A Fast Algorithm for Solving Scalar Wave Scattering Problem by Billions of Particles

A. G. Ramm[1], N. T. Tran[1,*]

Mathematics Department, Kansas State University, Manhattan, KS 66506-2602, USA

Abstract

Scalar wave scattering by many small particles of arbitrary shapes with impedance boundary condition is studied. A fast algorithm for solving this wave scattering problem by billions of particles is presented. A recipe for creating materials with a desired refraction coefficient is given. An example of creating a material with negative refraction coefficient is described.

Keywords: wave scattering, small impedance particles, fast algorithm, billions of particles, negative refraction, meta-materials.
2000 MSC: 35J05, 35J25, 70-08, 68W10, 74Q15.

1. Introduction

Wave scattering is a natural phenomenon that happens in everyday life, for example, light scattering in the atmosphere, light scattering by cosmic dust and by the dust in atmosphere, sound wave scattering by packs of fish in the ocean, etc. Studying wave scattering is a subject that has attracted much attention from scientists and engineers since it has many practical applications, for example, in medical image processing, geophysical prospecting, quantum theory, materials science. By studying wave scattering one can understand the interaction of waves with matter, the scattering of waves by the materials. The wave scattering theory gives insights into the structure of the materials. Wave scattering by small particles was studied by Lord Rayleigh, see [10], who understood that the main term in the scattered field is the dipole radiation. For particles of an arbitrary shape he did not give formulas for calculating the induced dipole moment with

[*] Corresponding author.
[1] Email addresses: ramm@math.ksu.edu (A. G. Ramm), nhantran@math.ksu.edu (N. T. Tran)

a desired accuracy for bodies of arbitrary shapes. This was done in [11], [12]. Scalar wave scattering by small impedance particles has practically important physical features: the field scattered by such particle is $O(a^{2-\kappa})$, as $a \to 0$, which is much larger than the field in Rayleigh scattering which is $O(a^3)$. Here it is assumed that the boundary impedance of a particle is $\zeta = ha^{-\kappa}$, where h and $\kappa \in [0,1)$ are constants. The theory of wave scattering by many small impedance particles of an arbitrary shape has been developed in [13] and [14], and is a basis for the computational results in this paper. In [13] and [14] the developed theory has been applied to materials science: it was proved that by distributing small particles with prescribed boundary impedances in a given bounded domain, one can create materials with any desired refraction coefficient, in particular, with negative refraction coefficient, which is of interest for the theory of metamaterials, see [4].

Earlier numerical results on wave scattering by one million particles and below, based on the above theory, were reported in [16]. In the current paper, a completely new algorithm is presented for solving wave scattering problem by billions of particles. The numerical solution of the wave scattering problem with so many small particles is obtained, apparently, for the first time.

In Section 2 the theory, on which the computational results are based, is outlined. In Section 3 the new algorithm for solving wave scattering problem with many small impedance particles of arbitrary shapes is described. The new algorithm is based on three-dimensional (3D) convolution, Fast Fourier Transforms (FFT), and Conjugate Gradient method (CG). It exploits the structure of the Green's function of the Helmholtz equation in the wave scattering problem and drastically reduces the total number of operations required for solving this problem. Numerical examples are presented in Section 4 to illustrate the practical usage of the algorithm. In these numerical examples the algorithm is implemented in parallel and the scalar wave scattering problem is solved with one, four, seven, and ten billions of particles using Gordon super computer at the Extreme Science and Engineering Discovery Environment (XSEDE).

2. Scalar wave scattering by many small impedance particles

Consider a bounded domain $\Omega \subset \mathbb{R}^3$ filled with a material whose refraction coefficient is $n_0(x)$. The assumptions on this coefficient are formulated below (2.5). Suppose there are M small particles D_m distributed in Ω so that the minimal distance between neighboring particles, d, is much greater than the maximal radius of the particles, $a = \frac{1}{2}\max_{1\le m\le M} \text{diam} D_m$, and much less than the wave length, λ, $a \ll d \ll \lambda$. Let D be the union of D_m, $D := \bigcup_{m=1}^{M} D_m$, $D \subset \Omega$, and $D' := \mathbb{R}^3 \setminus D$ is the exterior domain. Suppose that the boundary impedance of the m$^{\text{th}}$ particle is ζ_m, $\zeta_m = \frac{h(x_m)}{a^\kappa}$, where $h(x)$ is a given continuous function in D such that Im $h \le 0$ in D, and x_m is a point inside D_m. This point gives the position of the m$^{\text{th}}$ particle in $\mathbb{R}^3$. Let κ be a given constant, $\kappa \in [0, 1)$. The scattering problem is formulated as follows:

$$(\nabla^2 + k^2 n_0^2(x))u = 0 \quad \text{in } D', \quad k = \text{const} > 0, \quad ka \ll 1 \tag{2.1}$$

$$u_N = \zeta_m u \quad \text{on } S_m := \partial D_m, \quad \text{Im}\, \zeta_m \le 0, \quad 1 \le m \le M, \tag{2.2}$$

$$u(x) = u_0(x) + v(x), \tag{2.3}$$

$$u_0(x) = e^{ik\alpha\cdot x}, \quad |\alpha| = 1, \tag{2.4}$$

$$v_r - ikv = o(1/r), \quad r := |x| \to \infty. \tag{2.5}$$

Here k is the wave number, $k = 2\pi/\lambda$, u_0 is the incident plane wave, v is the scattered wave, α is the direction of the incident wave, $\vec{N}$ is the outer unit normal to S_m, the refraction coefficient $n_0(x) = 1$ in $\Omega' := \mathbb{R}^3 \setminus \Omega$. It is assumed that $n_0(x)$ is a Riemann-integrable function and that Im $n_0^2(x) \ge 0$ in Ω. Equation (2.5) is called the radiation condition. It was proved in Ramm [14] that if Im $n_0^2(x) \ge 0$ and Im $h(x) \le 0$, then the scattering problem (2.1)-(2.5) has a unique solution of the form

$$u(x) = u_0(x) + \sum_{m=1}^{M} \int_{S_m} G(x,t)\sigma_m(t)dt, \tag{2.6}$$

where $G(x, y)$ is the Green's function of the Helmholtz equation (2.1), G satisfies the equation

$$[\nabla^2 + k^2 n_0^2(x)]G = -\delta(x - y) \quad \text{in } \mathbb{R}^3 \tag{2.7}$$

and the radiation condition (2.5). The functions $\sigma_m(t)$ are unknown continuous functions. These functions are uniquely defined by the boundary condition (2.2), see [13]. If $n_0^2 = 1$ in $\mathbb{R}^3$, then

$$G(x, y) = \frac{e^{ik|x-y|}}{4\pi|x-y|}, \tag{2.8}$$

The assumption $n_0^2 = 1$ in $\mathbb{R}^3$ is not a restriction in the problem of creating materials with a desired refraction coefficient. In the general case, when n_0^2 is a function of x, the Green's function G has to be computed.

