N_{mvm} denotes the number of matrix-vector multiplies and T_{mvm} denotes the average time required per matrix-vector multiply. All of these runs were performed with a relative residual tolerance of 3.25e−7, for which GMRES requires 20 iterations, which matches the number of iterations reported in the last row of Table 1. In the $P = 32$ and $P = 8$ cases each pair of processors shared 1 GB of memory. In the $P = 2$ cases, in turn, each one of the two processors was furnished with 2GB of memory; the actual memory used by GMRES in this case was 3.5GB, whereas BiCGSTAB used 2.2GB of memory.

4.3. Array of potentials

In Fig. 2 and Table 4, we present the results for the $5 \times 5 \times 5$ array of smooth inhomogeneous scatterers. (This scattering configuration is meant to demonstrate the capabilities of our method for the evaluation of scattering by an array of quantum scattering potentials.) The centers of the scatterers are separated by 1λ, where λ is the wavelength of the incident field, and the function $m(x) = 1 - n^2(x)$ for each of these scatterers is given by a spherically symmetric function of the form (22), with $r = 0$, $R = 0.5\lambda$, and $m = -1$ at the center of each scatterer. Therefore, the support of the array is contained in a $5\lambda \times 5\lambda \times 5\lambda$ box. Since $m \in C^\infty$, we do not replace m by $\widetilde{m}$, but instead integrate directly with the trapezoidal rule on the $5\lambda \times 5\lambda \times 5\lambda$ integration domain ($\tilde{a} = (-2.5\lambda, -2.5\lambda, -2.5\lambda)$ and $\tilde{b} = 2.5\lambda, 2.5\lambda, 2.5\lambda$). For the partition of unity function p, we set $r = 0.5\lambda$ and $R = 2.5\lambda$. Since no analytical solution for this scattering configuration is known, the near field reference solution for this example was computed on a $320 \times 320 \times 320$ mesh, and the corresponding linear system was solved using GMRES with a relative residual tolerance of 1e−10. The maximum near field error was evaluated on the computational mesh, which, for each value of $\overline{N}$ in Table 4, is a subset of the reference solution mesh. The reference far field was evaluated from the reference near field on a 64×32 mesh on the unit sphere. As expected, we observe a very rapid convergence rate.

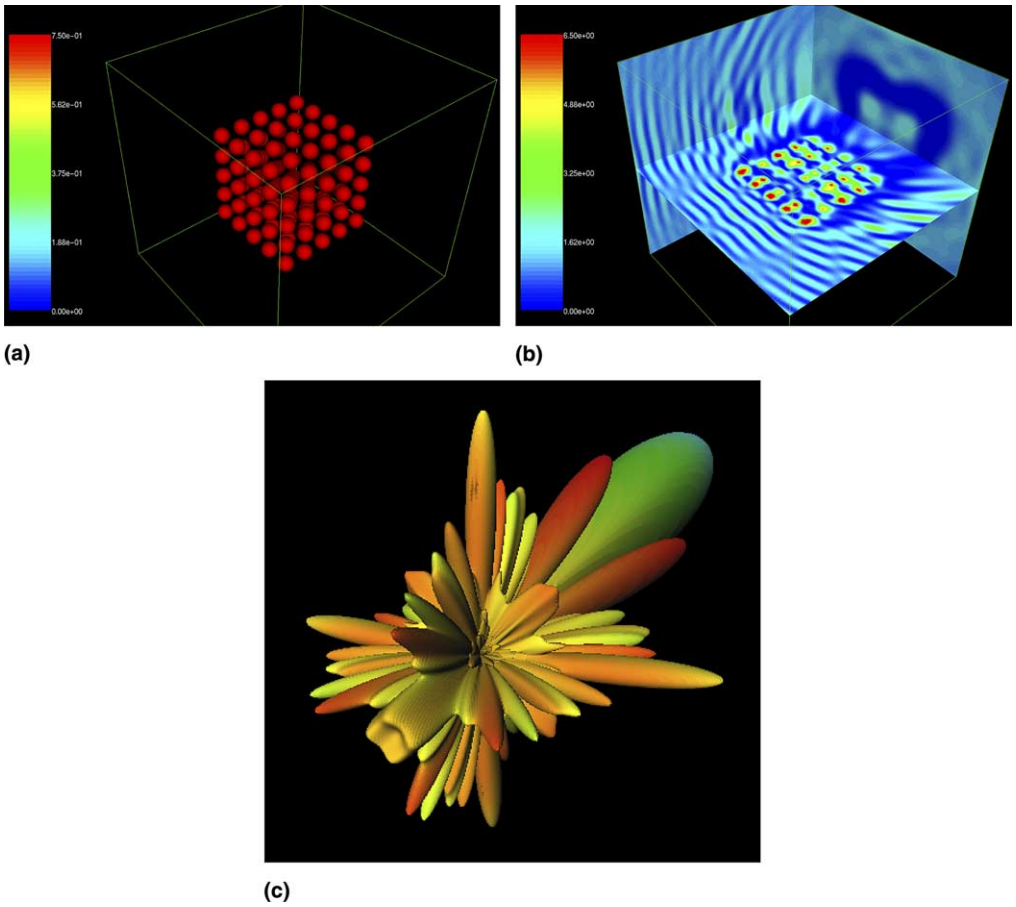

Fig. 2. Visualizations for the array of smooth scatterers. (a) Scatterer ($q = n^2 - 1$); (b) near field intensity ($|u|^2$); (c) far field ($|u_\infty|$).

