Journal of Computational Physics 230 (2011) 5807–5821

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A Fourier-series-based kernel-independent fast multipole method

Bo Zhang [a,*], Jingfang Huang [b], Nikos P. Pitsianis [a,c], Xiaobai Sun [a]

[a] Department of Computer Science, Duke University, Durham, NC 27708, USA
[b] Department of Mathematics, University of North Carolina at Chapel Hill, CB #3250, Phillips Hall, Chapel Hill, NC 27599, USA
[c] Department of Electrical and Computer Engineering, Aristotle University, Thessaloniki 54124, Greece

ARTICLE INFO

Article history:
Received 13 October 2010
Received in revised form 13 February 2011
Accepted 25 March 2011
Available online 6 April 2011

Keywords:
Fast multipole method
Green's function
Compressive approximation
Fourier series

ABSTRACT

We present in this paper a new kernel-independent fast multipole method (FMM), named as FKI-FMM, for pairwise particle interactions with translation-invariant kernel functions. FKI-FMM creates, using numerical techniques, sufficiently accurate and compressive representations of a given kernel function over multi-scale interaction regions in the form of a truncated Fourier series. It provides also economic operators for the multipole-to-multipole, multipole-to-local, and local-to-local translations that are typical and essential in the FMM algorithms. The multipole-to-local translation operator, in particular, is readily diagonal and does not dominate in arithmetic operations. FKI-FMM provides an alternative and competitive option, among other kernel-independent FMM algorithms, for an efficient application of the FMM, especially for applications where the kernel function consists of multi-physics and multi-scale components as those arising in recent studies of biological systems. We present the complexity analysis and demonstrate with experimental results the FKI-FMM performance in accuracy and efficiency.

1. Introduction

We present in this paper a new kernel-independent fast multipole method (FMM), which we refer to as FKI-FMM, for efficient evaluation of the pairwise interactions in a large ensemble of N particles governed by

$$\Phi(\mathbf{x}_i) = \sum_{j=1, j\neq i}^{N} q_j K(\mathbf{x}_i - \mathbf{x}_j), \quad i = 1, \ldots, N, \tag{1}$$

using $O(N)$ arithmetic operations, where the translation-invariant kernel function K describes how the particle located at $\mathbf{x}_i$ with charge q_i interacts with the other particles. FKI-FMM uses a compressive approximation of the kernel function in the form of a truncated Fourier series. It provides an alternative and competitive option, among other kernel-independent algorithms, for applications where the kernel function is composed of multi-physics and multi-scale interacting terms.

FKI-FMM draws upon three major advances in the design, development, and applications of the FMM invented in 1987 by Greengard and Rokhlin [1–3]. These advances are on (i) the adaptation to the particle distributions, (ii) the diagonalization of the multipole-to-local translation operator, and (iii) the development of the kernel-independent approximation mechanism. The adaptive FMM extended the efficient use of the FMM to applications with non-cubical geometry or with non-uniform sample distributions. The diagonalization of the multipole-to-local translation operator using the plane wave expansions [4] minimized the factor with N in the arithmetic complexity. The resulting algorithm, known as the new version of the

* Corresponding author. Tel.: +1 919 660 6558; fax: +1 919 260 3645.
E-mail addresses: zhangb@cs.duke.edu (B. Zhang), huang@amath.unc.edu (J. Huang), nikos.pitsianis@eng.auth.gr (N.P. Pitsianis), xiaobai@cs.duke.edu (X. Sun).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.03.049

FMM [4], dramatically outperformed many other types of fast algorithms, especially those with $O(N\log N)$ arithmetic complexity. It has since enabled many large-scale simulations in science and engineering, ranging from computational electromagnetics, solid and fluid dynamics, to biophysics and biochemistry applications, and beyond (see, for example, [5–12] and the references therein).

Kernel-independent FMM algorithms are intended for extending the FMM to the practical situations where the eigenfunctions of a kernel function over the far-field interaction regions are analytically unavailable or numerically expensive to evaluate. In the existing kernel-independent expansion mechanisms, we have seen the use of Taylor's expansions [13,14], certain orthogonal polynomials [15–17] and wavelet decompositions [18–20]. It is not unexpected that when applied to kernel functions with available eigen-expansions, purely for the purpose of making comparisons, these algorithms may use more arithmetic operations, because the analytical kernel-specific eigen-expansion renders the minimal number of retaining terms with respect to any prescribed accuracy. Incredibly, however, the kernel-independent algorithm by Ying, Biros and Zorin (YBZ) proved highly competitive in its accuracy and efficiency performance [21]. The YBZ algorithm assumes that the kernel is the Green's function of certain partial differential equation to which the potential theory can be applied so that an *equivalent source representation* can be derived on the circle or surface in the 2D or 3D case, respectively.

FKI-FMM relaxes on such kernel conditions while maintaining competitive performance in accuracy and efficiency. The relaxation on the kernel condition is motivated or entailed by many recent scientific simulations based on emerging multi-physics and multi-scale models. The kernel function typically couples more than one heterogeneous interaction terms, each term may be associated with a particular range in the space or spatial frequency domain, see [22] for example. Multiple term-specific applications of the FMM to such a kernel function inhibit the compression across the terms and complicate the computation. Although oblivious to any additional kernel property, the arithmetic complexity of the translation operators is comparable to both the YBZ algorithm and the eigen-expansion based FMM algorithms. By detailed analysis, the arithmetic complexity of d-dimensional FKI-FMM is $O(p^d H_d N)$, or $O(p^{d-1}(p+H_d)\log(p+H_d)N)$ by an alternative algorithmic arrangement, where p is the (equal) length of the kernel expansion in each spatial dimension and H_d is the length in the frequency scaling expansion. Here, H_d depends only on the accuracy requirement, it increases much slower than p with the number of digits in accuracy requirement. By our numerical experiments with a preliminary FKI-FMM implementation, the crossover point in the number of particles, and by the execution time, between the direct evaluation method and FKI-FMM with 3-digit accuracy for the 3D Laplacian potential is no larger than 3000. It is a clear advantage of FKI-FMM that the same arithmetic complexity holds for a translation-invariant kernel that couples multi-physics and multi-scale terms, without teasing out the terms and their respective interacting regions.

FKI-FMM consists of two stages. In the first or precomputation stage, it provides a compressive approximation of a given kernel function over the far-field interaction regions and a set of translation operators, with respect to a specific accuracy requirement. Distinctively, FKI-FMM approximates the kernel function in the simple and familiar form of a truncated Fourier series, with what we refer to as the adaptive sampling algorithm. While the approximation mechanism is kernel independent, the Fourier terms (or modes) and the associated coefficients are kernel specific. The operators are used, and re-used, in the subsequent execution stage, namely, calculating $\Phi(\mathbf{x})$ in (1) within any given particle ensemble, following the typical procedure of the FMM. The translation operations have the following unique and desirable properties. (a) The multipole-to-local translations, which dominate the computation in both complexity and practical computation time in many other FMM algorithms, become highly economic as the operator is readily diagonal in all spatial dimensions. (b) The multipole-to-multipole and local-to-local translation operators, depending only on the accuracy requirement, are independent of the kernel-specific expansion coefficients. (c) The evaluation of the trigonometric functions is simple, robust and supported ubiquitously in software or hardware. (d) The algorithm structure is pleasantly simple, easy in adaptation to the sample distribution, easy in adoption of the "merge-and-shift" technique to reduce the computational redundancy among the multipole-to-local translations [4], and relatively easy in its transporting to a parallel implementation.

The rest of the paper is organized as follows: Section 2 reviews briefly the data structures of the FMM. Section 3 introduces the adaptive sampling approach for the kernel approximation. Section 4 describes the translation operators and the arithmetic complexity of each translation. Section 5 presents experimental results with a preliminary FKI-FMM implementation. Section 6 concludes the paper with additional remarks.

2. Adaptive data structures

FKI-FMM adopts the same data structures as in the adaptive FMM [2]. In this section, we briefly discuss on (i) the FMM tree structure that represents a spatial partition at multiple scales, and (ii) a set of associated lists with each tree node, representing the spatial relations of the node to some other nodes. We illustrate these structures in the 2D case.

2.1. The FMM tree: a spatial partition hierarchy

The root node at the top level (level 0) of the FMM tree represents the smallest bounding box in $\mathcal{R}^d$ ($d = 1,2,3$) that contains all the interacting particles. The box is partitioned equally along each dimension into 2^d smaller boxes. These child boxes are the tree nodes at level 1, each with an edge to the parent node. Each child box is then partitioned in the same way, having its own child boxes. A recursion of the partition process results in a quad-tree or an oct-tree in the 2D or 3D

case, respectively. The tree nodes at level $\ell > 0$ correspond to the boxes that partition the root node equally at scale level ℓ. The path from a node at level $\ell > 0$ to the root describes its partition lineage. The criterion for terminating the partition at a box is described by a preset parameter $s > 0$. If a box contains no more than s particles, it becomes a leaf node with no further partition. Empty boxes are removed from the tree. When the particles are uniformly distributed in the root box, the tree is full and has about $\log_{2^d}(N/s)$ levels. We refer to this case as the uniform version. Otherwise, the FMM tree is adaptive to the sparsity or clusters in the particle distribution. For example, when the particles are located on the surface of a sphere, the subtrees located inside the sphere become empty, and the depth of the entire tree is larger than $\log_{2^d}(N/s)$.