From (2.6), one gets

$$u(x) = u_0(x) + \sum_{m=1}^{M} G(x, x_m)Q_m + \sum_{m=1}^{M} \int_{S_m} [G(x,t) - G(x, x_m)]\sigma_m(t)dt, \qquad (2.9)$$

where

$$Q_m := \int_{S_m} \sigma_m(t)dt. \qquad (2.10)$$

It is proved in [14] that in (2.9)

$$|G(x, x_m)Q_m| \gg \left| \int_{S_m} [G(x,t) - G(x, x_m)]\sigma_m(y)dt \right|, \qquad (2.11)$$

as $a \to 0$ and $|x - x_m| \geq a$. Therefore, the solution to the scattering problem can be well approximated by the sum

$$u(x) \sim u_0(x) + \sum_{m=1}^{M} G(x, x_m)Q_m. \qquad (2.12)$$

Thus, instead of finding the unknown functions $\sigma_m(t)$, as is usually done when one solves a wave scattering problem, we just need to find the unknown numbers Q_m to get the accurate approximation of the solution. This makes it possible to solve problems with so large number of particles that it was not possible to do earlier.

To find the numbers Q_m, let us define the effective field $u_e(x)$. The effective field acting on the j$^{\text{th}}$ particle is defined as follows

$$u_e(x_j) := u(x) - \int_{S_j} G(x_j, t)\sigma_j(t)dt, \qquad (2.13)$$

or equivalently

$$u_e(x_j) = u_0(x_j) + \sum_{m=1, m\neq j}^{M} \int_{S_m} G(x_j, t)\sigma_m(t)dt, \qquad (2.14)$$

where x_j is a point in D_j. The asymptotic formula for Q_m is derived in [14]:

$$Q_m = -c_S a^{2-\kappa} h(x_m) u_e(x_m)[1 + o(1)], \quad a \to 0, \qquad (2.15)$$

where $c_S > 0$ is a constant depending on the shape of the particle,

$$|S_m| = c_S a^2, \tag{2.16}$$

where $|S_m|$ is the surface area of S_m. If S_m is a sphere of radius a, then $c_S = 4\pi$. We assume for simplicity that c_S does not depend on m, that is, all the particles are of the same shape.

Let us derive a formula for the effective field. From (2.14)-(2.15) one gets

$$u_e(x_j) \simeq u_0(x_j) - c_S \sum_{m=1, m\neq j}^{M} G(x_j, x_m) h(x_m) u_e(x_m) a^{2-\kappa}, \tag{2.17}$$

as $a \to 0$ and $1 \le j \le M$.

Denote $u_j := u_e(x_j), u_{0j} := u_0(x_j), G_{jm} := G(x_j, x_m)$, and $h_m := h(x_m)$. Then (2.17) can be rewritten as a linear algebraic system for the unknown numbers u_m:

$$u_j = u_{0j} - c_S \sum_{m=1, m\neq j}^{M} G_{jm} h_m a^{2-\kappa} u_m, \quad \text{as } a \to 0, \quad 1 \le j \le M. \tag{2.18}$$

In (2.18), the numbers u_j, $1 \le j \le M$, are unknowns. We call (2.18) the *original linear algebraic system* (ori). It was proved in Ramm [14] that under the assumptions

$$d = O\left(a^{\frac{2-\kappa}{3}}\right), \quad \text{and } M = O\left(\frac{1}{a^{2-\kappa}}\right), \quad \text{for } \kappa \in [0,1), \tag{2.19}$$

the numbers u_j, $1 \le j \le M$, can be uniquely found by solving (ori) for all sufficiently small a. If the numbers u_m are known, then the numbers Q_m can be calculated by formula (2.15) and the approximate solution to the wave scattering problem (2.1)-(2.5) can be computed by (2.12). *This solution is asymptotically exact as $a \to 0$.*

The method for solving many-body wave scattering problem, described above, differs in principle from the Fast Multipole Method (FMM), used in many papers, of which we mention just two: [6], [7]. The difference between FMM and our method briefly can be explained as follows: our method is developed for scattering by small impedance particles of arbitrary shapes and is based on the asymptotically exact formula for the field, scattered by one small particle, and on the assumption $d \gg a$; we derive an integral equation for the limiting field in the medium consisting of many small particles as $a \to 0$; we give a recipe for creating materials with a desired refraction coefficient. We do not use multipole expansions.

Next, let us derive the reduced order linear system for solving the wave scattering problem. Let Δ be an arbitrary subdomain of Ω. Assume that the distribution of particles in Δ satisfies this law

$$\mathcal{N}(\Delta) = \frac{1}{a^{2-\kappa}} \int_{\Delta} N(x)dx[1+o(1)], \quad \text{as } a \to 0. \tag{2.20}$$

Here $N(x) \geq 0$ is a given continuous function in Ω. The function $N(x)$ and the number $\kappa \in [0,1)$ can be chosen by the experimenter as he(she) desired. The number $\mathcal{N}(\Delta)$ is the total number of the embedded particles in Δ.

Let Ω be partitioned into P non-intersecting sub-cubes Δ_p's of side b such that $b \gg d \gg a$, where $b = b(a)$, $d = d(a)$, and $\lim_{a\to 0} \frac{d(a)}{b(a)} = 0$. Here $P \ll M$, and each sub-cube contains many particles. If the function $N(x)$ in (2.20) is continuous and $b \ll 1$, then

$$\mathcal{N}(\Delta_p) a^{2-\kappa} = N(x_p)|\Delta_p|[1+o(1)] = \sum_{x_m \in \Delta_p} 1, \quad \text{as } a \to 0, \tag{2.21}$$

where $|\Delta_p|$ is the volume of Δ_p and $x_p \in \Delta_p$ is an arbitrary point, for example, the center of Δ_p. Thus, (2.18) can be rewritten as

$$u_q = u_{0q} - c_S \sum_{p=1, p\neq q}^{P} G_{qp} h_p N_p u_p |\Delta_p|, \quad \text{for } 1 \leq q \leq P, \tag{2.22}$$

where $N_p := N(x_p)$ and x_p is a point in Δ_p, for example, the center of Δ_p. We call (2.22) the *reduced linear algebraic system* (red). This system is much easier to solve since $P \ll M$.

Let $|\Delta_p| \to 0$. Then it follows from (2.22) that the limiting integral equation for $u = u(x)$ holds

$$u(x) = u_0(x) - c_S \int_D G(x,y)h(y)N(y)u(y)dy, \quad \text{for } x \in \mathbb{R}^3, \tag{2.23}$$

if the assumption (2.20) is satisfied. The sum in (2.22) is the Riemannian sum for the integral in (2.23) which converges to this integral when $\max_p |\Delta_p| \to 0$ (see [14] for the proof of convergence).

