Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 211 (2006) 289–299
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Fast evaluation of electro-static interactions in multi-phase dielectric media

Per-Gunnar Martinsson *

Department of Mathematics, Yale University, P.O. Box 208283, New Haven, CT 06520-8283, United States

Received 30 November 2004; received in revised form 9 May 2005; accepted 11 May 2005
Available online 12 July 2005

Abstract

A method is presented for the rapid evaluation of electro-static interactions in a dielectric medium containing inclusions with different dielectric properties. The principal observation is that in environments where several potential evaluations are required for a fixed geometry, it is advantageous to precompute compressed representations of the inverses of certain operators. Numerical examples in two dimensions show a speed-up of a factor of 10–100 when compared to the fastest existing methods. Moreover, the method presented is based on direct (as opposed to iterative) techniques, making it preferable to existing methods in environments involving ill-conditioned problems.

1. Introduction

We consider the problem of evaluating the electro-static interactions between a number of charged particles distributed in a domain consisting of two (or more) regions with different dielectric constants, as illustrated in Fig. 1. This evaluation problem occurs in many areas of science and engineering and is frequently the most expensive part of computational simulations. The dielectrics problem that we focus on is commonly used to model electrochemical interfaces, semiconductor junctions, the solvation of macromolecules, ion channels, etc., see, e.g. [3]. Furthermore, the problem we consider is mathematically equivalent to the conduction problem for a material containing inclusions with different conductivities. It is also very similar to the problem of solving Stokes' equation for viscous fluid suspensions, see [12,13], and to the problem of determining the stresses in an elastic body having inclusions with different material properties, see [4,7].

* Tel.: +1 203 432 1277.
E-mail address: per-gunnar.martinsson@yale.edu.

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.05.018

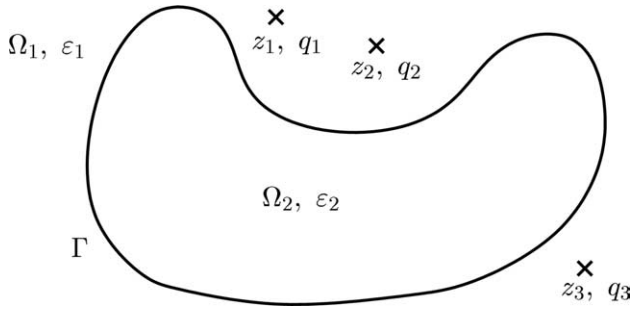

Fig. 1. The domain $\Omega = \mathbb{R}^2 = \Omega_1 \cup \Omega_2$. In the sub-domain Ω_i, the dielectric constant is ε_i. In Ω_2, there are $M = 3$ point particles; particle number i is located at the position $z_i \in \mathbb{R}^2$ and has the electric charge q_i. The interface is called Γ.

Accelerated solution techniques for the problem described in the previous paragraph tend to fall in two categories. Methods in the first category (sometimes called "P^3M" methods) are based on fast Poisson solvers; these methods interpolate the point charges to a continuous charge density on some mesh imposed on the domain, and incorporate boundary conditions that enforce the continuity of electric fluxes across the interfaces between the domains, see e.g. [1,8,16]. Methods in the second category use classical potential theory to rewrite the evaluation problem as a set of two subproblems; (i) determine the induced charges on the domain interfaces by solving an integral equation, and (ii) evaluate the force on each particle as the sum of the force resulting from the interaction with the induced charges and the force resulting from all pairwise interactions with other particles, see [3,12,14].

One feature of the technique based on boundary methods is that it can easily be accelerated in the environment where several force calculations need to be performed for a fixed geometry. It was observed in [3], that in this situation one can pre-compute the inverse of the operator in the integral equation that determines the induced charges. Once this has been done, the induced forces can be determined through a simple matrix–vector multiplication. The computational cost for each force evaluation is then $O(M^2 + N^2)$, where M is the number of degrees of freedom used in the discretization of the integral operator and N is the number of charged particles.

