SIAM J. Sci. Comput.
Vol. 38, No. 3, pp. A1538–A1560

A FINITE ELEMENT BASED $\mathrm{P}^3\mathrm{M}$ METHOD FOR N-BODY PROBLEMS[*]

NATALIE N. BEAMS[†], LUKE N. OLSON[‡], AND JONATHAN B. FREUND[§]

Abstract. In this paper we introduce a new mesh-based method for N-body calculations. The method is founded on a particle-particle–particle-mesh ($\mathrm{P}^3\mathrm{M}$) approach, which decomposes a potential into rapidly decaying short-range interactions and smooth, mesh-resolvable long-range interactions. However, in contrast to the traditional approach of using Gaussian screen functions to accomplish this decomposition, our method employs specially designed polynomial bases to construct the screened potentials. Because of this form of the screen, the long-range component of the potential is then solved accurately with a finite element method, leading ultimately to a sparse matrix problem that is solved efficiently with standard multigrid methods, though the short-range calculation is now more involved than $\mathrm{P}^3\mathrm{M}$ particle-mesh-Ewald (PME) methods. We introduce the method, analyze its key properties, and demonstrate the accuracy of the algorithm.

Key words. N-body, finite element, multigrid, $\mathrm{P}^3\mathrm{M}$, PME, multipole methods

AMS subject classifications. 70-08, 70F10, 65N30, 65N99

DOI. 10.1137/15M1014644

1. Introduction.

N-body interactions arise in a range of applications, including molecular dynamics, plasma dynamics, vortex methods, and viscous flow: systems that are described by a Green's function solution to the Poisson equation or its derivatives. We focus on three-dimensional electrostatic-like $1/R$ interactions, where R is the distance to a particle; this is the simplest kernel in three dimensions and well known for this class of problems. However, the resulting algorithm we describe extends to other systems. We first consider a periodic domain, which is commonly used to model extensive systems, and discuss a straightforward extension to other boundary conditions in section 2.6. We consider a cubic cell of length L containing N point charges, which has the total electrostatic potential energy

$$\mathcal{U} = \frac{1}{2}\sum_{i=1}^{N} Q_i \Phi_i, \tag{1.1}$$

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section March 30, 2015; accepted for publication (in revised form) March 7, 2016; published electronically May 26, 2016. This work was supported by the Computational Science & Engineering program at the University of Illinois at Urbana-Champaign, grants NSF 09–32607, 13–36972, and 07–46676. This work was also sponsored by the Air Force Office of Scientific Research under grant FA9550–12–1–0478. This material is based in part upon work supported by the Department of Energy, National Nuclear Security Administration, under award DE-NA0002374.

http://www.siam.org/journals/sisc/38-3/M101464.html

[†] Department of Mechanical Science & Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (beams2@illinois.edu).

[‡] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (lukeo@illinois.edu).

[§] Department of Mechanical Science & Engineering, and Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (jbfreund@illinois.edu).

where Φ_i is the electrostatic potential at location $\mathbf{x}_i$ of particle i with charge Q_i. The potential in the specific case of a periodic domain is

$$\Phi(\mathbf{x}_i) = \underbrace{\sum_{\substack{j=1 \\ i\neq j}}^{N} \frac{Q_j}{|\mathbf{x}_i - \mathbf{x}_j|}}_{\text{charges}} + \underbrace{\sum_{\substack{\mathbf{n}=-\boldsymbol{\infty} \\ \mathbf{n}\neq 0}}^{\boldsymbol{\infty}} \sum_{j=1}^{N} \frac{Q_j}{|\mathbf{x}_i - \mathbf{x}_j + \mathbf{n}L|}}_{\text{periodic images}} . \tag{1.2}$$

We also write the potential in more general terms,

$$\Phi(\mathbf{x}_i) = \sum_{\substack{j=1 \\ j\neq i}}^{N} Q_j \, G(\mathbf{x}_i;\, \mathbf{x}_j), \tag{1.3}$$

where $G(\mathbf{x}_i;\, \mathbf{x}_j)$ is the Green's function describing the influence of charge j at the location of charge i, subject to boundary conditions. The fairly slow $1/R$ decay rate of the Green's function at large distances is the central challenge in the computation of (1.1).

There are a number of approaches for efficiently evaluating (1.3). The most widely used methods are generally classified as either tree-based, such as the fast multipole method (FMM) [11], or mesh-based (sometimes called "particle-in-cell"), such as the particle-particle–particle-mesh (P^{3}M) method [16] and its popular variant, the particle-mesh-Ewald (PME) method [5, 7]. In the FMM, particles are grouped within multipole expansions to provide an accurate representation of their combined influence at a distance, thus limiting the number of terms needed in the computation. The resulting algorithm scales with $\mathcal{O}(N)$ complexity, although the coefficient in this scaling can be large, especially if a high-order multipole expansion is required for the desired accuracy [12]. Efficient implementations are intricate—especially in parallel—but demonstrated, and the FMM has been shown to be effective as an adaptive three-dimensional algorithm [4]. The method also extends to systems with more complicated kernels, such as Stokes flow [28, 30, 29].

In comparison, mesh-based methods also reduce the number of explicit particle pair calculations but achieve this by splitting the potential into a rapidly decaying component Φ^{sr}, which is accurately calculated with inclusion of only a few short-range interactions, and a smooth part Φ^{sm}, which is solved on a mesh covering the domain [16]. It is instructive to view this splitting as the addition and subtraction of strategically selected "screening" functions, so that the potential in (1.3) decomposes as

$$\Phi_i = \underbrace{\Phi_i - \Phi_i^{\text{sm}}}_{\Phi_i^{\text{sr}}} + \Phi_i^{\text{sm}}. \tag{1.4}$$

The PME method [5] bases this decomposition directly on the Ewald summation [8] for (1.3) and uses Lagrangian interpolation to move between particle locations and the mesh, while the smooth PME (SPME) uses B-spline interpolants, similar to those proposed in the P^{3}M method [7]. PME-based algorithms use Gaussian screening functions, as illustrated in Figure 1. Here, the screen is designed to yield a Φ^{sr} that is straightforward to calculate within a prescribed cutoff at radius R_c, while the long-range portion of the potential remains smooth.

In PME-based methods, the Gaussian screen yields a Φ^{sm} that is accurately solved by fast Fourier transforms (FFTs). To do this, the screen is interpolated to a regular mesh and the Poisson (or similar) operator is inverted. For computational efficiency it is desirable that these screens be as compact as possible and barely resolved on the mesh, since this maximizes the decay of the screened potential $\Phi^{\rm sr}$. Fast decay allows for a small point-to-point interaction cutoff distance R_c, which reduces the number of interactions that need to be explicitly computed for the targeted accuracy. The ideal wavenumber resolution of the FFT provides accurate representation of the most compact screens possible. The FFT also makes these methods most natural for periodic domains, but they can be extended to free space [24, 9].

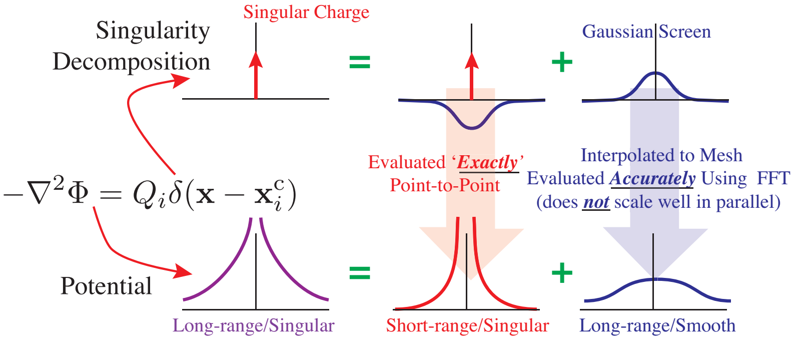


FIG. 1. *Introduction of a Gaussian screen to define* (1.4) *and resulting decomposition of the potential of the singular charge.*

Here we propose a fundamentally new decomposition that is constructed within a P^{3}M-type framework. The method incorporates screening functions that are selected to yield mesh-based potentials, which has many benefits. The screens are designed for a mesh and thus have no explicit dependence on problem geometry; this suggests complex geometries as well as more general boundary conditions fit naturally within this method. In addition, calculating the mesh potential is recast as a sparse matrix problem where multigrid methods are known (and shown in section 4) to be effective and scale to high core counts [1]. As a result, since the method does not rely on the Fourier resolution for an accurate mesh solution, a global FFT can be avoided, which may be beneficial at extreme scales. Indeed, while multigrid methods are ultimately latency bound, they do not exhibit the strong dependence on a machine's half-bandwidth, which is a limiting factor of using multidimensional FFTs a large core counts [10].

Previous work on improving the performance of Ewald-type schemes ranges from utilizing coarsening strategies to reduce the number of grid points by using a staggered mesh [3] to multilevel approaches [2] that yield increased locality in the FFT calculations while resulting in only a small increase in total work. Other methods such as the Multilevel Summation Method (MSM) [25, 26, 27, 17] take a different approach to operator splitting altogether.

The new decomposition we propose does come with the cost of representing more intricate short-range interactions. The calculations are more involved than the simple isotropic point-to-point interactions of PME, but are tractable and, more importantly, *local*. As we highlight in the following sections, the short-range potential also has fast but algebraic decay (up to $1/R^6$ in our examples), which is less attractive than the exponential decay seen in PME, thus possibly leading to more local interactions.

In summary, the goal of this paper is to detail a method that incorporates mesh-based screens and to investigate the accuracy of such an approach. In section 2, we develop the mathematical construction of each component. In particular, we detail the screen functions that lead to the exact sparse linear system for Φ^{sm} and the local evaluation of Φ^{sr}. In section 3, we develop a performance model for the method and discuss its implications in a parallel setting. Numerical experiments are shown in section 4 to confirm the accuracy of our method and demonstrate extension to different boundary conditions. Additional considerations and possible extensions are discussed in section 5.