Table 4
Convergence for the array of smooth scatterers

$\overline{N}$	P	T_{setup} (s)	N_{iter}	T_{iter} (s)	ϵ_u^{nf}	ϵ_u^{ff}
$20 \times 20 \times 20$	1	4.46	15	0.44	0.600	5.56
$40 \times 40 \times 40$	1	23.51	25	3.64	6.49e−3	5.48e−2
$80 \times 80 \times 80$	1	171.88	30	28.43	1.66e−4	4.58e−4
$160 \times 160 \times 160$	32	48.31	35	8.74	2.08e−6	1.62e−5

4.4. Complex scatterer

The next scattering geometry is displayed, in two orthogonal slices, in Fig. 3(a). This complex, discontinuous scatterer is created by adding together a cube, two spheres, two layered spheres, and six smooth inhomogeneities similar to those in the previous example. In detail, the scatterer of Fig. 3 is constructed as follows: beginning with a cube of side 4 centered at the origin and $m = -1$, we add two unit spheres each with $m = +1$ and centered at $(0, -1, 0)$ and $(0, 1, 0)$, respectively; this essentially cuts two spheres out of the cube ($m = 0$ insides those spheres). Then, two layered spheres of unit radius (with $a_1 = 0.5$, $m_1 = -1.25$, $a_2 = 1$, and $m_2 = -1$) are placed tangent to the two faces of the cube that are orthogonal to the y-axis.

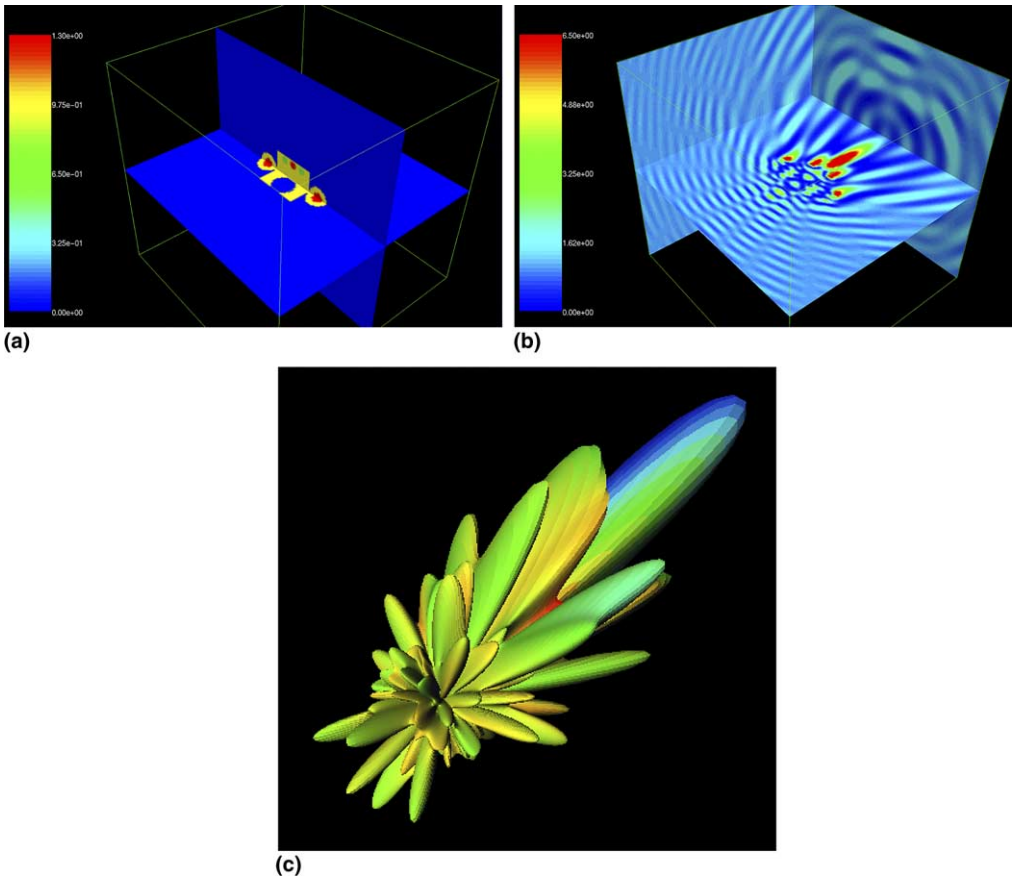

Fig. 3. Visualizations for the complex scatterer. (a) Scatterer ($q = n^2 - 1$); (b) near field intensity ($|u|^2$); (c) far field ($|u_\infty|$).

Finally, we add two sets of three smooth, inhomogeneous scatterers, with $r = 0$, $R = 0.5$, and $m = \pm 0.5$ at the centers, which lie in the cube on the yz-plane. With $\kappa = 4$, the length of this scatterer along the longest dimension is $7.64\lambda_{\text{int}}$, where λ_{int} is the wavelength corresponding to inner layer of the two layered spheres. We set $\tilde{a} = (-2.5, -5, -2.5)$, $\tilde{b} = (2.5, 5, 2.5)$, $r = 0.5$ and $R = 2.5$.

In accordance with (18), our solver makes use of the smoothed version

$$\widetilde{m}(y) = p_{m_1}(y) m_1^{\mathrm{F}}(y) + p_{m_2}(y) m_2^{\mathrm{F}}(y) + \cdots \tag{34}$$

of the function m, where m_j^{F} is the truncated Fourier series for the jth discontinuous component of the scatterer. (As in the previous example, we do not replace the C^∞ components of the scatterer by their Fourier-smoothed approximations.) As shown in (34), it suffices to compute the Fourier coefficients of simple building blocks, i.e., the cube, the spheres, and the layered spheres – the corners, cusps, and inhomogeneities present *no additional difficulties*. This example illustrates one of the powerful features of this method: it can treat complicated scatterers through simple addition of Fourier coefficients.

The reference near field solution for this example was evaluated on a $256 \times 512 \times 256$ mesh using GMRES with a relative residual tolerance of 1e−10. The corresponding reference far field was computed on a 64×32 mesh on the unit sphere. The computational results for this example are contained in Fig. 3 and Table 5; again, we observe higher-order convergence in the near and far fields.