In this paper, we demonstrate that further acceleration can be obtained by using multilevel "fast" techniques for both the inversion of the integral operator, and for the evaluation of charge–charge interactions. The resulting method has a computational complexity of $O(M + N)$ for each force evaluation. We use the fast multipole method, see [6], to rapidly evaluate the pairwise interactions between the particles and the fast direct solver of [9] to evaluate the induced charges. While FMM-accelerated methods could be used for both these tasks, the method of [9] is considerably faster, see Remark 4 and Table 3. Moreover, all FMM accelerated methods for solving integral equations are based on iterative procedures, which compare unfavorably with direct methods in terms of sensitivity to ill-conditioning, the possibility of reusing computed data to solve slightly perturbed problems, etc.

The accelerated technique described above has been implemented and tested on a number of two-dimensional problems. We present examples showing that: (i) The break-even point where acceleration techniques start paying off is less than one thousand particles. (ii) Using boundary integral equations, it is feasible to achieve solutions that are accurate to ten digits, or more. (This property makes validation of computer calculations very simple.) (iii) Since the complexity of the accelerated method scales linearly with problem size, very large problems can be solved even on modest computers. We show that at 10 digits accuracy, a problem involving one hundred thousand particles and a contour discretized into 25,000 nodes can be solved in a couple of seconds on an ordinary desktop PC.

The paper is structured as follows: Section 2 introduces a model problem involving a dielectric interface and derives an integral equation that describes the induced charges on the interface. Section 3 describes effi-

cient procedures for solving the equations given in Section 2 numerically. Section 4 presents the results from several numerical examples and Section 5 summarizes the main findings.

2. A model problem

In this section, we introduce a simple electro-statics problem involving a distribution of point charges in a two-phase dielectric medium. We describe the problem and derive equations that determine the electro-static forces on all particles.

The geometry of the problem is shown in Fig. 1; a number of charged particles are distributed in an unbounded dielectric medium containing a finite inclusion consisting of a different dielectric medium. When the system is in electro-static equilibrium, the two dielectric media will be polarized in such a fashion that there will appear to be a monopole charge distribution – the "induced" charges – on the boundary between the two media. The electro-static force experienced by any particle can then be viewed as a combination of the force arising from direct pairwise interactions with the other particles and the force arising from interactions with the induced charges on the interface.

For notational simplicity, we will in the remainder of this section focus on equations set in two dimensions even though equations in higher dimensions can be handled analogously.

In order to determine the electro-static forces described in the previous paragraph, we need to introduce some notation. We let $\Omega = \mathbb{R}^d$ denote the full domain, while Ω_2 denotes the inclusion and $\Omega_1 = \Omega \backslash \Omega_2$, see Fig. 1. The dielectric constant of the domain Ω_j is ε_j for $j = 1,2$. There are M charged particles with charges q_i, located at positions $z_i \in \Omega_1$, for $i = 1, \ldots, M$. We let u denote the electric potential, $u^{(1)}$ its restriction to Ω_1 and $u^{(2)}$ its restriction to Ω_2. When the system is in electro-static equilibrium, u satisfies the equations

$$-\varepsilon_1 \Delta u^{(1)}(x) = \sum_{i=1}^{M} q_i \delta(x - z_i), \quad x \in \Omega_1, \tag{1}$$

$$-\varepsilon_2 \Delta u^{(2)}(x) = 0, \quad x \in \Omega_2. \tag{2}$$

Letting n denote the outward unit normal of the interface $\Gamma = \partial\Omega_2$, the boundary conditions on Γ read

$$u^{(1)}(x) = u^{(2)}(x), \quad x \in \Gamma, \tag{3}$$

$$\varepsilon_1 \frac{\partial u^{(1)}}{\partial n}(x) = \varepsilon_2 \frac{\partial u^{(2)}}{\partial n}(x), \quad x \in \Gamma. \tag{4}$$

The final condition on u is a decay condition at infinity ensuring that the solution has finite total energy. In two dimensions, the condition is that there must exist a constant c such that

$$\lim_{|x| \to \infty} \left| u^{(1)}(x) - c \log |x| \right| = 0. \tag{5}$$

Eqs. (1)–(5) uniquely define the electric potentials $u^{(1)}$ and $u^{(2)}$. (In Eq. (5), it turns out that $c = -\sum_{i=1}^{M} q_i / 2\pi\varepsilon_1$, as could be expected.)