2.2. The interaction lists: uniform vs. adaptive

Each and every tree node has a set of associated lists that specify the spatial relationships with certain nodes other than its parent and children. The lists among all the tree nodes form the coordination among the far-field interactions at each scale level and across multiple scale levels. The following two basic relations are used to describe the more complex relationships. First, two non-overlapping boxes are *near neighbors* if the minimal spatial distance between them is zero. The neighbors of a box B include its siblings and other boxes at the spatial boundary of B, the latter may be topologically distant along the tree edges. Second, two boxes at the same level are considered *well-separated* if they are not near neighbors.

We now describe the associated lists with each node. In the uniform version, each box B is associated with only one list, namely, the interaction list of the boxes at the same scale that are well-separated from B but whose parents are near neighbors of B's parent. The interaction lists for the spatially interior nodes have the same pattern, regardless of location and orientation. See the 2D illustration in Fig. 1. In the adaptive version, each box B is associated with up to five lists of boxes denoted by $L_i(B)$, $1 \leqslant i \leqslant 5$, specifying different types of relations detailed as follows. See also Fig. 2.

1. List $L_1(B)$ is empty if B is not a leaf node; otherwise, it consists of B itself and its near-neighbors that are leaf nodes.
2. List $L_2(B)$ is the *interaction list* as defined in the uniform version. Its pattern, however, may vary from node to node in the adaptive version.
3. List $L_3(B)$ is empty if B is not a leaf node; otherwise, it consists of all descendants of B's near neighbors, the descendants themselves are not near neighbors of B.
4. List $L_4(B)$ is the reversal of $L_3(B)$, $C \in L_4(B)$ if and only if $B \in L_3(C)$.
5. List $L_5(B)$ consists of all the boxes which are well-separated from B's parent.

3. The adaptive sampling algorithm for kernel approximation

For a specific kernel function $K(\mathbf{x})$ and a prescribed accuracy requirement, FKI-FMM provides in the precomputation stage a compressive approximation in the form of a truncated d-dimensional Fourier series, over the far-field interaction region at every level of the spatial partition, level by level.

3.1. The kernel approximation region

We determine first a unified region at each level that includes all the boxes on the interaction list of any box B at this level. At any fixed level ℓ, we may assume that the boxes are of unit length in each side, by a scaling. Then, for any box at level ℓ, the distance between B and any other box on its interaction list is between 1 and 4. It is therefore sufficient to approximate the kernel function over the following region,

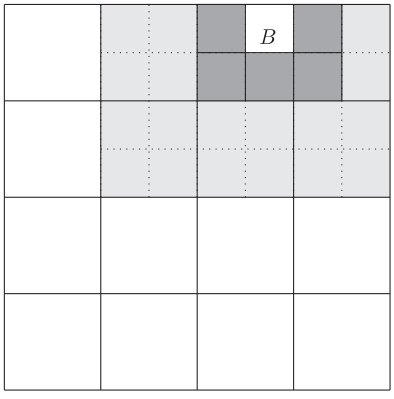

Fig. 1. A 2D illustration of the uniform version: the darker shaded boxes are near-neighbors of box B, and the lighter shaded are well separated from B and are on the interaction list of B.

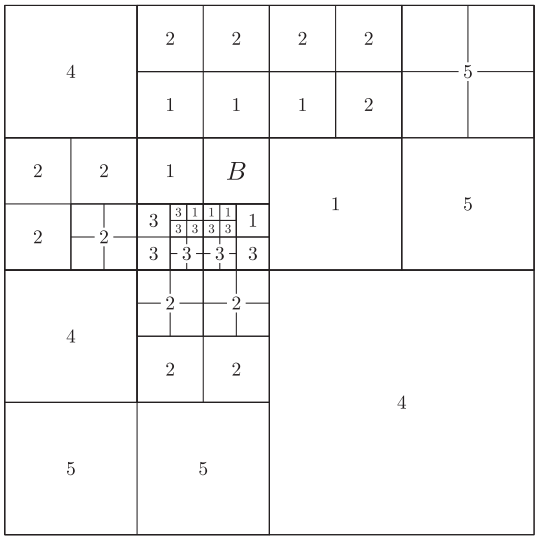

Fig. 2. A 2D illustration of the adaptive version: the boxes with label i are on List-i of box B, $i = 1, 2, \ldots, 5$.

$$\Omega_\ell = [-4,4]^d \setminus [-1,1]^d, \quad d = 1,2,3.$$

The subscript of Ω_ℓ denotes the particular scaling at level ℓ. See the 2D illustration in Fig. 3.

For any given translation-invariant kernel function, a truncated Fourier series representation over the kernel approximation region can be precomputed, level by level, and saved. Many frequently occurring kernels have an additional property, namely, the scaling property. A 3D kernel K is said to have the scaling property if, in the spherical coordinates, the following identity holds

$$K(\omega\rho,\theta,\phi) = \omega^\sigma K(\rho,\theta,\phi), \tag{2}$$

for some real number σ. The kernel functions with the scaling property include the free-space Green's functions for the Laplace, Stokes, and biharmonic equations. For such kernels, the approximation at one level can be easily converted to that at another level. Let ϵ be a prescribed error bound. Suppose the kernel is approximated at level ℓ by

$$\widetilde{K}(\mathbf{x};\boldsymbol{\alpha},\{\gamma_{\mathbf{j}}\}) = \sum_{|\mathbf{j}|\leqslant\mathbf{p}} \gamma_{\mathbf{j}} e^{i(\mathbf{j}*\boldsymbol{\alpha})\cdot\mathbf{x}}, \tag{3}$$

where the d-dimensional vector $\boldsymbol{\alpha}$ denotes the fundamental frequency in all dimensions, and $*$ denotes the Hadamard product. Assume that the kernel is nonzero over Ω_ℓ and the pointwise relative errors in the approximation are well bounded by ϵ,

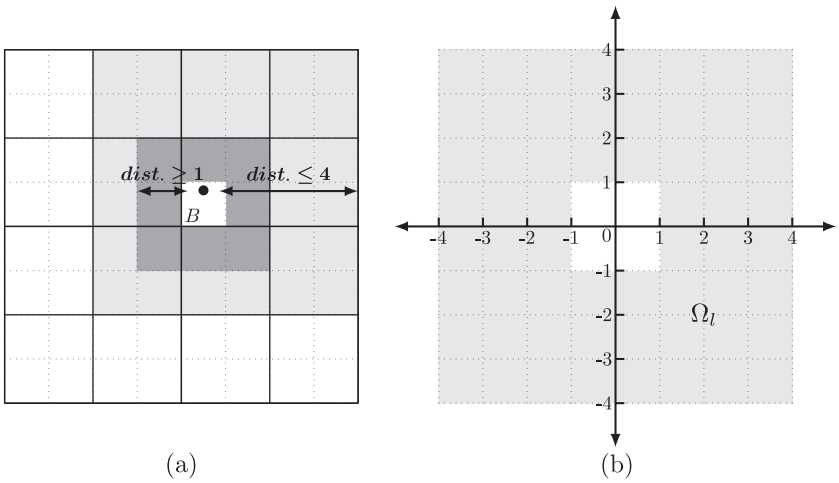

Fig. 3. A 2D illustration of the kernel approximation region.

$$\frac{K(\mathbf{x}) - \widetilde{K}(\mathbf{x}; \boldsymbol{\alpha}, \{\gamma_{\mathbf{j}}\})}{K(\mathbf{x})} = \eta(\mathbf{x}), \quad \mathbf{x} \in \Omega_\ell, \quad |\eta(\mathbf{x})| \leqslant \epsilon. \tag{4}$$

Then, at level $\ell' \neq \ell$,

$$\frac{K(\mathbf{x}') - \widetilde{K}(\mathbf{x}'; \omega\boldsymbol{\alpha}, \{\gamma_{\mathbf{j}}/\omega^\sigma\})}{K(\mathbf{x}')} = \eta(\omega\mathbf{x}'), \quad \mathbf{x}' \in \Omega_{\ell'}, \ \ |\eta| \leqslant \epsilon, \tag{5}$$

where ω is the frequency scaling factor between levels ℓ and ℓ'. In other words, the Fourier representation at level ℓ' is derived by rescaling the fundamental frequency number at level ℓ by ω, and the Fourier coefficients by $1/\omega^\sigma$. The scaling property simplifies the precomputation across multiple scales, only one set of expansion coefficients is to be computed and saved in the memory.

3.2. The adaptive sampling algorithm

We consider the kernel approximation over the approximation region at a fixed level. Assume again that the boxes at this level are scaled to 1 in the side length. We denote the kernel approximation region simply by Ω. There are two familiar approaches for finding a kernel representation as in (3) over Ω. In the first approach, one extends the kernel function over Ω to a cuboid so that the extended function is smooth at the boundaries of Ω and the resulting Fourier series converges fast. We note that the fundamental period $\mathbf{T}$, or the fundamental frequency number $\boldsymbol{\alpha} * \mathbf{T} = 2\pi$, is bounded from below, or above, respectively,

$$\mathbf{T} \geqslant 8, \quad \boldsymbol{\alpha} \leqslant \frac{\pi}{4}. \tag{6}$$

In the second approach, one decomposes Ω into regular cubical subdomains and approximates the kernel over each subdomain, possibly with an extension as well, resulting in a piecewise approximation over Ω. While one may resort to numerical techniques for computing the expansion coefficients, the process of finding a kernel extension is predominantly kernel specific.