Let

$$p(x) := c_S N(x) h(x). \tag{2.24}$$

Then (2.23) can be written as

$$u(x) = u_0(x) - \int_D G(x,y) p(y) u(y) dy, \quad \text{for } x \in \mathbb{R}^3. \tag{2.25}$$

Here $u = u(x)$ is the limiting field in the medium created by embedding many small impedance particles distributed according to equation (2.20). We call (2.25) the *limiting integral equation* (ie).

Now, applying the operator $(\nabla^2 + k^2 n_0^2)$ to (2.25) and using the equation $(\nabla^2 + k^2 n_0^2) G(x,y) = -\delta(x-y)$, one gets

$$(\nabla^2 + k^2 n_0^2) u(x) = p(x) u(x). \tag{2.26}$$

This implies

$$(\nabla^2 + k^2 n^2) u = 0, \tag{2.27}$$

where

$$n^2(x) := n_0^2(x) - k^{-2} p(x), \tag{2.28}$$

and $n(x)$ *is the new refraction coefficient of the limiting medium.* Since $\text{Im}h(x) \le 0$ and $\text{Im}n_0^2(x) \ge 0$, $\text{Im}n^2(x) \ge 0$. From (2.28), one gets

$$p(x) = k^2 [n_0^2(x) - n^2(x)]. \tag{2.29}$$

By equation (2.24), one has $p(x) = c_S N(x) h(x)$, and $h(x)$ can be computed as

$$h(x) = \frac{p(x)}{c_S N(x)}. \tag{2.30}$$

This gives a method for creating new materials with a desired refraction coefficient $n(x)$ by embedding many small impedance particles into a given material with the original refraction coefficient n_0 using the distribution law (2.20).

3. A fast algorithm for solving wave scattering problem by billions of particles

Our goal is to develop a fast algorithm for solving (ori) in order to get the solution of the scattering problem (2.1)-(2.5). Our algorithm is a combination of the Conjugate Gradient (CG) method, 3D convolution, and Fast Fourier Transform

(FFT). When one solves a linear algebraic system using iterations, matrix-vector multiplications are carried out in the iterative process. These multiplications take most of the computation time. If the linear system is very large, it takes a huge amount of time to finish only one matrix-vector multiplication in a standard way, since this multiplication is of the order $O(n^2)$. In some cases it is practically impossible to perform such computations, for example, when the system is dense and has more than one billion equations and unknowns. In this section we present an algorithm that greatly reduces the total number of operations (from $O(n^2)$ to $O(n\log n)$) and decreases the overall computation time of the iterative process by handling the matrix-vector multiplication by using 3D convolution and FFT.

There are numerous methods which also employ FFT to solve different problems, for example, Precorrected-FFT method for electrostatic analysis of complicated 3D structures [9], or Particle mesh Ewald method for Ewald sums in large systems [3], etc. Nevertheless, none of them deal with the scale that we face when solving the scalar wave scattering problem with ten billion particles, i.e. we have to work on a $10^{10} \times 10^{10}$ linear system. In order to reach our goal at such a large scale, we have to develop a new algorithm that can solve two major problems in our computing: memory and time. First, it is impossible to store a $10^{10} \times 10^{10}$ dense matrix in any current state of the art super computer in the world. Suppose we use only single precision, it would take 800 million terabytes of memory to store only one matrix to do the computation, since each complex number is 8 bytes. Furthermore, we will painfully suffer network latency and traffic jam which cause a halt in our computation if we use such an amount of memory in any parallel cluster. Second, it is impossible to do the computation at order $O(n^2)$ for a $10^{10} \times 10^{10}$ linear system in a reasonable and permitted time. We deal with the first computing problem, memory, by finding a way to store the $n \times n$ matrix in a 3D cube which is equivalent to only one $n \times 1$ vector in size. The second computing problem, time, is resolved by reducing the number of operations from $O(n^2)$ to $O(n\log n)$. The details on how to attack these two major computing problems at our scale and how to set up the wave scattering problem in order to solve it in parallel clusters are described in this section.

3.1. Conjugate Gradient method (CG)

The Conjugate Gradient is one of the widely used methods for solving large symmetric positive definite linear systems, see [8]. It is also applicable for solving symmetric systems with complex coefficients, see [17] and [2].

CG is a projection method for solving a linear algebraic system

$$Ax = b$$

by projecting the residual of the solution onto the m$^{\text{th}}$ Krylov subspace

$$K_m(A, r_0) = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}, \tag{3.1}$$

where r_0 is the initial residual, $r_0 = b - Ax_0$, and x_0 is the initial guess of the solution.

Suppose A is an $n \times n$ matrix. Define its characteristic polynomial:

$$\det(\lambda I - A) = h_0 + h_1\lambda + h_2\lambda^2 + \dots + h_n\lambda^n. \tag{3.2}$$

By the Cayley-Hamilton's theorem one has

$$h_0 I + h_1 A + h_2 A^2 + \dots + h_n A^n = 0. \tag{3.3}$$

Multiply both sides of (3.3) by A^{-1}:

$$h_0 A^{-1} + h_1 + h_2 A + \dots + h_n A^{n-1} = 0. \tag{3.4}$$

The matrix A^{-1} has the form:

$$A^{-1} = -h_0^{-1}(h_1 I + h_2 A + h_3 A^2 + \dots + h_n A^{n-1}). \tag{3.5}$$

The solution x of the linear system $Ax = b$ can be written as

$$x = A^{-1}b = -h_0^{-1}(h_1 I + h_2 A + h_3 A^2 + \dots + h_n A^{n-1})b. \tag{3.6}$$

Formula (3.6) shows that the elements of the Krylov subspace can approximate the solution of the linear system.

Let us recall the idea of the standard CG method. Two non-zero vectors u and v are called conjugate with respect to an $n \times n$ matrix A if

$$u^T Av = (u, Av) = 0. \tag{3.7}$$

If A is symmetric and positive definite then $u^T Av$ defines an inner product:

$$(u, v)_A := (Au, v) = (u, A^T v) = (u, Av) = u^T Av. \tag{3.8}$$

Define P as a basis of $\mathbb{R}^n$ orthogonal in the sense of inner product (Au, v) but not normalized with respect to this product:

$$P := \{p_i : (p_i, p_k)_A = 0 \text{ if } i \neq k, 1 \leq i, k \leq n\}. \tag{3.9}$$

The solution x of the linear system $Ax = b$ can be written as

$$x = \sum_{i=1}^{n} \alpha_i p_i. \tag{3.10}$$

Let us find α_i, $1 \leq i \leq n$. The right-hand-side b can be computed as follows

$$b = Ax = \sum_{i=1}^{n} \alpha_i A p_i. \tag{3.11}$$

For any $p_k \in P$,

$$p_k^T b = \sum_{i=1}^{n} \alpha_i p_k^T A p_i = \sum_{i=1}^{n} \alpha_i (p_k, p_i)_A = \alpha_k p_k^T A p_k. \tag{3.12}$$

