SIAM J. Sci. Comput. © 1998 Society for Industrial and Applied Mathematics
Vol. 19, No. 2, pp. 635–656, March 1998 019

PROVABLY GOOD PARTITIONING AND LOAD BALANCING ALGORITHMS FOR PARALLEL ADAPTIVE N-BODY SIMULATION*

SHANG-HUA TENG[†]

Abstract. We present an efficient and provably good partitioning and load balancing algorithm for parallel adaptive N-body simulation. The main ingredient of our method is a novel geometric characterization of a class of communication graphs that can be used to support hierarchical N-body methods such as the fast multipole method (FMM) and the Barnes–Hut method (BH). We show that communication graphs of these methods have a good partition that can be found efficiently sequentially and in parallel. In particular, we show that an N-body communication graph (either for BH or for FMM) can be partitioned into two subgraphs with equal computation load by removing only $O(\sqrt{n \log n})$ and $O(n^{2/3}(\log n)^{1/3})$ number of nodes, respectively, for two and three dimensions. These bounds on node-partition imply bounds on edge-partition of $O(\sqrt{n}(\log n)^{3/2})$ and $O(n^{2/3}(\log n)^{4/3})$, respectively, for two and three dimensions. To the best of our knowledge, this is the first theoretical result on the quality of partitioning N-body communication graphs for nonuniformly distributed particles. Our results imply that parallel adaptive N-body simulation can be made as scalable as computation on regular grids and as efficient as parallel N-body simulation on uniformly distributed particles.

Key words. adaptive computing, the fast multipole method, hierarchical methods, load balancing, N-body simulation, partitioning, parallel processing, scientific computing, tree-codes

AMS subject classifications. 05C50, 65F50, 68R10

PII. S1064827595288942

1. Introduction.

The most commonly used approaches for numerical simulation of complex physical problems are the mesh-based numerical approach (to solving partial differential equations) [29] and the particle simulation method [14]. In addition to its direct applications in astrophysics, molecular dynamics, plasma physics, and fluid dynamics, the particle methods, especially the recently developed N-body techniques, have been applied to boundary element methods and the eigensystem analysis of tridiagonal linear systems.

A major challenge of real-world scientific problems is solving very large scale computation problems. For example, we need to simulate hundreds of millions or even thousands of millions of particles for most of the applications mentioned above. The use of high-performance computers, in the form of either tightly coupled parallel machines or relatively loosely coupled workstation clusters, is inevitable.

However, equally importantly, the use of more advanced and more efficient methods is critical to solving large-scale problems on parallel machines. For N-body simulation, though the $O(n^2)$-time direct method can be easily parallelized, more sophisticated methods such as hierarchical methods are needed to reduce the total number of operations to much more slowly growing functions (e.g., $O(n \log n)$ or $O(n)$).

The hierarchical methods include the $O(n)$-time algorithms of Greengard and Rokhlin [13], Anderson [1], and Zhao [33], and the $O(n \log n)$-time algorithms (with

*Received by the editors July 5, 1995; accepted for publication (in revised form) May 7, 1996.
http://www.siam.org/journals/sisc/19-2/28894.html

[†]Department of Computer Science, University of Minnesota, Minneapolis, MN 55455 (steng@cs.umn.edu). This research was supported in part by NSF CAREER award CCR-9502540, an Alfred P. Sloan Research Fellowship, and an Intel research grant. Part of this work was done while the author was at Xerox Palo Alto Research Center, Palo Alto, CA, and at the Department of Mathematics and the Laboratory for Computer Science, MIT, Cambridge, MA.

a smaller constant in the big-O) of Appel[1] [2] and Barnes and Hut [3]. These methods were originally designed for sequential machines and have been implemented on various parallel platforms by several groups of researchers [5, 12, 15, 16, 17, 23, 24, 26, 28, 34].

In order to fully utilize a parallel machine with a distributed memory to solve a large problem, we need to efficiently (in parallel) decompose the problem into a given number of subproblems, each with roughly equal computational requirements, to balance the load. However, to achieve a close-to-optimal speed-up on a parallel machine, in addition to balancing load, we need to simultaneously preserve locality and minimize the communication overhead. The load balancing and partitioning problems are much better understood for uniformly distributed particles but are much harder for nonuniformly distributed particles, where adaptive hierarchical methods are required. Several heuristics have been developed and evaluated experimentally [24, 28]. However, to the best of our knowledge, no provably good bounds are known for these heuristics.

The time complexity of the N-body algorithms mentioned above is valid only for “reasonably” uniform distributions of particles. The algorithms of Appel [2] and Barnes and Hut [3] are readily extended to nonuniformly distributed particles. Carrier, Greengard, and Rokhlin [7] presented an adaptive version of Greengard and Rokhlin’s fast multipole method (FMM) (see Procedure Interaction-Pair in section 2.4). Similar extensions can be made to Anderson’s [1] and Zhao’s [33] algorithms. As particles move close to each other nonuniformly, the time complexity of these extensions becomes superlinear—even exceeding $O(n \log n)$ for some distributions. There is an elegant $O(n \log n)$-time algorithm given by Callahan and Kosaraju [6] which is independent of particle distributions. They obtained this complexity using the notion of well-separated-pair decomposition of particles. Some ideas of their construction, such as shrinking boxes, were originally proposed by Vaidya in the context of computing nearest neighbor graphs [35]. They also show that their algorithm can be parallelized to run on a PRAM of n processors in $O(\log n)$ time.