2. Description of method. The Ewald decomposition is often viewed through the construction of a screen potential to define the corresponding short-range and long-range potentials. The usual PME formulation is consistent with the original Ewald decomposition in that it uses a Gaussian screen function

$$\rho_i^{\mathrm{G}}(\mathbf{x}) = Q_i \rho_0 e^{-a^2|\mathbf{x}-\mathbf{x}_i|^2}. \tag{2.1}$$

This screen, as depicted in Figure 1, yields a short-range potential so that $\Phi^{\mathrm{sr}} \propto \mathrm{erfc}(a|\mathbf{x}-\mathbf{x}_i|)/|\mathbf{x}-\mathbf{x}_i|$, which is straightforward to compute. The resulting mesh potential satisfies the Poisson problem,

$$-\nabla \cdot \nabla \Phi^{\mathrm{sm}} = \sum_i \rho_i(\mathbf{x}), \tag{2.2}$$

which is then optimally solved using FFTs on a mesh.

We instead propose screening functions $\rho_i(\mathbf{x})$ that are order q piecewise polynomials on the mesh, as shown in Figure 2.

The screens are thus represented exactly by polynomial finite element basis functions of order $\geq q$. The corresponding potential is then solved with (2.2) using a finite element method with basis functions of order $p = q + 2$.

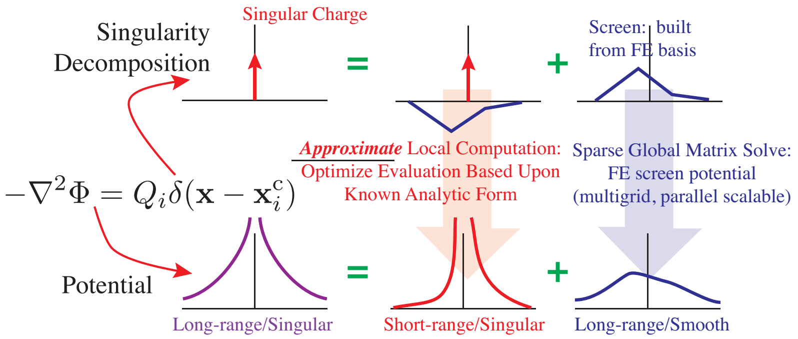

Singular Charge
Singularity Decomposition
=
+
Screen: built from FE basis
Approximate Local Computation: Optimize Evaluation Based Upon Known Analytic Form
Sparse Global Matrix Solve: FE screen potential (multigrid, parallel scalable)
$-\nabla^2\Phi = Q_i\delta(\mathbf{x}-\mathbf{x}_i^{\mathrm{c}})$
Potential
=
+
Long-range/Singular
Short-range/Singular
Long-range/Smooth

FIG. 2. *Introduction of a polynomial screen and resulting decomposition of the potential of the singular charge.*

Next, we describe the details of the method, following the four basic steps of P^{3}M methods: Assignment of charges to the mesh, solving for the smooth potential on the mesh, transferring the potential back to the charge locations, and calculating the point-to-point (short-range) interactions. A high-level synopsis of the algorithm is described in Algorithm 1 to illustrate the structural pieces of our approach.

We assume a collection of N charges $\mathcal{Q} \equiv \{Q_i\}_{i=1}^N$ located at $\mathbf{x}_i^{\mathrm{c}}$ in a cube $\Omega = [0, L]^3$ (see Figure 3). A mesh with $N_{\mathrm{el}} = n_{\mathrm{el}}^x \times n_{\mathrm{el}}^y \times n_{\mathrm{el}}^z$ elements is constructed to conform to the domain, and a uniform mesh is assumed in each direction for simplicity of presentation—i.e., $n_{\mathrm{el}} = n_{\mathrm{el}}^x = n_{\mathrm{el}}^y = n_{\mathrm{el}}^z$. Finally, the collocation points for q-order basis functions on the mesh are denoted $\mathbf{x}_j^{\mathrm{m}}$, with $j = 1, \ldots, M \equiv (qn_{\mathrm{el}} + 1)^3$.

Algorithm 1: Polynomial screen method for calculating potential.

```
Input: A mesh of elements $e_j$ and a group of point charges $Q_i$
Return: Potential at locations of charges
for each charge $Q_i$                                   {charge assignment, section 2.1}
    place $Q_i$ in element
    solve for screen                                    {see (2.10) or (2.19)}
for each element $e_j$
    if element $e_j \in$ surface
        adjust boundary conditions as necessary         {see (2.26)}
    apply charge assignment operator to form $\rho_\mathrm{m}$   {see (2.9)}
perform multigrid solve of $-\nabla^2\Phi^\mathrm{sm} = \rho_\mathrm{m}$     {see (2.2)}
for each charge $Q_i$                                   {evaluations, sections 2.3 and 2.4}
    $\Phi_i \leftarrow \Phi_i^\mathrm{sr} + \Phi_i^\mathrm{sm}$             {mesh-to-charge assignment}
```

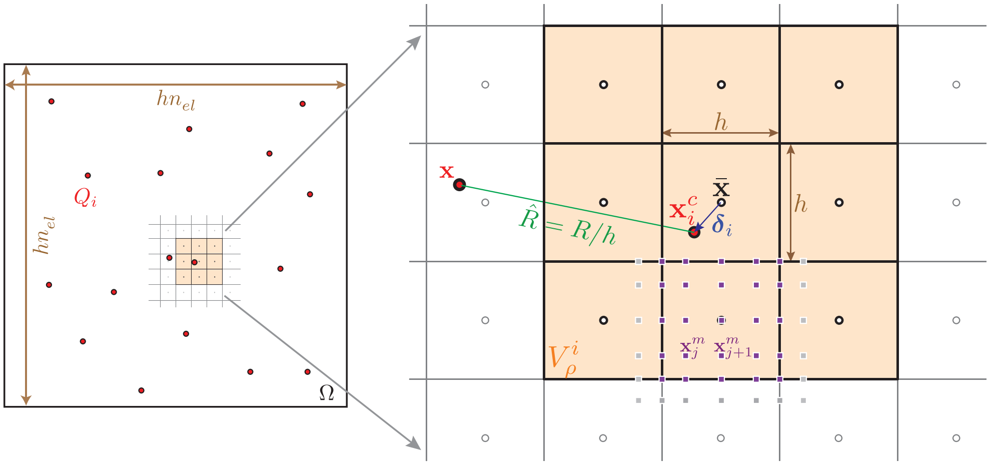


FIG. 3. *Schematic configuration showing N charges of strength Q_i at locations $\mathbf{x}_i^c$ distributed in the cubic domain Ω of size $hn_{el} \times hn_{el} \times hn_{el}$, where h is the edge length of the cubic finite elements. The cell centers are $\bar{\mathbf{x}}$ and the collocation points are $\mathbf{x}_j^m$.*

2.1. Charge assignment.

A central component of particle-mesh methods is the assignment of singular charges to the mesh, yielding a mesh-based charge density function, $\rho_\mathrm{m}(\mathbf{x}^\mathrm{m})$. In particular, we seek an assignment function $\mathcal{W}(\mathbf{x})$ that reflects our specially selected screen functions and provides a weighting that distributes a charge Q_i at $\mathbf{x}_i^\mathrm{c}$ to each collocation point $\mathbf{x}_j^\mathrm{m}$ of the basis functions:

$$\rho_\mathrm{m}(\mathbf{x}_j^\mathrm{m}) = \sum_{i=1}^{N} Q_i \mathcal{W}(\mathbf{x}_j^\mathrm{m}; \mathbf{x}_i^\mathrm{c}). \tag{2.3}$$

Existing methods use Lagrange polynomials (PME [5]) or B-splines (P^{3}M [16] and SPME [7]) for this weighting, where B-splines work particularly well with FFTs. The charge assignment function impacts both accuracy and efficiency of the method. In our approach we design an assignment operator based directly on polynomial basis functions for compatibility with a finite-element-based Poisson solver.

2.1.1. Defining the polynomial screens. We define our screen density function for a single charge $Q_i \in \mathcal{Q}$ as

$$\rho_i(\mathbf{x}) = \sum_j c_j \psi_j(\mathbf{x}), \tag{2.4}$$

with linear superposition providing the extension to multiple charges. As suggested by the form of (2.3), we will consider unit charges in our rules for screen construction (defining $\mathcal{W}(\mathbf{x})$), with the result then scaled by any charge strength Q. Here $\psi_j(\mathbf{x})$ are a collection of q-order Lagrange basis functions over an index set determined as follows. If charge Q_i is located within element τ_j of the mesh, we choose $V_\rho^i = \cup_{\tau\cap\tau_j\neq 0}\tau$ to be the interpolation support of the charge assignment operator. That is, the support is the union of the element of the mesh that includes the charge along with all neighboring elements, leading to a support of 27 elements in three dimensions. Generalization to other choices for this support is briefly discussed in section 5. To construct the polynomial screen, we consider the degrees of freedom which are interior to or on the faces of the element containing the charge. For q-order interpolating polynomials, this leads to $\dim(V_\rho^i) = (q+1)^3$ degrees of freedom. These degrees of freedom are determined so that the charge-screen combination has a potential that decays rapidly in space by considering the multipole expansion of the screen for a point well outside the screen, given by

$$\Phi_i^{\mathrm{sc}}(\mathbf{x}) = \frac{1}{\hat{R}}\int_{V_\rho^i} \rho_i(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} - \frac{1}{\hat{R}^2}\int_{V_\rho^i} \rho_i(\boldsymbol{\xi})\;(\boldsymbol{\xi}\cdot\hat{\mathbf{r}})\;\mathrm{d}\boldsymbol{\xi} + \frac{1}{2\hat{R}^3}\int_{V_\rho^i} \rho_i(\boldsymbol{\xi})[3(\boldsymbol{\xi}\cdot\hat{\mathbf{r}})^2 - |\boldsymbol{\xi}|^2]\,\mathrm{d}\boldsymbol{\xi} + \cdots, \tag{2.5}$$

where $\hat{\mathbf{R}} = (\mathbf{x}-\mathbf{x}_i)/h$, where $\mathbf{x}$ is the observation point, $\mathbf{x}_i$ is the charge location, h is the mesh size, and $\hat{R} = |\hat{\mathbf{R}}|$. The quantity $\hat{\mathbf{r}}$ is the unit direction vector $\hat{\mathbf{R}}/\hat{R}$.