Table 5
Convergence for the complex scatterer

$\overline{N}$	P	T_{setup} (s)	N_{iter}	T_{iter} (s)	ϵ_u^{nf}	ϵ_u^{ff}
$16\times32\times16$	1	6.81	75	0.45	0.309	0.673
$32\times64\times32$	1	36.01	75	3.55	1.73e−2	3.37e−2
$64\times128\times64$	8	41.10	100	4.60	3.51e−3	1.29e−3
$128\times256\times128$	32	83.99	100	9.68	8.95e−4	1.11e−4

4.5. Large four-layer sphere

Finally, we consider a large spherically layered sphere, each one of whose layers is filled by a material with complex refractive index. This scatterer was considered in [12] and is, to our knowledge, the largest inhomogeneous scatterer considered in the literature to date. In detail, the frequency of the incident field is $f = 1$ GHz (yielding $\kappa \approx 20.9$ m^{-1}); the diameters of the layers are 1.515, 2.115, 2.415, and 2.865 m, respectively; and the dielectric constants and conductivities of the layers are $\epsilon_r = 4, 2.56, 4, 2.25$ and $\sigma = 0.1, 0.07, 0.1, 0.02$ S/m, respectively, where the complex permittivity is defined as $\epsilon = \epsilon_r\epsilon_0 + i\sigma/\omega$ with $\epsilon_0 \approx 8.85\times10^{-12}$ F/m and $\omega = 2\pi f$. The complex refractive index, in turn, is given by $n^2 = \epsilon/\epsilon_0$. Thus, in terms of the minimum wavelength λ_1 (the wavelength in the first and third layers), this layered sphere has a volume of $3648\lambda_1^3$. For this example we used the algorithm parameters $\tilde{a} = (-1.75\text{ m}, -1.75\text{ m}, -1.75\text{ m})$, $\tilde{b} = (1.75\text{ m}, 1.75\text{ m}, 1.75\text{ m})$, $r = 0.5$ m, and $R = 1.5$ m.

With $\overline{N} = (192, 192, 192)$ (7.19 million unknowns) and a relative residual tolerance of 0.5% = 0.005 (the same tolerance and roughly the same number of points per wavelength as those used in [12] for this scatterer), use of GMRES and BiCGSTAB require, respectively, 88 iterations (88 matrix-vector multiplies) and 71 iterations (142 matrix-vector multiplies) to match the prescribed residual tolerance. The resulting values of the maximum far field error are 1.27e−1 and 4.87e−2, (computed as the maximum difference between the numerical and the analytical solution at 1024 angles; see Fig. 4). The maximum value 4.87e−2 of the far field error resulting from BiCGSTAB occurs near the location of the minimum modulus of the far field, while the 0.127 GMRES maximum far field error occurs near the location of the maximum modulus of the far field. Therefore, in a log-scale plot the RCS produced by GMRES matches more closely the exact solution than the BiCGSTAB solution does, although the former actually contains a larger maximum error.

The time statistics for these runs are as follows: using 32 processors, the setup time was 85.7 s and the average time per matrix-vector multiply was 15.96 s. Thus, if the GMRES computation were performed on a single processor (which was not possible because of memory limitations), it would require no more than 32×15.96 s = 8.51 min per iteration; the BiCGSTAB calculation, in turn, would have required no more than 2×8.51 min = 17.02 min per iteration.

Ours being a higher-order method, these results may be considered to compare quite favorably with the 64.68 minutes per iteration required in [12], even when we take into account the factor of three more unknowns (21.23 million unknowns) required to compute the vector-valued solution to the integral equation considered in that paper. The advantages provided by the present higher-order approach are realized more fully as the residual tolerance is decreased somewhat – which shows that the higher-order discretization considered here actually approximates the integral operator much more closely than suggested by results provided by previous methods. Indeed, using a residual tolerance of 1e−5 and after 177 BiCGSTAB iterations (344 matrix-vector products; 2.5 as many iterations as those required by the 0.005 tolerance considered above) our method produces the maximum error of 3.08e−4: two full orders of magnitude smaller error than that resulting from the larger tolerance, and, from our reading of the graphical results of [12], at least that much smaller than those provided by the previous approach for the same number of points per wavelength.

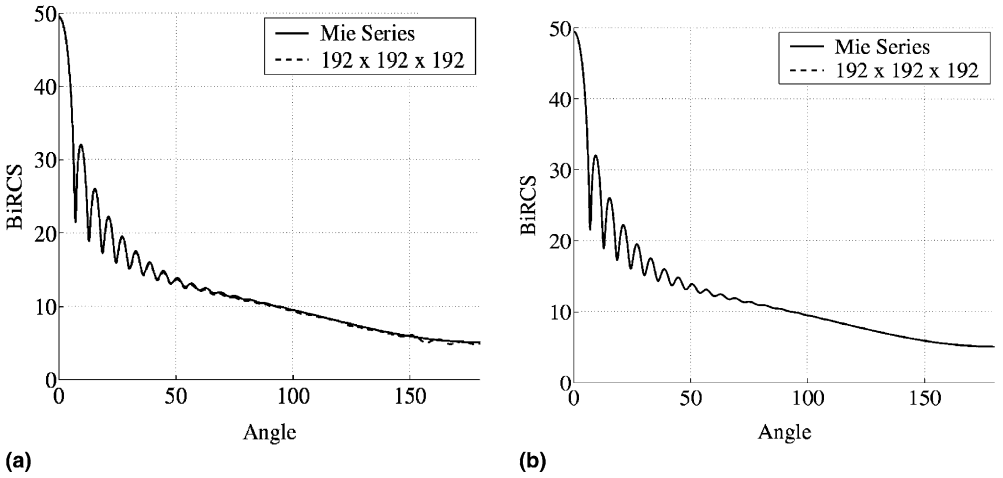

Fig. 4. BiRCS for four-layer sphere with $\overline{N} = (192, 192, 192)$. (a) Residual tolerance = 5e−3. (b) Residual tolerance = 1e−5.

Acknowledgement

The original implementation of the method described in the appendix (for computing the Fourier coefficients of the singular part of the Green's function) was written at Caltech by Samba Ba, a visiting undergraduate from École Polytechnique, France. Color visualizations of the scatterers and the near field intensities were generated with the VTK-based visualization tool Vizamrai, developed by Steven Smith at the Center for Applied Scientific Computing (CASC) at Lawrence Livermore National Laboratory. Far field visualizations were created with SceneViewer, a 3D viewer for Open Inventor scenes; the scene graph files were generated with software written by Randy Paffenroth at Caltech.

Appendix A. Higher-order integration of Fourier-smoothed integrands in one dimension

As discussed in Section 2.3, the higher-order accurate approximation of an integral of a discontinuous function through trapezoidal rule integration of the Fourier-smoothed version of the function is a central aspect of our approach. In this appendix, we provide a simple example and a proof of this fact in one dimension See Fig. A.1.