However, in practice, the depth of the hierarchical structure used in these N-body algorithms is limited by the machine precision as pointed out by Greengard. Consequently, the benefit from Callahan and Kosaraju becomes apparent on machines with high precision. Callahan and Kosaraju’s parallel N-body algorithm runs on the PRAM model, and hence the communication cost and the load balancing issue are not considered explicitly. In this paper, we develop a partitioning and load balancing algorithm which can be applied to adaptive hierarchical N-body methods on “practical” particle distributions and machine architectures. We will make an assumption on nonuniformity (see section 2.3) which allows us to derive bounds for a variety of distributions while retaining the applicability to real-life particle problems.

There are two major difficulties in load balancing adaptive N-body methods. First, the communication pattern (graph) is not explicitly known a priori. This makes the load balancing problem for the parallel N-body simulation harder than that for parallel unstructured-mesh computations. Second, the communication pattern in nonuniform N-body simulations is much more irregular than that of unstructured-mesh computations—not all communications are between near neighbors. So the partitioning problem is much harder even though the communication pattern is known in advance.

This paper makes two key contributions:

[1] The complexity of Appel’s algorithm has been shown to be of $O(N)$ [9].

1. We show that the communication graphs (implicitly) used in FMM and the Barnes–Hut (BH) method for nonuniform distributed particles have a good partition. In particular, we show that an N-body communication graph (either a BH graph or a FMM graph) can be partitioned into two subgraphs of roughly equal computation load by removing only $O(\sqrt{n \log n})$ and $O(n^{2/3}(\log n)^{1/3})$ number of nodes, respectively, for two and three dimensions. These bounds on node-partition imply bounds on edge-partition of $O(\sqrt{n}(\log n)^{3/2})$ and $O(n^{2/3}(\log n)^{4/3})$, respectively. Our result implies, from the load balancing viewpoint, that any N-body communication graph has a partition of quality almost as good as that for regular grids. Therefore, parallel adaptive N-body simulation can, in theory, be as scalable as computations on regular grids and as efficient as N-body simulation on uniformly distributed particles.
2. We develop an efficient parallel algorithm for finding such a partition. More significantly, our algorithm does not need an explicit representation of the N-body communication graph. Our algorithm can partition an N-body communication graph directly from the input positions of particles. We then use our partitioning algorithm in the design of an efficient parallel divide-and-conquer algorithm for constructing the N-body communication graph.

The main ingredient of our method is a novel geometric characterization of the communication graphs used in hierarchical N-body methods. Such a geometric characterization not only enables us to prove rigorously that an N-body communication graph has a good partition but also provides an efficient parallel method for finding such a partition.

In section 2, we briefly review hierarchical N-body methods and introduce notation that will be used in this paper. In section 3, we define and characterize the communication graphs used in FMM and BH. We present a refinement of FMM to make it more suitable for nonuniformly distributed particles. In section 4, we study the geometric structure of N-body communication graphs and show that any such communication graph can be partitioned into two subgraphs of roughly equal size by removing only $O(n^{1-1/d}(\log n)^{1/d})$ $(d = 2, 3)$ nodes. In section 5, we incorporate the computational requirement into the communication graph and present an efficient parallel load balancing algorithm. In section 6, we exploit our partitioning method to develop an efficient parallel algorithm for computing the communication graph directly from the positions of the particles. In section 7, we discuss our avenues for future research plans and give some open questions.

2. Hierarchical methods for N-body simulation.

The crux of hierarchical N-body methods is to decompose the potential at a point $\boldsymbol{x}$, $\phi(\boldsymbol{x})$, into the sum of two potentials: $\phi_N(\boldsymbol{x})$, the potential induced by "near-field" particles; and $\phi_F(\boldsymbol{x})$, the potential due to "far-field" particles [2, 3, 13, 33]. In these methods, $\phi_N(\boldsymbol{x})$ is computed exactly, while $\phi_F(\boldsymbol{x})$ is computed approximately.

2.1. Well-separated clusters for far-field approximation.

The approximation is based on a notion of well-separated clusters [3, 13]. Suppose we have two clusters A and B, whose "centers" are distance r away. See Fig. 1.

Direct potential calculation for all particles in A induced by particles in B (and vice versa) requires $O(|A||B|)$ operations. If r is much larger than both r_1 and r_2; i.e., A and B are "well separated," then we can use the pth-order multipole expansion to express the pth-order approximation of potential due to all particles in B. Let $\Phi_B^p(\boldsymbol{x})$ denote such a multipole expansion (for details see [1, 13, 33]). To (approximately)

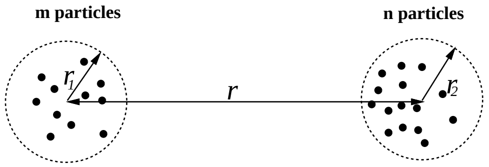

FIG. 1. *Well-separated clusters.*

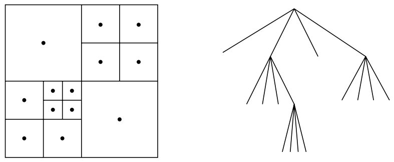

FIG. 2. *Quadtree.*

compute the potential at particles in A, we simply evaluate $\Phi_B^p()$ at each particle in A. Suppose $\Phi_B^p()$ has $g(p, d)$ terms. Using multipole expansion, we reduce the number of operations to $g(p, d)(|A| + |B|)$. The error of the multipole expansion depends on p and the ratio $\max(r_1, r_2)/r$. We say A and B are *β-well-separated*, for a $\beta > 2$, if $\max(r_1, r_2)/r \leq 1/\beta$. As shown in [13], the error of the pth-order multipole expansion is bounded by $(1/(\beta - 1))^p$.