For a charge Q_i located at $\mathbf{x}_i = (x_i, y_i, z_i)$ in element τ_j (see Figure 3; in the interest of readability, we drop superscript "c" when it is clear that the reference is to a charge location), we denote the offset $\boldsymbol{\delta}_i = (\delta_i^x, \delta_i^y, \delta_i^z) = \mathbf{x}_i - \bar{\mathbf{x}}_j$ with respect to the center of the element $\bar{\mathbf{x}}_j$, and define the (l,m,n)-moment and *centered* (l,m,n)-moment of the screen function as

$$\rho_i^{(l,m,n)} = \int_{V_\rho^i} (x-\delta_i^x)^l (y-\delta_i^y)^m (z-\delta_i^z)^n \rho_i(\mathbf{x})\,\mathrm{d}\mathbf{x}, \tag{2.6}$$

$$\bar{\rho}_i^{(l,m,n)} = \int_{V_\rho^i} x^l y^m z^n \rho_i(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{2.7}$$

If we take the origin as the element center $\bar{\mathbf{x}}_j$, dividing $\rho_i^{(0,0,0)} = \int_{V_\rho^i} \rho_i(\mathbf{x})\,\mathrm{d}\mathbf{x}$ by $\hat{R} = |\mathbf{x}-\mathbf{x}_i|/h$ gives the first term of the screen's multipole expansion from (2.5). Thus, requiring $\rho_i^{(0,0,0)} = 1$ (for a unit charge) guarantees that the combined point-charge and screen have a potential that decays at least as fast as $1/R^2$ with distance from the point charge. Likewise, zeroing higher moments of the screen enforces the cancellation of dipole and higher-order terms, making the far-field behavior of the screen follow the behavior of a point charge by increasing its symmetry. This accelerates the long-range decay rate of the point-charge-and-screen combination and reduces the number of interactions that must be explicitly represented by point-to-point computations. With the available degrees of freedom, a screen of order q cancels all terms up to $R^{-(q+1)}$, leaving $\Phi^{\mathrm{sr}} = 1/R - \Phi^{\mathrm{sc}} \sim R^{-(q+2)}$ for a unit charge. This is summarized in Table 1, which shows the moments that result from performing the vector operations in the integrands of (2.5).

Table 1
Polynomial terms in multipole expansion.

Power of R	Single terms	Mixed terms
R^{-1}	1	—
R^{-2}	x, y, z	—
R^{-3}	x^2, y^2, z^2	xy, xz, yz
R^{-4}	x^3, y^3, z^3	$x^2y, xy^2, x^2z, xz^2, y^2z, yz^2, xyz$
$\vdots$	$\vdots$	$\vdots$
R^{-N}	$x^{N-1}, y^{N-1}, z^{N-1}$	$x^l y^m z^n$, with $1 \le l,\ m,\ n \le N-2$, and $l+m+n = N-1$

Constructing the screen. The goal is to perform multipole cancellations with screens that are also compatible with the basis functions of our finite element discretization. With this description, each screen is composed of $N_{\text{sc}} = n_{\text{sc}}^3 = (q+1)^3$ nodal screen basis functions, ψ:

$$\rho_i(\mathbf{x}) = \sum_{j=0}^{N_{\text{sc}}-1} c_j \psi_j(\mathbf{x}). \tag{2.8}$$

Thus, revisiting (2.3), the assignment operator $\mathcal{W}$ is

$$\mathcal{W}(\mathbf{x}_j^{\text{m}}; \mathbf{x}_i^{\text{c}}) = \begin{cases} \sum_{k=0}^{N_{\text{sc}}-1} c_k(\boldsymbol{\delta}_i)\psi_k(\mathbf{x}_j^{\text{m}}), & \mathbf{x}_j^{\text{m}} \in V_\rho^i, \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

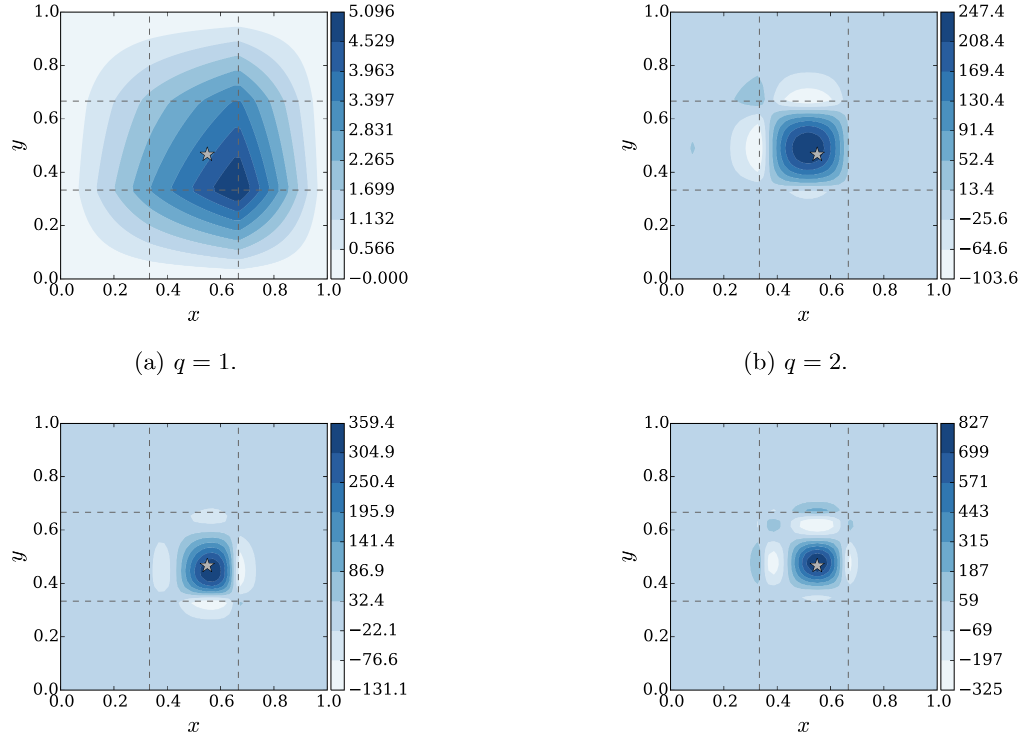
Restricting the first N_{sc} moments leads to a $N_{\text{sc}} \times N_{\text{sc}}$ linear system for the coefficients $\mathbf{c}$ in (2.8):

$$\begin{bmatrix} \psi_0^{(0,0,0)} & \psi_1^{(0,0,0)} & \cdots & \psi_{N_{\text{sc}}-1}^{(0,0,0)} \\ \psi_0^{(1,0,0)} & \psi_1^{(1,0,0)} & \cdots & \psi_{N_{\text{sc}}-1}^{(1,0,0)} \\ \vdots & & \ddots & \vdots \\ \psi_0^{(q,q,q)} & \psi_1^{(q,q,q)} & \cdots & \psi_{N_{\text{sc}}-1}^{(q,q,q)} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{N_{\text{sc}}-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.10}$$

where $\psi_j{}^{(l,m,n)}$ is the (l,m,n)-moment, as defined in (2.6) for ρ, of the jth screen basis function $\psi_j(\mathbf{x})$.

Figure 4 shows cross-sections of example screens constructed using $q = 1, \ldots, 4$. Each screen's peak is attained near the marked charge, and the screens are constructed to decay to zero at the edge of V_ρ. The screens have support in the active screen region V_ρ^i and for $q > 1$, the screens are in general nonmonotone.

We confirm in Figure 5 that screens constructed in this fashion yield potentials with the expected behavior: in all cases, the far-field behavior approximates $1/R$. In Figure 6, we estimate mean and peak errors incurred for point-to-point interaction truncation at a distance $\hat{R}_c = R_c/h$. To do this, the short-range potential is constructed for $N = 84$ charge locations within an element and sampled in 42 directions; the behavior of $|\Phi^{\text{sr}}|$ is shown in the figure as the weighted average $|\Phi^{\text{sr}}|_{\text{avg}}$ and the maximum from all samples $|\Phi^{\text{sr}}|_{\max}$. In addition, the radius from a charge location is also normalized as $\hat{R} = R/h$, and is denoted $\hat{R}$ (see Figure 3). We see that twice



(a) $q = 1$.

(b) $q = 2$.

(c) $q = 3$.

(d) $q = 4$.

FIG. 4. *Example of linear through quartic screens for a single charge (marked by a star).*

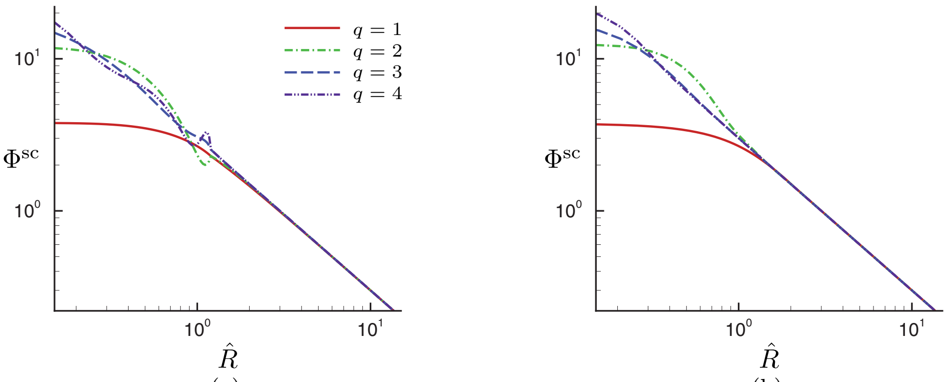
the screen-size scale $\hat{R} \approx 3$ corresponds to the expected start of the asymptotic decay behavior. This is the distance at which a multipole expansion is generally considered "well-separated" and expected to show convergence with R. Both the mean and peak errors show the expected behavior for increasing q beyond this distance.