Thus, α_k can be derived as

$$\alpha_k = \frac{p_k^T b}{p_k^T A p_k} = \frac{(p_k, b)}{||p_k||_A^2}. \tag{3.13}$$

CG method uses Gram-Schmidt procedure to form an orthogonal basis of the Krylov subspace. In practice CG method is often implemented as an iterative algorithm. The CG method converges to a solution within n iterations. When the matrix is complex and symmetric the corresponding method is Conjugate Orthogonal Conjugate Gradient (COCG), which coincides with the standard CG algorithm for real-valued systems, see [2]. For COCG, the inner product is replaced by the bilinear form

$$[x, y] := \sum_{i=1}^{n} x_i y_i = (\bar{x}, y), \tag{3.14}$$

where $(\cdot, \cdot)$ stands for the standard inner product, see [17], and so it coincides with the real-valued inner product. The corresponding condition to (3.7) for complex vectors u and v is

$$u^T Av = (\bar{u}, Av) = 0, \tag{3.15}$$

which is called conjugate orthogonality. The convergence behavior of COCG method is similar to that of BiConjugate Gradients method, see [17].

The usage of CG method is not strictly required in our paper. For example, one can use GMRES method, see [15], which is designed for solving large general linear systems whose matrices are not necessarily positive definite. We choose CG over GMRES because CG does not require to store the basis of the Krylov subspaces. Consequently, much less memory space is used.

3.2. Discrete convolution theorem

The convolution of discrete and periodic signals $x[\mathbf{m}]$ and $y[\mathbf{m}]$, where $\mathbf{m} = (m_1, m_2, \ldots, m_n) \in l^n$, is defined as

$$(x * y)[\mathbf{m}] = \sum_{\mathbf{k=0}}^{\mathbf{N\text{-}1}} x[\mathbf{k}] y[\mathbf{m} - \mathbf{k}]. \tag{3.16}$$

Here $\mathbf{0} := (0, 0, \ldots, 0)$ and $\mathbf{N\text{-}1} := (N-1, N-1, \ldots, N-1)$ in l^n, $l^n := \{(m_1, m_2, \ldots, m_n) : m_i \in \mathbb{N}, 0 \le m_i \le N-1, 1 \le i \le n\}$.

The convolution theorem states:

$$\mathscr{F}(x * y) = \mathscr{F}(x) \cdot \mathscr{F}(y). \tag{3.17}$$

Its proof is standard:

$$\begin{align}
\mathscr{F}(x * y)[\mathbf{m}] &= \sum_{\mathbf{k=0}}^{\mathbf{N\text{-}1}} (x * y)[\mathbf{k}] e^{-i\frac{2\pi}{N}\mathbf{m}\cdot\mathbf{k}} \tag{3.18}\\
&= \sum_{\mathbf{k=0}}^{\mathbf{N\text{-}1}} \sum_{\mathbf{s=0}}^{\mathbf{N\text{-}1}} x[\mathbf{s}] y[\mathbf{k} - \mathbf{s}] e^{-i\frac{2\pi}{N}\mathbf{m}\cdot\mathbf{k}} \tag{3.19}\\
&= \sum_{\mathbf{s=0}}^{\mathbf{N\text{-}1}} x[\mathbf{s}] e^{-i\frac{2\pi}{N}\mathbf{m}\cdot\mathbf{s}} \sum_{\mathbf{k=0}}^{\mathbf{N\text{-}1}} y[\mathbf{k} - \mathbf{s}] e^{-i\frac{2\pi}{N}\mathbf{m}\cdot(\mathbf{k}-\mathbf{s})} \tag{3.20}\\
&= \sum_{\mathbf{s=0}}^{\mathbf{N\text{-}1}} x[\mathbf{s}] e^{-i\frac{2\pi}{N}\mathbf{m}\cdot\mathbf{s}} \sum_{\mathbf{l=0}}^{\mathbf{N\text{-}1}} y[\mathbf{l}] e^{-i\frac{2\pi}{N}\mathbf{m}\cdot\mathbf{l}} \tag{3.21}\\
&= \mathscr{F}(x) \cdot \mathscr{F}(y). \tag{3.22}
\end{align}$$

3.3. 3D convolutions and Fourier transforms

Consider the following summation in the original linear system (2.18) of the wave scattering problem

$$\sum_{m=1,m\neq j}^{M} G(\mathbf{x_j},\mathbf{x_m})\,u(\mathbf{x_m}), \tag{3.23}$$

where by bold letters vectors are denoted. Since $h(\mathbf{x}) = \frac{p(\mathbf{x})}{c_S N(\mathbf{x})}$ where $p(\mathbf{x}) = k^2[n_0^2(\mathbf{x}) - n^2(\mathbf{x})]$, $h(\mathbf{x})$ is a constant function if $n_0(\mathbf{x})$, $n(\mathbf{x})$, and $N(\mathbf{x})$ are constant functions. $N(\mathbf{x}) = const$ means the distribution of particles is uniform, which we will assume in this problem for simplicity. Therefore, we can view $h(\mathbf{x})$ as a constant function in the summation (3.23). Then G in equation (3.23) is the Green function of the form

$$G(\mathbf{x_j},\mathbf{x_m}) = \frac{e^{ik|\mathbf{x_j}-\mathbf{x_m}|}}{4\pi|\mathbf{x_j}-\mathbf{x_m}|}, \tag{3.24}$$

where $\mathbf{x_j},\mathbf{x_m}$ are the positions of the j$^{\text{th}}$ and m$^{\text{th}}$ particles in $\mathbb{R}^3$, respectively. If we write $G(\mathbf{x}-\mathbf{y}) := G(\mathbf{x},\mathbf{y})$, the summation in (3.23) will be

$$\sum_{m=1,m\neq j}^{M} G(\mathbf{x_j}-\mathbf{x_m})\,u(\mathbf{x_m}), \tag{3.25}$$

which is a discrete convolution of G and u, $G * u$.

In the linear system (2.18), G is an $M \times M$ matrix, where M is the total number of particles, and u is an $M \times 1$ vector. When solving the linear system (2.18) using CG iterative algorithm, the matrix-vector multiplication in (3.25) need to be executed. If we do this matrix-vector multiplication in the standard way, it would take $O(M^2)$ operations. This is very expensive in terms of computation time if M is very large, for example, $M \geq 10^9$. Therefore, we have to find a new way to do the matrix-vector multiplication.