The solution of the system (1)–(5) can be decomposed into a sum of a potential caused directly by the point charges, and another potential caused by the induced charges on the interface Γ. Formally speaking, we make the ansatz

$$u = v + w, \tag{6}$$

where v is the potential caused directly by the point-charges,

$$v(x) = -\frac{1}{2\pi\varepsilon_1}\sum_{i=1}^{M} q_i \log|x - z_i|, \tag{7}$$

and w is the potential caused by the induced charge distribution σ,

$$w(x) = \int_\Gamma \left(-\frac{1}{2\pi\varepsilon_1}\log|x-y|\right)\sigma(y)\,\mathrm{d}s(y). \tag{8}$$

The function u defined by (6)–(8) automatically satisfies the conditions (1), (2), (3) and (5). The remaining condition, (4), is used to construct an equation that determines the induced charge distribution σ in (8). To this end, we fix an $x \in \Gamma$, and note that the derivative of $u^{(1)}$ in the direction of the normal $n(x)$ is given by the limit

$$\varepsilon_1 \frac{\partial u^{(1)}}{\partial n}(x) = \lim_{y\to x, y\in\Omega_1} \varepsilon_1 \nabla u(y)\cdot n(x). \tag{9}$$

It is a standard result in the study of integral equations that under minimal smoothness assumptions on the boundary, the above limit evaluates as follows:

$$\varepsilon_1 \frac{\partial u^{(1)}}{\partial n}(x) = \varepsilon_1 \frac{\partial v}{\partial n}(x) - \frac{1}{2}\sigma(x) + [D\sigma](x), \tag{10}$$

where D denotes the "double layer operator",

$$[D\sigma](x) = -\frac{1}{2\pi}\int_\Gamma \frac{n(x)\cdot(x-y)}{|x-y|^2}\sigma(y)\,\mathrm{d}s(y). \tag{11}$$

For a discussion of such limits, see [10].
Similarly, we find that

$$\varepsilon_2 \frac{\partial u^{(2)}(x)}{\partial n} = \varepsilon_2 \frac{\partial v}{\partial n}(x) + \frac{\varepsilon_2}{\varepsilon_1}\frac{1}{2}\sigma(x) + \frac{\varepsilon_2}{\varepsilon_1}[D\sigma](x). \tag{12}$$

In view of (10) and (12), we find that (4) is satisfied if and only if the induced charge distribution σ satisfies

$$-\frac{\varepsilon_1+\varepsilon_2}{2(\varepsilon_1-\varepsilon_2)}\sigma(x) + [D\sigma](x) = -\varepsilon_1\frac{\partial v}{\partial n}(x), \qquad x \in \Gamma, \tag{13}$$

where v is defined by (7). We remark that Eq. (13) is an example of what is called a "second kind Fredholm equation" and that such equations typically are very benign; for instance, it is known that (13) has a unique solution whenever $\varepsilon_1 \neq \varepsilon_2$, see [11]. Moreover, in most situations of practical interest, the ratio between the largest and the smallest singular values (the "condition number") of (13) tends to be small, which is very beneficial when the equation is discretized and solved numerically. (An important exception is the case where $\varepsilon_2 \gg \varepsilon_1$, discussed in Remark 2.)

Once the induced charge distribution σ has been determined by solving (13), the electro-static force acting on particle i is given by the formula

$$F_i = F_i^{(\text{particles})} + F_i^{(\text{induced})}, \tag{14}$$

where

$$F_i^{(\text{particles})} = \sum_{i\neq j} \frac{q_i q_j}{2\pi\varepsilon_1}\frac{z_i - z_j}{|z_i - z_j|^2}, \tag{15}$$

and

$$F_i^{(\text{induced})} = -q_i \nabla w(z_i) = \frac{q_i}{2\pi\varepsilon_1} \int_\Gamma \frac{z_i - y}{|z_i - y|^2} \sigma(y)\, \mathrm{d}s(y). \tag{16}$$

Remark 1 (*Generalizations*). For clarity, the solution technique described in this section was presented in the simplest possible setting. It can without difficulty be extended to environments involving:

(1) Particles with other types of charges (e.g., dipole charges).
(2) Several inclusions (possibly with different dielectric constants).
(3) Inclusions that contain charged particles.
(4) Different boundary conditions (e.g., condition (5) can be replaced by periodic boundary conditions).