Consider the integral on the interval $[-1, 1]$ of the product of a discontinuous and piecewise-smooth function

$$\phi(x) = \begin{cases} 1 & \text{if } |x| \leqslant 2/3 \\ 0 & \text{otherwise,} \end{cases}$$

and a C^1, piecewise-smooth and periodic function ψ, which is defined on its period $[-1, 1]$ as

$$\psi(x) = \begin{cases} 9(x+1)^2 & \text{if } -1 \leqslant x \leqslant -2/3 \\ -\frac{9}{2}x^2 + 3 & \text{if } -2/3 \leqslant x \leqslant 2/3 \\ 9(x-1)^2 & \text{if } 2/3 \leqslant x \leqslant 1. \end{cases}$$

We replace ϕ by its truncated Fourier series on $[-1, 1]$, i.e., the same period as ψ,

$$\phi^{\mathrm{F}}(x) = \sum_{\ell=-F}^{\mathrm{F}} \phi_\ell \, \mathrm{e}^{\pi \mathrm{i} \ell x},$$

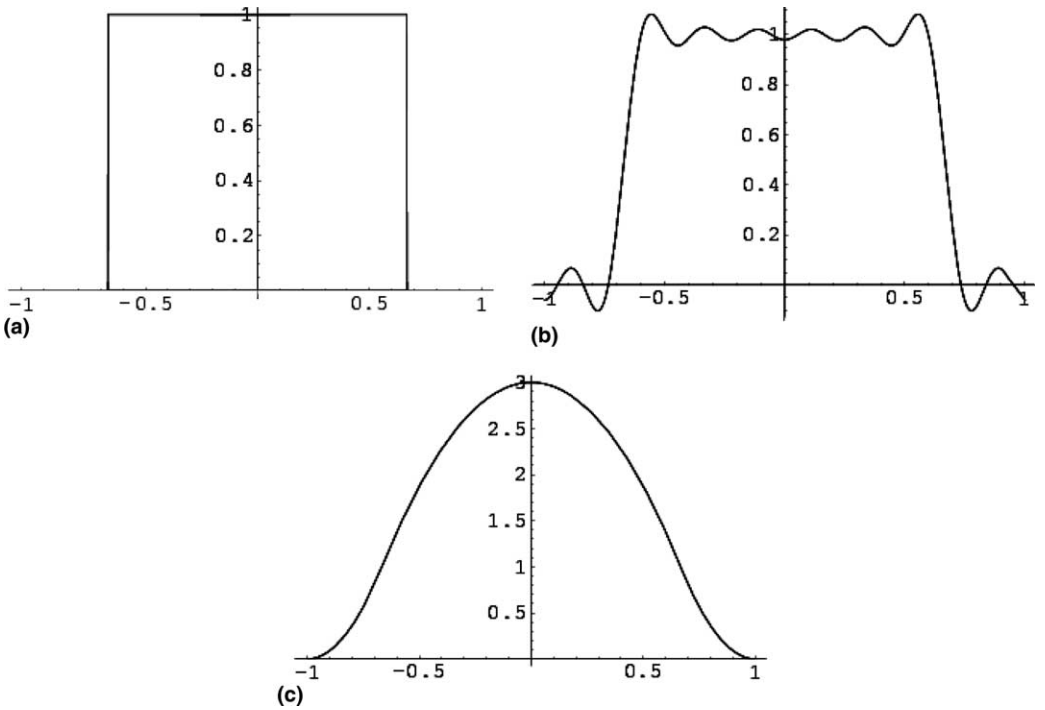

Fig. A.1. Example of Fourier smoothing. (a) Discontinuous function ϕ. (b) Fourier-smoothed function ϕ^F. (c) C^1 function ψ.

where the Fourier coefficients ϕ_ℓ are easily found analytically. (In general, we assume that the Fourier coefficients of the inhomogeneity m are either known analytically, as in this case, or have been computed accurately.) Table A.1 compares the accuracy obtained by means of the trapezoidal rule with and without the substitution of ϕ by ϕ^F. As expected, without the Fourier smoothing, one obtains only first-order convergence. With the Fourier smoothing, on the other hand, we observe approximately *third-order* convergence *to the true value of the integral* (see also [20,21]).

This rather surprising result can be proven in general for a piecewise-smooth $\phi(x)$ and a C^1, piecewise-smooth and periodic $\psi(x)$ on an interval $[a,b]$. As above, we replace ϕ by ϕ^F and approximate the resulting integral by means of the trapezoidal rule with N points. We denote the error in this approximation by $\varepsilon(N)$, i.e.,

$$\varepsilon(N) = \int_a^b \phi(x)\psi(x)\mathrm{d}x - h\sum_{n=0}^{N-1} \phi^{\mathrm{F}}(nh)\psi(nh),$$

where $h = \frac{b-a}{N}$. The error, $\varepsilon_{\mathrm{trap}}(N)$, in the trapezoidal rule integration of a given periodic function f (with Fourier coefficients f_ℓ) over its period $[a,b]$ is given by

$$\varepsilon_{\mathrm{trap}}(f,N) = \int_a^b f(x)\mathrm{d}x - h\sum_{n=0}^{N-1} f(nh) = (b-a)f_0 - (b-a)\sum_{p=-\infty}^{\infty} f_{pN} = -(b-a)\sum_{p\neq 0} f_{pN}. \tag{A.1}$$

Here we have used the fact that

$$\frac{1}{N}\sum_{n=0}^{N-1} \mathrm{e}^{2\pi \mathrm{i}\ell n/N} = \begin{cases} 1 & \text{if } \ell = pN \text{ for } p \in \mathbb{Z} \\ 0 & \text{otherwise.} \end{cases}$$