Let $\mathbf{m} = (m_x, m_y, m_z)$ be the position of the m$^{\text{th}}$ particle in $\mathbb{R}^3$, where m_x, m_y, and m_z are real numbers. If all the particles are distributed uniformly in a unit cube, this cube is placed in the first octant and the origin is one of the cube's vertices, then $\mathbf{m}$ can be rewritten as a product of the scalar factor $d > 0$ and a vector (m_1, m_2, m_3):

$$\mathbf{m} = d(m_1, m_2, m_3), \tag{3.26}$$

where d is the distance between neighboring particles, a scalar, (m_1, m_2, m_3) is a vector whose components m_1, m_2, and m_3 are integers in $[0, b)$, and $b = M^{1/3}$ is the number of particles on a side of the cube.

In the convolution (3.25) suppose that $\mathbf{x_j} = d(j_1, j_2, j_3)$ and $\mathbf{x_m} = d(m_1, m_2, m_3)$ and write (3.25) as

$$G * u = \sum_{m=1, m\neq j}^{M} G(\mathbf{x_j} - \mathbf{x_m})u(\mathbf{x_m}) \tag{3.27}$$

$$= \sum_{\substack{m_1, m_2, m_3 = 0 \\ (m_1, m_2, m_3) \neq (j_1, j_2, j_3)}}^{b-1} G(j_1 - m_1, j_2 - m_2, j_3 - m_3)u(m_1, m_2, m_3). \tag{3.28}$$

This is a 3D convolution of G and u.

This convolution can be carried out by using Convolution theorem as follows:

$$G * u = \mathscr{F}^{-1}(\mathscr{F}(G * u)) = \mathscr{F}^{-1}(\mathscr{F}(G) \cdot \mathscr{F}(u)), \tag{3.29}$$

where the $\cdot$ stands for the component-wise multiplication of two vectors and its result is a vector.

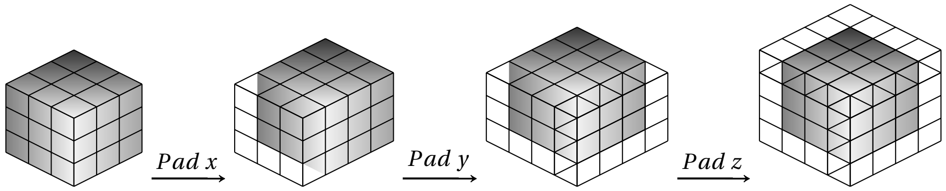
In order to do this convolution, we need to store matrix G as a vector. This reduces drastically the amount of memory for storing the original $M \times M$ matrix to a much smaller amount for storing an $M \times 1$ vector, which is denoted G. Since this vector depends on three components, $G = G(j_1 - m_1, j_2 - m_2, j_3 - m_3)$ where $j_1, m_1, j_2, m_2, j_3, m_3$ are integers in $[0, b)$ and $(j_1, j_2, j_3) \neq (m_1, m_2, m_3)$, we can alternatively store it as a cube of size $b \times b \times b$. Similarly, vector $u(m_1, m_2, m_3)$ is also stored as a cube of size $b \times b \times b$.

When implementing the 3D convolution we need to pad the cubes G and u as follows:

- Pad in x-direction

- Pad in y-direction

- Pad in z-direction



For each direction, the padding is illustrated by this example
For G:

$$\boxed{\begin{matrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{matrix}} \xrightarrow{Pad} \boxed{\begin{matrix} 1 & 2 & 3 & 4 & 3 & 2 \\ 5 & 6 & 7 & 8 & 7 & 6 \\ 9 & 10 & 11 & 12 & 11 & 10 \\ 13 & 14 & 15 & 16 & 15 & 14 \end{matrix}}$$

For u:

$$\boxed{\begin{matrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{matrix}} \xrightarrow{Pad} \boxed{\begin{matrix} 1 & 2 & 3 & 4 & 0 & 0 \\ 5 & 6 & 7 & 8 & 0 & 0 \\ 9 & 10 & 11 & 12 & 0 & 0 \\ 13 & 14 & 15 & 16 & 0 & 0 \end{matrix}}$$

This means that we pad G using its entries and pad u with zeros. As described in the example above, for padding G we copy all columns except the first and the last ones and put them symmetrically through the last column. This will create a periodic signal G. In fact, if one places padded G continuously, one can see a periodic signal. Since we only need to perform linear convolution on M-length vectors, the result we need is an M-length vector. All the entries after the M-th entry in the convolution result will be discarded. So, we pad u with zeros just to have the same length with the padded G to do the cyclic convolution in computer. Cyclic convolutions allow us to compute linear convolutions by means of Discrete Fourier Transforms (DFT). After padding G and u will have size $(2b-2)^3$.

The Fourier transform and inverse Fourier transform are of order $O(n \log n)$, and vector pointwise multiplication is of order $O(n)$ if the vectors are $n \times 1$. In our case the total number of operations for computing $G * u = \mathscr{F}^{-1}(\mathscr{F}(G) \cdot \mathscr{F}(u))$ is

$$n \log n + n \log n + n + n \log n = O(n \log n), \quad n = (2b-2)^3, \tag{3.30}$$

since the Fourier transforms $\mathscr{F}(G)$ and $\mathscr{F}(u)$ are of order $O(n \log n)$, the vector point-wise multiplication $\mathscr{F}(G) \cdot \mathscr{F}(u)$ is of order $O(n)$, and the inverse Fourier transform $\mathscr{F}^{-1}(\mathscr{F}(G)\cdot\mathscr{F}(u))$ is of order $O(n\log n)$. If we compare this with the standard matrix-vector multiplication which takes M^2 operations ($M = b^3$), this is a huge reduction of the number of operations and computation time, when M is very large, say $M \geq 10^9$.

This algorithm is applicable not only to solving scalar wave scattering problems but also to other PDE problems.

4. Numerical examples

The algorithm described in Section 3 is implemented in parallel using PETSC library developed at Argonne National Laboratory (ANL), see [1]. For implementing FFT, Fastest Fourier Transform in the West (FFTW) library is used, see [5]. The wave scattering problem is solved using Gordon super computer at XSEDE. "Gordon is a dedicated XSEDE cluster designed by Appro and SDSC consisting of 1024 compute nodes and 64 I/O nodes. Each compute node contains two 8-core 2.6 GHz Intel EM64T Xeon E5 (Sandy Bridge) processors and 64 GB of DDR3-1333 memory", see [18]. Table 1 shows the technical information of one compute node in Gordon.

Table 1: Compute node Intel EM64T Xeon E5

System Component	Configuration
Sockets	2
Cores	16
Clock speed	2.6 GHz
Flop speed	333 Gflop/s
Memory capacity	64 GB DDR3-1333
Memory bandwidth	85 GB/s
STREAM Triad bandwidth	60 GB/s

"The network topology of Gordon is a 4x4x4 3D torus with adjacent switches connected by three 4x QDR InfiniBand links (120 Gbit/s). Compute nodes (16 per switch) and I/O nodes (1 per switch) are connected to the switches by 4x QDR (40 Gbit/s). The theoretical peak performance of Gordon is 341 TFlop/s", see [18]. Table 2 shows information about the network of Gordon.