It is also possible to replace the Laplace equation by any other equation that has a translation invariant fundamental solution with an appropriate singularity at the origin (examples include Helmholtz' equation, Yukawa, etc). The numerical technique to be described in Section 3 works well under any of the generalizations 1–4 but it cannot handle problems with highly oscillatory fundamental solutions. In particular, for certain wave propagation problems, other techniques would be needed.

Remark 2 (*High contrast ratios*). Some complications arise when Eq. (13) is to be solved for an inclusion for which ε_2 is much larger than ε_1. The condition number of (13) then scales as $\varepsilon_2/\varepsilon_1$ and care must be taken to avoid incurring numerical errors, see [17]. Since we use direct methods to solve (13), this is less of a concern than it would have been if iterative methods had been used. However, if $\varepsilon_2/\varepsilon_1$ is truly very large, it may be necessary to reformulate Eq. (13) slightly in order to avoid incurring prohibitively large round-off errors, see [5]. Table 4 gives quantitative numbers for a particular geometry. (We note that the case where $\varepsilon_2 \ll \varepsilon_1$ presents no difficulties, and that the case $\varepsilon_1 \approx \varepsilon_2$ is extremely benign; in fact, as $\varepsilon_2 \to \varepsilon_1$, the condition number of (13) approaches 1, and $\|\sigma\| \to 0$.)

3. Numerical solution techniques

In this section, we discuss numerical techniques for solving the electro-statics problem described in Section 2. The goal is to compute the electric forces specified by Eq. (14), given a set of particle positions and charges $\{z_i, q_i\}_{i=1}^M$, two dielectric constants ε_1 and ε_2 and an interface Γ. We do this by performing the following four steps:

(1) Evaluate $\partial v/\partial n$ on Γ, where v is given by (7).
(2) Solve Eq. (13) to determine the induced charges.
(3) Evaluate the forces caused by the induced charges, cf. (15).
(4) Evaluate the forces caused by particle–particle interactions, cf. (16).

While the question of how to discretize the integral equation (13) is generally very important, it is not the focus of the current work. For now, we simply assume that it has somehow been discretized using a Nyström method on the points $\{x_i\}_{i=1}^N \subset \Gamma$, and note that such schemes are in most environments stable and highly accurate, see [2].

The evaluation of the sum (16) in Step (4) is a well-known numerical problem. If N is small, it is simply solved by direct evaluation using $O(N^2)$ operations. If N is large, the Fast Multipole Method, see [6], solves this problem in $O(N)$ operations.

When the Nyström method is used to discretize (13), the evaluation problems in Steps (1) and (3) are almost entirely analogous to the evaluation problem in Step (4) (the only difference being that the set of evaluation points is different from the set of source points). If direct calculations are used, $O(NM)$ arithmetic operations are required; if the FMM is used, only $O(N + M)$ operations are required.

It remains to discuss Step (2). If Eqs. (1)–(5) need to be solved only once, it is well-understood how to efficiently solve the system of equations obtained when Eq. (13) is discretized: If N is small, use Gaussian elimination, otherwise use an iterative solver (such as GMRES, see [15]) combined with a fast method for applying the integral operator to a vector, such as, e.g., the fast multipole method. A question that seems to be less well-understood is how to accelerate Step (2) in an environment where the interface Γ is fixed (or changes only infrequently) while Eqs. (1)–(5) need to be solved for several distributions $\{z_i, q_i\}_{i=1}^{M}$ of charged particles. It was observed in [3] that in this environment, it is advantageous to pre-compute the inverse of the matrix associated with Eq. (13), since then a simple matrix–vector multiplication is all that is needed to determine the induced charges in Step (2). The main point of the current work is to illustrate that by using a fast inversion technique presented in [9], this step can be accelerated further by computing a compressed version of this inverse that can be applied to a vector using $O(N)$ arithmetic operations.

Remark 3. The proposed method has a potential drawback in that it forces us to choose the discretization points $\{x_i\}_{i=1}^{N}$ at the time of the pre-computation. This could potentially cause a problem if (say during the course of a particle simulation) a particle position z_i happens to get close to the interface. If the particle is only moderately close, the problem can be addressed by placing an image charge at the opposite side of the interface. Should the method of image charges not be sufficient, it is possible to cheaply update the computed inverse to incorporate a locally more refined mesh.