The above example illustrates the *particle–cluster* interaction between two well-separated clusters. Greengard and Rokhlin [13] showed that the *cluster–cluster* intersection among well-separated clusters can further improve the hierarchical method. Suppose we have k clusters $B_1, \ldots, B_k$ that are well separated from a cluster A. Let $\Phi_i^p()$ be the pth-order multipole expansion of B_i. Using particle–cluster interaction to approximate the far-field potential at A, we need to perform

$$g(p, d)|A|(|B_1| + |B_2| + \cdots + |B_k|)$$

operations. Greengard and Rokhlin showed that from $\Phi_i^p()$ we can efficiently compute a local expansion $\Psi_i^p()$ centered at the centroid of A that approximates $\Phi_i^p()$. Such an operation of transforming $\Phi_i^p()$ to $\Psi_i^p()$ is called a *FLIP*. The cluster–cluster interaction first flips $\Phi_i^p()$ to $\Psi_i^p()$; we then compute $\Psi_A^p() = \sum_{i=1}^k \Psi_i^p()$ and use $\Psi_A^p()$ to evaluate the potential at each particle in A. This reduces the number of operations to the order of

$$g(p, d)(|A| + |B_1| + |B_2| + \cdots + |B_k|).$$

2.2. Hierarchical decomposition for a canonical set of boxes.

Hierarchical N-body methods use quadtree (for two dimensions) and octree (for three dimensions) to generate a canonical set of boxes to define clusters. The number of boxes is typically linear in the number of particles, i.e., $O(n)$.

A *quadtree* [25] is a recursive partition of a region of the plane into axis-aligned squares (see Fig. 2). One square, the *root*, covers the entire set of particles. It is often chosen to be the smallest (up to a constant factor) square that contains all particles. A square can be divided into four *child* squares by splitting it with horizontal and vertical line segments through its center. The collection of squares then forms a tree, with smaller squares at lower levels of the tree. The recursive decomposition is often adaptive to the local geometry. The most commonly used termination condition is

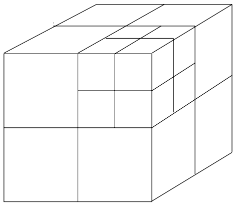

Fig. 3. *Octree.*

as follows: the division stops when a box contains less than some constant (typically $m = 100$) number of particles. *Octree*s are the three-dimensional version of quadtrees (see Fig. 3). The root of an octree is a box covering the entire set of particles. An octree is constructed by recursively and adaptively dividing a box into eight child-boxes, by splitting it with hyperplanes normal to each axis through its center (see Fig. 3).

In the remainder of the paper, we refer to both quadtree and octree as *hierarchical trees.*

For a set of uniformly distributed n points, the leaf-boxes of the hierarchical tree form a regular grid. The height of the tree, the tree distance from a leaf to the root, is $\log_4(n/m)$ and $\log_8(n/m)$, respectively, for two and three dimensions.

In practical simulations, particles are usually not uniformly distributed. Particles may be highly clustered in some regions and relatively scattered in some other regions. Thus, the hierarchical tree is adaptively generated, with smaller boxes for regions of clustered particles. The computation and communication patterns of a hierarchical method become more complex and often are not known explicitly in advance.

2.3. An assumption on nonuniform distributions. In this paper, we use the following notion of nonuniformity: we say a point set $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ in d dimensions is *μ-nonuniform* if the height of the hierarchical tree generated for P is $\log_{2^d}(n/m) + \mu$. In other words, the ratio of the size of smallest leaf-box to the root-box is $1/2^{\log_{2^d}(n/m)+\mu}$. In practice, μ is less than 100.

Our notion of nonuniformity is very closely related with the precision of a point set. One measurement of the precision, in the context of hierarchical decomposition, is height of the hierarchical tree that separates all points. Another common measurement is the logarithm of the ratio of the distance between the farthest two points to the distance between the closest two points.

2.4. The BH and FM methods. We now review the basic steps of BH[2] and FMM for evaluating far-field interactions. Both methods start with the construction of a hierarchical tree, followed by the computation of the multipole expansion at each box in the hierarchical tree. BH uses the *particle–box* interaction, while FMM uses the *box–box* interaction, i.e., FLIP, to further reduce the number of operations.

The original particle–box evaluation procedure of BH can be described as follows. To evaluate the potential for a particle at $\boldsymbol{p}$, we start with the root-box of the hierarchical tree and recursively perform the following operations: if $\boldsymbol{p}$ is well separated from the current box $\boldsymbol{b}$, then evaluate $\boldsymbol{p}$ against the multipole expansion of $\boldsymbol{b}$ and add the contribution of $\boldsymbol{b}$ to the potential of $\boldsymbol{p}$; otherwise, if $\boldsymbol{b}$ is not a leaf-box and is

[2] The original BH uses the total mass and the center of mass to approximate a box-cluster. It can be shown that such an approximation is second order. In this paper, we consider its general higher-order formulation.

not well separated from $\boldsymbol{p}$, then recursively compare $\boldsymbol{p}$ with all child-boxes of $\boldsymbol{b}$. If $\boldsymbol{b}$ is a leaf-box, then all particles of $\boldsymbol{b}$ are near-field particles of $\boldsymbol{p}$, and hence direct calculations are performed.

We now describe the adaptive FMM for nonuniformly distributed particles due to Carrier, Greengard, and Rokhlin [7]. The method uses the box–box interaction. FMM tries to maximize the number of FLIPs among large boxes and also tries to FLIP between roughly equal sized boxes, a philosophy which can be described as follows: *let parents do as much work as possible and then do as much of the left-over work as possible before passing to the next generation.* We first describe a somewhat simpler algorithm than the original algorithm given in [7]. We will say more about that algorithm in section 3.1, where we will give a refined adaptive algorithm. Let $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{2^d}$ be the set of child-boxes of the root-box of the hierarchical tree. FMM generates the set of all *interaction-pairs* of boxes by taking the union of Interaction-pair$(\boldsymbol{c}_i, \boldsymbol{c}_j)$ for all $1 \leq i < j \leq 2^d$, using the interaction-pair procedure defined below.