A fast algorithm for screen construction. Solving (2.10) directly requires $\mathcal{O}(N_{\text{sc}}^3)$ operations for each screen, which is feasible, but is not necessary in general. In the following, we design a fast algorithm for computing screens, which follows from a generalization of the parallel axis theorem applied to moments used in the system. First, we note that the moments are additive. For example, the first moment in variable x of basis function ψ_j satisfies

$$\begin{aligned} \psi_j^{(1,0,0)} &= \int_{V_\rho} x\psi(\mathbf{x})\,\mathrm{d}\mathbf{x} - \delta_i^x \int_{V_\rho} \psi(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= \bar{\psi}_j^{(1,0,0)} - \delta_i^x \psi_j^{(0,0,0)}, \end{aligned} \tag{2.11}$$

where $\bar{\psi}_j\,^{(1,0,0)}$ is the *centered* moment of the jth basis function as in (2.7). Therefore, the second row of (2.10) is equivalent to

$$\sum_{j=0}^{N_{\text{sc}}-1} c_j \bar{\psi}_j^{(1,0,0)} = \delta_i^x \sum_{j=0}^{N_{\text{sc}}-1} c_j \psi_j^{(0,0,0)} = \delta_i^x. \tag{2.12}$$



(a)

(b)

FIG. 5. *Screen potential in two directions:* (a) $\mathbf{x} - \mathbf{x}_i \propto (1,0,0)$ *and* (b) $\mathbf{x} - \mathbf{x}_i \propto (1,1,1)$.

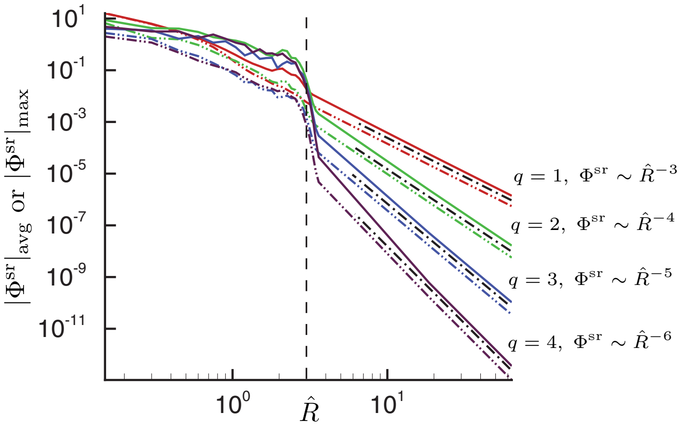


FIG. 6. *Maximum (solid) and average (dash-dot-dot) short-range potentials for screens with* $q = 1$ *to* 4*. The straight dash-dot lines denote the expected slopes.*

Similarly, the second moment of ψ_j satisfies

$$\begin{aligned}\psi_j^{(2,0,0)} &= \int_{V_\rho} x^2 \psi(\mathbf{x})\ \mathrm{d}\mathbf{x} - 2\delta_i^x \int_{V_\rho} x\psi(\mathbf{x})\ \mathrm{d}\mathbf{x} + (\delta_i^x)^2 \int_{V_\rho} \psi(\mathbf{x})\ \mathrm{d}\mathbf{x} \\ &= \bar{\psi}_j^{(2,0,0)} - 2\delta_i^x \bar{\psi}_j^{(1,0,0)} + (\delta_i^x)^2 \psi_j^{(0,0,0)},\end{aligned} \tag{2.13}$$

so the third row of (2.10) becomes

$$\begin{aligned}\sum_{j=0}^{N_{\rm sc}-1} c_j \bar{\psi}_j^{(2,0,0)} &= 2\delta_i^x \sum_{j=0}^{N_{\rm sc}-1} c_j \bar{\psi}_j^{(1,0,0)} - (\delta_i^x)^2 \sum_{j=0}^{N_{\rm sc}-1} c_j \psi_j^{(0,0,0)} \\ &= 2(\delta_i^x)^2 - (\delta_i^x)^2 = (\delta_i^x)^2.\end{aligned} \tag{2.14}$$

Continuing this procedure for other moments in (2.10) yields

$$\begin{bmatrix} \psi_0^{(0,0,0)} & \psi_1^{(0,0,0)} & \cdots & \psi_{N_{\mathrm{sc}}-1}^{(0,0,0)} \\ \bar{\psi}_0^{(1,0,0)} & \bar{\psi}_1^{(1,0,0)} & & \\ \vdots & & & \vdots \\ \bar{\psi}_0^{(q,0,0)} & & \ddots & \bar{\psi}_{N_{\mathrm{sc}}-1}^{(q,0,0)} \\ \bar{\psi}_0^{(0,1,0)} & & & \\ \vdots & & & \vdots \\ \bar{\psi}_0^{(q,q,q)} & & \cdots & \bar{\psi}_{N_{\mathrm{sc}}-1}^{(q,q,q)} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \\ \vdots \\ \\ \\ c_{N_{\mathrm{sc}}-1} \end{bmatrix} = \begin{bmatrix} 1 \\ \delta_i^x \\ \vdots \\ (\delta_i^x)^q \\ \delta_i^y \\ \vdots \\ (\delta_i^x)^q(\delta_i^y)^q(\delta_i^z)^q \end{bmatrix}, \tag{2.15}$$

which we write compactly as $\mathcal{C}\mathbf{c} = \mathbf{f}$. An advantage of this form is that for a uniform mesh, the matrix $\mathcal{C}$ is the same for each screen, since the moments reference the center of the element. As a result, the matrix can be prefactorized, leading to a complexity of only $\mathcal{O}(N_{\mathrm{sc}}^2)$ to solve for each screen. Furthermore, this reduces setup cost by eliminating the need to calculate the moment integral entries of the matrix in (2.10) for every screen.

For small q this yields a small operation count, yet the computation is further reduced if $\psi(\mathbf{x})$ is separable, as is the case for the regular cubic mesh shown in Figure 3. In this case,

$$\psi_\kappa(\mathbf{x}) = \omega_i(x)\omega_j(y)\omega_k(z), \tag{2.16}$$

where ω_i are the one-dimensional nodal basis functions for a mesh size h and $\kappa = i + (q+1)j + (q+1)^2 k$ with $i, j, k \in [0, q]$. With ψ separable, the moment integrals are also separable:

$$\begin{aligned} \bar{\psi}_\kappa^{(l,m,n)} &= \int_{V_\rho^i} x^l y^m z^n \psi_\kappa(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= \left(\int_{-3h/2}^{3h/2} x^l \omega_i(x)\,\mathrm{d}x\right)\left(\int_{-3h/2}^{3h/2} y^m \omega_j(y)\,\mathrm{d}y\right)\left(\int_{-3h/2}^{3h/2} z^n \omega_k(z)\,\mathrm{d}z\right). \end{aligned} \tag{2.17}$$

Following the notation of (2.7), we define

$$\bar{\omega}_i^{(l)} = \int_{-3h/2}^{3h/2} x^l \omega_i(x)\,\mathrm{d}x \tag{2.18}$$

and likewise for the one-dimensional y and z centered moments. We take $c_\kappa = w_i^x w_j^y w_k^z$ and recognize that the right-hand side of (2.15) is also separable as $f_\mu = (\delta^x)^l(\delta^y)^m(\delta^z)^n$ with $\mu = l+(q+1)m+(q+1)^2 n$. This yields three equivalent $(q+1)^2$ systems of the form:

$$\sum_{i=0}^{q} \bar{\omega}_i^{(l)}(x) w_i^x = (\delta^x)^l, \quad l = 0, \ldots, q, \tag{2.19}$$

which are solved independently. The inverse of the matrix is computed once and applied for all right-hand sides, resulting in only $\mathcal{O}(n_{\mathrm{sc}}^2)$ operations per screen. This method also extends to regular rectangular meshes, where h is not necessarily equal in each direction.

2.2. Solution of the mesh potential. Given our construction of ρ_m through the screens ρ_i built from the finite element basis functions via (2.9), the solution of Φ^{sm} is straightforward. Since the screens are polynomial of degree q, we use a finite element solver with basis functions u^{m} of order $p = q+2$, which results in a symmetric, positive definite sparse linear system (under realistic assumptions regarding the boundary conditions).

Multigrid preconditioners are effective for this problem, even for high-order bases, and allow the sparse matrix problem to be solved to any level of accuracy. As an example, consider the case of a high-order finite element discretization of the Poisson problem with Dirichlet boundary conditions. Figure 7 shows the convergence history of a multigrid preconditioned conjugate gradient method for basis functions of order $p = 1$ through 6. An algebraic multigrid preconditioner, based on smoothed aggregation using a more general strength measure [22] and optimal interpolation operator [21], is used. We observe only a weak dependence on p. Moreover, more advanced multigrid techniques have shown still better scalings for both Poisson and other elliptic problems such as for Stokes flow [15, 20]. Importantly, multigrid preconditioners are also well known to exhibit high parallel efficiency [10, 1]. In the following tests, we use AMG through the BoomerAMG package [13].

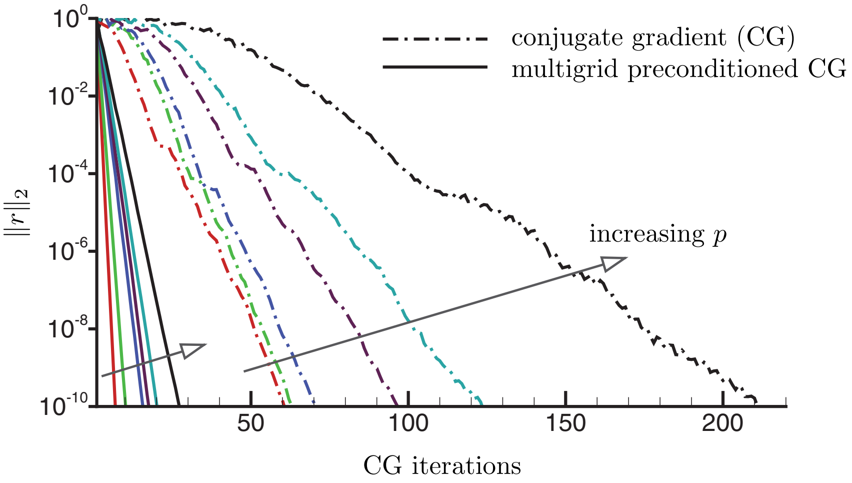


FIG. 7. *Convergence history for the Poisson problem using basis elements of order $p = 1, \ldots, 6$ for both the conjugate gradient method (dot-dashed) and multigrid preconditioning (solid).*

2.3. Evaluation of the smooth potential at charge locations. The next step is to evaluate Φ^{sm} at the charge locations. For standard P^{3}M and PME implementations, this involves interpolation with the Lagrangian or B-spline basis functions from the charge assignment. In contrast, our method requires no interpolation, though interpolation can be used to speed up calculations, if desired. Since the smooth potential exists in each element as a linear combination of coefficients—i.e., the values of Φ^{sm} at $\mathbf{x}^{\mathrm{m}}$ for all N_{m} points in an element—and the basis functions u^{m}, the smooth potential is expressed at any point as

$$\Phi^{\mathrm{sm}}(\mathbf{x}) = \sum_{j=1}^{N_{\mathrm{m}}} \Phi^{\mathrm{sm}}(\mathbf{x}_j^{\mathrm{m}}) u_j^{\mathrm{m}}(\mathbf{x}), \tag{2.20}$$

where N_{m} is the number of collocation points in an element. Direct evaluation at the charge locations $\mathbf{x} = \mathbf{x}_i^{\mathrm{c}}$ is straightforward.

