SCIENCE CHINA
Mathematics

• ARTICLES •

May 2012 Vol. 55 No. 5: 1043–1051
doi: 10.1007/s11425-012-4392-0

A pedestrian introduction to fast multipole methods

YING Lexing

Department of Mathematics and ICES, University of Texas at Austin, Austin TX 78712, USA

Email: lexing@math.utexas.edu

Received January 19, 2012; accepted January 31, 2012; published online March 2, 2012

Abstract This paper provides a conceptual and non-rigorous description of the fast multipole methods for evaluating convolution kernel functions with source distributions. Both the non-oscillatory and the oscillatory kernels are considered. For non-oscillatory kernel, we outline the main ideas of the classical fast multipole method proposed by Greengard and Rokhlin. In the oscillatory case, the directional fast multipole method developed recently by Engquist and Ying is presented.

Keywords fast multipole method, non-oscillatory kernels, oscillatory kernels, multiscale methods

MSC(2010) 65N38, 65R20

Citation: Ying L. A pedestrian introduction to fast multipole methods. Sci China Math, 2012, 55(5): 1043–1051, doi: 10.1007/s11425-012-4392-0

1 Introduction

Many computational physics problems require the calculation of the so-called N-body problem: given a set $X \subset \mathbb{R}^d$ of N target points, a set $Y \subset \mathbb{R}^d$ of N source points, a kernel function $G(x, y)$, and a set $\{f(y) : y \in Y\}$ of weights at source points, we compute for each $x \in X$ the potential $u(x)$ defined by

$$u(x) = \sum_{y \in Y} G(x, y) f(y).$$

There are many fields in which this kind of problem appears. In astrophysics, $X = Y$ represents the locations of N stars, and $G(x, y) = 1/|x - y|$ is the gravitational potential. In electrostatics, Y and X represent the locations of N charges and N probes, respectively, and the kernel $G(x, y) = 1/|x - y|$ is the Coulumb potential. In numerical solution of wave scattering in time harmonic regime, $X = Y$ is often a set of discretization points and $G(x, y) = \exp(i\omega|x - y|)/|x - y|$, where ω is the angular frequency of the wave field.

Direct computation of $u(x)$ takes $O(N^2)$ steps, which can be quite costly when N is large. Fast multipole methods (FMMs) were developed in the past two decades and they compute an approximate solution with tunable accuracy in only $O(N \log^\alpha N)$ number of steps for $\alpha = 0, 1$. There are arguably two classes of fast multipole methods: those for non-oscillatory kernels, and those for oscillatory kernels. In this paper, we provide a conceptual and non-rigorous exposition for one member of each class: the classical FMM algorithm for the non-oscillatory Coulumb potential [7, 8] and the directional FMM algorithm for oscillatory Helmholtz kernel [5, 6].

2 Non-oscillatory kernels

Let $X = Y = P$ be a set of N points distributed quasi-uniformly inside the unit box $\Omega = [0,1]^2$ and let $G(x,y) = 1/|x-y|$ (see Figure 1). Our goal is to rapidly compute all pairwise interactions.

Let us start by considering a slightly simpler problem, where B and A are two disjoint squares of the same size, each containing only $O(n)$ points. Direct computation of the potential at points in A induced by the points in B takes $O(n^2)$ steps. However, when A and B are well-separated, there is an easy procedure to approximate the calculation. Let us imagine A and B as two galaxies. When A and B are far away from each other, instead of considering all pairwise interactions, one can sum up the mass in B to obtain $f_B = \sum_{y\in B\cap P} f(y)$ and place it at the center c_B of B, evaluate the potential $u_A = G(c_A, c_B)f_B$ at the center c_A of A as if all the mass is located at c_B, and finally use u_A as the approximation of the potential at each point x in A. A graphical description of this three-step procedure is given in Figure 2. The cost of the three-step procedure is much lower, as it takes only $O(n)$ numbers of steps instead of $O(n^2)$. This procedure works well when A and B are sufficiently far away from each other. However, for the time being, let us assume that this procedure gives a valid approximation as long as A and B are well separated in the sense that the distance between A and B is greater than or equal to the width of A and B.