Remark 4 (*Comparison with FMM*). Instead of using the fast inversion scheme of [9] to compute the induced charges in Step (2), it is possible to use an iterative solver in combination with a fast algorithm for applying the operator in (13) such as the fast multipole method, see [14]. Both approaches have a computational complexity of $O(N + M)$, but it is our experience that the method using a pre-computed inverse is significantly faster in typical environments (see Table 3 for a numerical comparison). However, in situations where the geometry changes frequently, iterative solvers have an advantage in that they do not rely on any pre-computation.

Remark 5 (*Accelerated FMM*). In an environment where pre-computation is possible, one can construct very fast algorithms for applying the original operator to a vector. Such methods could alternatively be used to accelerate Step (2) and would then result in a method that is about as fast as the direct method proposed here provided that Eq. (13) is very well-conditioned, see Table 3. However, there are few obvious advantages to such an approach.

4. Numerical experiments

In this section, we illustrate through numerical examples the relative performance of the numerical techniques for solving Eqs. (1)–(5) that were discussed in Section 3. We consider only the environment where the interface Γ is fixed and some pre-computation of the inverse of the matrix associated with (13) can be performed.

We compared three different combinations of acceleration techniques for performing the calculations referred to as Steps (1), (2), (3) and (4) in Section 3. The gradient calculations of Steps (1), (3) and (4) were done either via direct computations, or via the FMM. The problem of determining the induced charges in

Step (3) was solved either by multiplying by the full inverse of the matrix associated with Eq. (13), or by multiplying by the compressed inverse. We considered the following combinations:

	Steps (1), (3), (4)	Step (2)
Method I	Direct calculation	Full inverse
Method II	Direct calculation	Compressed inverse
Method III	FMM	Compressed inverse

Letting t_α denote the CPU time required for method α, we expect that

$$t_{\mathrm{I}} = \mathrm{O}(N^2 + M^2),$$
$$t_{\mathrm{II}} = \mathrm{O}(N^2 + NM),$$
$$t_{\mathrm{III}} = \mathrm{O}(N + M),$$

where N denotes the number of point charges, and M denotes the number of points used to discretize the interface.

Eq. (13) was discretized using the Nyström method with the trapezoidal rule. We considered smooth contours only, meaning that this method has an exponential rate of convergence (note that the kernel in (11) is in fact smooth), see [14]. In the tests presented, the discretization error was less than 10^{-10}, provided that all charged particles were separated from the interface by at least a distance of $5h$, where h is the distance between discretization nodes on the contour. Some experiments were run at a relative accuracy of 10^{-5}, in which case an interface-particle separation of $2.5h$ was sufficient.

The algorithms were implemented in Fortran 77 and run on a desktop PC with a 2.8 GHz Pentium IV processor and 512 Mb of RAM memory.

4.1. A simple channel-type geometry

In the experiments presented in this section, Eqs. (1)–(5) were solved to a relative accuracy of 10^{-10} on the geometry shown in Fig. 2. The CPU times required by the three methods under consideration are shown in Table 1. The table shows that if N and M are both on the order of hundreds, the acceleration techniques presented here are not helpful. However, if either of these numbers exceed about 1000, the savings very rapidly become significant. (At lower accuracies, the break-even points are of course lower.)

The approach of this paper requires some pre-computation. The corresponding CPU times are presented in Table 2.

Table 3 gives the time required for an FMM-accelerated matrix–vector multiply and the time required to apply the compressed inverse to a vector at a number of different problem sizes; note that the former is much larger than the latter, see Remark 4. Table 3 also reports the CPU times required for a matrix–vector multiply using an FMM accelerated via pre-computation, as discussed in Remark 5. (The actual algorithm used here has not yet been published.)

The CPU time requirements reported are entirely independent of the values of the dielectric constants ε_1 and ε_2. However, these numbers do influence the condition number of the equations that we solve, and thus the precision of the computed results, see Remark 2. Table 4 reports the numerical errors incurred in the accelerated inversion of (13) for a range of different ratios $\varepsilon_2/\varepsilon_1$. These numbers show that since we use direct methods, even quite ill-conditioned problems can be solved rapidly and accurately.