2.4. Short-range potential. Our formulation for the mesh-based potential yields a more complex short-range interaction than PME. In addition to R, the short-range interaction now also depends on the position of the charge relative to the underlying mesh. Consequently, additional effort is required to evaluate the short-range interaction. However, the calculation is *local*, which is an added benefit in a manycore setting.

The short-range potential at point $\mathbf{x}$ due to a charge Q_i located at $\mathbf{x}_i$ is

$$\Phi_i^{\mathrm{sr}}(\mathbf{x}) = \frac{Q_i}{|\mathbf{x} - \mathbf{x}_i|} - \Phi_i^{\mathrm{sc}}(\mathbf{x}), \tag{2.21}$$

where

$$\Phi_i^{\mathrm{sc}}(\mathbf{x}) = Q_i \int_{V_\rho^i} \frac{\rho_i(\boldsymbol{\xi})}{|\mathbf{x} - \boldsymbol{\xi}|} \mathrm{d}\boldsymbol{\xi}. \tag{2.22}$$

Recall that Φ^{sm}, the smooth potential, includes the sum of all individual screen potentials Φ_i^{sc} and the boundary effects (here, the periodic images of the charges). Though feasible, performing accurate quadrature for each screen individually is computationally expensive. We therefore shift a significant portion of this computational effort to a preprocessing step, for which there are multiple options.

One approach is to consider a look-up table of precomputed values for the screen potential evaluated at $\mathbf{x}_j$ due to a charge at $\mathbf{x}_i^{\mathrm{c}}$. These values comprise a six-dimensional look-up table, $\Phi^{\mathrm{sc}}(\mathbf{x}_j - \mathbf{x}_i^{\mathrm{c}}; \boldsymbol{\delta}_i)$, since they are a function of the difference between the evaluation point and the charge location, and also the offset of the charge within its element (which determines the screen).

With some additional computation, but still without resorting to direct evaluation of (2.22), it is possible to remove the charge offset interpolation to reduce errors. We accomplish this by recognizing the screen's formulation as a linear combination of basis functions,

$$\Phi_i^{\mathrm{sc}}(\mathbf{x}) = Q_i \sum_{j=0}^{N_{\mathrm{sc}}-1} c_j(\boldsymbol{\delta}_i) \int_{V_\rho^i} \frac{\psi_j(\boldsymbol{\xi})}{|\mathbf{x} - \boldsymbol{\xi}|} \mathrm{d}\boldsymbol{\xi} \tag{2.23}$$

$$= Q_i \sum_{j=0}^{N_{\mathrm{sc}}-1} c_j(\boldsymbol{\delta}_i) \Phi_j^{\mathrm{b,sc}}(\mathbf{x} - \mathbf{x}_i^{\mathrm{c}}). \tag{2.24}$$

This approach yields N_{sc} look-up tables for basis-function potential values $\Phi^{\mathrm{b,sc}}(\mathbf{x}_j - \mathbf{x}_i^{\mathrm{c}})$. However, for $q \geq 2$ the polynomial nature of the screen leads to nonmonotonic decay for some directions within the region where the screen is active, as shown in Figure 5. Consequently, a direct implementation of a look-up table for such functions requires sufficient resolution, which is harder to achieve for larger q. In contrast to Ewald-based methods, where short-range computations are efficient by design, the cost of these look-up tables may be nontrivial. Therefore, for good performance, knowledge of the underlying structure of the screen potentials should be used to inform both the storage locations for the look-up table values and the interpolation method.

2.5. A note about the self term. If the point $\mathbf{x}$ is the location of a charge, we do not wish to include the potential due to this charge in our calculation. However, we do still need to subtract the screen potential from the charge's own screen, which is included in Φ^{sm}; this is sometimes called the "self" term. We can allow for this by amending our short-range potential expression to include both cases:

$$\Phi_i^{\mathrm{sr}}(\mathbf{x}) = \begin{cases} \frac{Q_i}{|\mathbf{x}-\mathbf{x}_i|} - \Phi_i^{\mathrm{sc}}(\mathbf{x}), & |\mathbf{x}-\mathbf{x}_i| > 0, \\ -\Phi_i^{\mathrm{sc}}(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{2.25}$$

2.6. Alternative boundary conditions. The formulation above is presented under the assumption of periodic boundary conditions, which is the simplest case and important for a range of applications. It is straightforward to generalize boundary conditions via the mesh potential Φ^{sm}. This is accomplished by adjusting for short-range effects present at the boundary and then proceeding in the usual manner for a finite element problem with the given type of boundary conditions. For example, for a Dirichlet boundary condition of $\Phi = g$ on ∂V, the condition for our mesh problem becomes

$$\Phi^{\mathrm{sm}}|_{\partial V} = g - \Phi^{\mathrm{sr}}|_{\partial V}, \tag{2.26}$$

which leads to

$$\Phi|_{\partial V} = \Phi^{\mathrm{sm}}|_{\partial V} + \Phi^{\mathrm{sr}}|_{\partial V} = g. \tag{2.27}$$

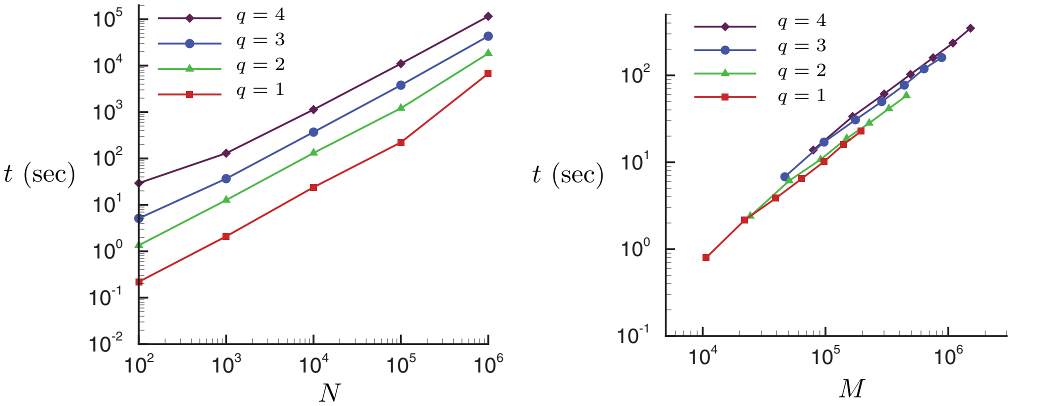
A similar approach is used in [14]. This also extends to the case of Neumann or mixed-type boundary conditions, with the usual constraint to address the nonuniqueness of the fully Neumann problem. However, free-space conditions require careful treatment, as in any mesh-based discretization scheme.

3. Performance model.

The computational cost of the method for N charges is formulated as $\mathcal{O}(N)+\mathcal{O}(M)$, where $M = (pn_{\mathrm{el}}+1)^3$ is the total number of degrees of freedom in the mesh. Example CPU time scalings for the major N-related components is illustrated in Figure 8a, with the M-dependent mesh solve times shown in Figure 8b. Given N and M and assuming on average $N_{\mathrm{el}}^{\mathrm{sr}}$ neighboring elements in the short-range interaction list for each charge, then it is possible to express the coefficients in the linear $\mathcal{O}(N)+\mathcal{O}(M)$ operation count in terms of p. Such a formulation provides a more detailed description of the actual costs of each component of the method and their relationships to the order of the screens.

3.1. Breakdown of costs. Screens $\rho_i(\mathbf{x})$ of order q are built out of $(q+1)^3 = N_{\mathrm{sc}}$ basis functions—recall that $p = q+2$. The corresponding finite element solve associated with these screens involves $(p+1)^3$ degrees of freedom per element and a total of M degrees of freedom. We also define the average number of charges per element as $\tilde{N} = N/N_{\mathrm{el}}$.

3.1.1. Screen construction. For each evaluation, the element containing each charge is identified, and the offsets from the center of these elements determined. This incurs a small $\mathcal{O}(N)$ cost, which we designate $C_1 N$. The screen coefficients are then



(a) Total CPU time vs. N for the main N-dependent components of the algorithm, including screen creation, short-range calculation, basis function evaluation, and combination of short-range and mesh potentials at charge locations.

(b) CPU time to solve for mesh potential versus number of mesh points M. The number of elements per coordinate direction is varied from 7 to 19.