We introduce the adaptive sampling algorithm (ASA) as a numerical and kernel-independent approach for the kernel approximation on Ω. In developing the ASA we made use of three ideas from the analytical approaches. In smoothly extending the kernel function, one attempts to minimize the maximal error pointwise in a truncated series. Once the retaining Fourier terms are determined, the coefficients are simply the least squares (LS) solution. When the expressions for the LS coefficients are not readily available for direct evaluation, one resorts to numerical means, particularly, numerical sampling and integration, to obtain the coefficients. Without seeking explicitly for a kernel extension first, the ASA couples, back and forth, the search for the Fourier modes, the computation of the expansion coefficients, and the error estimation, with the objective of finding the shortest expansion length subject to a preset error threshold. The same approach can be applied piecewise to the subdomains of Ω.

The Fourier terms or modes in (3) at a fixed level ℓ are described by two d-dimensional vectors $\mathbf{T}$ and $\mathbf{p}$. For simplicity in expressions and discussion, we assume through the rest of the paper the simplest case where the vectors are constant across the spatial dimensions and scales. Thus, the Fourier modes are described by only two scalar parameters, T and p. The ASA consists of two layers of sampling and search for the mode parameters and the coefficients. It searches at the outer layer for a pair of Fourier mode parameters (T,p) and renders at the inner layer the expansion coefficients $\gamma_{\mathbf{j}}$ for any pair (T,p) and the error estimation.

3.2.1. The Inner layer: computing the expansion coefficients

At its inner layer, for a specific parameter pair (T,p), the ASA determines the expansion coefficients $\gamma_{\mathbf{j}}$ with an adaptive sampling process. A pseudocode is listed in Fig. 4. Let Q be an initial and finite set of sample points in Ω. At each adaptation step, the algorithm forms and solves a weighted least squares (LS) problem on the sample set,

$$\Gamma_* = \operatorname*{argmin}_{\Gamma} \|K(Q) - \widetilde{K}(Q;\Gamma)\|_{w,2}^2 = \sum_{\mathbf{x}\in Q} w(\mathbf{x})(K(\mathbf{x}) - \widetilde{K}(\mathbf{x};\Gamma))^2, \tag{7}$$

where $\widetilde{K}(\mathbf{x};\Gamma)$ is the approximation function with the coefficient set $\Gamma = \{\gamma_{\mathbf{j}}\}$. The weight function $w(\mathbf{x})$ is positive on the sample points. Then, the ASA estimates the pointwise errors, which are expressed either in the absolute magnitude,

$$e_{\text{abs}}(\mathbf{x}) = |\widetilde{K}(\mathbf{x};\Gamma_*) - K(\mathbf{x})|, \quad \mathbf{x} \in \Omega, \tag{8}$$

or, in the relative magnitude if the kernel function is away from zero over Ω,

$$e_{\text{rel}}(\mathbf{x}) = \frac{|\widetilde{K}(\mathbf{x};\Gamma_*) - K(\mathbf{x})|}{|K(\mathbf{x})|}, \quad \mathbf{x} \in \Omega. \tag{9}$$

The error estimation is based on the pointwise errors at a finite set Q_{eval} of evaluation points, which are sufficiently dense in Ω. Based on the error evaluation and estimation, the procedure `adapt` makes changes in the samples and weights for the LS optimization at the next step. We elaborate the adaptation process in Section 3.2.2.

The inner layer of the adaptive sampling algorithm.
For computing the coefficients with a set of specified Fourier terms and estimating the approximation errors.

Input
K: an evaluation procedure for a given kernel function
(T, p): the parameters for specifying the Fourier terms
(Q, w): the initial samples and weights
Q_{eval}: the set of evaluation points for error estimation
i_{max}: the maximal number of adaptation steps
Adaptive sampling
$i = 0$
while $(i{+}{+} < i_{\max}$ & {some other criteria}$)$
find the LS solution as described by (7)
estimate $e_{\max}(\Omega)$ from sampled error evaluation $e(Q_{\text{eval}})$
$[Q, w] := \text{adapt}(Q, w, e(Q_{\text{eval}}))$
endwhile
Output $(\Gamma, e_{\max})$ with the min-max error among the sampling trials.

Fig. 4. An outline of the inner layer process of the adaptive sampling algorithm.

As shown in Fig. 4, the inner procedure terminates and returns to the outer layer search when the number of adaptation steps exceeds a preset upper bound, or when the LS problem becomes extremely ill-conditioned, as a result of samples clustering at certain location(s). At the termination, the LS solution with the min-max error among all the sampling trials is returned to the outer layer as a feedback to the search for the mode parameters.

3.2.2. The sampling scheme

The pointwise errors at the sample points in Q and the evaluation points in Q_{eval} depend on three particular attributes in the LS approximation (7). The first is the ratio of the number of samples to the number of the expansion coefficients. It is always set greater than 1 and therefore referred to as the *oversampling factor*. The second attribute is the sample distribution, and the third is the weight function.

The oversampling factor is initially set close to 1 and increases in the adaptive sampling process. The initial sample distribution can be chosen among several options, including equispaced nodes, the Chebyshev nodes, or randomly located nodes following a certain distribution. As the kernel approximation is precomputed, one may also utilize kernel-specific information in the initial sampling. Here, the sampling consists of both the sample points and the associated weights. The initial weights are set to the constant 1 as a default in the ASA or as provided otherwise. In the adaptive sampling process, the procedure `adapt` adds to Q more samples at locations where the errors are large. In a particular implementation, we update Q, followed by the coefficient computation and error evaluation, and then update the weights. We found it substantially more effective to use the pointwise error distribution over the updated sample set as the new weight distribution, in comparison to the constant weight.

We illustrate the adaptive sampling process with the approximation of the 1D kernel function $K(x) = 1/x^2$. The kernel is even, only the cosine terms are needed in the Fourier representation. Fig. 5 shows the effect of the adaptive sampling procedure on the changes in the over-sampling factor, sample distribution, and maximal errors, with $T = 8.253$ and $p = 9$. We present the curves of K and $\widetilde{K}$ over $[1,4]$ and the errors at the sample points and evaluation points. We show also the extended periodic function over $[-5.25, 5.25]$, which covers $[-T/2, T/2]$. Initially, 12 equispaced samples were used for the case shown on the left of Fig. 5, and 12 Chebyshev nodes were used for the case on the right. The pointwise errors were used as the feedback to adjust the weights over the samples. The initial error is smaller when the Chebyshev nodes are used. As the samples and weights are adapted, both adaptation sequences reach to about the same accuracy. The final sample distributions are very similar, not sensitive to the initial distribution, as long as the initial samples are sparse and leave room for the adaptation. The adaptive sampling scheme is also effectively used in the development of the FKI-FMM operators for coefficient translations across spatial scales, which we introduce in Section 4.

3.2.3. The outer layer: searching for the Fourier modes

At the outer layer, the ASA searches for the optimal pair of mode parameters (T, p) in the sense that p is minimal over the search domain subject to the accuracy requirement, see (3). The algorithm places an initial grid over an estimated range of (T, p). It then narrows down the search region and refines the search direction and step size, based on the min–max error feedback from the inner layer procedure.

We show in Fig. 6 the effect of the fundamental period on the constrained minimization. We use the same kernel function $K(x) = 1/x^2$ and set $p = 9$, as in Fig. 5. We use different values of T. The Fourier coefficients are computed by the inner layer routine. When T is closer to the lower bound of the fundamental frequency, $T = 8.0111$ for instance, more terms are needed for a smooth extension at the boundary, see the extended function at the bottom left plot of Fig. 6. When T is larger, $T \geqslant 9.4342$ for instance, the fundamental frequency number α becomes smaller. More terms are needed to capture the necessary high-frequency components. After certain point in T, the extended function starts to have negative values outside the