From an algebraic point of view, the three-step procedure is a rank-1 approximation of the interaction between A and B:

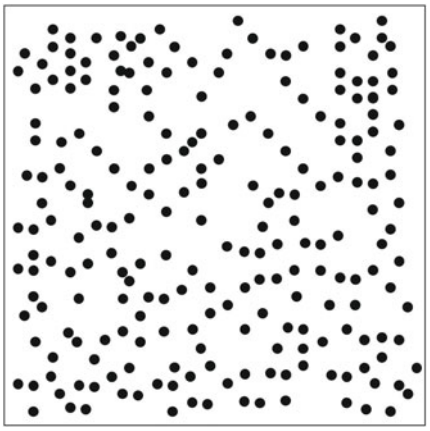

Figure 1 N points quasi-uniformly distributed in a unit box $[0,1]^2$.

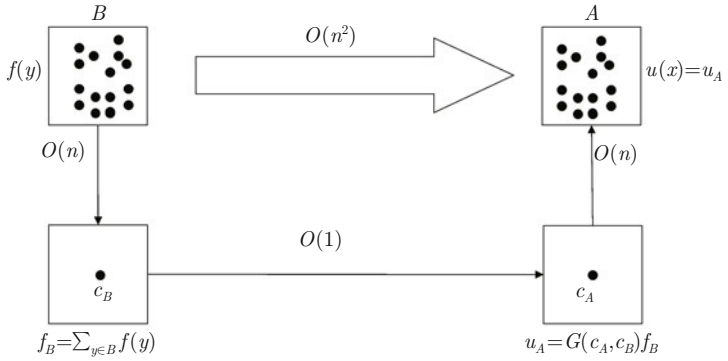

Figure 2 A three-step procedure which efficiently approximates the potential in A induced by the sources in B. The computational cost is reduced from $O(n^2)$ to $O(n)$.

$$G(A,B) \approx \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \cdot G(c_A, c_B) \cdot (1, \ldots, 1), \tag{2.1}$$

and f_A and u_B are just the intermediate results that we get while applying this rank-1 approximation.

This simple problem only considers the potential in A from points in B. However, we are interested in the interaction between all points, so this three-step procedure only partially solve our problem. To get around this, we partition the domain hierarchically with a quadtree structure until the number of points in each leaf box is less than a prescribed $O(1)$ constant (see Figure 3 for an illustration). The whole quadtree then has $O(\log N)$ levels, and we denote the top level as level 0. At level ℓ, there are 4^ℓ squares and each square has $O(N/4^\ell)$ points due to the quasi-uniform point distribution.

The algorithm starts from level 2. Let B be one of the boxes on level 2 (see Figure 4 (top-left)). Let B's *near field* $N(B)$ be the union of B and its neighbors and the *far field* $F(B)$ be the complement of the near field. We define B's *interaction list* to be the set of boxes in B's far field. For a box A in B's far field, the interaction from B to A can be accelerated with the three-step procedure. There are 4^2 possibilities for B and for each B there are $O(1)$ choices for A (see Figure 4 (top-left)). Since both A and B contain $O(N/4^2)$ points due to the quasi-uniformity assumption, the cost for all the interaction that

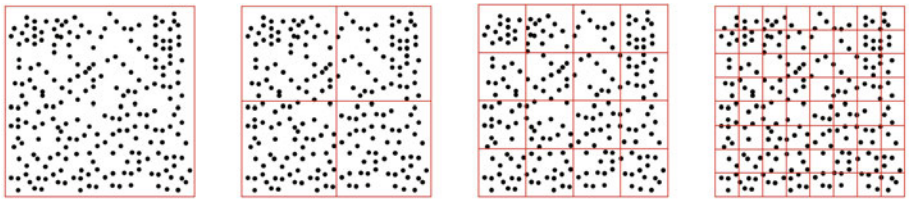

Figure 3 The domain is partitioned with a quadtree structure until the number of points in each leaf node is bounded by a small constant.