Procedure Interaction-Pair $(\boldsymbol{b}_1, \boldsymbol{b}_2)$

- If $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are β-well-separated, then $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ is an interaction-pair.
- Else if both $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are leaf-boxes, then particles in $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are near-field particles.
- Else if neither $\boldsymbol{b}_1$ nor $\boldsymbol{b}_2$ is a leaf-box, without loss of generality, assuming that $\boldsymbol{b}_2$ is at least as large as $\boldsymbol{b}_1$ and letting $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{2^d}$ be the child-boxes of $\boldsymbol{b}_2$, then recursively decide interaction-pair by calling interaction-pair($\boldsymbol{b}_1$,$\boldsymbol{c}_i$) for all $1 \leq i \leq 2^d$.
- Else if one of $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ is a leaf-box, without loss of generality, assuming that $\boldsymbol{b}_1$ is a leaf-box and letting $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{2^d}$ be the child-boxes of $\boldsymbol{b}_2$, then recursively decide interaction-pairs by calling interaction-pair($\boldsymbol{b}_1$,$\boldsymbol{c}_i$) for all $1 \leq i \leq 2^d$.

FMM calculates the far-field interactions as follows: for each interaction-pair $(\boldsymbol{b}_1, \boldsymbol{b}_2)$, letting $\Phi_i^p()$ $(i = 1, 2)$ be the multipole expansion of $\boldsymbol{b}_i$, FLIP $\Phi_1^p()$ to $\boldsymbol{b}_2$ and add to $\boldsymbol{b}_2$'s potential local expansion. Similarly, FLIP $\Phi_2^p()$ to $\boldsymbol{b}_1$ and add to $\boldsymbol{b}_1$'s potential local expansion. Then traverse down the hierarchical tree in a pre-ordering, shift, and add the potential local expansion of the parent-box of a box to its own local expansion.

Note that FMM for uniformly distributed particles has a much simpler description (see [13, 33]) than what we described above. In Carrier, Greengard, and Rokhlin [7] the set of intersection pairs is defined more directly. Furthermore, their set of intersection pairs is a subset of the one constructed by the procedure above, and hence their formulation requires fewer operations than the simple formulation that we give above. See section 3.1 for more details.

3. Communication graphs for N-body simulations.

In order to efficiently implement an N-body method on a parallel machine, we need to understand its communication pattern, which can be described by a graph that characterizes the pattern of information exchange during the execution of the method. The communication graph is defined on basic computational elements of the method. The basic elements of hierarchical N-body methods are boxes and points, where points give the locations of particles and boxes are generated by the hierarchical method. Formally, the communication graph is an edge-weighted directed graph, where the edges describe the

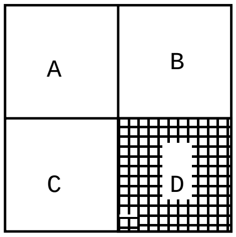

FIG. 4. *A nonuniform example.*

pattern of communication and the weight on an edge specifies the communication requirement along the edge.

In this section, we define the communication graph for parallel N-body simulation. In section 3.1, we show how to apply the idea of BH to improve parallel adaptive FMM. In section 3.2, we show how to use FMM communication patterns to support parallel BH implementation. In the next section, we will study the geometric structure of this class of graphs and design a provably good partitioning and load balancing algorithm.

3.1. A refined FMM for nonuniform distributions. *For parallel implementation, it is desirable to have a communication graph that uses small edge-weights and has small in- and out-degrees.* However, some boxes in the set of interaction-pairs defined in the last section may have large degree! Such an example is given in Fig. 4.

Suppose the root-box is divided into four child-boxes A, B, C, and D. Assume further that boxes A, B, and C contains less than m (< 100) particles, and most particles, say n of them, are uniformly distributed in D. In the FMM algorithm of the previous section, we further recursively divide D by $\log_4(n/m)$ levels. Notice that A, B, and C are not well separated from any box in D. Hence the FMM algorithm of the previous subsection will declare all particles of D as near-field particles of A, B, and C (and vice versa). The drawback is two-fold:

1. From the computational viewpoint, we cannot take advantage of the hierarchical tree of D to evaluate potentials in A, B, and C.
2. From the communication viewpoint, boxes A, B, and C have a large in-degree in the sense that each particle in these boxes needs to receive information from all n particles in D, making partitioning and load balancing harder.

Carrier, Greengard, and Rokhlin [7], from the computational viewpoint, presented an efficient way to handle this case when a large leaf-box is next to many small leaf-boxes. Instead of assigning all small leaf-boxes (such as those in D) to be near-field boxes of the large leaf-box (e.g., C in our example), they only make those small leaf-boxes adjacent to the large box to be its near-field boxes. (See list 1, denoted by U, defined in [7].) Particle–box interactions similar to BH are then used to evaluate particles in the large box against the hierarchical structure built on other smaller leaf-boxes. This approach clearly reduces the number of operations. However, the degree of the large leaf-box in the near-field interaction can still be very large. In the example above, such contribution to the degree of D is about $\sqrt{n/m}$.

We now show that the particle–box interaction idea of BH adopted in [7] can be used to further reduce the degree of the large box to $O(\log n)$.

Notice that in BH most boxes of D are well separated from particles in A, B, and C. Hence the well-separation condition is different in BH: *because BH uses the particle–box interaction, the well-separation condition is measured with respect to the size of the boxes in D. Thus most boxes are well separated from particles in A, B,* *and C. In contrast, because FMM applies the FLIP operation, the well-separation condition must measure up against the size of the larger box. Hence no box in D is well separated from A, B, and C.*

Our refined FMM algorithm circumvents this problem by incorporating the well-separation condition of BH into Procedure Interaction-Pair: if $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are not well separated, and $\boldsymbol{b}_1$, the larger of the two, is a leaf-box, then we use a well-separation condition with respect to $\boldsymbol{b}_2$, instead of to $\boldsymbol{b}_1$, and apply the FLIP operation directly onto particles in the leaf-box $\boldsymbol{b}_1$ rather than $\boldsymbol{b}_1$ itself.