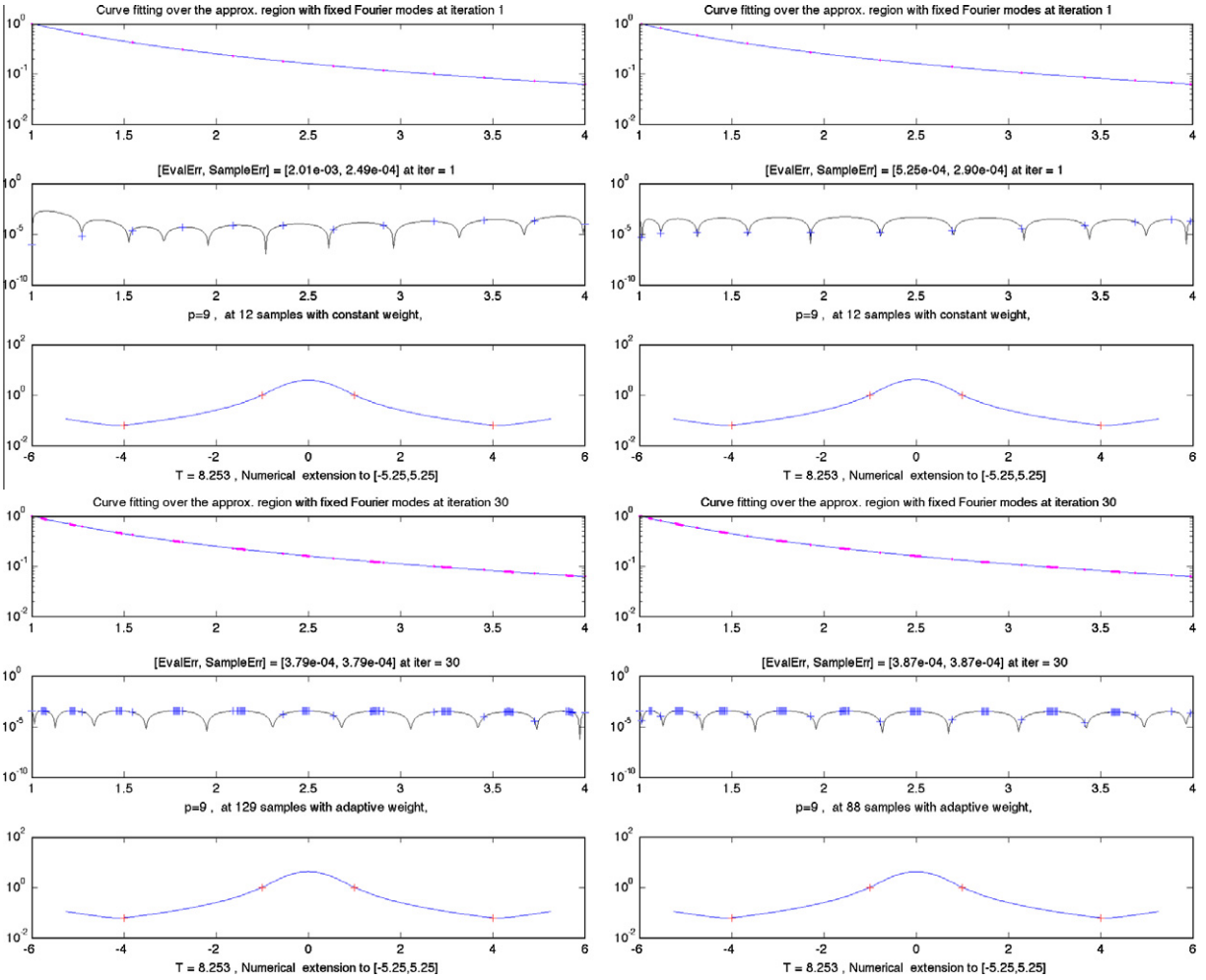

Fig. 5. Approximating $K(x) = 1/x^2$ with $T = 8.2530$ and $p = 9$, with initial samples equispaced (left) or at the Chebyshev nodes (right), at adaptation steps 1 (top) and 30 (bottom).

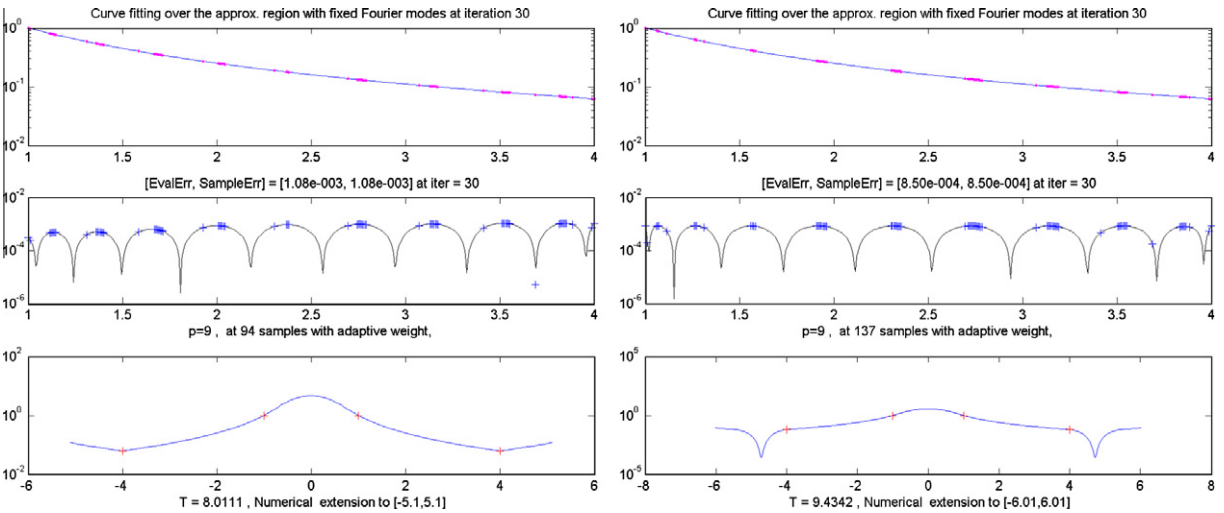

Fig. 6. Approximating $K(x) = 1/x^2$ with $p = 9$ and $T = 8.0111$ (left) and $T = 9.4342$ (right).

approximation region. Clearly, the extended functions in Fig. 5 are smoother than those in Fig. 6, and are better approximations over Ω.

3.3. The dyadic frequency structure across scales

We have discussed the kernel approximation at any spatial scale. We comment now on the structure of the fundamental frequency numbers across scales. The fundamental frequency number at one spatial scale differs from, but relates to, that at another scale.

Consider first the case that the kernel function has the scaling property, see (2). Assume that at level ℓ the frequency number is α and the expansion coefficients are $\gamma_{\mathbf{j}}$. Then, by (4) and (5), the frequency number at level ℓ' is $2^{\ell'-\ell}\alpha$ and the kernel expansion coefficients are $\gamma_{\mathbf{j}}/2^{\sigma(\ell'-\ell)}$. For kernels without this scaling property, the dyadic frequency structure across scales can still be maintained. Once the frequency number at a fixed level is determined, the largest value of p is to be determined across the dyadic frequency scales. This dyadic frequency structure significantly simplifies the inter-scale translations of the coefficients that carry the particle information, which we shall introduce in Sections 4.4 and 4.5.

4. Encoding and translating the particle information

In this section, we describe the second stage of FKI-FMM, the calculation of particle interactions. Specifically, we describe the operations for generating and translating the multipole and local coefficients that encode and decode, respectively, the particle information and interaction in the Fourier terms chosen for the kernel approximation. We give the arithmetic complexity for each operation. We elaborate the representation and translation formulas for the 2D case only, which can be extended to the 3D case straightforwardly. The complexity analysis is for any dimension d, $d = 1, 2, 3$.

We assume that an approximation $\widetilde{K}$ to the given kernel function has been obtained in the form of a truncated Fourier series over the approximation region at every level,

$$\widetilde{K}(\mathbf{x}) = \sum_{|\mathbf{j}|\leqslant p} \gamma_{\mathbf{j}} e^{\imath\alpha\mathbf{j}\cdot\mathbf{x}}, \quad \mathbf{x}\in\Omega, \tag{10}$$

where $\alpha T = 2\pi$ and the relative error criterion in (9) is used.

4.1. Forming the multipole expansion

The following theorem provides the formulas for the multipole expansion. It can be verified straightforwardly with some algebraic manipulations.

Theorem 1 (Generating the multipole coefficients). *Assume that there are M source points at locations $\mathbf{x}_m$ with charges q_m, $m = 1,\ldots,M$, inside a unit box A centered at $\mathbf{x}_a$. Let $\mathbf{x}$ be any target point in the interaction list region of A. By the kernel approximation in (10), the potential at $\mathbf{x}$ due to the M source charges can be represented by a "multipole expansion",*

$$\sum_{m=1}^{M} q_m K(\mathbf{x}-\mathbf{x}_m) = \sum_{|\mathbf{j}|\leqslant p} \gamma_{\mathbf{j}} M_{\mathbf{j}} e^{\imath\alpha\mathbf{j}\cdot(\mathbf{x}-\mathbf{x}_a)} + r(\mathbf{x},A), \tag{11}$$

where $\gamma_{\mathbf{j}}$ are the precomputed coefficients in the kernel approximation, $M_{\mathbf{j}}$ are the multipole coefficients,

$$M_{\mathbf{j}} = \sum_{m=1}^{M} q_m e^{\imath\alpha\mathbf{j}\cdot(\mathbf{x}_a-\mathbf{x}_m)}, \tag{12}$$

the residual function is bounded from above as follows,

$$|r(\mathbf{x},A)| < \epsilon \sum_{m=1}^{M} |q_m|\cdot|K(\mathbf{x}-\mathbf{x}_m)|, \quad \mathbf{x}-\mathbf{x}_m\in\Omega,$$

and ϵ is the error tolerance in the kernel approximation.

By (12), the source information q_m at location $\mathbf{x}_m$ is translated to $\mathbf{x}_a$, the center of box A, in terms of the multipole coefficients $M_{\mathbf{j}}$. It is worth noting that the computation in (12) is based only on the Fourier modes determined in the kernel approximation, independent of the kernel expansion coefficients. We denote by $\mathcal{T}_{\mathrm{SM}}$ the operator in (12) for generating the multipole coefficients. The complexity of the coefficient generation per source point is $O(p^d)$.

4.2. The multipole-to-local coefficient translation

In FKI-FMM, as in the basic FMM, every box C collects from all the boxes on its interaction list the particle information encoded in the multipole coefficients and expresses the accumulated information in terms of the so-called local coefficients at box C.