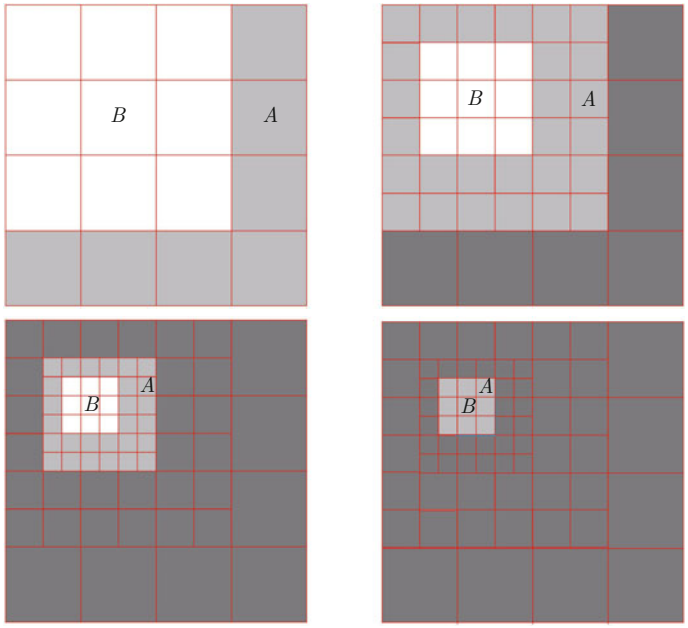

Figure 4 The algorithm at different levels. B stands for a source box. A is a target box for which the interaction with B is processed at the current level. Dark-gray stands for the boxes for which the interaction has already been considered by the previous level. Light-gray stands for the boxes for which the interaction is being considered at the current level. For the first three plots, three-step procedure is used to accelerate the interaction between well-separated boxes. For the last plot, the nearby interaction is handled directly at the leaf level.

can be taken care of on this level is

$$4^2 \cdot O(1) \cdot O(N/4^2) = O(N).$$

For the interaction between B and its near field, we cannot process it on this level, and hence we go down one level in the quadtree.

On the next level, again let B be one square on this level (see Figure 4 (top-right)). We do not need to consider B's interaction with the far field of B's parent since it has already been taken care of by the previous level. Therefore, we only need to consider the interaction between B and its parent's near field. On this level, there are 6^2 boxes in its parent's near field on this level. Out of these boxes, typically 27 of them are well-separated from A and hence the interaction between B and these boxes can be accelerated using the three-step procedure. We define B's interaction list to be the set of these boxes. Since each box on this level contains $O(N/4^3)$ boxes and there are 4^3 possibilities for B, the total cost for the three-step procedures performed on this level is

$$4^3 \cdot O(1) \cdot O(N/4^3) = O(N)$$

again. However, for the interaction of B and its near field, we need to go down again.

For a general level ℓ, there are 4^ℓ choices for B (see Figure 4 (bottom-left)). For each B, there are at most 27 possibilities for A. Since each box on this level has $O(N/4^\ell)$ points, the cost of far field computation is

$$4^\ell \cdot O(1) \cdot O(N/4^\ell) = O(N).$$

Once we reach the leaf level, there is still the interaction between a leaf box B and its neighbors that need consideration (see Figure 4 (bottom-right)). For that, we just use direct computation. Since there are $O(N)$ leaf boxes, each containing $O(1)$ points and having $O(1)$ neighbors, the total cost of direct computation is

$$O(N) \cdot O(1) \cdot O(1) = O(N).$$

For the three-step procedure between a pair of well-separated boxes A and B, it is clear that the first step only depends on B and the last step only depends on A. Therefore, there is an opportunity for reusing computation. Taking this observation into consideration, we can write this algorithm as follows.

- For each level ℓ and each box A on level ℓ, set u_A to be zero.
- For each level ℓ and each box B on level ℓ, compute $f_B = \sum_{y \in B \cap P} f(y)$.
- For each level ℓ and each box B on level ℓ, and for each box A in B's interaction list, update $u_A := u_A + G(c_A, c_B) f_B$.
- For each level ℓ and each box A on this level, update $u(x) := u(x) + u_A$ for each $x \in A \cap P$.
- For each box B on the leaf level, update $u(x) := u(x) + \sum_{y \in N(B) \cap P} G(x, y) f(y)$.

Since the cost at each level is $O(N)$ and there are $O(\log N)$ levels, the whole cost of the algorithm is $O(N \log N)$. This algorithm is in fact the famous tree code, for example see [2].

The question now is whether we can do it in fewer steps. The answer is positive and it is based on the following simple observation: Let $B_1, \ldots, B_4$ be B's children. From $B = B_1 \cup B_2 \cup B_3 \cup B_4$ and all B_i being disjoint, we conclude that (see Figure 5 (left))

$$f_B = f_{B_1} + f_{B_2} + f_{B_3} + f_{B_4}.$$

Therefore, assuming that f_{B_i} are ready, using the previous line to compute f_B is much more efficient than summing over all $f(y)$ in B.