Table A.1
Higher-order trapezoidal rule integration via Fourier smoothing

N		Abs. error
Convergence for $\int_{-1}^{1}\phi(x)\psi(x)\mathrm{d}x$		
4		0.264
8		6.42×10^{-2}
16		4.71×10^{-2}
32		1.20×10^{-2}
64		1.07×10^{-2}
128		5.13×10^{-3}
256		2.62×10^{-3}
	F	Abs. error
Convergence for $\int_{-1}^{1}\phi^{F}(x)\psi(x)\mathrm{d}x$		
4	2	6.93×10^{-2}
8	4	4.11×10^{-4}
16	8	4.87×10^{-4}
32	16	3.86×10^{-5}
64	32	4.96×10^{-6}
128	64	7.25×10^{-7}
256	128	6.68×10^{-8}

By integrating by parts, it follows that the Fourier coefficients of ϕ and ψ on the interval $[a,b]$ satisfy

$$\phi_\ell = \mathcal{O}\left(\frac{1}{|\ell|}\right) \tag{A.2}$$

$$\psi_\ell = \mathcal{O}\left(\frac{1}{|\ell|^3}\right) \tag{A.3}$$

as $\ell \to \infty$. Using (A.1) as well as (A.2) and (A.3), we see that

$$\begin{aligned}\varepsilon(N) &= \int_a^b (\phi(x)-\phi^{\mathrm{F}}(x))\psi(x)\mathrm{d}x + \varepsilon_{\mathrm{trap}}(\phi^{\mathrm{F}}\psi,N) = (b-a)\sum_{|\ell|>F}\phi_\ell\psi_{-\ell} + \varepsilon_{\mathrm{trap}}(\phi^{\mathrm{F}}\psi,N)\\ &= \mathcal{O}\left(\frac{1}{F^3}\right) + \varepsilon_{\mathrm{trap}}(\phi^{\mathrm{F}}\psi,N).\end{aligned}$$

The Fourier coefficients of $\phi^F\psi$, which are needed to to bound $\varepsilon_{\mathrm{trap}}(\phi^F\psi,N)$, are given by

$$(\phi^{\mathrm{F}}\psi)_\ell = \sum_{|k|\leqslant F}\phi_k\psi_{\ell-k}.$$

Thus, choosing $N = sF$ for a fixed integer $s \geqslant 2$, we finally obtain

$$\begin{aligned}\varepsilon_{\mathrm{trap}}(\phi^{\mathrm{F}}\psi,sF) &= \sum_{p\neq0}\sum_{|k|\leqslant F}\phi_k\psi_{psF-k}\\ &\leqslant C_1\sum_{p=1}^{\infty}\sum_{k=1}^{\mathrm{F}}\frac{1}{k}\frac{1}{(psF-k)^3}\\ &\leqslant C_1\sum_{p=1}^{\infty}\frac{1}{(psF-F)^3}\sum_{k=1}^{\mathrm{F}}\frac{1}{k}\\ &\leqslant C_2\frac{\log F}{F^3},\end{aligned} \tag{A.4}$$

for some positive constants C_1 and C_2, which are independent of F. Therefore, as seen in the numerical experiments, this integration scheme yields nearly third-order accuracy, i.e.,

$$\varepsilon(N) = \mathcal{O}\left(\frac{\log N}{N^3}\right) \tag{A.5}$$

as $N \to \infty$, where $N = sF$ for some fixed integer $s \geqslant 2$. It is easy to see that significantly higher-orders of convergence are obtained for more regular integrands.

Appendix B. Efficient, high-order accurate evaluation of fourier integrals

Given a smooth, compactly supported, real-valued function $f(t)$ for $t \in \mathbb{R}$, we seek to compute the integral

$$I(\omega) = \int_a^b f(t) \mathrm{e}^{\mathrm{i}\omega t}\, \mathrm{d}t \tag{B.1}$$

for various values of $\omega \in [\omega_{\min}, \omega_{\max}]$. Since $I(-\omega) = \overline{I(\omega)}$, we restrict our attention to $\omega_{\min} \geqslant 0$.

To do this, we introduce a modified version of the method described in [27, pp. 577–584] and the references therein. First, to obtain a high-order approximation of the function f, we use piecewise polynomials interpolants $\psi(s)$ of order q where $-q \leqslant s \leqslant q$ such that $\psi(0) = 1$ and $\psi(s) = 0$ for integer values $s = -q, \ldots, q$:

$$f(t) \approx \sum_{k=-(q-1)}^{N+(q-1)} f_k \psi\left(\frac{t - t_k}{\delta}\right),$$

where $\delta = (b - a)/N$, $t_k = a + k\delta$ and $f_k = f(t_k)$, and where, for simplicity, we assume that the functions $\psi(s)$ are even; see Sections B.1 and B.2 for specific choices of $\psi(s)$. (Note that this approximation requires knowledge of f outside of the interval $[a,b]$; this presents no difficulties in our application, however, since the integrands $p(\rho)$ and $\rho p(\rho)$ are given by analytic expressions; see Section 2.4.)

Then, after some simplification, the integrals (B.1) become

$$I(\omega) \approx \delta\, \mathrm{e}^{\mathrm{i}\omega a}\left[W(\theta)S(\theta) + v(\theta) + \mathrm{e}^{\mathrm{i}\omega(b-a)}\overline{\mu(\theta)}\right],$$

where $\theta = \omega\delta$,

$$S(\theta) = \sum_{k=0}^{N} f_k\, \mathrm{e}^{\mathrm{i}\theta k},$$

$$W(\theta) = \int_{-p}^{p} \psi(s) \cos(\theta s) \mathrm{d}s,$$

$$v(\theta) = f_0\gamma_0(\theta) + \sum_{k=1}^{q-1}\left[f_k\gamma_k(\theta) - f_{-k}\overline{\gamma_k(\theta)}\right],$$

$$\mu(\theta) = f_N\gamma_0(\theta) + \sum_{k=1}^{q-1}\left[f_{N-k}\gamma_k(\theta) - f_{N+k}\overline{\gamma_k(\theta)}\right],$$

and

$$\gamma_k(\theta) = \mathrm{e}^{\mathrm{i}\theta k} \int_k^q \psi(s) \mathrm{e}^{\mathrm{i}\theta s}\, \mathrm{d}s.$$

Note that since ψ is defined analytically, the functions $W(\theta)$ and $\gamma_k(\theta)$ can be computed exactly for each choice of ψ.