We will define this new well-separation condition shortly. First, we make the following observation about Procedure Interaction-Pair. We can prove, by a simple induction, the following fact: if $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are an interaction-pair and both $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are not leaf-boxes, then $1/2 \leq size(\boldsymbol{b}_1)/size(\boldsymbol{b}_2) \leq 2$. This is precisely the condition that FMM would like to maintain. For uniformly distributed particles, such a condition is always true between any interaction-pair (even if one of them is a leaf-box). However, for nonuniformly distributed particles, if $\boldsymbol{b}_1$, the larger box, is a leaf-box, then $\boldsymbol{b}_1$ could be much larger than $\boldsymbol{b}_2$.

The new *β-well-separation* condition, when $\boldsymbol{b}_1$ is a leaf-box, is then defined as follows: $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are *β-well separated* if $\boldsymbol{b}_2$ is well separated from all particles of $\boldsymbol{b}_1$ (as in BH). Notice, however, with the new condition, that we can no longer FLIP the multipole expansion of $\boldsymbol{b}_1$ to a local expansion for $\boldsymbol{b}_2$. Because $\boldsymbol{b}_1$ has only a constant number of particles, we can directly evaluate the potential induced by these particles for $\boldsymbol{b}_2$. This new condition makes the FLIP operation of this special class of interaction-pairs *unidirectional*: we only FLIP $\boldsymbol{b}_2$ to $\boldsymbol{b}_1$.

We can describe the refined Procedure Interaction-Pair using the modified well-separation condition when one box is a leaf-box.

Procedure Refined Interaction-Pair ($\boldsymbol{b}_1$, $\boldsymbol{b}_2$)

- If $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are β-well-separated and $1/2 \leq size(\boldsymbol{b}_1)/size(\boldsymbol{b}_2) \leq 2$, then $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ is a bidirectional interaction-pair.
- Else if the larger box, without loss of generality, $\boldsymbol{b}_1$, is a leaf-box, then the well-separation condition becomes: $\boldsymbol{b}_2$ is well separated from all particles of $\boldsymbol{b}_1$. If this condition is true, then $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ is a unidirectional interaction-pair from $\boldsymbol{b}_2$ to $\boldsymbol{b}_1$.
- Else if both $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are leaf-boxes, then particles in $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are near-field particles.
- Else if neither $\boldsymbol{b}_1$ nor $\boldsymbol{b}_2$ is a leaf-box, without loss of generality, assuming that $\boldsymbol{b}_2$ is at least as large as $\boldsymbol{b}_1$ and letting $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{2^d}$ be the child-boxes of $\boldsymbol{b}_2$, then recursively decide interaction-pairs by calling interaction-pair($\boldsymbol{b}_1$,$\boldsymbol{c}_i$) for all $1 \leq i \leq 2^d$.
- Else, if one of $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ is a leaf-box, without loss of generality, assuming that $\boldsymbol{b}_1$ is a leaf box and letting $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{2^d}$ be the child-boxes of $\boldsymbol{b}_2$, then recursively decide interaction pairs by calling interaction-pair($\boldsymbol{b}_1$,$\boldsymbol{c}_i$) for all $1 \leq i \leq 2^d$.

Let $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{2^d}$ be the set of child-boxes of the root-box of the hierarchical tree. Then the set of all interaction-pairs can be generated as the union of refined interaction-pair$(\boldsymbol{c}_i, \boldsymbol{c}_j)$ for all $1 \leq i < j \leq 2^d$.

The refined FMM for far-field calculation can then be defined as follows: for each bidirectional interaction pair $(\boldsymbol{b}_1, \boldsymbol{b}_2)$, letting $\Phi_i^p()$ $(i = 1, 2)$ be the multipole expansion of $\boldsymbol{b}_i$, FLIP $\Phi_1^p()$ to $\boldsymbol{b}_2$ and add to $\boldsymbol{b}_2$'s potential local expansion. Similarly, FLIP $\Phi_2^p()$ to $\boldsymbol{b}_1$ and add to $\boldsymbol{b}_1$'s potential local expansion. Then traverse down the hierarchical tree in a pre-ordering, shift, and add the potential local expansion of the parent-box of a box to its own local expansion. For each unidirectional interaction pair $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ from $\boldsymbol{b}_2$ to $\boldsymbol{b}_1$, letting $\Phi_2^p()$ be the multipole expansion of $\boldsymbol{b}_2$, evaluate $\Phi_2^p()$ directly for each particle in $\boldsymbol{b}_1$ and add its potential.

In section 4.3, we will show that the degree of each box in this set of interaction-pairs is "small." Hence the new set of interaction-pairs is more suitable for parallel implementation.

Remark 3.1. Several heuristics have been proposed in the literature to reduce the number of operations in N-body simulations. The commonly used ones include collapsing boxes when a box has only one nonempty child-box, and avoiding translation of particle expansion coefficients to multipole expansions and avoiding FLIPs whenever such operations are inefficient. See Nabors et al. [22] for an example of exploiting these heuristics and the heuristics used in [7]. Moreover, almost all N-body codes have their own set of heuristics based on various intuitions. These heuristics may be used in our formulation to further reduce the number of operations and the amount of communication; conversely, our analysis or some variants of it could be used in the evaluation of formulations that have been proposed and implemented.