FIG. 8. *Total CPU time for* (a) *N-related components, and* (b) *mesh solve (M-related).*

calculated. As shown in (2.19), assuming precomputed inverses, this amounts to three matrix-vector multiplications of size $q + 1 = p - 1$, for a cost of $6(p-1)^2 - 3(p-1)$. We then multiply the one-dimensional weights, resulting in two additional floating point multiplications. The total cost for determining the screen coefficients is thus

$$\text{(screen construction)} \qquad \sim [6(p-1)^2 - 3(p-1) + 2(p-1)^3]N.$$

3.1.2. Short-range potential. The cost of evaluating the short-range potential depends on the method chosen for calculating Φ^{sc}, as discussed in section 2.4. In addition, there is a cost of $\mathcal{O}(N)$ due to the singular part of the short-range calculations, which we denote SN. For a general six-dimensional look-up table, the cost of calculating Φ^{sc} at a point due to all charges in the short-range interaction volume is $C_2\tilde{N}N_{\text{el}}^{\text{sr}}$, where C_2 depends on the order of interpolation used. If N_{sc} three-dimensional look-up tables are used, as we have done in the example calculations of section 4, then the interpolation is repeated for $(q+1)^3$ tables and combined by an inner product with the screen coefficients and a multiplication by Q_i for a total of $[C_2(p-1)^3 + 2(p-1)^3]\tilde{N}N_{\text{el}}^{\text{sr}}$. The cost for the short-range calculation is then

$$\text{(point-to-point evaluation)} \qquad \sim SN + [C_2(p-1)^3 + 2(p-1)^3]\tilde{N}N_{\text{el}}^{\text{sr}}N.$$

Since $\tilde{N} = N/N_{\text{el}}$, this expression is also written in terms of N^2. However, we assume that in practice, $\tilde{N}N_{\text{el}}^{\text{sr}}$ is chosen to be small enough to render this effectively as $\mathcal{O}(N)$. Furthermore, if $\tilde{N}$ is > 1, this cost is reduced further by calculating the effects of all charges in an element at once in an "element-to-point" operation. To do this, we compile a combined list of $Q_i\mathbf{c}_i$ for all the charges in any given element, so that the screen potential for this sum at a point as calculated by (2.24) is the same as if the charges were handled individually. The cost then is then reduced by a factor of $\tilde{N}$ yielding

$$\text{(element-to-point evaluation)} \qquad \sim SN + [C_2(p-1)^3 + 2(p-1)^3 - 1]N_{\text{el}}^{\text{sr}}N.$$

3.1.3. Mesh solve. The "transfer" of the order-q screens to a representation in order $p = q + 2$ basis functions by (2.9) to construct the source ρ_{m} in the right-hand side of the finite element solve (2.2) requires an inner product between a vector containing the screen coefficients $\mathbf{c}(\boldsymbol{\delta})$ with the evaluation of the order-q basis functions at the collocation points, followed by a multiplication by Q_i. This is done at each degree of freedom within an active screen area, for a total of $(3p+1)^3 \times [2(p-1)]N$ operations. The multigrid solve for the finite element problem is $\mathcal{O}(M)$, with a coefficient C that depends on the convergence of the iterations, but is considered low in practice. Overall the mesh solve thus has complexity

$$\text{(mesh solve)} \qquad \sim \{(3p+1)^3 \times [2(p-1)^3]\}N + CM.$$

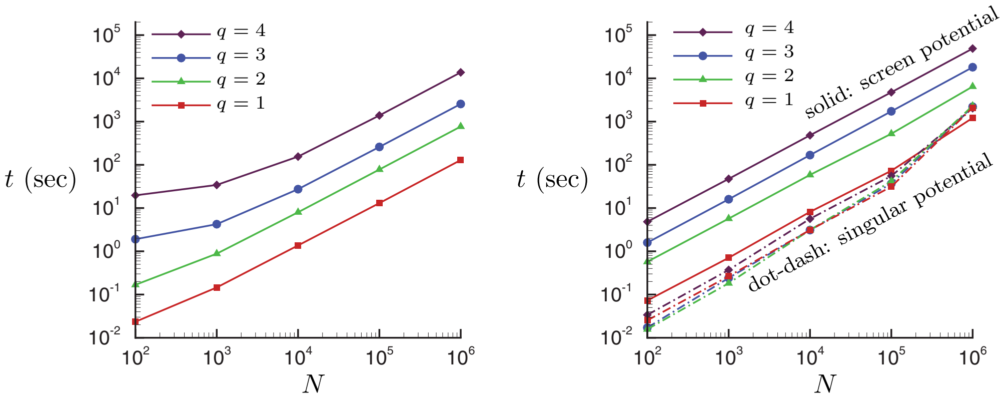
3.1.4. Evaluation. The smooth potential is written as a combination of basis functions at the location of each charge, as in (2.20). Thus evaluation involves $(p+1)^3$ basis functions at a cost of $2p$ operations for each function. However, empirically we find that this cost is minimal in terms of CPU time.

3.2. Summary. The screen creation (mostly due to the "transfer" portion) and short-range interaction calculations are the most costly even for modest values of p given the scaling shown above. The relative costs of these two portions of the algorithm depend on choices in short-range calculation method, mesh size, and q. For any given cutoff error, decreasing mesh spacing decreases the number of short-range interactions, but results in an increased number of collocation points M in the mesh solve. Likewise, increasing q also decreases the number of short-range interactions, but at the price of the increased cost of constructing and manipulating screens for larger q. Calculating the short-range effects of each individual charge becomes more costly with increased q, though at a slower rate than the transfer. The scaling of these components is shown in Figure 9 for cases of $N = 10^2$ to 10^6 randomly distributed particles in a triply-periodic box with 6 859 elements. The tests were done using a dual, quad-core Intel Xeon E5506 CPU with 48 GB of main memory. It is noted that once $\tilde{N} \gg 1$, the singular short-range calculation loses its linearity in N. However, the screen potential portion of the short-range calculation retains its linearity due utilization of the "element-to-point" evaluation method.

4. Example calculations.

4.1. Periodic test case. We first consider cases with N ranging from 10^2 to 10^5 unit charges placed in a triply-periodic unit cube of elements with $h = 0.06\overline{6}$. The exact positions are selected randomly, but distributed so that any given charge experiences both long-range interactions, on the scale of the overall periodic domain size, and short-range interactions of comparable magnitude. This is done to provide a balanced test of both the short-range and smooth portions of our decomposition. To achieve this, the charges are randomly distributed within two smaller cubes: $[0, 1/2]^3$ is biased toward positive charges, 55% to 45%, and $[1/2, 1]^3$ is biased equally strongly toward negative charges. This setup is visualized in Figure 10a for $N = 100$.

The potential is then calculated using a short-range interaction of $7 \times 7 \times 7 = 343$ elements (corresponding to a minimum possible value of 3 for the cutoff distance $\hat{R}_c$)



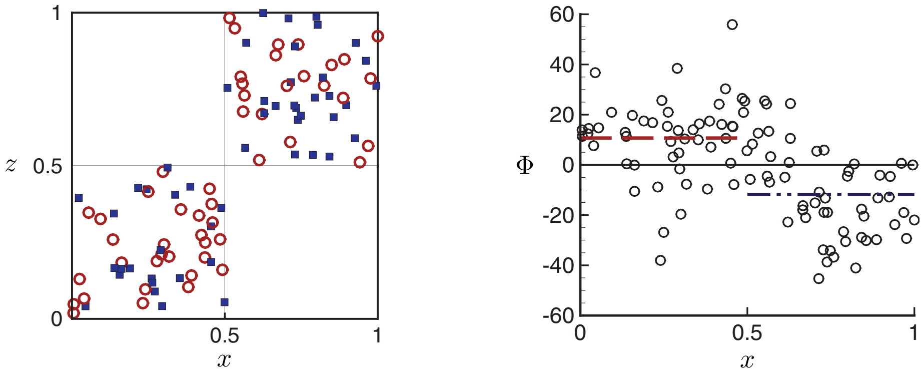
(a) (b)

FIG. 9. *Example CPU time vs. N for the two most costly local portions of the algorithm:* (a) *creation of the screens,* (b) *calculation of short-range interactions for* $N_{el}^{sr} = 7 \times 7 \times 7$. *The dot-dash lines show the time associated with calculating singular potentials, while solid lines show times calculating element-to-point screen potentials. At large N, there is an expected breakdown in linearity for the singular potential calculations.*

for linear through quartic screens. This short-range cutoff is chosen to ensure that the short-range potential of every charge near the cutoff exhibits asymptotic behavior. The short-range calculation uses the approach of (2.24), with N_{sc} look-up tables.

Remark 1. In our current implementation, we use a variation (but equivalent form) to this construction, in which the values stored in the tables are for "basis screens" instead of screen basis functions. These basis screens, ρ^{basis}, are the polynomial screens associated with each node in an order-q finite element. The values of each table are computed as a Dirichlet finite element solution for Poisson's equation, with $-\nabla^2\Phi_i^{\mathrm{basis}} = \rho_i^{\mathrm{basis}}$. The computation is completed in a domain larger than the size that will be kept in the look-up table to minimize boundary effects. Because these basis screens follow our moment-canceling rules, they have long-range decay $\sim \hat{R}^{-(q+2)}$, and the boundary conditions are accurately set by the first terms of the multipole expansion (2.5). The finite element solver uses basis functions of order p, and the look-up tables are stored in terms of their order-p basis functions, allowing them to be evaluated and combined in the same way as Φ^{sm} for all charge locations. We note that because the number of tables and coefficients is unchanged, the computational complexity for the short-range calculation is not altered by this variation.

Upon calculation the potential is compared, allowing for a constant which is included in a potential and in this case is equal to the average value of Φ^{sm} throughout the computational domain, with that of an Ewald summation Φ^{E} with large enough resolution that we consider it the "exact" solution. This uses two periodic images in physical space with $a^2 = 6.25$ and four modes for each direction in the Fourier sum. As we see for a representative calculation in Figure 11, the method has superalgebraic convergence with q. It is important to note that convergence in this case with respect to mesh spacing, h, is not easily predicted. Indeed, since decreasing h with a fixed number of charges N results in few charges per element (on average), the short-range error has fewer charges contributing to its total error in the case of a mesh based screen. Similarly, extending the short-range interaction more broadly would result in little impact from changing h.