Theorem 2 (The multipole-to-local translation). *Assume that box B is on the interaction list of C. The multipole expansion of box B is given by (12) with the fundamental frequency β. Then, the potential at any* $\mathbf{x}$ *in C due to charges in box B can be accurately represented by a "local expansion"*

$$\sum_{\mathbf{x}'\in B} q(\mathbf{x}')K(\mathbf{x}-\mathbf{x}') \approx \sum_{|\mathbf{j}|\leqslant p} L_{\mathbf{j}}^{C} e^{\imath\beta\mathbf{j}\cdot(\mathbf{x}-\mathbf{x}_c)}, \tag{13}$$

where the local coefficients $L_{\mathbf{j}}^{C}$ are computed from the multipole coefficients as follows,

$$L_{\mathbf{j}}^{C} = \gamma_{\mathbf{j}}^{C} e^{\imath\beta\mathbf{j}\cdot(\mathbf{x}_c-\mathbf{x}_b)} M_{\mathbf{j}}^{B}. \tag{14}$$

We denote by $\mathcal{T}_{\mathrm{ML}}$ the multipole-to-local translation operator in (14). The operator is readily diagonal. The complexity of the multipole-to-local coefficient translation per box is $O(p^d)$. Across the boxes at each level of the FMM tree, the "merge-and-shift" technique initially introduced in [4] can be easily adopted in the FKI-FMM to reduce the complexity of the $\mathcal{T}_{\mathrm{ML}}$ operations per level.

A few remarks are in order. Based on the above, one can develop two "tree-code" algorithms, each with $O(N\log N)$ arithmetic complexity. The first follows the structure of the Barnes–Hut algorithm. The value of $\Phi(\mathbf{x})$, see (1), can be obtained by collecting and accumulating the interactions at all levels in terms of the multipole expansions. Only the $\mathcal{T}_{\mathrm{SM}}$ operations are used. In the second algorithm, $\mathcal{T}_{\mathrm{ML}}$ translations are used at each level before the accumulation across the scales. The algorithm is symmetric in the sense that the particle information is encoded and decoded with the same set of Fourier modes before and after the aggregated interactions. Following the basic algorithmic structure in the FMM, we introduce next two inter-scale translation operators in order to reduce the total complexity to $O(N)$.

4.3. Dyadic frequency scaling

We introduce first the scaling relationship among the Fourier modes between two levels. Fix any specific level below level 2. Assume that the side length of a box at this level is scaled to 1. Then, by the discussion in Section 3.1, the fundamental period T for the kernel expansion at this level is bounded from below by 8 in each dimension, and equivalently, the fundamental frequency number α is bounded from above by $\pi/4$. Based on the discussion in Section 3.3, we may assume further that the fundamental frequency number at the parent level is $\alpha/2$. The following lemma describes how to represent the Fourier terms at the parent level in terms of those at the child level. It plays the same important role as the addition theorems used for the multipole-to-multiple and local-to-local translations in the FMM for the particle interaction governed by the Laplacian potential [1,2].

Lemma 3 (The dyadic frequency scaling expansion). *Let $\alpha \in (0,\pi/4)$. For any $\delta > 0$, there exists an integer $H_k > 0$ and a set of coefficients τ_h^k, $-H_k \leqslant h \leqslant H_k$, such that*

$$\max_{x\in[-k,k]} \left| e^{\imath(\alpha/2)x} - \sum_{h=-H_k}^{H_k} \tau_h^k e^{\imath\alpha hx} \right| < \delta, \quad k = 1,2,3. \tag{15}$$

The coefficients τ_h^k, $k = 1,2,3$, in the frequency scaling expansion (15) can be obtained, in the precomputation stage, by three different approaches. The first approach is by the Fourier series expansion, it suffers from slow convergence. The second approach involves solving a system of linear equations derived from two Taylor's expansions at the parent and child levels, respectively. The resulting expansion is much shorter, subject to the same accuracy requirement. Based on the second approach, the third approach gives further compressed expansion by using the ASA introduced in Section 3.2. The bound H_k on the length of frequency scaling expansion over the range $[-k,k]$ decreases with k, see Table 1, with respect to any particular accuracy requirement δ. The bounds in the table are obtained at the extreme frequency number $\alpha = \pi/4$. The reason for having three different sets of the frequency scaling coefficients over three different ranges of variable x shall become clear in the later discussion on the inter-scale translations.

4.4. The multipole-to-multipole translation

Let B be a non-leaf node box. Using the dyadic frequency scaling expansion, the following theorem shows how the multipole expansion at B can be derived from those at its child boxes.

Table 1
Bounds on the length of dyadic frequency scaling expansions for various accuracy requirements.

$-\log_{10}(\delta)$	1	2	3	4	5	6	7	8	9	10	11	12
H_1	1	1	2	2	3	4	4	5	6	7	7	8
H_2	1	2	4	5	6	7	8	10	11	12	14	15
H_3	3	5	8	10	13	16	18	21	24	27	30	32

Theorem 4 (The Multipole-to-multipole translation). *Let B be the parent box of a unit-length box A. Assume that by the kernel approximation levelwise, the frequency number is α at the child level and $\beta = \alpha/2$ at the parent level. Assume that the multipole expansion of box A as in (12) is available. Then, the multipole expansion of the parent box B at* $\mathbf{x}$ *due to the source charges in A is as follows,*

$$\sum_{\mathbf{x}'\in A} q(\mathbf{x}')K(\mathbf{x}-\mathbf{x}') \approx \sum_{|\mathbf{j}|\leqslant p} \gamma_{\mathbf{j}}^{B} M_{\mathbf{j}}^{B} e^{\imath\beta\mathbf{j}\cdot(\mathbf{x}-\mathbf{x}_b)}, \tag{16}$$

where $\mathbf{x}$ *is an arbitrary location in the interaction list region of B, $\gamma_{\mathbf{j}}^{B}$ are the precomputed Fourier coefficients at the parent level, and the multipole coefficients of B can be obtained from those of A with sufficient accuracy, instead of the direct evaluation by (12),*

$$M_{\mathbf{j}}^{B} \approx \sum_{h\in\mathcal{I}_{\mathrm{M2M}}} \tau_h^{\nu} M_{\mathbf{k}+h(\mathbf{j}-2\mathbf{k})}^{A} e^{\imath\alpha(\mathbf{k}+h(\mathbf{j}-2\mathbf{k}))\cdot(\mathbf{x}_b-\mathbf{x}_a)}. \tag{17}$$

Here, $\mathbf{k} = floor(\mathbf{j}/2)$, $\nu = \|\mathbf{j}-2\mathbf{k}\|_1$, *and the summation index set is*

$$\mathcal{I}_{\mathrm{M2M}} = \{h : |h| \leqslant H_\nu, |(\mathbf{j}-2\mathbf{k})h+\mathbf{k}| \leqslant p\}.$$

The parameter H_ν and coefficients τ_h^ν for $\nu > 0$ are determined as in Lemma 3 and Table 1 and $H_0 = 0$, $\tau_0^0 = 1$.

Proof. We consider the translated expansion at B due to charges inside A. We select $H_\nu > 0$, $\nu = 1,2$, such that the error in the dyadic scaling expansion (15) is within a fraction of the error bound ϵ imposed on the kernel approximation. The proof of (17) is divided into four cases, depending on the polarity of the Fourier mode associated with the integer multipliers $\mathbf{j} = (j_1, j_2)$. In each case, we start with the direct generation of the multipole coefficients by (12),

$$M_{\mathbf{j}}^{B} = \sum_{m=1}^{M} q_m e^{\imath\beta\mathbf{j}\cdot(\mathbf{x}_b-\mathbf{x}_m)},$$

assuming that A contains M charged particles.

CASE EE: $(j_1, j_2) = (2k_1, 2k_2)$.

$$M_{\mathbf{j}}^{B} = \left[\sum_{m=1}^{M} q_m e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_a-\mathbf{x}_m)}\right] e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_b-\mathbf{x}_a)} = M_{\mathbf{k}}^{A} e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_b-\mathbf{x}_a)}.$$

CASE OE: $(j_1, j_2) = (2k_1+1, 2k_2)$.

$$M_{\mathbf{j}}^{B} = \sum_{m=1}^{M} q_m e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_b-\mathbf{x}_m)} e^{\frac{\imath\alpha}{2}(x_b-x_m)} \approx \sum_{m=1}^{M} q_m e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_b-\mathbf{x}_m)} \sum_{h=-H_1}^{H_1} \tau_h^1 e^{\imath\alpha h(x_b-x_m)} = \sum_{h=-H_1}^{H_1} \tau_h^1 \left[\sum_{m=1}^{M} q_m e^{\imath\alpha(\mathbf{k}+h(\mathbf{j}-2\mathbf{k}))\cdot(\mathbf{x}_a-\mathbf{x}_m)}\right] e^{\imath\alpha(\mathbf{k}+h(\mathbf{j}-2\mathbf{k}))\cdot(\mathbf{x}_b-\mathbf{x}_a)}$$
$$\approx \sum_{h\in\mathcal{I}_{\mathrm{M2M}}} \tau_h^1 M_{\mathbf{k}+h(\mathbf{j}-2\mathbf{k})}^{A} e^{\imath\alpha(\mathbf{k}+h(\mathbf{j}-2\mathbf{k}))\cdot(\mathbf{x}_b-\mathbf{x}_a)}.$$

There are two sources of errors introduced in this case. The first lies in the frequency scaling of the modes with odd indices. The frequency scaling errors can be made sufficiently small by properly choosing H_ν, see Lemma 3 and Table 1. The second is the absence of the multipole coefficients associated with the indices out of the range. The truncation errors are already under control by the ASA and Theorem 2. These comments apply to the next two cases as well.