The only approximation in this method is the high-order interpolation of $f(t)$. As a result, only accurate polynomial approximations of the function $f(t)$ are needed, and no polynomial approximations of the highly oscillatory function $f(t)\mathrm{e}^{\mathrm{i}\omega t}$ need to be produced. As a result, this method evaluates the integrals $I(\omega)$ with an accuracy that is *independent of* ω: for a fixed value of q, and given any $\epsilon > 0$, one can choose N sufficiently large so that the error in the computed values of $I(\omega)$ is less than ϵ, *uniformly in* ω.

As can be easily demonstrated, the convergence rate depends on q in much the same way as with Newton-Cotes integration methods [36], i.e., for q odd, the error decays like $\mathcal{O}(N^{-(q+1)})$, and, for q even, the error decays like $\mathcal{O}(N^{-(q+2)})$. Hence, we choose q to be even, our most common choices being $q = 2$ (fourth-order convergence) or $q = 4$ (sixth-order convergence). The values of $W(\theta)$ and $\gamma_k(\theta)$ corresponding to $q = 2$ and $q = 4$ are found in Sections B.1 and B.2, respectively.

In general, it may be necessary to evaluate $I(\omega)$ for many different values of ω. (In our application, $\omega = \kappa \pm 2\pi|d_\ell|$ with $(d_\ell)_q = \ell_q/(B_q - A_q)$ and where $|\ell_q| < \widetilde{N}_q/2$ for $q = 1, 2, 3$.) It is not difficult to obtain $W(\theta)$, $v(\theta)$, and $\mu(\theta)$ for all the necessary values of ω since these functions involve only a few of the coefficients f_k. A straightforward evaluation of the sum $S(\theta)$, on the other hand, requires $\mathcal{O}(N^2)$ operations.

To reduce this complexity, we first use an FFT to evaluate $S(\theta)$ at $\theta_n = 2\pi n/N_F$ for $n = 0, \ldots, N_F - 1$, where $N_F > N$. In detail, in this first step, we compute

$$S(\theta) = \sum_{k=0}^{N} f_k\, \mathrm{e}^{\mathrm{i}\theta_n k} = \sum_{k=0}^{N_F-1} f_k\, \mathrm{e}^{2\pi \mathrm{i} k n/N_F}$$

by means of an FFT, where we set $f_k = 0$ for $k > N$. Since $S(\theta)$ is periodic in θ with period 2π, we thereby obtain the value of the $S(\theta)$ at $\theta = \theta_n + 2\pi r$, $r \in \mathbb{Z}$. Then, the desired values $S(\theta)$ for $\theta = \omega\delta$ are obtained through interpolation of the FFT values. These values, together with those of $W(\theta)$, $v(\theta)$, and $\mu(\theta)$, give us the needed values $I(\omega)$.

The number of interpolation points N_p determines the order of the interpolation. To avoid instabilities we generally choose $N_p \leqslant 10$. Furthermore, although increasing the value of N_F also increases the accuracy of the interpolated value $S(\theta)$, the actual value of N_F is less important than the "oversampling rate" $\beta = N_F/N$. This is the number of points per wavelength with which the most oscillatory mode in $S(\theta)$ is sampled. We have found that for the function (22) the values $q = 4$, $N = 1024$, $\beta = 128$, $N_p = 10$ as well as $q = 2$, $N = 8192$, $\beta = 128$, $N_p = 10$ give us nearly full double precision accuracy. Either of these methods may outperform the other, however, depending on the problem size: the FFT requires a smaller computing time for the first set of parameters than it does for the second set since $N_F = \beta N$ is smaller for the first set. On the other hand, the interpolation uses less time for the second set than it does for the first set, since the endpoint corrections v and μ are simpler for the second set. Hence, in smaller problems (fewer unknowns), which require less interpolation, we prefer the first set of parameters, and in larger problems, which require more interpolation, we prefer the second set of parameters.

B.1. Second-order interpolating polynomials

For the case of $q = 2$, $\psi(s)$ is taken as a sum of second-order Lagrange interpolating polynomials:

$$\psi_1(s) = \begin{cases} \frac{(s+2)}{2}\frac{(s+1)}{1}, & \text{if } -2 \leqslant s \leqslant 0, \\ 0, & \text{otherwise,} \end{cases}$$

$$\psi_2(s) = \begin{cases} \frac{(s+1)}{1}\frac{(s-1)}{-1}, & \text{if } -1 \leqslant s \leqslant 1, \\ 0, & \text{otherwise,} \end{cases}$$

$$\psi_3(s) = \begin{cases} \frac{(s-1)}{-1}\frac{(s-2)}{-2}, & \text{if } 0 \leqslant s \leqslant 2, \\ 0, & \text{otherwise.} \end{cases}$$

Notice that ψ_1 and ψ_3 are the usual second-order Lagrange interpolation scheme when the point $s = 0$ lies on the boundary of two subintervals. On the other hand, ψ_2 is the usual Lagrange interpolating polynomial when the point $s = 0$ lies at the center of the subinterval. Addition and normalization leads to

$$\psi(s) = \frac{1}{2}[\psi_1(s) + \psi_2(s) + \psi_3(s)].$$

The functions $W(\theta)$ and $\gamma_k(\theta)$ in this case are given by

$$W(\theta) = \frac{4\sin^3(\theta/2)[2\cos(\theta/2) + \theta\sin(\theta/2)]}{\theta^3},$$

$$\gamma_0(\theta) = -\frac{2\mathrm{i} + (3 + 4\mathrm{i}\theta)\theta - 4(\theta + \mathrm{i})\mathrm{e}^{\mathrm{i}\theta} + (\theta + 2\mathrm{i})\mathrm{e}^{2\mathrm{i}\theta}}{4\theta^3},$$

$$\gamma_1(\theta) = -\frac{\mathrm{e}^{\mathrm{i}\theta}[-2\mathrm{i} + \theta + (2 + \mathrm{i}t)\mathrm{e}^{\mathrm{i}\theta}]}{4\theta^3}.$$

It is important to note that for $\theta \ll 1$ the numerical evaluation of these functions can produce a significant amount of cancellation error. To avoid this problem, for sufficiently small θ, we approximate $W(\theta)$ and $\gamma_k(\theta)$ with a power series. Through experiment, we have determined the value of θ at which to switch from one method to the other, while ensuring double precision accuracy. For example, for the function $W(\theta)$ above, we switch to the power series method for $\theta < 10^{-4}$; and for $\gamma_1(\theta)$, we switch for $\theta < 8/10$.