Similarly, for each A, we update $u(x) := u(x) + u_A$ for each $x \in A$. Assume that $A_1, \ldots, A_4$ are the children boxes of A. Then, since we perform the same step for each A_i and each x belongs to one such A_i, we can simply update $u_{A_i} := u_{A_i} + u_A$ instead, which is much more efficient (see Figure 5 (right)).

Notice that in order to carry out these two improvements, we make the assumption that for f_B we visit the parent after the children, while for u_A we visit the children after the parent. This requires

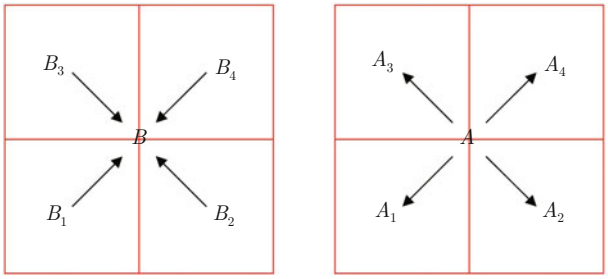

Figure 5 Basic observation that speeds the computation regarding f_B and u_A. Left: f_B can be computed directly from f_{B_i}, where B_i are B's children. Right: instead of adding u_A directly to all the points in A, we only add to u_{A_i}, where A_i are A's children.

one traversing the quadtree with different orders at different stages of the algorithm. Putting the pieces together, we have the following algorithm.

- For each level ℓ and each box A on level ℓ, set u_A to be zero.
- For level $L-1$ to level 0 and for each box B, if B is a leaf box, set $f_B = \sum_{y\in B\cap P} f(y)$. If B is not a leaf box, set $f_B = f_{B_1} + f_{B_2} + f_{B_3} + f_{B_4}$.
- For each level ℓ, for each B, and for each A in B's interaction list, update $u_A := u_A + G(c_A, c_B)f_B$.
- For level 0 to level $L-1$ and for each box A, if A is not a leaf box, update $u_{A_i} := u_{A_i} + u_A$ for each child A_i of A. If A is a leaf box, update $u(x) := u(x) + u_A$.
- For each box B on the leaf level, update $u(x) := u(x) + \sum_{y\in N(B)\cap P} G(x, y)f(y)$.

Steps 1, 3, and 5 are the same as the previous algorithm and their cost is $O(N)$ each. For the second and fourth steps, since there are at most $O(N)$ boxes in the tree and the algorithm spends $O(1)$ steps per box, the cost is again $O(N)$. As a result, the total cost is $O(N)$ as promised. This is essentially the classical FMM algorithm proposed in [7, 8], but significantly simplified.

There are two important points that require some attention. First, f_B represents approximately the charge distribution inside the box B. It is valid for any box A in $F(B)$ when one comes to consider the interaction between A and B, however, it is independent from the specific choice of A. Second, f_B is a compact representation of the source distribution in B, valid only in the far field $F(B)$. In order to be able to compute f_B directly from f_{B_i}, one needs to make sure that f_{B_i} is valid in the far field $F(B)$, or equivalently, $F(B) \subset F(B_i)$ for each B_i. This is indeed true and we call this the nested property of the far field (see Figure 6).

We have been ignoring the accuracy issue so far. In fact, if we use only f_B and u_A, the accuracy will be terrible since A and B are only one box away from each other. Recall that the three-step procedure that we used so far is a poor rank-1 approximation of the interaction between A and B (see (2.1)). In practice, much better low-rank approximations are available and f_B and u_A are just intermediate results.

For example, in the classical FMM algorithm [7, 8], one constructs the low-rank approximation based on analytic properties of the kernel (i.e., complex analysis in 2D and spherical harmonics in 3D). The resulting f_B and u_A are called multipole expansion and local expansions, respectively. In the kernel-independent FMM algorithm (KIFMM) [13], the low-rank approximation is based on the observation that one can reproduce the potential in A induced by points in B by placing some equivalent charges either in B or on its boundary. In this case, the resulting f_B and u_A are respectively called equivalent charges and check potentials. We shall not go into the details about these representation except for two essential points.