3.2. Hierarchical neighboring graphs. Hierarchical methods explicitly use two graphs: the *hierarchical tree* which connects each box to its parent box and each particle to its leaf-box, and the *near-field graph* which connects each box with its near-field boxes. The hierarchical tree is generated and used in the first step to compute the multipole expansion induced by each box. We can use a bottom-up procedure to compute these multipole expansions: first compute the multipole expansions at leaf-boxes and then *SHIFT* the expansion to the parent boxes and then up the hierarchical tree until multipole expansions for all boxes in the hierarchical tree are computed.

The near-field graph can also be generated by Procedure Refined Interaction-Pair. In section 3.4, we will formally define the near-field graph.

We now define the *hierarchical-neighboring graph*, which is closely related to the near-field graph. This class of graphs is much simpler than both far-field and near-field graphs and will be used as examples in the illustration of our geometric structure lemmas in the next section.

The hierarchical-neighboring graph represents the neighboring relation among leaf-boxes. Give a hierarchical tree T, let V be the set of all leaf-boxes of T. The tree T defines a graph, the *hierarchical-neighboring graph*, on V, where two boxes are joint by an edge if and only if the two boxes share a common point (i.e., they are neighboring boxes). This graph is the nine-point stencil grid for a completely balanced tree, i.e., each leaf-box is connected with its eight neighbors.

3.3. Far-field graphs. We now define the communication graphs used in BH and in the refined FMM.

Fast multipole graphs (FM). The fast multipole graph, FM^β, models the communication pattern of the refined FMM. It is a graph defined on the set of boxes and particles in the hierarchical tree. Two boxes $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are connected in FM^β if and only if (1) $\boldsymbol{b}_1$ is the parent-box of $\boldsymbol{b}_2$ or, vice versa, in the hierarchical tree; or (2) $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ is an interaction-pair generated by Procedure Refined Interaction-Pair defined in section 3.1. The edge is bidirectional for a bidirectional interaction-pair and unidirectional for a unidirectional interaction-pair. Furthermore, each particle is connected with the box that contains the particle.

The following lemma will be useful in the next section.

LEMMA 3.2. *The refined FMM flips the multipole expansion of* $\boldsymbol{b}_2$ *to* $\boldsymbol{b}_1$ *if and only if*

1. $\boldsymbol{b}_2$ *is well separated from* $\boldsymbol{b}_1$*, and*
2. *the parent of* $\boldsymbol{b}_2$ *is not well separated from* $\boldsymbol{b}_1$*, and* $\boldsymbol{b}_2$ *is not well separated from the parent of* $\boldsymbol{b}_1$*.*

Proof. The lemma follows from the analysis of [13] and Procedure Refined Interaction-Pair: if $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are not well separated, then clearly they do not form an interaction-pair. On the other hand, if the parent of $\boldsymbol{b}_1$ is well separated from $\boldsymbol{b}_2$, then FMM flips the multipole expansion of $\boldsymbol{b}_2$ to the parent of $\boldsymbol{b}_1$, and hence there is no need to flip $\boldsymbol{b}_2$ to $\boldsymbol{b}_1$. Similarly, if $\boldsymbol{b}_1$ is well separated from the parent of $\boldsymbol{b}_2$, then FMM flips the parent of $\boldsymbol{b}_2$ to $\boldsymbol{b}_1$. □

In section 4.3 we will show that both in- and out-degrees of FM^β are small.

BH graphs. BH defines two classes of communication graphs: BH_S^β and BH_P^β. BH_S^β models the sequential communication pattern and BH_P^β is more suitable for parallel implementation. The letters S and P, in BH_S^β and BH_P^β, respectively, stand for "sequential" and "parallel."

We first define BH_S^β and show why parallel computing requires a different communication graph BH_P^β to reduce total communication cost.

The graph BH_S^β of a set of particles P contains two sets of vertices: P, the particles, and B, the set of boxes in the hierarchical tree. The edge set of the graph BH_S^β is defined by the communication pattern of the sequential BH. A particle $\boldsymbol{p}$ is connected with a box $\boldsymbol{b}$ if in BH, we need to evaluate $\boldsymbol{p}$ against $\boldsymbol{b}$ to compute the force or potential exerted on $\boldsymbol{p}$. So the edge is directed from $\boldsymbol{b}$ to $\boldsymbol{p}$. Notice that if $\boldsymbol{p}$ is connected with $\boldsymbol{b}$, then $\boldsymbol{b}$ must be well separated from $\boldsymbol{p}$. Moreover, the parent of $\boldsymbol{b}$ is not well separated from $\boldsymbol{p}$. Therefore, if $\boldsymbol{p}$ is connected with $\boldsymbol{b}$ in BH_S^β, then $\boldsymbol{p}$ is not connected to any box in the subtree of $\boldsymbol{b}$ nor to any ancestor of $\boldsymbol{b}$.

In addition, each box is connected directly with its parent-box in the hierarchical tree and each point $\boldsymbol{p}$ is connected to its leaf-box. Both types of edges are bidirectional.

LEMMA 3.3. *Each particle is connected to at most* $O(\log n + \mu)$ *number of boxes. So the in-degree of* BH_S^β *is bounded by* $O(\log n + \mu)$*.*

Proof. It directly follows from the analysis of BH for uniformly distributed particles [3]. □

Notice, however, that BH_S^β is not suitable for parallel implementation. It has a large out-degree. This major drawback can be illustrated by the example of n uniformly distributed particles in two dimensions. Assume we have four processors. Then the "best" way to partition the problem is to divide the root-box into four boxes and map each box onto a processor. Notice that in the direct parallel implementation of BH, as modeled by BH_S^β, each particle needs to access the information of at least $\Omega(1)$ boxes in each of the other processors. Because each processor has $n/4$ particles, the total communication overhead is $\Omega(n)$, which is very expensive.