The program code is written in C & C++, compiled with Intel compiler, and linked with MPI library MVAPICH2. The code uses 64-bit integers and single precision. The relative error used for the convergence of CG iterations is 10^{-3}.

Table 2: Network summary

QDR InfiniBand Interconnect	
Topology	3D Torus
Link bandwidth	8 GB/s (bidirectional)
MPI latency	1.3 μs

We assume that the domain Ω, which contains all the particles, is a unit cube, placed in the first octant such that the origin is one of its vertices, and particles are distributed uniformly in Ω. Suppose we want to create a new meta-material with the refraction coefficient $n(x) = -1$ in Ω given a material with the refraction coefficient $n_0(x) = 1$ by embedding many small particles into the given material. We assume the particles are spheres, so $c_S = 4\pi$. The new refraction coefficient is computed by the following formula

$$n(x) = [n_0^2(x) - k^{-2} c_S h(x) N(x)]^{1/2}, \tag{4.1}$$

where $N(x)$ and $h(x)$ are at our choices. For simplicity we choose $N(x) = 1$. The choice of $h(x)$ is subject to the physical condition $\text{Im}h \le 0$. If $\text{Im}h(x) \le 0$ and $\text{Im}n_0^2 \ge 0$, then $\text{Im}n^2(x) \ge 0$. The square root in formula (4.1) is of the form

$$z^{1/2} = |z|^{1/2} e^{i\frac{\phi}{2}}, \quad \phi := \arg z, \quad \phi \in [0, 2\pi]. \tag{4.2}$$

Formula (4.2) defines a one-valued branch of analytic function $z^{1/2}$ in the complex plane with the cut $[0, +\infty)$. If one wants to get $n = Be^{i(\pi-\epsilon)}$, where $B > 0$ and $\epsilon > 0$, then $n^2 = B^2 e^{i(2\pi - 2\epsilon)}$. When $\epsilon > 0$ is very small, one gets practically negative refraction coefficient n. In this experiment, we choose $\text{Im}n = 0.001$. This violates the assumption $\text{Im}h(x) \le 0$. To justify this violation for very small values of $\text{Im}h(x)$ we argue as follows. The integral equation (2.23) is an equation with compact integral operator T

$$Tu := c_S \int_D G(x, y) h(y) N(y) u(y) dy \tag{4.3}$$

It is of Fredholm type with index zero. It is proved in [14] that equation (2.23) has at most one solution for $\text{Im}h \le 0$. Therefore, the inverse operator $(I + T)^{-1}$ is bounded for $\text{Im}h \le 0$. The set of boundedly invertible operators is open. Therefore the inverse operator $(I + T)^{-1}$ exists and is bounded also for sufficiently small Im$h \geq 0$.

The radius a of the particles and the distance d between neighboring particles are chosen so that

$$d = \frac{1}{M^{1/3}} = a^{\frac{2-\kappa}{3}}, \quad \text{and} \quad M = \frac{1}{a^{2-\kappa}}, \tag{4.4}$$

where M is the total number of particles embedded in the domain Ω. To solve (ie), we use a collocation method, dividing the domain into many sub-cubes, taking the collocation points as the centers of these cubes, and then approximating the integral equation by the corresponding Riemannian sum.

The following physical parameters are used to conduct the experiment:

- Speed of wave, $v = 34400$ cm,
- Frequency, $f = 1000$ Hz,
- Wave number, $k = 0.182651$ cm^{-1},
- Direction of plane wave, $\alpha = (1,0,0)$,
- The constant $\kappa = 0.5$,
- Volume of the domain that contains all particles, $|\Omega| = 1$ cm^3,
- Distribution of particles, $N = Ma^{2-\kappa}/|\Omega| = 1$, i.e. particles are distributed uniformly in the unit cube,
- Function $h(x) = 2.65481\text{E-}09 + i5.30961\text{E-}06$,
- Original refraction coefficient, n_0 = 1+i0,
- Desired refraction coefficient, n = -1+i0.001 ($n = [n_0^2(x) - k^{-2}c_S h(x)N(x)]^{1/2} = [1^2 - 0.182651^{-2} 4\pi(2.65481\text{E-}09 + i5.30961\text{E-}06)1]^{1/2}$ = -1+i0.001),
- Number of small subcubes after partitioning the domain Ω for solving (red), $P = 8000$.
- Number of collocation points for solving (ie), $C = 64000$.

Table 3 and figure 1 show the time usage in Gordon for solving the wave scattering problem with 1, 4, 7, and 10 billion particles using the algorithm described in Section 3. The computation time is measured by SUs, 1 SU corresponds to 1 hour/core.

Table 3: Time usage for solving the wave scattering problem

Number of particles	1 billion	4 billions	7 billions	10 billions
Time usage (second)	103	1076	2082	1674
Node usage	8	28	49	74
Number of SUs	3.66	133.90	453.41	550.56

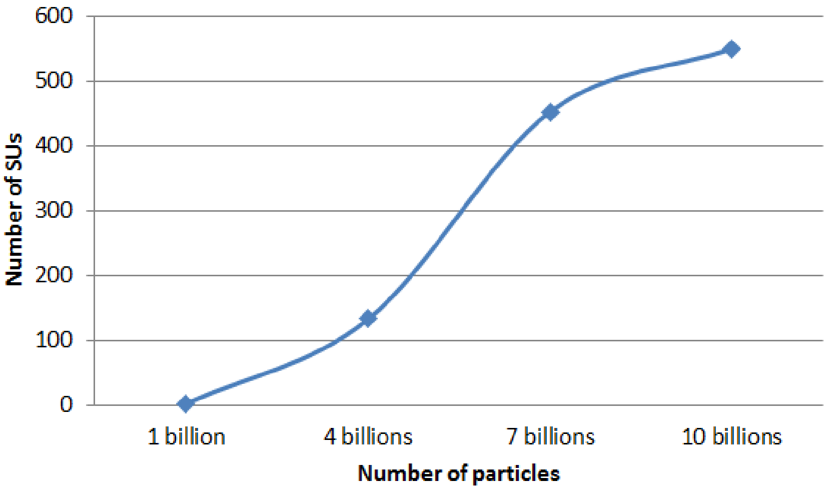


Figure 1: Time usage for solving the wave scattering problem

Table 4 and figure 2 show the differences (errors) between the solutions of (ori) vs. (red), (red) vs. (ie), and (ori) vs. (ie). Since the numbers of unknowns in (ori), (red), and (ie) are different, $M \gg P$ and $P < C$, we use interpolation procedure to compare their solutions. For example, let x and y be the solutions of (ori) and (red), respectively. We find all the particles x_i that lie in the subcube Δ_q corresponding to y_q and then find the difference $|x_i - y_q|$. Then, we compute

$$\sup_{y_q} \frac{1}{\mathcal{N}(\Delta_q)} \sum_{x_i \in \Delta_q} |x_i - y_q|, \tag{4.5}$$

where $\mathscr{N}(\Delta_q)$ is the number of particles in the subcube Δ_q. This gives the difference between the solutions of (ori) and (red). The solution differences between (red) vs. (ie) and (ori) vs. (ie) are computed similarly. The numbers in table 4 are rounded to the nearest tenth thousandths.