- All f_B and u_A contain $O(1)$ numbers, so they are very effective.
- There are operators $T_{B;B_i}$ that take f_{B_i} to f_B ($f_B = \sum_i T_{B;B_i} f_{B_i}$), operators $T_{A_i;A}$ that take u_A to u_{A_i} ($u_{A_i} := u_{A_i} + T_{A_i;A}u_A$), and operators $T_{A;B}$ that take f_B to u_A ($u_A := u_A + T_{A;B}f_B$). All of these operators can be implemented efficiently. Following the conventions of classical FMM, we call $T_{B;B_i}$ the M2M operator, $T_{A_i;A}$ the L2L operator, and $T_{A;B}$ the M2L operator.

There are many other ways to implement the low-rank approximation, besides the two mentioned

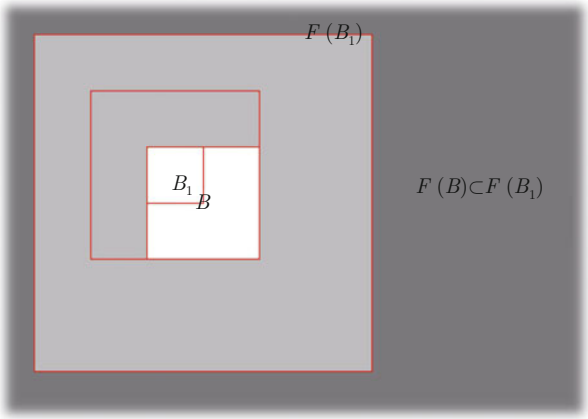

Figure 6 Nested property of the far field. The far field $F(B)$ of B is contained in the far field $F(B_i)$ for any child B_i of B.

above. Well-known examples include H^2-matrices [3] and the multipole method without multipole [1].

3 Oscillatory kernels

In this section, we consider the oscillatory kernel $G(x,y) = \exp(i2\pi K|x-y|)/|x-y|$, where K is a large constant. The points $X = Y = P$ are a set of $N = O(K)$ points sampled quasi-uniformly from a smooth curve in $[0,1]^2$ (see Figure 7). The wavelength λ is $1/K$ and the average distance between adjacent points is $O(\lambda)$. An $O(N \log N)$ algorithm for this problem was first discovered in [10, 11]. In this section, we will describe an alternative with the same computational cost, presented in [5, 6].

Based on the discussion of the non-oscillatory case, we need to have a notion of well-separatedness and for well-separated regions their interaction must be of numerically low-rank. Let us see whether the geometric configuration used in the non-oscillatory kernel is able to yield a low-rank approximation here. Consider a box B and its far field $F(B)$: for the Helmholtz kernel, the numerical rank for the interaction between in fact scales like $O(\text{diam}(B)/\lambda)$ [10]. Therefore, the rank goes higher for larger boxes, and if B is a constant fraction of the domain, then the rank is $O(K)$. Hence, the three-step procedure is no longer effective and we have to come up with something different.

The solution is to consider a different geometric configuration (see Figure 8). Let B be a box with diameter $s\lambda$. We define $W_{B,d}$ to be a wedge in the direction of d, of distance $s^2\lambda$ away from B, and with an opening angle of size $O(1/s)$. Hence, when B gets larger and larger, the wedge becomes further and further away quadratically with a smaller opening angle. It is proved in [5,6] that the interaction between B and $W_{B,d}$ for any direction d is numerical low rank now. If we define $F(B)$ now to be the set which is $O(s^2\lambda)$ away from B, then $F(B)$ is a disjoint union of $O(s)$ wedges $\{W_{B,d}\}$. The complement of $F(B)$ is the new near field $N(B)$. The low-rank approximation actually used is very similar to the equivalent charges used in the kernel-independent FMM. So we will continue to call the intermediate results $f_{B,d}$ the directional equivalent charges and $u_{A,d}$ the directional check potentials.

The nested property of the far fields still hold in this case: for any $W_{B,d}$ of a box B, there exists a direction d' such that for each child B_i we have $W_{B,d} \subset W_{B_i,d'}$ (see Figure 9). This property ensures that, given $f_{B_i,d'}$, we can compute $f_{B,d}$ efficiently from them. We can also design efficient translation operators: $T_{B_i,d';B,d}$ from $f_{B,d}$ to $f_{B_i,d'}$, $T_{A,d;A_i,d'}$ from $u_{A,d}$ to $u_{A_i,d'}$, and $T_{A,-d;B,d}$ from $f_{B,d}$ to $u_{A,-d}$. All of these operators are of low computational cost.