CASE EO: $(j_1, j_2) = (2k_1, 2k_2+1)$. Similar to the OE case,

$$M_{\mathbf{j}}^{B} \approx \sum_{h\in\mathcal{I}_{\mathrm{M2M}}} \tau_h^1 M_{\mathbf{k}+h(\mathbf{j}-2\mathbf{k})}^{A} e^{\imath\alpha(\mathbf{k}+h(\mathbf{j}-2\mathbf{k}))\cdot(\mathbf{x}_b-\mathbf{x}_a)}.$$

CASE OO: $(j_1, j_2) = (2k_1+1, 2k_2+1)$.

$$M_{\mathbf{j}}^{B} = \sum_{m=1}^{M} q_m e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_b-\mathbf{x}_a)} e^{\frac{\imath\alpha}{2}(x_b-x_m+y_b-y_m)} \approx \sum_{m=1}^{M} q_m e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_b-\mathbf{x}_a)} \sum_{h=-H_2}^{H_2} \tau_h^2 e^{\imath\alpha h(x_b-x_m+y_b-y_m)} = \sum_{h=-H_2}^{H_2} \tau_h^2 M_{\mathbf{k}+h(\mathbf{j}-2\mathbf{k})}^{A} e^{\imath\alpha(\mathbf{k}+h(\mathbf{j}-2\mathbf{k}))\cdot(\mathbf{x}_b-\mathbf{x}_a)}.$$

Note that in the 2D case, the longest expansion is in Case OO, bounded by H_2. In the 3D case, the longest expansion is Case OOO, bounded by H_3. □

We denote by $\mathcal{T}_{\mathrm{MM}}$ the multipole-to-multipole translation operator in (17). The computation by (17) requires $O(p^d H_d)$ arithmetic operations. The translation can be alternatively computed via the use of the convolution theorem and the fast Fourier transform (FFT). This alternative algorithm arrangement has the asymptotic complexity $O(p^{d-1}(p+H_d)\log(p+H_d))$. The latter approach is better only when H_d is large in order to meet a high accuracy requirement.

4.5. The local-to-local translation

In the downward pass of the FMM, each box D gets from its parent box C the far-field information at all the ancestor levels. As the local expansion of C is valid in the child boxes, the information relay is simply by a scaling in the frequency number from the parent level to the child level and a translation of the local expansion from the center of C to that of each child box.

Theorem 5 (The local-to-local translation). *Let D be a unit box with parent C. Assume C has the local expansion about its center with fundamental frequency β as in (13). Then, the local expansion of C can be translated to a local expansion about the center of D as follows,*

$$\sum_{\mathbf{x}'\notin L_1(B)\cup L_2(B)} q(\mathbf{x}')K(\mathbf{x}-\mathbf{x}') \approx \sum_{\mathbf{k}\in\mathcal{I}^2_{\mathrm{L2L}}\times\mathcal{I}^2_{\mathrm{L2L}}} L^D_{\mathbf{k}} e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}_d)}, \quad \alpha = 2\beta,$$

where the expansion index set is $\mathcal{I}^2_{\mathrm{L2L}} = \{k : |2k+1| \leqslant \min(p+2H_2, 2p)\}$,

$$\begin{aligned}
L^D_{\mathbf{k}} &\approx e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_d-\mathbf{x}_c)}\Big[L^C_{ee}(\mathbf{k}) + L^C_{oe}(\mathbf{k}) + L^C_{eo}(\mathbf{k}) + L^C_{oo}(\mathbf{k})\Big], \qquad (18)\\
L^C_{ee}(\mathbf{k}) &= \chi(k_1)\chi(k_2)L^C_{2\mathbf{k}},\\
L^C_{oe}(\mathbf{k}) &= \chi(k_2)\sum_{h\in\mathcal{H}(k_1)} \tau^1_h L^C_{2\mathbf{k}-2\mathbf{e}_1 h+\mathbf{e}_1}, \quad \mathbf{e}_1 = (1,0),\\
L^C_{eo}(\mathbf{k}) &= \chi(k_1)\sum_{h\in\mathcal{H}(k_2)} \tau^1_h L^C_{2\mathbf{k}-2\mathbf{e}_2 h+\mathbf{e}_2}, \quad \mathbf{e}_2 = (0,1),\\
L^C_{oo}(\mathbf{k}) &= \sum_{h\in\mathcal{H}(k_1)\cap\mathcal{H}(k_2)} \tau^2_h L^C_{2\mathbf{k}-2\mathbf{e}h+\mathbf{e}}, \quad \mathbf{e} = (1,1),
\end{aligned}$$

with the boolean function $\chi(k) = (|2k| \leqslant p)$ and the index set $\mathcal{H}(k) = \{h : |2(k-h)+1| \leqslant p\}$.

Before giving the proof, we note that the length of the local expansion of D due to the far-field particle influence is shorter than p when H_v is smaller than $p/2$, $1 \leqslant v \leqslant d$. This does not mean a loss of the high frequency components in terms of the fundamental frequency number α. Box D does not inherit the high frequency information from its ancestors as they don't have such. Instead, D gets the high frequency information from the boxes in its interaction list at the same level via the multipole-to-local translation by Theorem 2.

Proof. Similar to the construction of $\mathcal{T}_{\mathrm{MM}}$ operator, we use the dyadic frequency scaling expansions in Lemma 3. We split the local expansion of C into four segments according to the polarity of the Fourier modes, i.e, the even–odd patterns of the indices,

$$\sum_{|\mathbf{j}|\leqslant p} L^C_{\mathbf{j}} e^{\imath\beta\mathbf{j}\cdot(\mathbf{x}-\mathbf{x}_c)} \stackrel{def}{=} L_{ee} + L_{oe} + L_{eo} + L_{oo}.$$

We use the following notation for the index sets

$$\mathcal{I}_o = \{k : |2k+1| \leqslant p\}, \quad \mathcal{I}_e = \{k : |2k| \leqslant p\}.$$

Considering the translation segment by segment, we have

$$L_{ee} = \sum_{\mathbf{j}=2\mathbf{k}} L^C_{\mathbf{j}} e^{\imath\beta\mathbf{j}\cdot(\mathbf{x}-\mathbf{x}_c)} = \sum_{\mathbf{k}\in(\mathcal{I}_e,\mathcal{I}_e)} \Big(L^C_{2\mathbf{k}} e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_d-\mathbf{x}_c)}\Big) e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}_d)}, \qquad (19)$$

$$\begin{aligned}
L_{oe} &= \sum_{\mathbf{j}=2\mathbf{k}+\mathbf{e}_1} L^C_{\mathbf{j}} e^{\imath\beta\mathbf{j}\cdot(\mathbf{x}-\mathbf{x}_c)} = \sum_{\mathbf{k}\in(\mathcal{I}_o,\mathcal{I}_e)} L^C_{2\mathbf{k}+\mathbf{e}_1} e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}_c)} \sum_{h=-H_1}^{H_1} \tau^1_h e^{\imath\alpha h(x-x_c)}\\
&= \sum_{\mathbf{k}\in(\mathcal{I}^1_{\mathrm{L2L}},\mathcal{I}_e)} \left(\sum_{h\in\mathcal{H}(k_1)} \tau^1_h L^C_{2\mathbf{k}-2\mathbf{e}_1 h+\mathbf{e}_1} e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_d-\mathbf{x}_c)}\right) e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}_d)}, \qquad (20)
\end{aligned}$$

$$L_{eo} = \sum_{\mathbf{j}=2\mathbf{k}+\mathbf{e}_2} L^C_{\mathbf{j}} e^{\imath\beta\mathbf{j}\cdot(\mathbf{x}-\mathbf{x}_c)} = \sum_{\mathbf{k}\in(\mathcal{I}_e,\mathcal{I}^1_{\mathrm{L2L}})} \left(\sum_{h\in\mathcal{H}(k_2)} \tau^1_h L^C_{2\mathbf{k}-2\mathbf{e}_2 h+\mathbf{e}_2} e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_d-\mathbf{x}_c)}\right) e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}_d)}, \qquad (21)$$

$$L_{oo} = \sum_{\mathbf{j}=2\mathbf{k}+\mathbf{e}} L^C_{\mathbf{j}} e^{\imath\beta\mathbf{j}\cdot(\mathbf{x}-\mathbf{x}_c)} = \sum_{\mathbf{k}\in(\mathcal{I}^2_{\mathrm{L2L}},\mathcal{I}^2_{\mathrm{L2L}})} \left(\sum_{h\in\mathcal{H}(k_1)\cap\mathcal{H}(k_2)} \tau^2_h L^C_{2\mathbf{k}-2\mathbf{e}h+\mathbf{e}} e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_d-\mathbf{x}_c)}\right) e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}-\mathbf{x}_d)}. \qquad (22)$$

The local coefficient $L^D_{\mathbf{k}}$ accumulates the corresponding coefficients in L_{ee}, L_{oe}, L_{eo}, and L_{oo} as follows,

1. When $(\chi(k_1), \chi(k_2)) = (1,1), L^D_{\mathbf{k}}$ collects from all four terms.
2. When $(\chi(k_1), \chi(k_2)) = (1,0), L^D_{\mathbf{k}}$ collects from terms L_{eo} and L_{oo}.
3. When $(\chi(k_1), \chi(k_2)) = (0,1), L^D_{\mathbf{k}}$ collects from terms L_{oe} and L_{oo}.
4. When $(\chi(k_1), \chi(k_2)) = (0,0), L^D_{\mathbf{k}}$ collects from term L_{oo} only.