B.2. Fourth-order interpolating polynomials

For $q = 4$, we similarly construct $\psi(s)$ as a sum of fourth-order Lagrange interpolating polynomials:

$$\psi_1(s) = \begin{cases} \frac{(s+4)}{4}\frac{(s+3)}{3}\frac{(s+2)}{2}\frac{(s+1)}{1}, & \text{if } -4 \leqslant s \leqslant 0, \\ 0, & \text{otherwise,} \end{cases}$$

$$\psi_2(s) = \begin{cases} \frac{(s+3)}{3}\frac{(s+2)}{2}\frac{(s+1)}{1}\frac{(s-1)}{-1}, & \text{if } -3 \leqslant s \leqslant 1, \\ 0, & \text{otherwise,} \end{cases}$$

$$\psi_3(s) = \begin{cases} \frac{(s+2)}{2}\frac{(s+1)}{1}\frac{(s-1)}{-1}\frac{(s-2)}{-2}, & \text{if } -2 \leqslant s \leqslant 2, \\ 0, & \text{otherwise,} \end{cases}$$

$$\psi_4(s) = \begin{cases} \frac{(s+1)}{1}\frac{(s-1)}{-1}\frac{(s-2)}{-2}\frac{(s-3)}{-3}, & \text{if } -1 \leqslant s \leqslant 3, \\ 0, & \text{otherwise,} \end{cases}$$

$$\psi_5(s) = \begin{cases} \frac{(s-1)}{-1}\frac{(s-2)}{-2}\frac{(s-3)}{-3}\frac{(s-4)}{-4}, & \text{if } 0 \leqslant s \leqslant 4, \\ 0, & \text{otherwise.} \end{cases}$$

Then, the function $\psi(s)$ is given by the normalized sum of these piecewise polynomials

$$\psi(s) = \frac{1}{4}\sum_{j=0}^{5}\psi_j(s).$$

In this case $W(\theta)$ and $\gamma_k(\theta)$ are given by

$$W(\theta) = \frac{4\sin^5\left(\frac{\theta}{2}\right)}{3\theta^5}\left\{2\theta\left[12-\theta^2+3(6-\theta^2)\cos\theta\right]\sin\left(\frac{\theta}{2}\right)+(12+\theta^2)\cos\left(\frac{\theta}{2}\right)+(12-11\theta^2)\cos\left(\frac{3\theta}{2}\right)\right\},$$

and it is important to note that for $\theta \ll 1$ the numerical evaluation of these functions can produce a significant amount of cancellation error. To avoid this problem, for sufficiently small θ, we approximate $W(\theta)$ and $\gamma_k(\theta)$ with a power series. Through experiment, we have determined the value of θ at which to switch from one method to the other, while ensuring double precision accuracy. For example, for the function $W(\theta)$ above, we switch to the power series method for $\theta < 10^{-4}$; and for $\gamma_1(\theta)$, we switch for $\theta < 8/10$.

$$\begin{aligned}\gamma_0 = {} & \frac{1}{48\theta^5}\left[\left(12\mathrm{i}+30\theta-35\mathrm{i}\theta^2+25\theta^3-48\mathrm{i}\theta^4\right)+\left(-48\mathrm{i}-108\theta+104\mathrm{i}\theta^2+48\theta^3\right)\mathrm{e}^{\mathrm{i}\theta}\right. \\ & +\left(72\mathrm{i}+144\theta-114\mathrm{i}\theta^2-36\theta^3\right)\mathrm{e}^{2\mathrm{i}\theta}+\left(-48\mathrm{i}-84\theta+56\mathrm{i}\theta^2+16\theta^3\right)\mathrm{e}^{3\mathrm{i}\theta} \\ & \left.+\left(12\mathrm{i}+18\theta-11\mathrm{i}\theta^2-3\theta^3\right)\mathrm{e}^{4\mathrm{i}\theta}\right],\end{aligned}$$

$$\begin{aligned}\gamma_1 = {} & \frac{1}{48\theta^5}\left[\left(-36\mathrm{i}-66\theta+33\mathrm{i}\theta^2-29\theta^3\right)\mathrm{e}^{\mathrm{i}\theta}+\left(72\mathrm{i}+144\theta-114\mathrm{i}\theta^2-36\theta^3\right)\mathrm{e}^{2\mathrm{i}\theta}.\right. \\ & \left.+\left(-48\mathrm{i}-84\theta+56\mathrm{i}\theta^2+16\theta^3\right)\mathrm{e}^{3\mathrm{i}\theta}+\left(12\mathrm{i}+18\theta-11\mathrm{i}\theta^2-3\theta^3\right)\mathrm{e}^{4\mathrm{i}\theta}\right],\end{aligned}$$

$$\begin{aligned}\gamma_2 = {} & \frac{1}{48\theta^5}\left[\left(36\mathrm{i}+42\theta+3\mathrm{i}\theta^2+7\theta^3\right)\mathrm{e}^{2\mathrm{i}\theta}+\left(-48\mathrm{i}-84\theta+56\mathrm{i}\theta^2+16\theta^3\right)\mathrm{e}^{3\mathrm{i}\theta}\right. \\ & \left.+\left(12\mathrm{i}+18\theta-11\mathrm{i}\theta^2-3\theta^3\right)\mathrm{e}^{4\mathrm{i}\theta}\right],\end{aligned}$$

$$\gamma_3 = \frac{1}{48\theta^5}\left[\left(-12\mathrm{i}-6\theta-i\theta^2-\theta^3\right)\mathrm{e}^{3\mathrm{i}\theta}+\left(12\mathrm{i}+18\theta-11\mathrm{i}\theta^2-3\theta^3\right)\mathrm{e}^{4\mathrm{i}\theta}\right].$$

References

[1] D. Colton, R. Kress, Inverse Acoustic and Electromagnetic Scattering Theory, second ed., Springer-Verlag, Berlin, Heidelberg, New York, 1998.
[2] A. Kirsch, P. Monk, An analysis of the coupling of finite-element and Nyström methods in acoustic scattering, IMA J. Numer. Anal. 14 (1994) 523–544.
[3] A. Kirsch, P. Monk, Convergence analysis of a coupled finite element and spectral method in acoustic scattering, IMA J. Numer. Anal. 9 (1990) 425–447.
[4] B. Yang, D. Gottlieb, J.S. Hesthaven, Spectral simulations of electromagnetic wave scattering, J. Comput. Phys. 134 (2) (1997) 216–230.
[5] A. Ditkowski, K. Dridi, J.S. Hesthaven, Convergent Cartesian grid methods for Maxwell's equations in complex geometries, J. Comput. Phys. 170 (1) (2001) 39–80.