The hierarchical decomposition still uses a quadtree structure, adaptively partitioned until each leaf box has $O(1)$ points. Since the points are quasi-uniformly distributed on the curve, we can assume that all leaf boxes are at the same level $L - 1 = O(\log N) = O(\log K)$. On a level ℓ, the number of non-empty boxes is $O(2^\ell)$. The algorithm now starts from the level with boxes of width $\sqrt{K}\lambda$. The reason is that for larger boxes, the far field is outside the domain $[0,1]^2$ and hence we do not need to do any work. The

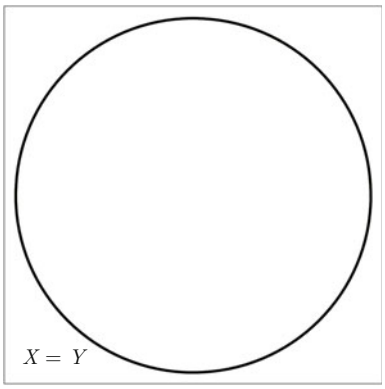

Figure 7 Points in $X = Y$ are samples from a smooth curve in $[0, 1]^2$.

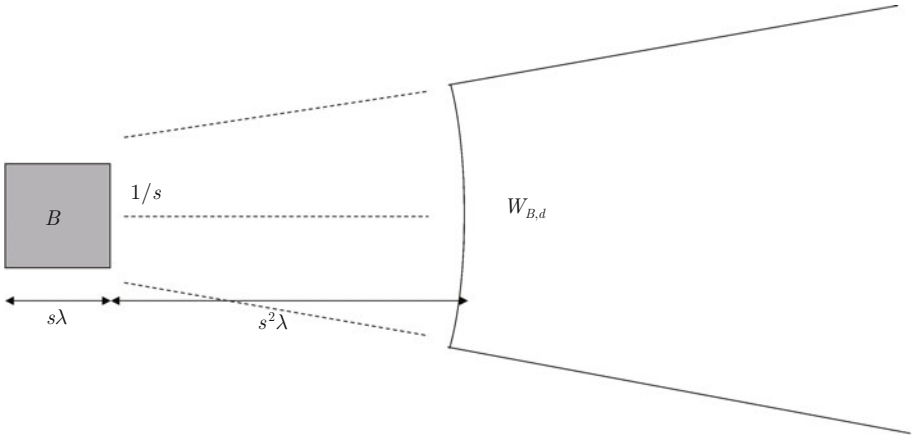

Figure 8 For box B and $W_{B,d}$, the interaction between them is numerically low-rank. Suppose the width of B is $s\lambda$. Then $W_{B,d}$ is at least $s^2\lambda$ away and with opening angle of size $O(1/s)$.

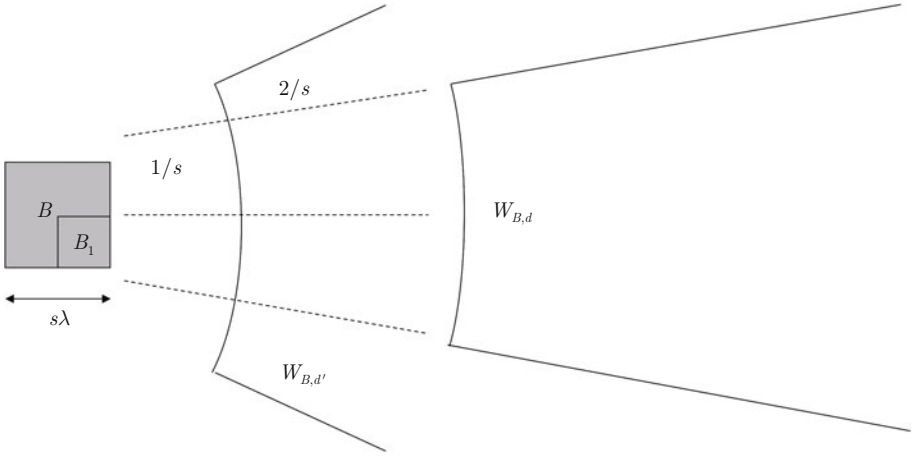

Figure 9 The nested property still holds between B and its children.

algorithm proceeds as follows.