By Eqs. (19)–(22), translating the local expansion from C to D involves only the terms with indices in $(\mathcal{I}^2_{\mathrm{L2L}}, \mathcal{I}^2_{\mathrm{L2L}})$. □

We denote by $\mathcal{T}_{\mathrm{LL}}$ the local-to-local translation operator in (18). The operator $\mathcal{T}_{\mathrm{LL}}$ is symmetric to $\mathcal{T}_{\mathrm{MM}}$ in the sense that they share the same sets of dyadic frequency scaling coefficients as well as the shifting factors $e^{\imath\alpha\mathbf{k}\cdot(\mathbf{x}_d-\mathbf{x}_c)}$. The $\mathcal{T}_{\mathrm{LL}}$ complexity is $O(p^d H_d)$. The comments on the alternative implementation of the $\mathcal{T}_{\mathrm{MM}}$ operation apply to the $\mathcal{T}_{\mathrm{LL}}$ operation as well.

4.6. A restriction on the adaptive FKI-FMM

The kernel approximation in FKI-FMM is restricted to the approximation region at every interaction level, unlike the eigen-expansion in the spherical harmonics for the Laplacian potential over the entire far-field interaction in the original FMM. This piecewise approximation has consequently a limitation in the adaptive version. We consider for example List 3, see Section 2, following the discussion in [23]. Suppose that boxes C_1 and C_2 are both on $L_3(B)$, the List 3 of box B, see Fig. 7. For the interaction of B with C_2, we may choose the more efficient one between the direct interaction and the indirect approach via the multipole expansion at particle locations in B. We do not have the indirect option for C_1 since its approximation region does not contain B entirely.

An alternative approach is to impose an additional constraint on the adaptive tree structure. Specifically, we require that two near-neighbor boxes be no more than one level apart. This condition guarantees that the boxes on Lists 3 and 4 are in the approximation region. This imposed constraint was previously used elsewhere, such as in the traditional adaptive mesh refinement (AMR) techniques for solving systems of partial differential equations governed by hyperbolic conservation laws, see [24,25], in the new version of an FMM-based 2D Poisson solver [26], and in a Yukawa solver [27].

Except for this constraint, FKI-FMM follows the same structure as the original and new versions of the FMM, the detail of which can be found in [1–4] and the references therein.

5. Experimental results

In this section, we report experimental results obtained with an FKI-FMM implementation in C on a workstation with dual Xeon 5520 processors and 48 GB memory. The C code employs the "merge-and-shift" technique in $\mathcal{T}_{\mathrm{ML}}$ translations. In the experiments, the particles were randomly located within a unit box $[0,1]^d$, $d = 2,3$, following a uniform distribution. The particle charges were generated randomly within the range $(0,1)$.

5.1. 2D experiments

We present two sets of 2D experiments, sets A and B, for accuracy assessment. The kernel functions used are $K(\mathbf{x}) = 1/r^2$ in set A and $K(\mathbf{x}) = x^2/r^4$ in set B. We report the results at the precomputation stage followed by the execution stage. For each set, the kernel approximation is by the absolute error criterion (8) with 6-digit accuracy requirement. We provide in Table 2 for each case the chosen values for α and p in (10). We show also the maximal error in the resulting approximation and the numerical range of the kernel expansion coefficients $\gamma_{\mathbf{j}}$. The kernel expansion coefficients are used in the $\mathcal{T}_{\mathrm{ML}}$ translation, they are in a modest numerical range. The errors in each of the kernel approximations are plotted in Fig. 8.

The experimental results at the execution stage are summarized in Table 3, where N is the number of particles for each experiment, *nlev* is the maximum refinement level.

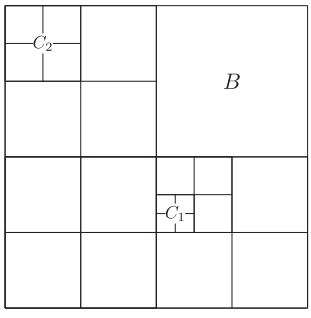

Fig. 7. Boxes C_1 and C_2 at different scale levels are on the List 3 of box B, $L_3(B)$.

Table 2
Numerical properties of the kernel approximations in sets A and B with the ASA.

Set	α	p	Max error	Range of $\gamma_{\mathbf{j}}$
A	$2\pi/9.05$	21	$3.6356 \cdot 10^{-7}$	$[-1.449 \cdot 10^{-7}, 2.278 \cdot 10^{-1}]$
B	$2\pi/8.85$	24	$5.0338 \cdot 10^{-7}$	$[-3.389 \cdot 10^{-5}, 1.203 \cdot 10^{-1}]$

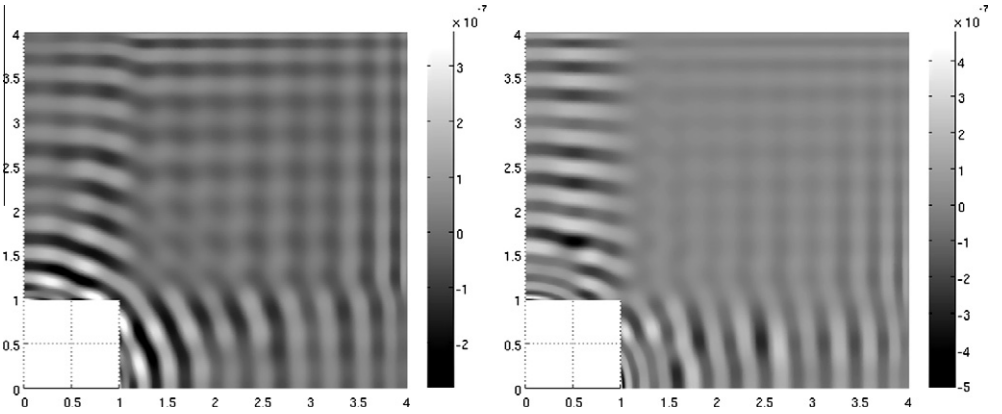

Fig. 8. The absolute errors over the first quadrant of the approximation region for the kernel $K(\mathbf{x}) = 1/r^2$ (left) and $K(\mathbf{x}) = x^2/r^4$ (right), respectively.

Table 3
A summary of the accuracy results with FKI-FMM for the 2D kernels $K(\mathbf{x}) = 1/r^2$ (left) and $K(\mathbf{x}) = x^2/r^4$ (right).

N	$K(\mathbf{x}) = 1/r^2$		$K(\mathbf{x}) = x^2/r^4$	
	nlev	E_{200}	*nlev*	E_{200}
1000	2	$1.1305 \cdot 10^{-10}$	2	$1.4026 \cdot 10^{-9}$
2000	2	$3.3964 \cdot 10^{-10}$	2	$7.8194 \cdot 10^{-10}$
3000	3	$9.0019 \cdot 10^{-10}$	3	$1.6959 \cdot 10^{-9}$
4000	3	$4.5964 \cdot 10^{-10}$	3	$7.2624 \cdot 10^{-10}$
5000	3	$5.0148 \cdot 10^{-10}$	3	$1.3418 \cdot 10^{-9}$
6000	3	$5.6749 \cdot 10^{-10}$	3	$9.5257 \cdot 10^{-10}$
7000	3	$4.7154 \cdot 10^{-10}$	3	$1.1746 \cdot 10^{-9}$
8000	3	$3.8653 \cdot 10^{-10}$	3	$1.2554 \cdot 10^{-9}$

We compared the difference at 200 target locations between the potential $\Phi(\mathbf{x}_i)$ by the direct method and $\widetilde{\Phi}(\mathbf{x}_i)$ by FKI-FMM, and calculated the error according to

$$E_{200} = \sqrt{\sum_{i=1}^{200} |\Phi(\mathbf{x}_i) - \widetilde{\Phi}(\mathbf{x}_i)|^2 \Big/ \sum_{i=1}^{200} |\Phi(\mathbf{x}_i)|^2}. \tag{23}$$

The FKI-FMM satisfies the desired accuracy requirement.

5.2. 3D experiments

In the 3D experiments we make performance assessment in efficiency as well as in accuracy. We applied FKI-FMM to the Laplacian potential $K(\mathbf{x}) = 1/r$. The kernel approximation is by the absolute criterion (8) with 3-digit accuracy. We used the following particular parameter values for the truncated Fourier series, $\alpha = 2\pi/9.05$ and $p = 9$. The maximum error in the resulting approximation is $2.1673 \cdot 10^{-4}$. We summarize the accuracy-efficiency results in Table 4, in a similar setting as in Table 3 with the additional timing information, where t_{direct} is the computation time in seconds by the direct method, t_{fkifmm} is the time by FKI-FMM with the leaf boxes at level *nlev*. The FKI-FMM time grows nearly linearly with N and surpasses the direct method when N reaches about 3000.

More complex than the direct evaluation in algorithm and programming structures, the FKI-FMM code in C is yet to be tuned with software programming technique for better performance in efficiency.

Table 4
A summary of the accuracy-efficiency results with FKI-FMM for the 3D kernel $K(\mathbf{x}) = 1/r$. The time is in seconds.