(a) (b)

FIG. 10. *Configuration of demonstration calculation:* (a) *Location of* ○ *positive and* □ *negative charges for* $N = 100$, (b) Φ *at each charge location for the* 100 *charge case, arranged by the charge's location in* x*; the red dashed and blue dot-dashed lines mark the average potential value at locations in the positively biased group and negatively biased group, respectively. (Figure appears in color electronically only.)*

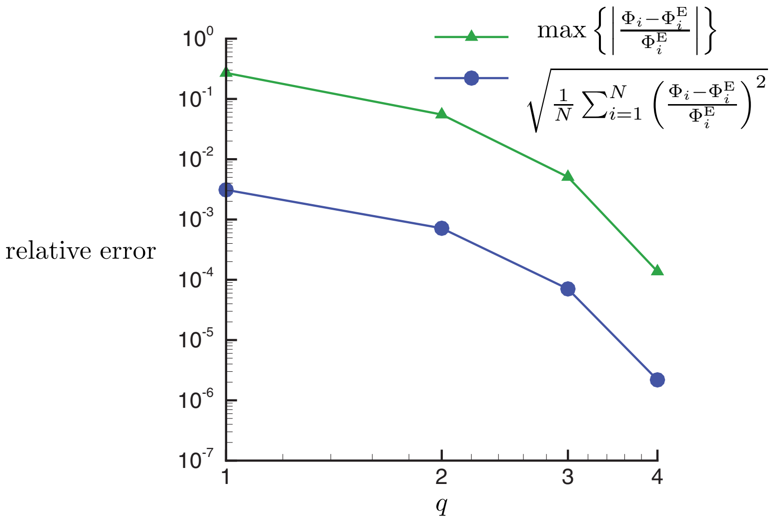


FIG. 11. *Convergence of the max and RMS measure of relative error in* Φ *versus* q *for a sample case with* $N = 10\,000$.

In each of these tests, we use an algebraic multigrid preconditioned GMRES solver with single precision residual tolerance—i.e., $1e-7$. BoomerAMG [13] is used and the resulting method yields eight or fewer iterations in each of the tests reported above. The timing dependence on mesh size is reported in Figure 8b, where we see that the solver exhibits $\mathcal{O}(M)$ scaling.

4.2. Estimated memory requirements. The memory requirements of the method in our example calculations are classified as finite element matrices or particle-related arrays. As the number of elements increases, the finite element matrices comprise a majority of the total allocated memory, as demonstrated in the following for the case of $N = 10^6$:

Order of	Particle	FE matrices	FE matrices
screens	arrays	($N_{\text{el}} = 9 \times 9 \times 9$)	($N_{\text{el}} = 25 \times 25 \times 25$)
linear ($q = 1$):	184 MB	39 MB ($\approx$ 18%)	844 MB ($\approx$ 81%)
quartic ($q = 4$):	1120 MB	1290 MB ($\approx$ 52%)	27 600 MB ($\approx$ 95%)

As the polynomial order increases—e.g., $q = 4$, which corresponds with a 6th order basis for the finite element solve—this effect increases as expected. We note that this memory footprint is typical for high-order FEM, but more optimal methods do exist [18].

4.3. Extension to Dirichlet boundary conditions. We now demonstrate the modification for Dirichlet boundary conditions described in section 2.6. For these tests, we consider the unit cube with particles uniformly distributed within the smaller cube defined by $[0.1, 0.9]^3$. The particles are 55% positive and 45% negative; a net charge is allowed with Dirichlet boundary conditions, unlike the periodic problem. We first consider homogeneous boundary conditions on all faces of the domain. In this case, an exact solution is defined by the method of images. In order to compute this result, we turn the Dirichlet problem in $[0, 1]^3$ into a periodic problem in $[-1, 1]^3$. This periodic problem has eight times as many particles as the original problem, with seven sets of image charges chosen to have the proper charge strengths and reflected coordinates required for the method of images. We then solve with a high-resolution Ewald solver as before. The errors for the homogeneous Dirichlet case are shown in Table 2 for $N = 10\,000$. In addition, the values for the maximum relative errors in the $N = 10\,000$ case from the previous periodic test (section 4.1 and Figure 11) are shown for comparison. We see similar errors for both test cases.

Table 2

Errors for the homogeneous Dirichlet and periodic test cases with $N = 10\,000$, $h = 0.06\bar{6}$, and screen order ranges from $q = 1$ to $q = 4$.

	Maximum relative error in FE-based method	
q	Homogeneous Dirichlet (sect. 4.3)	Periodic (sect. 4.1)
1	9.579×10^{-2}	2.701×10^{-1}
2	1.927×10^{-2}	5.507×10^{-2}
3	2.983×10^{-3}	5.077×10^{-3}
4	1.917×10^{-4}	1.464×10^{-4}

Next, we consider a nonhomogeneous Dirichlet boundary condition, where $\Phi = g$ on ∂V and g is

$$g(\mathbf{x}) = \begin{cases} 2500\,[1 + 0.9\sin(2\pi y)\sin(\pi z)] & \text{when x} = 0, \\ 2500 & \text{otherwise.} \end{cases}$$

The particle distribution is the same as the homogeneous Dirichlet case (the schematic of this case is depicted in Figure 12, with the gray box indicating the region where the particles are placed). The method of images alone can no longer solve this problem. Any method (Ewald-based, FMM, etc.) that relies on symmetry and images in order to implement different boundary conditions would not be able to complete this test on its own, but would need to be coupled with a Laplace solver capable of handling general boundary conditions. Our method, in contrast, only requires changing the description of the boundary condition from the previous homogeneous case.

A slice of the resulting potential is shown on the right side of Figure 12 and we note the expected behavior. For example, the overall potential is positive, with the peak value near the center of the box. Moreover, the contours flatten in the particle-free zone around the edge of the domain, as there are no longer large fluctuations from very near particles. The gradient of the potential (the electric field) becomes perpendicular to the walls in the center of each of the three constant-valued edges, as expected along conducting surfaces. Finally, the sinusoidal boundary condition "pulls" the contour lines along the $x = 0$ face of the domain.

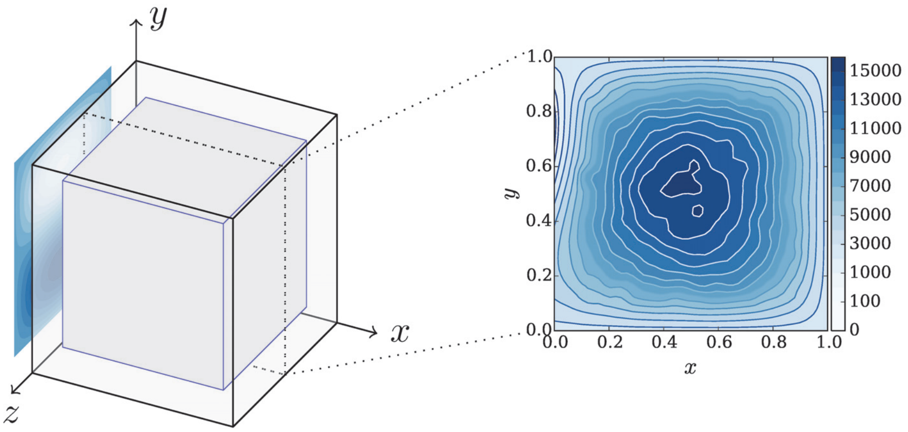


FIG. 12. *Schematic for nonhomogeneous Dirichlet test case. The charges are distributed within the gray box; the nonconstant boundary condition is shown at $x = 0$. A slice of the potential taken at $z = 0.5$ is shown on the right.*

5. Discussion.

5.1. Comparison to PME. While the method proposed here incorporates several advantageous features of PME, there are notable differences that may offer benefits in certain settings. Our method no longer relies on the FFT, which may be limiting at extreme scales (in comparison to other Poisson solvers) and forces an assumption of structure on the compute geometry. The key is the introduction of a mesh-based screen, which introduces additional complexities locally, but also allows for a more general decomposition of the problem. There are particle-mesh variants that use finite elements—e.g., some PIC methods [6, 23]—but these have been proposed with a symmetric screen, which must be resolved on the mesh. We avoid this approximation, but at the cost of more intricate screen functions, which are constructed with (and the resulting potentials evaluated by) using memory-local operations.

These new screens incur an $\mathcal{O}(p^6)$ cost when represented by p-order basis functions as discussed in section 3. This is larger than the $\mathcal{O}(\tilde{p}^3)$ cost of the $\tilde{p}$-order B-spline interpolations in PME. However, the polynomial order p in the present scheme and the B-spline order $\tilde{p}$ in PME are only loosely related. The B-spline order affects the overall accuracy of the PME method since it affects the resolution of the mesh description of the smooth potential. The polynomial order p in the present method does not have the same effect, since the screens are exactly represented on the mesh. Instead, p affects $\hat{R}_c$ via the decay of the screened potential as shown in Figure 6. This is important, since for uniform charge density the cost of point-to-point interactions scales with volume $\sim \hat{R}_c^3$. An independent Ewald splitting parameter sets the corresponding truncation error at fixed cut-off radius for PME.

Similarly, the mesh density has different implications in the two methods. As with the B-spline order, the mesh density in PME affects the accuracy by providing more resolution for the potential. A denser mesh does not affect the short-range calculation, but requires more global communication for the FFT. In contrast, the mesh density in the present scheme decreases the communication burden for the short-range component of the calculation by reducing the number of interactions included for a given $\hat{R}_c$, since the cut-off radius is scaled by the mesh size, unlike in PME. The communication required of the mesh solver is that of multigrid.

The simple periodic test case depicted in section 4.1 allows computation of the "exact" answer using the Ewald sum in order to report errors. A triply-periodic box, however, is an ideal case for PME, and we would expect its performance in terms of total CPU time to be superior to other methods, including the method presented here. This is confirmed for the case with $N = 10\,000$ particles that is shown in Figure 11, with timing results summarized in Table 3. Here, p is the order of interpolation for a standard PME method with $\alpha = 100$ and cutoff radius $R_c = 0.23$. This was selected to approximately equal the size used in our FE-based method. The number of mesh points for the PME method was $M = 97336$, which corresponds to the number of degrees of freedom in the mesh for the FE-based case with $q = 1$.

Table 3
CPU times for major components of a periodic potential calculation.

Method		Max relative abs. error	Short-range/ self time (sec)	Mesh solve time (sec)
FE-based screens	$q = 1$	2.70×10^{-1}	7.58×10^{0}	1.06×10^{1}
	$q = 2$	5.51×10^{-2}	2.51×10^{1}	4.48×10^{1}
	$q = 3$	5.08×10^{-3}	7.22×10^{1}	6.49×10^{1}
	$q = 4$	1.46×10^{-4}	2.17×10^{2}	1.66×10^{2}
PME	$p = 1$	5.47×10^{-1}	1.12×10^{0}	6.00×10^{-3}
	$p = 2$	2.35×10^{-2}	1.53×10^{0}	8.00×10^{-3}
	$p = 3$	6.47×10^{-3}	1.21×10^{0}	6.00×10^{-3}

The PME short-range/self time is nearly the same for all values of p, since the short-range term in PME does not depend on the order of interpolation. In both PME and our FE-based method, we expect the short-range time to be drastically reduced when the problem size is large enough to allow a high degree of parallelization. Furthermore, both methods are significantly faster than the standard Ewald sum for this problem, which took $\sim 3 \times 10^3$ seconds.