At the level where all boxes have width $\sqrt{K}\lambda$, there are $K/\sqrt{K}$ boxes, each with $O(\sqrt{K})$ boxes in the far field (see Figure 10 (top-left)). The interaction between each well-separated pair takes only $O(1)$ steps, assuming the directional equivalent sources are ready. Therefore, the cost is

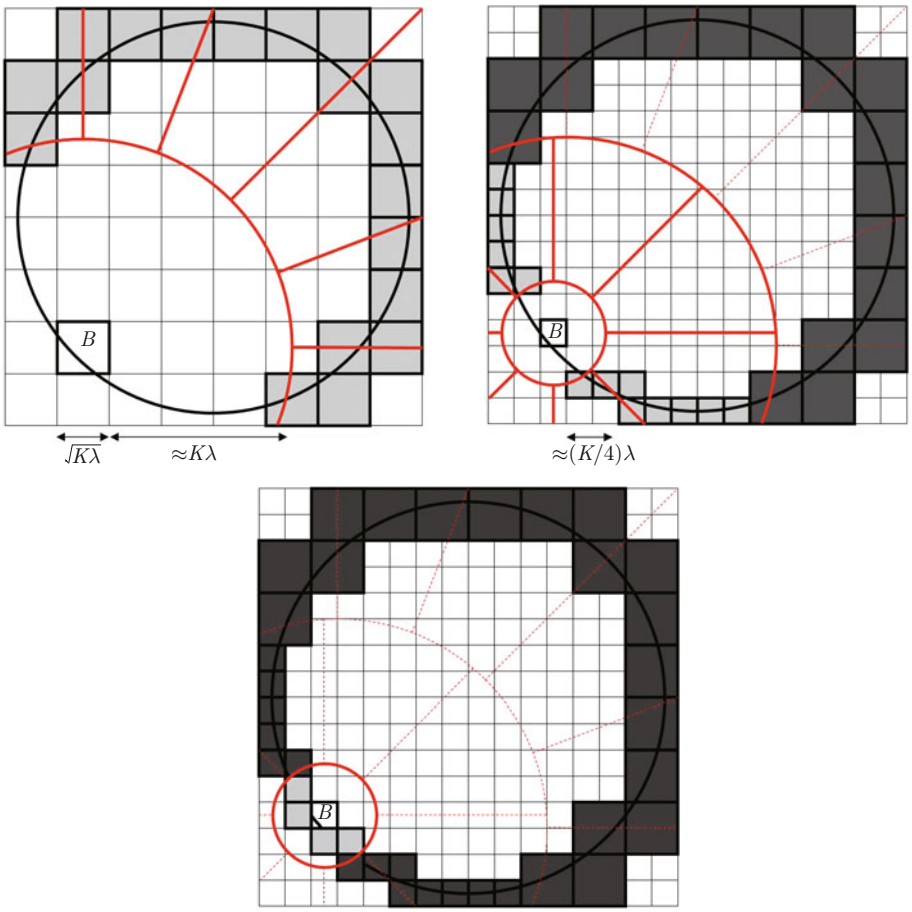

Figure 10 The algorithm at different levels. B stands for a source box at the current level. Its far field is partitioned into wedges, each of which contains several non-empty boxes. Dark-gray stands for the boxes for which the interaction has already been considered. Light-gray stands for the boxes for which the interaction is being considered on the current level. For the last plot, the nearby interaction is considered at the leaf level.

$$O(\sqrt{K}) \cdot O(\sqrt{K}) \cdot O(1) = O(K).$$

For each B, there are boxes in the near field and the interaction associated with them needs to be taken care in the next level.