The main problem with BH_S^β is that many particles from a processor need to access the information of the same box in some other processors (which contributes to the large out-degree). We show that a combination technique can be used to reduce the out-degree. The idea is to *combine* the "same" information from a box and send the information as one unit to another box on a processor that needs the information. We will show, as a consequence of our main result, that this combination technique reduces the total communication cost to $O(\sqrt{n \log n})$ for the four-processor example, and to $O(\sqrt{pn \log n})$ for p processors. Similarly, in three dimensions, the combination technique reduces the volume of messages from $\Omega(n \log n)$ to $O(p^{1/3} n^{2/3} (\log n)^{1/3})$.

We can define a graph BH_P^β to model the communication and computation pattern that uses this combination technique. Our definition of BH_P^β is inspired by the communication pattern of the refined FMM. We will show that the communication pattern of the refined FMM can be used to guide the message combination for the parallel implementation of the BH method!

The combination technique is based on the following observation: suppose $\boldsymbol{p}$ is well separated from $\boldsymbol{b}_1$ but not from the parent of $\boldsymbol{b}_1$. Let $\boldsymbol{b}$ be the largest box that contains $\boldsymbol{p}$ such that $\boldsymbol{b}$ is well separated from $\boldsymbol{b}_1$, using the well-separation definition in section 3.1. If $\boldsymbol{b}$ is not a leaf-box, then $(\boldsymbol{b}, \boldsymbol{b}_1)$ is a bidirectional interaction-pair in the refined FMM. If $\boldsymbol{b}$ is a leaf-box, then $(\boldsymbol{b}, \boldsymbol{b}_1)$ is a unidirectional interaction-pair from $\boldsymbol{b}_1$ to $\boldsymbol{b}$. Hence $(\boldsymbol{b}, \boldsymbol{b}_1)$ is an edge of FM^β. Then, any other particle $\boldsymbol{q}$ contained in $\boldsymbol{b}$ is well separated from $\boldsymbol{b}_1$ as well. Hence we can combine the information from $\boldsymbol{b}_1$ to $\boldsymbol{p}$ and $\boldsymbol{q}$ and all other particles in $\boldsymbol{b}$ as follows: $\boldsymbol{b}_1$ sends its information (just one copy) to $\boldsymbol{b}$ and $\boldsymbol{b}$ forwards the information down the hierarchical tree, to both $\boldsymbol{p}$ and $\boldsymbol{q}$ and all other particles in $\boldsymbol{b}$. This combination-based communication scheme defines a new communication graph BH_P^β for parallel BH: the nodes of the graph are the union of particles and boxes, i.e., $P \cup B(P)$. Each particle is connected to the leaf-box it belongs to. Two boxes are connected if and only if they are connected in the FM graph. However, to model the communication cost, we must introduce a weight on each edge along the hierarchical tree embedded in BH_P^β, to be equal to the number of data units needed to be sent along that edge. We will show the following.

LEMMA 3.4. *The weight on each edge in* BH_P^β *is at most* $O(\log n + \mu)$.

Proof. See Lemma 4.9 of the next section. □

It is worthwhile to point out the difference between the comparison and communication patterns in BH. In the sequential version of BH, if $\boldsymbol{p}$ is connected with $\boldsymbol{b}$, then we have to compare $\boldsymbol{p}$ against all ancestors of $\boldsymbol{b}$ in the computation. The procedure is to first compare $\boldsymbol{p}$ with the root of the hierarchical tree, and then recursively move the comparison down the tree: if the current box compared is not well separated from $\boldsymbol{p}$, then we will compare $\boldsymbol{p}$ against all its child-boxes. However, in terms of force and potential calculation, we only evaluate a particle against the first box down a path that is well separated from the particle. The graphs BH_S^β and BH_P^β capture the communication pattern rather than the comparison pattern. The communication is more essential to force or potential calculation. In section 5, we will give an efficient parallel algorithm for finding the communication graph directly without going through the comparison phase. The construction of the communication graph has been one of the bottlenecks in load balancing BH and FMM on a parallel machine.

3.4. Near-field graphs.

The near-field graph, denoted NF^β, is defined over all leaf-boxes. A leaf-box $\boldsymbol{b}_1$ is a *near-field neighbor* of a leaf-box $\boldsymbol{b}$ if $\boldsymbol{b}_1$ is not well separated from some particles of $\boldsymbol{b}$. Thus, FMM and BH directly compute the potential at particles in $\boldsymbol{b}$ induced by particles of $\boldsymbol{b}_1$.

There are two basic cases: (1) if $size(\boldsymbol{b}_1) \leq size(\boldsymbol{b})$, then we call $\boldsymbol{b}_1$ a *geometric near-field neighbor* of $\boldsymbol{b}$; (2) if $size(\boldsymbol{b}_1) > size(\boldsymbol{b})$, then we call $\boldsymbol{b}_1$ a *hierarchical near-field neighbor* of $\boldsymbol{b}$. In the example of section 3.1, A, B, and C are hierarchical near-field neighbors of all leaf-boxes in D, while A, B, and C have some geometric near-field neighbors in D.

We introduce some notation. The *geometric in-degree* of a box $\boldsymbol{b}$ is the number of its geometric near-field neighbors. The *geometric out-degree* of a box $\boldsymbol{b}$ is the number of boxes to which b is the geometric near-field neighbor. The *hierarchical in-degree* of a box $\boldsymbol{b}$ is the number of its hierarchical near-field neighbors. We will define the *hierarchical out-degree* of a box shortly.

In section 4.3, we will show that the geometric in-degree, geometric out-degree, and hierarchical in-degree are small. However, in the example of section 3.1, A, B, and C are hierarchical near-field neighbors for all leaf-boxes in D. Hence the number of leaf-boxes to which a box is a hierarchical near-field neighbor could be very large. So the near-field graph defined above can have a very large out-degree.