16, 32.

metries were

Table 2
$t_{\alpha,\text{pre-comp}}$ denotes the CPU time (in seconds) required for pre-computation when using method α = I, II. $t_{\alpha,\text{solve}}$ denotes the CPU time needed to solve Eq. (13) when using method α = I, II

M	400	800	1600	3200	6400	12,800	25,600	51,200	102,400
$t_{\text{I,pre-comp}}$	4.0e − 1	3.3e + 0	2.8e + 1	2.2e + 2	(1.7e + 3)	(1.4e + 4)	(1.1e + 5)	(8.8e + 5)	(7.1e + 6)
$t_{\text{I,solve}}$	8.0e − 4	3.2e − 3	1.2e − 2	4.8e − 2	(1.9e − 1)	(7.7e − 1)	(3.1e + 0)	(1.2e + 1)	(4.9e + 1)
$t_{\text{II,pre-comp}}$	4.1e − 1	8.4e − 1	1.1e + 0	1.2e + 0	1.4e + 0	1.9e + 0	2.9e + 0	4.7e + 0	9.5e + 0
$t_{\text{II,solve}}$	2.5e − 3	5.5e − 3	6.2e − 3	8.0e − 3	1.2e − 2	1.9e − 2	3.4e − 2	6.1e − 2	1.2e − 1

Note that these times are the same for methods II and III. Numbers in parenthesis are extrapolated.

Table 3
t_{FMM} is the time (in seconds) required for an FMM matrix–vector multiply on the contour in Fig. 2, while $t_{\text{II,solve}}$ is the time required to apply the compressed inverse to a vector (both at a relative accuracy of 10^{-10})

M	1600	3200	6400	12,800	25,600	51,200	102,400
t_{FMM}	1.8e − 2	2.7e − 2	5.4e − 2	1.1e − 1	2.2e − 1	4.5e − 1	8.8e − 1
$t_{\text{II,solve}}$	6.2e − 3	8.0e − 3	1.2e − 2	1.9e − 2	3.4e − 2	6.1e − 2	1.2e − 1
$t_{\text{FMM,acc}}$	1.2e − 3	2.2e − 3	3.9e − 3	7.4e − 3	1.5e − 2	2.8e − 2	5.5e − 2

$t_{\text{FMM,acc}}$ is the time required for a matrix–vector multiply using an accelerated FMM, as described in Remark 5.

Table 4
The condition number c of Eq. (13) is reported for a range of ratios between the dielectric constants ε_1 and ε_2

$\varepsilon_2/\varepsilon_1$	1.0e − 9	1.0e − 6	1.0e − 3	1.1e + 0	1.0e + 3	1.0e + 6	1.0e + 9
c	1.0e + 1	1.0e + 1	1.0e + 1	1.1e + 0	1.9e + 3	1.9e + 6	1.9e + 9
E	8.0e − 9	8.0e − 9	8.0e − 9	3.8e − 10	2.9e − 7	3.0e − 4	2.9e − 1

For each ratio, we also report the error $E = \max_f |A^{-1}f - A_\varepsilon^{-1}f|/|A^{-1}f|$ which indicates how far the computed approximate inverse A_ε^{-1} is from the exact inverse A^{-1}. All experiments in this table relate to the contour shown in Fig. 2, which was discretized into 25,600 points.

N = 25,000, 50,000, 100,000 charged particles were placed outside the inclusions. The CPU times required to determine the forces on all particles using Method III are given in Table 5, from which it is easily seen that the CPU times scale linearly with problem size.

We remark that when a contour takes on an "area-filling" character like the contours in this example, the simple compression algorithm used here requires $O(M^{3/2})$ arithmetic operation to pre-compute the inverse of the matrix. Once the inverse has been computed, its application to a vector is still very cheap, as Table 5 shows. (Moreover, we expect that more sophisticated inversion techniques that are currently under development will soon reduce the cost of the pre-computation from $O(M^{3/2})$ to $O(M \log M)$.)