In the next level, the boxes are of width $(\sqrt{K}/2)\lambda$ and there are so $O(K/(\sqrt{K}/2)) = O(2\sqrt{K})$ of them (see Figure 10 (top-right)). For each of them, the boxes in the interaction list are the ones which are at least $(\sqrt{K}/2)^2\lambda$ away but within distance $\sqrt{K}^2\lambda$. Since the points are sampled on a curve, the number of non-empty boxes on this level fall in this range is of size $O(K\lambda/(\sqrt{K}/2\lambda)) = O(\sqrt{K}/2)$. Therefore, the total cost on this level is

$$O(2\sqrt{K}) \cdot O(\sqrt{K}/2) \cdot O(1) = O(K).$$

As one goes down the tree, the number of non-empty boxes on a level doubles. On the other hand, the number of non-empty boxes in any box's interaction list halves due to the quadratic nature of the far field. Therefore, their product is always $O(K)$ and hence the cost is always $O(K)$ as well.

Finally, when one reaches the leaf level (see Figure 10 (bottom)), we need to perform nearby calculation for leaf boxes and their neighbors. Similar to the non-oscillatory case, this cost is $O(K)$, the same. Adding all these components together we see that the overall cost is $O(K \log K) = O(N \log N)$. More specifically, the algorithm proceeds as follows.

- Go up the tree. For each leaf box B and direction d, form $f_{B,d} = \sum_{y \in B \cap P} f(y)$. For each non-leaf box B, form $f_{B,d}$ from its children boxes using $f_{B_i,d'}$.

• Go through the tree. For each pair B and A with A in B's interaction list, update $u_{A,-d} := u_{A,-d} + T_{A,-d;B,d} f_{B,d}$.

• Go down the tree. For each non-leaf box A and direction d, transform $u_{A,d}$ and add to $u_{A_i,d'}$ for each child A_i of A. If A is a leaf, add $u_{A,d}$ to $u(x)$ for all $x \in A \cap P$.

• For each leaf B, update $u(x) = u(x) + \sum_{y \in N(B) \cap P} G(x,y) f(y)$.

This is essentially a simplified version of the directional FMM algorithm proposed in [5,6].

4 Conclusions

In this paper, we introduced two examples of the fast multipole methods, one with a non-oscillatory kernel and the other with an oscillatory kernel. Both of them are based on low-rank approximations between well-separated regions, with an appropriately definition of well-separatedness. In the second case, the low-rank approximation comes from a more delicate geometric configuration and is more complex implementation-wise. The combination of low-rank approximations and hierarchical decompositions provides us with a very powerful tool in developing efficient algorithms for various integral operators arising from partial differential equations, with recent examples including hierarchical matrices [3], butterfly algorithm for oscillatory interaction [4,9,12], and so on.

Acknowledgements This work was partially supported by the Sloan Foundation and the National Science Foundation of USA (CAREER Award DMS-0846501).

References

1 Anderson C R. An implementation of the fast multipole method without multipoles. SIAM J Sci Statist Comput, 1992, 13: 923–947
2 Barnes J, Hut P. A hierarchical $O(N \log N)$ force-calculation algorithm. Nature, 1986, 324: 446–449
3 Börm S, Grasedyck L, Hackbusch W. Hierarchical matrices. Technical Report 21, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, 2003
4 Candès E, Demanet L, Ying L. A fast butterfly algorithm for the computation of Fourier integral operators. SIAM Multiscale Model Simul, 2009, 7: 1727–1750
5 Engquist B, Ying L. Fast directional multilevel algorithms for oscillatory kernels. SIAM J Sci Comput, 2008, 29: 1710–1737
6 Engquist B, Ying L. A fast directional algorithm for high frequency acoustic scattering in two dimensions. Commun Math Sci, 2009, 7: 327–345
7 Greengard L. The rapid evaluation of potential fields in particle systems. ACM Distinguished Dissertations. Cambridge, MA: MIT Press, 1988
8 Greengard L, Rokhlin V. A fast algorithm for particle simulations. J Comput Phys, 1987, 73: 325–348
9 O'Neil M, Woolfe F, Rokhlin V. An algorithm for the rapid evaluation of special function transforms. Appl Comput Harmon Anal, 2010, 28: 203–226
10 Rokhlin V. Rapid solution of integral equations of scattering theory in two dimensions. J Comput Phys, 1990, 86: 414–439
11 Rokhlin V. Diagonal forms of translation operators for the Helmholtz equation in three dimensions. Appl Comput Harmon Anal, 1993, 1: 82–93
12 Ying L. Sparse Fourier transform via butterfly algorithm. SIAM J Sci Comput, 2009, 31: 1678–1694
13 Ying L, Biros G, Zorin D. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. J Comput Phys, 2004, 196: 591–626