Table 4: Solution differences (errors)

Number of particles	1 billion	4 billions	7 billions	10 billions
(ori) vs. (red)	0.0045	0.0045	0.0045	0.0045
(red) vs. (ie)	0.0022	0.0022	0.0022	0.0022
(ori) vs. (ie)	0.0022	0.0022	0.0022	0.0022

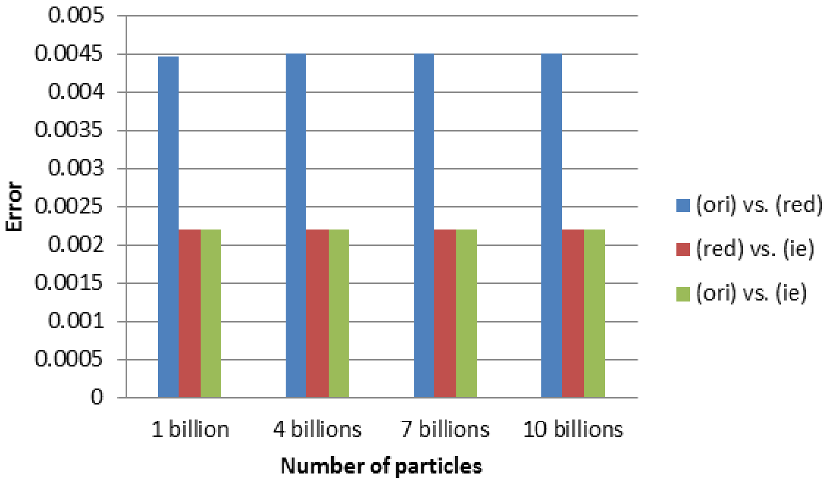
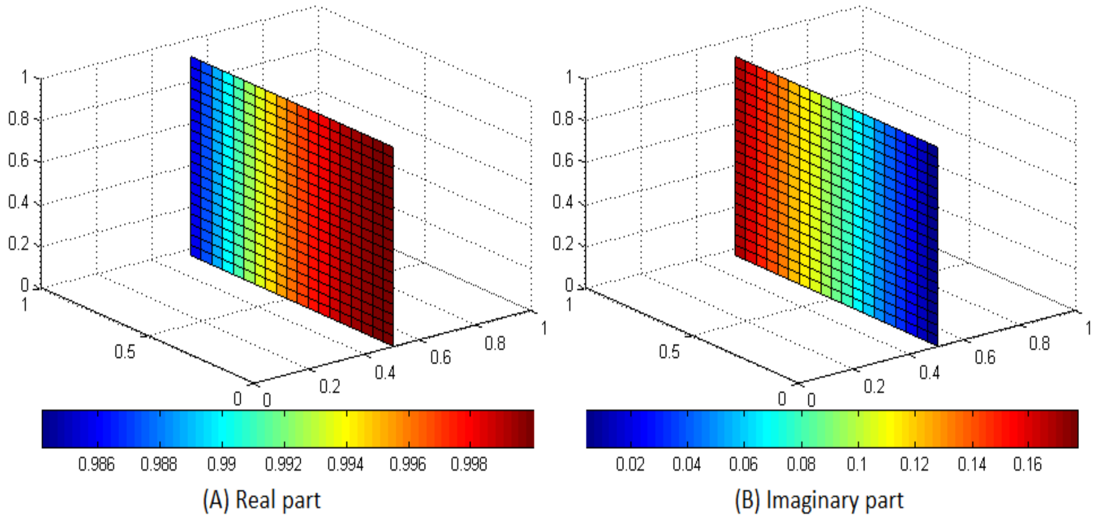


Figure 2: Solution differences (errors)

For example, figures 3, 4 and 5 display vertical slice planes of the solutions, scattering fields, of (ori), (red), and (ie), respectively, at the center of the domain Ω, when $M = 10^9$ particles, $P = 8000$ subcubes, and $C = 64000$ collocation points. The relative errors of the solutions to (ori), (red), and (ie) are 1.72448E-05, 1.94613E-05, and 1.93914E-05, respectively. The solution differences between (ori) vs. (red), (red) vs. (ie), and (ori) vs. (ie) are 0.0045, 0.0022, and 0.0022, respectively. The color bars indicate the values of the corresponding colors. The values used here are the real part and imaginary part of the scattering fields at the grid points on the slices. For reference, tables 5, 6, and 7 show the solutions of (ori), (red), and (ie), respectively, at the grid points 5x5x5 in the unit cube Ω. Since (red) is the Riemann sum of (ie), the solution difference between them is very small.

Conclusions: The numerical results in this paper allow one to solve (ori) for $10^6 < M \leq 10^{10}$. These results show that the solution by (red) for $M = 10^{10}$ agrees with the solution by (ori) with high accuracy (99.55 %), and agrees with the solution of (ie) also with high accuracy (99.78%). Therefore, practically for solving problems with $M > 10^6$ one may use (red) or (ie). For solving the scattering problem for $M < 10^6$ numerically one can use (ori). The accuracy of our numerical method is high if the quantity $ka + ad^{-1}$ is small.



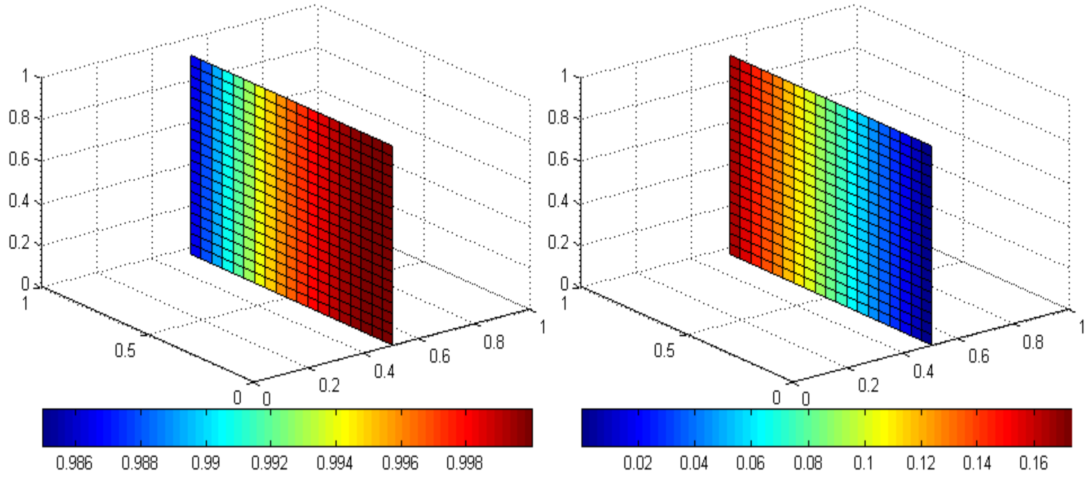
(A) Real part

(B) Imaginary part

Figure 3: Solution of (ori) when $M = 10^9$

Table 5: Solution of (ori) at the grid points 5x5x5 in the cube

0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i
0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i
0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i
0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i
0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i	0.999990+0.004392i
0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i
0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i
0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i
0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i
0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i	0.999163+0.040911i
0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i
0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i
0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i
0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i
0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i	0.997002+0.077375i
0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i
0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i
0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i
0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i
0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i	0.993511+0.113736i
0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i
0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i
0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i
0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i
0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i	0.988694+0.149945i