[6] W. Rachowicz, L. Demkowicz, An hp-adaptive finite element method for electromagnetics. Part1: Data structure and constrained approximation, Comput. Methods Appl. Mech. Engrg. 187 (2000) 307–335.
[7] J.S. Hesthaven, T. Warburton, Nodal high-order methods on unstructured grids. I. Time-domain solution of Maxwell's equations, J. Comput. Phys. 181 (1) (2002) 186–221.
[8] N.N. Bojarski, The k-space formulation of the scattering problem in the time domain, J. Opt. Soc. Amer. 72 (1982) 570–584.
[9] P. Zwamborn, P.V. den Berg, Three dimensional weak form of the conjugate gradient FFT method for solving scattering problems, IEEE Trans. Microwave Theory Techniques 40 (1992) 1757–1766.
[10] X.M. Xu, Q.H. Liu, Fast spectral-domain method for acoustic scattering problems, IEEE Trans. Ultrasonics, Ferroelectrics, Frequency Control 48 (2) (2001) 522–529.
[11] Z.Q. Zhang, Q.H. Liu, Three-dimensional weak-form conjugate- and biconjugate-gradient FFT methods for volume integral equations, Microwave Optical Technol. Lett. 29 (5) (2001) 350–356.
[12] Z.Q. Zhang, Q.H. Liu, X.M. Xu, RCS computation of large inhomogeneous objects using a fast integral equation solver, IEEE Trans. Antennas Propagation 51 (3) (2003) 613–618.
[13] G. Liu, S. Gedney, High-order Nyström solution of the volume EFIE for TM-wave scattering, Microwave Optical Technol. Lett. 25 (1) (2000) 8–11.
[14] S.D. Gedney, C.C. Lu, High-order solution for the electromagnetic scattering by inhomogeneous dielectric bodies, Radio Science 38 (1), article 1015.
[15] S. Gedney, A. Zhu, W.H. Tang, G. Liu, P. Petre, A fast, high-order quadrature sampled pre-corrected fast-Fourier transform for electromagnetic scattering, Microwave Optical Technol. Lett. 36 (5) (2003) 343–349.
[16] J. Strain, Locally corrected multidimensional quadrature rules for singular functions, SIAM J. Sci. Comput. 16 (4) (1995) 992–1017.
[17] J. Strain, 2D votex methods and singular quadrature rules, J. Comput. Phys. 124 (1996) 131–145.
[18] J. Strain, Fast adaptive 2D vortex methods, J. Comput. Phys. 132 (1997) 108–122.
[19] G.M. Vainikko, Fast solvers of the Lippmann–Schwinger equation, in: R.P. Gilbert, J. Kajiwara, Y.S. Xu (Eds.), Direct and inverse problems of mathematical physics (Newark, DE 1997), vol. 5 of International Society for Analysis, Applications and Computation, Kluwer Acad. Publ., Dordrecht, 2000, pp. 423–440.
[20] O.P. Bruno, A. Sei, A fast high-order solver for EM scattering from complex penetrable bodies: TE case, IEEE Trans. Antennas Propagation 48 (12) (2000) 1862–1864.
[21] O.P. Bruno, A. Sei, A fast high-order solver for problems of scattering by heterogeneous bodies, IEEE Trans. Antennas Propagation 51 (11) (2003) 3142–3154.
[22] O.P. Bruno, E.M. Hyde, Higher-order Fourier approximation in scattering by two-dimensional, inhomogeneous media, SIAM J. Numer. Anal., in press.
[23] A. Zygmund, Trigonometric Series, Second ed., Cambridge University Press, London, 1968.
[24] L. Bers, F. John, M. Schechter, Partial Differential Equations, John Wiley, New York, 1964.
[25] G.B. Folland, Introduction to Partial Differential Equations, Second ed., Princeton University Press, Princeton, 1995.
[26] D. Gilbarg, N.S. Trudinger, Elliptic Partial Differential Equations of Second Order, Springer-Verlag, Berlin, Heidelberg, New York, 1977.
[27] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes, second ed.Fortran 77: The Art of Scientific Computing, vol. 1, Cambridge University Press, Cambridge New York, 1992.
[28] K.E. Atkinson, An Introduction to Numerical Analysis, second ed., John Wiley, New York, 1989.
[29] O.P. Bruno, L.A. Kunyansky, A fast, high-order algorithm for the solution of surface scattering problems: Basic implementation, tests and applications, J. Comput. Phys. 169 (2001) 80–110.
[30] A. Greenbaum, Iterative Methods for Solving Linear Systemsvol. 17 of Frontiers in Applied Mathematics, SIAM, Philadelphia, 1997.
[31] M. Frigo, S.G. Johnson, FFTW home page. Available from: <http://www.fftw.org>.
[32] M. Frigo, S.G. Johnson, FFTW: An adaptive software architecture for the FFT, in: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 3, 1998, pp. 1381–1384.
[33] S. Balay, K. Buschelman, W.D. Gropp, D. Kaushik, L.C. McInnes, B.F. Smith, PETSc home page (2001). Available from: <http://www.mcs.anl.gov/petsc>.
[34] S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, PETSc users manual, Tech. Rep. ANL-95/11 - Revision 2.1.0, Argonne National Laboratory, 2001.
[35] S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A.M. Bruaset, H.P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser, Boston, 1997, pp. 163–202.
[36] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, second ed.no. 12 in Texts in Applied Mathematics, Springer-Verlag, Berlin Heidelberg New York, 1993.