5. Conclusions

In this paper, we investigated the usefulness of fast direct methods as a technique for evaluating the forces on a collection of charged particles distributed in a multi-phase dielectric medium. The basic idea is that by expending some computational effort analyzing a given geometry, it becomes possible to perform force-evaluations very rapidly. Through numerical examples, we demonstrated that two-dimensional problems involving tens of thousands of charged particles and geometries that require tens of thousands of

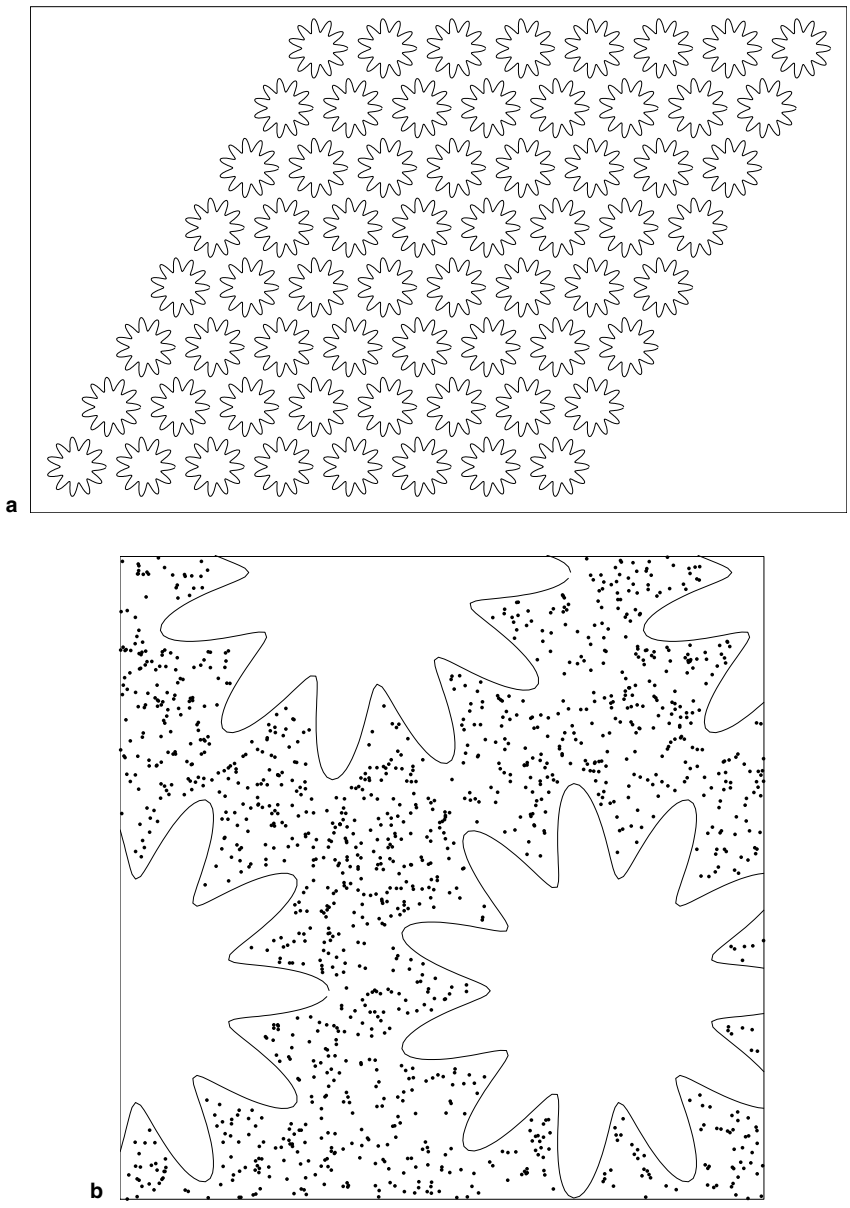

Fig. 3. (a) A domain with 64 dielectric inclusions. The interface was discretized using $M = 25{,}600$ points and $N = 100{,}000$ point charges were randomly distributed inside the rectangular box. Due to the number of particles, they are not shown in (a), but a close-up of the particle distribution is given in (b).

discretization points can be solved in about one second on a 2.8 GHz desktop PC (at a relative accuracy of 10^{-10}).

The techniques considered here can easily be extended to the solution of other Laplace-type problems (heat conduction, elasticity, Yukawa, etc.) on domains involving piecewise constant material properties. We expect this method to outperform existing methods in those situations where multiple problems need to be solved for a given geometry.

At this point, the techniques presented have been fully developed only in two dimensions. The extension to geometries involving surface interfaces in three dimensions is under way and will be reported at a later date.

Acknowledgments

The idea of using the fast inversion techniques of [9] to accelerate numerical simulations of the type con-