N	t_{direct}	t_{fkifmm}	$nlev$	E_{200}
1000	$2.1625 \cdot 10^{-2}$	$7.8552 \cdot 10^{-2}$	2	$6.5541 \cdot 10^{-6}$
1500	$4.9499 \cdot 10^{-2}$	$1.0403 \cdot 10^{-1}$	2	$5.1821 \cdot 10^{-6}$
2000	$8.5940 \cdot 10^{-2}$	$1.3195 \cdot 10^{-1}$	2	$4.3610 \cdot 10^{-6}$
2500	$1.3435 \cdot 10^{-1}$	$1.6240 \cdot 10^{-1}$	2	$4.1777 \cdot 10^{-6}$
3000	$1.9329 \cdot 10^{-1}$	$1.9677 \cdot 10^{-1}$	2	$3.9664 \cdot 10^{-6}$
3500	$2.6236 \cdot 10^{-1}$	$2.3090 \cdot 10^{-1}$	2	$3.9921 \cdot 10^{-6}$
4000	$3.4322 \cdot 10^{-1}$	$2.6939 \cdot 10^{-1}$	2	$3.8544 \cdot 10^{-6}$

6. Additional remarks

FKI-FMM has the potential to admit more kernel functions into the application paradigm of the FMM, with its relaxed conditions on the kernel function while remaining a competitive performance in accuracy and efficiency. The $\mathcal{T}_{9m}$ translation is diagonal, becoming the least costly translation. The kernel expansion coefficients are determined in precomputation once for each kernel function with respect to an accuracy requirement. They are the elements in the diagonal operator $\mathcal{T}_{9m}$. The translation operators $\mathcal{T}_{\text{MM}}$ and $\mathcal{T}_{\text{LL}}$ are determined also in precomputation once the Fourier modes for the kernel approximation are chosen. The complexity of the frequency scaling in $\mathcal{T}_{\text{MM}}$ or $\mathcal{T}_{\text{LL}}$ grows slowly with the number of digits in the accuracy requirement as well as with the dimension. These properties make the arithmetic complexity of FKI-FMM as low as $O(H_d p^d N)$, or $O(p^{d-1}(p + H_d)\log(p + H_d)N)$ with the use of the FFT in the operations with $\mathcal{T}_{\text{MM}}$ and $\mathcal{T}_{\text{LL}}$, see Section 4. It remains to be seen whether or not this is the lowest in asymptotic complexity under the same mild conditions. Our numerical experiments show that the use of the FFT is not numerically unstable, because the numerical ranges of the operators are modest, as illustrated in Table 2.

In addition, FKI-FMM has the simple structure and operations that are favorable on many modern computers. Specifically, during a computational execution, function evaluations take place only within each and every leaf node. Highly efficient evaluation of the trigonometric functions is broadly supported by software or even hardware. In the coefficient translations, FKI-FMM benefits numerically from the modest numerical range of the translation operators, namely, it is not sensitive to the architectural precision. These features promise greater potential in performance tuning to outweigh its relative longer expansion length in comparison to the eigen-expansion. Moreover, FKI-FMM can be transported with much ease to parallel computation on emerging parallel architectures, such as on a multi-core or many-core processor. A detailed description of FKI-FMM and its parallelization can be found in [28,29].

We comment on the essence of FKI-FMM. As discussed at the end of Section 4.2, FKI-FMM can lead to at least two tree-code algorithms of complexity $O(p^d N \log N)$, without inter-scale translations. The original FMM for particle simulation can lead to yet another algorithm with asymptotic complexity $O(N \log N)$. At the bottom level, the algorithm does a global convolution, via the use of FFT, for the interactions associated with each expansion term. There are as many independent global convolutions as the expansion terms. This is not so with FKI-FMM, because the approximation by FKI-FMM is piecewise over the approximation regions at different spatial scales. In fact, FKI-FMM is the economic alternative to the approach. It carries out a multi-level convolution, the spatial support at each level is local and within the kernel approximation region Ω_ℓ. The convolution at each level is done efficiently by the diagonal or element-wise multiplication in the Fourier domain. The Fourier transforms, to and from the Fourier domain, at all levels are more economically related than that in the multiple and independent FFTs.

Particularly, the dyadic scaling equation plays the central role in the Fourier transforms across the spatial scales. It bears similarity to the scaling or dilation equation in wavelets. It is a natural dual to the dyadic spatial partition typical in the FMM. This dual structure makes explicit, and intuitive to many computational practitioners, the relation between the spatial scales and frequency scales. The technique and analysis in Sections 3 and 4 extend straightforwardly to the situation where the ratios among the frequency numbers across the scales are not necessarily dyadic or constant.

Further improvements or extension of FKI-FMM are expected. Among the others, the Fourier-based structure holds the promise of admitting more kernel functions that are oscillatory beyond the modest extent within the reach of many existing FMM algorithms.

Acknowledgments

This work is supported in part by National Science Foundation, under NSF0905164, NSF0905473 and NSF0811130. The authors gratefully thank the referees for their valuable comments and generous suggestions on improving the initially submitted manuscript.

References

[1] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, Journal of Computational Physics 73 (1987) 325–348.
[2] J. Carrier, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm for particle simulations, SIAM Journal on Scientific and Statistical Computing 9 (1988) 669–686.

[3] L. Greengard, V. Rokhlin, The rapid evaluation of potential fields in three dimensions, Lecture Notes in Mathematics 1360 (1988) 121–141.
[4] L. Greengard, V. Rokhlin, A new version of the fast multipole method for the Laplace equation in three dimensions, Acta Numerica 6 (1997) 229–269.
[5] B. Lu, X. Cheng, J. Huang, J. McCammon, Order *N* algorithm for computation of electrostatic interactions in biomolecular systems, Proceedings of the National Academy of Sciences 103 (2006) 19314–19319.
[6] B. Lu, X. Cheng, J. Huang, J.A. McCammon, AFMPB: an adaptive fast multipole Poisson–Boltzmann solver for calculating electrostatics in biomolecular systems, Computer Physics Communications 181 (2010) 1150–1160.
[7] K. Nabors, S. Kim, J. White, Fast capacitance extraction of general three-dimensional structures, IEEE Transactions on Microwave Theory and Technology 40 (1992) 1496–1505.
[8] K. Nabors, J. White, FastCap: a multipole accelerated 3-D capacitance extraction program, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 10 (1991) 1447–1459.
[9] K. Nabors, J. White, Multipole-accelerated capacitance extraction algorithms for 3-D structures with multiple dielectrics, IEEE Transactions on Circuits and Systems 39 (1992) 946–954.
[10] T. Nishida, K. Hayami, Application of the fast multipole method to the 3-D BEM analysis of electron guns, Boundary Elements XIX (1997) 613–622.
[11] N. Nishimura, K. Yoshida, S. Kobayashi, A fast multipole boundary integral equation method for crack problem in 3D, Engineering Analysis with Boundary Elements 23 (1999) 97–105.
[12] J.S. Zhao, W.C. Chew, MLFMA for solving integral equations of 2-D electromagnetic problems from static to electrodynamic, Microwave and Optical Technology Letters 20 (1999) 306–311.
[13] V. Popov, H. Power, An *O*(*N*) Taylor series multipole boundary element method for three-dimensional elasticity problems, Engineering Analysis with Boundary Elements 25 (2002) 7–18.
[14] J. Tausch, The fast multipole method for arbitrary Green's functions, Contemporary Mathematics 329 (2003) 307–314.
[15] W. Fong, E. Darve, The black-box fast multipole method, Journal of Computational Physics 228 (2009) 8712–8725.
[16] Z. Gimbutas, V. Rokhlin, A generalized fast multipole method for nonoscillatory kernels, SIAM Journal on Scientific Computing 24 (2003) 796–817.
[17] P.G. Martinsson, V. Rokhlin, An accelerated kernel-independent fast multipole method in one dimension, SIAM Journal on Scientific Computing 29 (2007) 1160–1178.
[18] B. Alpert, C. Beylkin, R. Coifman, V. Rokhlin, Wavelet-like bases for the fast solution of second-kind integral equations, SIAM Journal on Scientific Computing 14 (1993) 159–184.
[19] G. Beylkin, R. Cramer, G. Fann, R. Harrison, Multiresolution separated representations of singular and weakly singular operators, Applied and Computational Harmonic Analysis 23 (2007) 235–253.
[20] R. Harrison, G. Fann, T. Yanai, Z. Gan, G. Beylkin, Multiresolution quantum chemistry: basic theory and initial applications, Journal of Chemical Physics 121 (2004) 11587–11598.
[21] L. Ying, G. Biros, D. Zorin, A kernel-independent adaptive fast multipole algorithm in two and three dimensions, Journal of Computational Physics 196 (2004) 591–626.
[22] G.A. Huber, J.A. McCammon, Browndye: a software package for Brownian dynamics, Computer Physics Communications 181 (2010) 1896–1905.
[23] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, Journal of Computational Physics 155 (1999) 468–498.
[24] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, Journal of Computational Physics 82 (1989) 64–84.
[25] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, Journal of Computational Physics 53 (1984) 484–512.
[26] F. Ethridge, L. Greengard, A new fast-multipole accelerated Poisson solver in two dimensions, SIAM Journal on Scientific Computing 23 (2001) 741–760.
[27] H. Cheng, J. Huang, T. Leiterman, An adaptive fast solver for the modified Helmholtz equation in two dimensions, Journal of Computational Physics 211 (2) (2006) 616–637.
[28] B. Zhang, Integral-Equation-Based Fast Algorithms and Graph-Theoretic Methods for Large-Scale Simulations, Ph.D. Dissertation, University of North Carolina at Chapel Hill, 2010.
[29] B. Zhang, J. Huang, N.P. Pitsianis, X. Sun, Dynamic Prioritization for Parallel Traversal of Irregularly Structured Spatio-Temporal Graphs, Tech. Rep. TR-2011-04, Department of Computer Science, Duke University, 2011.