5.2. Comparison to FMM. The method presented in this paper shares several attractive features of the fast multipole method, most notably the linear scaling. The relative merits in comparison to FMM are likely application dependent, and the preferred choice depends on several factors. Though intricate, the low communication burden of FMM leads to efficient implementations [19]. Both methods become expensive with increased p, the basis order in the present scheme or the multipole expansion order for FMM. Yet the highly local work load of the proposed high-order screens is more suitable for emerging architectures with accelerators.

5.3. Other considerations. For dynamic application, the conservation properties of the overall scheme are important, such as conservation of energy in molecular dynamics simulations. Since we are only evaluating potentials in this paper, we do not consider momentum or energy conservation in detail. For the formulation as presented, the operators we demonstrate do not exactly satisfy the symmetry discussed by Hockney and Eastwood [16], so exact momentum conservation is not anticipated. Moreover, as the basis functions are not differentiable at the collocation points, straightforward analytical differentiation of the potentials is not always possible.

The nature of the mesh solve in the presented method lends itself to varied boundary conditions since the fundamental formulation of the algorithm does not change when the boundary conditions are changed. As with PME, periodic boundary conditions are the simplest to implement in our method, and require no extra effort beyond creating a finite element matrix that honors the periodic structure of the mesh. As presented in section 2.6 and demonstrated in section 4.3, even nontrivial Dirichlet (or Neumann) boundary conditions simply require calculations to allow for any short-range effects already present on the surface before applying the conditions to the finite element problem. Furthermore, we are not restricted to boundary conditions which can be implemented with the method of images.

The method is also extensible without fundamental changes to nonuniform meshes common in finite element discretizations. The main difference for general meshes is in the cost of the method. The screens are still built in the same way, that is, they still solve (2.10). However, the discussed simplifications of the screen coefficient calculations depend on a regular, rectangular mesh and are not applicable to an unstructured mesh with general quadrilateral elements. Thus, the flexibility of a complex mesh is balanced with the benefits of localizing the mesh cells. Likewise, the short-range potential becomes more difficult to generalize due to the many different shapes a screen could take based on the shapes of the elements composing it. Gaining accurate values for the short-range potentials may require quadrature-based evaluations for each pair of interacting charges. However, the locality and structured character of these operations is expected to coincide with high-throughput accelerators.

In our demonstration, we have presented one choice for the support of the screens. Another possible variation of the method is to limit the screens to have support in only the element containing the charge, so that each screen includes only degrees of freedom interior to the element and not those on the faces. Using the same multipole representation to construct the screen, this choice results in a loss of two powers in the short-range decay of the screens—e.g., $q = 3$ for the screen yields a R^{-3} far-field decay instead of R^{-5}. However, the more compact screen provides an asymptotic decay rate starting at $\hat{R} \approx 1$ instead of $\hat{R} \approx 3$ (see Figure 6), which reduces the cost through reducing $\hat{R}_c$ for a certain target accuracy. The local composition of these one-element screens also facilitates the move to unstructured meshes, helping alleviate some of the additional cost in the screen-related calculations.

In constructing our screens, we have chosen to maximize the far-field decay rate. Some simulation goals may be better served by other choices—e.g., by a weighted objective function. In such cases, a least-squares optimization might provide screens with advantageous properties to meet overall simulation objectives. We have also restricted our discussion to purely polynomial basis functions. Given the regularity of the underlying Green's function, basis enrichment with specially designed functions chosen to increase the short-range decay would likely enhance the overall performance of the method.

Acknowledgment. We would like to thank the National Center for Supercomputing Applications (NCSA) for their computing resources.

REFERENCES

[1] A. H. Baker, R. D. Falgout, T. V. Kolev, and U. Meier Yang, *Scaling hypre's multigrid solvers to* 100, 000 *cores*, in High-Performance Scientific Computing, M. W. Berry, K. A. Gallivan, E. Gallopoulos, A. Grama, B. Philippe, Y. Saad, and F. Saied, eds., Springer London, 2012, pp. 261–279, http://dx.doi.org/10.1007/978-1-4471-2437-5_13.

[2] D. S. Cerutti and D. A. Case, *Multi-level Ewald: A hybrid multigrid/fast Fourier transform approach to the electrostatic particle-mesh problem*, J. Chem. Theory Comput., 6 (2010), pp. 443–458.

[3] D. S. Cerutti, R. E. Duke, T. A. Darden, and T. P. Lybrand, *Staggered mesh Ewald: An extension of the smooth Particle-Mesh Ewald method adding great versatility*, J. Chem. Theory Comput., 5 (2009), pp. 2322–2338.

[4] H. Cheng, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm in three dimensions*, J. Comput. Phys., 55 (1999), pp. 468–498.

[5] T. Darden, D. York, and L. Pedersen, *Particle mesh Ewald: An* $n \log(n)$ *method for Ewald sums in large systems*, J. Chem. Phys., 98 (1993), pp. 10089–10092.

[6] J. W. Eastwood, *Particle simulation methods in plasma physics*, Comput. Phys. Commun., 43 (1986), p. 89–106.

[7] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, *A smooth particle mesh Ewald method*, J. Chem. Phys., 103 (1995), pp. 8577–8593.

[8] P. Ewald, *Die berechnung optischer und elektrostatischer gitterpotentiale*, Ann. Phys., 369 (1921), pp. 253–287.

[9] J. B. Freund, *Electro-osmosis in a nanometer-scale channel studied by atomistic simulation*, J. Chem. Phys., 116 (2002), pp. 2194–2200.

[10] H. Gahvari and W. Gropp, *An introductory exascale feasibility study for FFTs and multigrid*, International Symposium on Parallel and Distributed Processing, IEEE, Washington, DC, 2010, pp. 1–9, http://doi.ieeecomputersociety.org/10.1109/IPDPS.2010.5470417.

[11] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 135 (1997), pp. 279–292.

[12] L. Greengard and V. Rokhlin, *A new version of the fast multipole method for the Laplace equation in three dimensions*, Acta Numer., 6 (1997), pp. 229–269.

[13] V. E. Henson and U. M. Yang, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177.

[14] J. P. Hernandez-Ortiz, J. J. de Pablo, and M. D. Graham, *Fast computation of many-particle hydrodynamic and electrostatic interactions in a confined geometry*, Phys. Rev. Let., 98 (2007), 140602.

[15] J. J. Heys, T. A. Manteuffel, S. F. McCormick, and L. N. Olson, *Algebraic multigrid for higher-order finite elements*, J. Comput. Phys., 204 (2005), pp. 520–532, http://dx.doi.org/10.1016/j.jcp.2004.10.021.

[16] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*, Taylor and Francis, New York, London, 1988.

[17] J. A. Izaguirre, S. S. Hampton, and T. Matthey, *Parallel multigrid summation for the N–body problem*, J. Parallel Distrib. Comput., 65 (2005), pp. 949–962.

[18] R. C. Kirby, *Fast simplicial finite element algorithms using Bernstein polynomials*, Numer. Math., 117 (2011), pp. 631–652, http://dx.doi.org/10.1007/s00211-010-0327-2.

[19] I. Lashuk, A. Chandramowlishwaran, H. Langston, T.-A. Nguyen, R. Sampath, A. Shringarpure, R. Vuduc, L. Ying, D. Zorin, and G. Biros, *A massively parallel adaptive fast multipole method on heterogeneous architectures*, Commun. ACM, 55 (2012), pp. 101–109.

[20] L. Olson, *Algebraic multigrid preconditioning of high-order spectral elements for elliptic problems on a simplicial mesh*, SIAM J. Sci. Comput., 29 (2007), pp. 2189–2209, http://dx.doi.org/10.1137/060663465.

[21] L. N. Olson, J. Schroder, and R. S. Tuminaro, *A new perspective on strength measures in algebraic multigrid*, Numer. Linear Algebra Appl., 17 (2010), pp. 713–733, http://dx.doi.org/10.1002/nla.669.

[22] L. N. Olson, J. B. Schroder, and R. S. Tuminaro, *A general interpolation strategy for algebraic multigrid using energy minimization*, SIAM J. Sci. Comput., 33 (2011), pp. 966–991, http://dx.doi.org/10.1137/100803031.

[23] A. C. J. Paes, N. M. Abe, V. A. Serrão, and A. Passaro, *Simulations of plasmas with electrostatic PIC models using the finite element method*, Braz. J. Phys., 33 (2003), pp. 411–417.

[24] E. Pollock and J. Glosli, *Comments on P^3M, FMM, and the Ewald method for large periodic coulombic systems*, Comput. Phys. Commun., 95 (1996), pp. 93–110.

[25] C. Sagui and T. Darden, *Multigrid methods for classical molecular dynamics simulations of biomolecules*, J. Chem. Phys., 114 (2001), pp. 6578–6591.

[26] B. Sandak, *Multiscale fast summation of long-range charge and dipolar interactions*, J. Comput. Chem., 22 (2001), pp. 717–731.

[27] R. D. Skeel, I. Tezcan, and D. J. Hardy, *Multiple grid methods for classical molecular dynamics*, J Comput Chem., 23 (2002), pp. 673–684.

[28] A.-K. Tornberg and L. Greengard, *A fast multipole method for the three-dimensional Stokes equations*, J. Comput. Phys., 227 (2008), pp. 1613–1619.

[29] S. Veerapaneni, A. Rahimian, G. Biros, and D. Zorin, *A fast algorithm for simulating vesicle flows in three dimensions*, J. Comput. Phys., 230 (2011), pp. 5610–5634.

[30] H. Wang, T. Lei, J. Li, J. Huang, and Z. Yao, *A parallel fast multipole accelerated integral equation scheme for 3D Stokes equations*, Internat. J. Numer. Methods Engrg., 70 (2007), pp. 812–839.