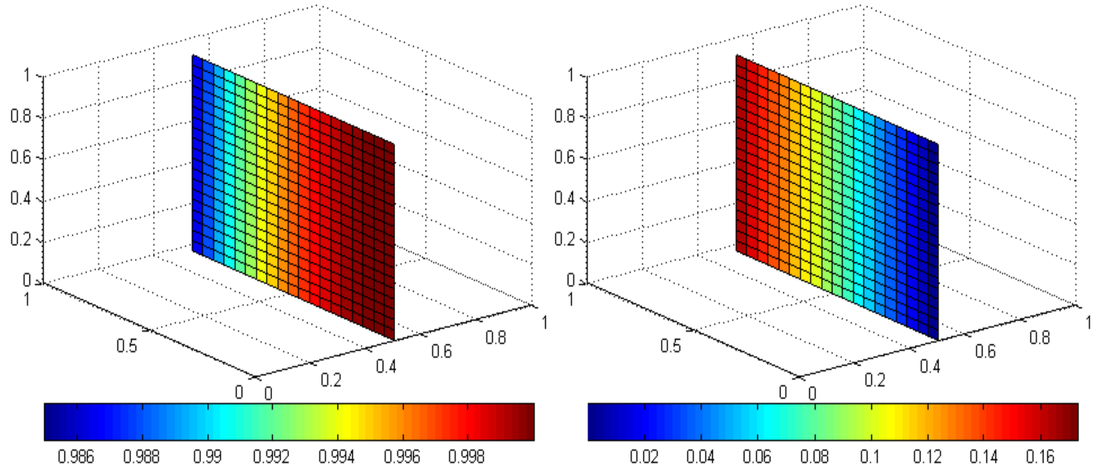
(A) Real part

(B) Imaginary part

Figure 4: Solution of (red) when $M = 10^9$ and $P = 8000$

Table 6: Solution of (red) at the grid points 5x5x5 in the cube

0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i
0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i
0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i
0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i
0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i	0.999999+0.000010i
0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i
0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i
0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i
0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i
0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i
0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i
0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i
0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i
0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i
0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i	0.997331+0.073005i
0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i
0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i
0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i
0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i
0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i	0.993999+0.109381i
0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i
0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i
0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i
0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i
0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i	0.989341+0.145611i



(A) Real part

(B) Imaginary part

Figure 5: Solution of (ie) when $M = 10^9$, $P = 8000$, and $C = 64000$

Table 7: Solution of (ie) at the grid points 5x5x5 in the cube

1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i
1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i
1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i
1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i
1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i	1.000000+0.000010i
0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i
0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i
0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i
0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i
0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i	0.999332+0.036532i
0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i
0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i
0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i
0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i
0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i	0.997332+0.073005i
0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i
0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i
0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i
0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i
0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i	0.994000+0.109381i
0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i
0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i
0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i
0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i
0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i	0.989342+0.145611i

Acknowledgements

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575.

References

[1] Balay, S., Brown, J., Buschelman, K., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Smith, B.F. & Zhang, H. (2013). PETSc Web page, http://www.mcs.anl.gov/petsc.

[2] Clemens, M., and Weiland, T. (1996). Iterative methods for the solution of very large complex symmetric linear systems of equations in electrodynamics. *Technische Hochschule Darmstadt, Fachbereich 18*. Web address: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.4577&rank=1

[3] Darden, T., York, D., and Pedersen, L. (1993). Particle mesh Ewald: An N·log (N) method for Ewald sums in large systems. The Journal of chemical physics, 98(12), 10089-10092.

[4] Eleftheriades, G.V., and Balmain, K.G. (2005). *Negative-refraction metamaterials: fundamental principles and applications*. Wiley-Interscience.

[5] Frigo, M., and Johnson, S. G. (1998). FFTW: An adaptive software architecture for the FFT. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on* Vol. 3, pp. 1381-1384, IEEE.

[6] Greengard, L., Rokhlin, V. (1987). A fast algorithm for particle simulations, J. Comp.Phys., 73, pp. 325-348.

[7] Greengard, L., Rokhlin, V. (1997). A new version of the fast multipole method for the Laplace equation in three dimensions, Acta numerica, 6, 629-669.

[8] Hestenes, M. R., and Stiefel, E. (1952). *Methods of conjugate gradients for solving linear systems* Vol. 49, pp. 409-436, the National Bureau of Standards Research Journal.

[9] Phillips, J. R., and White, J. K. (1997). A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 16(10), 1059-1072.

[10] Lord Rayleigh (1964), *Scientific papers*, Dover, New York.

[11] Ramm, A. G. (1970). Approximate formulas for polarizability tensors and capacitances of bodies of arbitrary shapes and applications. In *Doklady Acad. Sci. USSR* Vol. 195, pp. 1303-1306.

[12] Ramm, A. G. (2005). *Wave scattering by small bodies of arbitrary shapes*, World Sci. Publishers, Singapore.

[13] Ramm, A. G. (2013). Many-body wave scattering problems in the case of small scatterers. *Journal of Applied Mathematics and Computing*, 41(1-2), 473-500.

[14] Ramm, A.G. (2013). *Scattering of Acoustic and Electromagnetic Waves by Small Bodies of Arbitrary Shapes. Applications to Creating New Engineered Materials,* Momentum Press, New York.

[15] Saad, Y., and Schultz, M.H., 1986 GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on scientific and statistical computing,* **7.3**, 856-869.

[16] Tran, N. T. (2013). Numerical Solution of Many-body Wave Scattering Problem and Creating Materials with A Desired Refraction Coefficient, *International Journal of Structural Changes in Solids (IJSCS)*, **5**, 27-38.

[17] van der Vorst, H. A., and Melissen, J. B. (1990). A Petrov-Galerkin type method for solving Ax = b, where A is symmetric complex. *IEEE Transactions on Magnetics*, 26(2), 706-708.

[18] XSEDE (2012). Gordon Web page, https://www.xsede.org/sdsc-gordon