We can use the combination technique to reduce the degree when a box $\boldsymbol{b}$ is a hierarchical near-field neighbor of a box $\boldsymbol{b}_1$. Let $\boldsymbol{b}_2$ be the ancestor of $\boldsymbol{b}_1$ of the same size as $\boldsymbol{b}$. Instead of $\boldsymbol{b}$ sending its information directly to $\boldsymbol{b}_1$, $\boldsymbol{b}$ sends it to $\boldsymbol{b}_2$ and $\boldsymbol{b}_2$ then forwards the information down the hierarchical tree. Notice that $\boldsymbol{b}$ and $\boldsymbol{b}_2$ are not well separated. We will refer to this modified near-field graph as the near-field graph, denoted by NF^{β}. We also define the *hierarchical out-degree* of a box $\boldsymbol{b}$ to be the number of edges from $\boldsymbol{b}$ to the set of non-leaf-boxes constructed above. In section 4.3, we will show that the *hierarchical out-degree* is also small.

To model the near-field communication, similar to our approach for BH, we introduce a weight on the edges of the hierarchical tree. We will also show the following.

LEMMA 3.5. *The weight on each edge in NF^{β} is at most $O(\log n + \mu)$.*

Proof. See Lemma 4.9 in the next section. □

3.5. N-body communication graphs.

By abusing notation, let $FM^{\beta} = FM^{\beta} \cup NF^{\beta}$ and $BH_P^{\beta} = BH_P^{\beta} \cup NF^{\beta}$. So the communication graph we defined simultaneously supports near-field and far-field communication, as well as communication up and down the hierarchical tree. Hence by partitioning and load balancing FM^{β} and BH_P^{β}, we automatically partition and balance the hierarchical tree, the near-field graph, and the far-field graph.

4. The geometry of the N-body graph and its partitioning.

In this section, we prove our main partitioning theorem.

THEOREM 4.1 (main). *Let G be an N-body communication graph (either for BH or FMM) of a set of particles located at $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ in $\mathbb{R}^d$ ($d = 2$ or 3). If P is μ-nonuniform, then G can be partitioned into two equal sized subgraphs by removing at most $O(n^{1-1/d}(\log n + \mu)^{1/d})$ nodes. Moreover, such a partitioning can be computed in linear time sequentially and in parallel $O(n/p)$ time with p processors.*

We will use the following notation in the remainder of the paper: if $\alpha > 0$ and B is a box of size r, we define $\alpha \cdot B$ to be the box with the same center as B but size αr.

4.1. Box-graphs.

In order to prove Theorem 4.1, we introduce a class of geometric graphs that contain BH_P^{β} and FM^{β} as special cases. This class of graphs, called the *box-graph*, is defined on boxes in two or three dimensions.

A *box-system* in $\mathbb{R}^d$ is a set $B = \{B_1, \ldots, B_n\}$ of boxes. Let $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ be the centers of the boxes, respectively. For each integer k, the set B is a *k-ply box-system* if no point $\boldsymbol{p} \in \mathbb{R}^d$ is contained in more than k of $\mathrm{int}(B_1), \ldots, \mathrm{int}(B_n)$.

For example, the set of all leaf-boxes of a hierarchical tree forms a 1-ply box-system. The box-system is a variant of the neighborhood system of Miller et al. [19], where a neighborhood system is a collection of Euclidean balls in $\mathbb{R}^d$. Miller et al. used the neighborhood system to model unstructured finite-element meshes. In this section, we will show that box-systems can be used to model the communication graphs for parallel adaptive N-body simulation.

Given a box-system, it is possible to define the *overlap* graph associated with the system.

DEFINITION 4.2. *Let $\alpha \geq 1$ be given, and let $\{B_1, \ldots, B_n\}$ be a k-ply box-system. The α-overlap graph for this box-system is the undirected graph with vertices $V = \{1, \ldots, n\}$ and edges*

$$E = \{(B_i, B_j) : B_i \cap (\alpha \cdot B_j) \neq \emptyset \text{ and } (\alpha \cdot B_i) \cap B_j \neq \emptyset\}.$$

The edge condition is equivalent to: $(B_i, B_j) \in E$ if and only if the α dilation of the smaller box touches the larger box.

As shown in the author's Ph.D. thesis [30], the partitioning algorithm and theorem of Miller et al. can be extended to overlap graphs on box-systems. We include a high level description of the algorithm and theorems of Miller et al. in Appendix A. More detailed algorithms can be found in [19]. Practical implementation (and a pointer to our Matlab tool-box) and experimental results can be found in [11].

THEOREM 4.3. *Let G be an α-overlap graph over a k-ply box-system in $\mathbb{R}^d$; then G can be partitioned into two equal sized subgraphs by removing at most $O(\alpha k^{1/d} n^{1-1/d})$ vertices. Moreover, such a partitioning can be computed in linear time sequentially and in parallel $O(n/p)$ time with p processors.*

Proof. See Theorem 7.1 of [30]. □

4.2. Partitioning N-body simulation graphs. In this section, we show that if a set of n particles is μ-nonuniform, then its N-body communication graph is a subgraph of an α-overlap of a $(\log_{2^d} n + \mu)$-ply box-system. We then apply Theorem 4.3 to prove the main partitioning Theorem 4.1.

LEMMA 4.4. *Let $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ be a point set in $\mathbb{R}^d$ that is μ-nonuniform. Then the set of boxes $B(P)$ of the hierarchical tree of P is a $(\log_{2^d} n + \mu)$-ply box-system and $FM^\beta(P)$ is a subgraph of the 3β-overlap graph of $B(P)$.*

Proof. Because the height of the hierarchical tree is bounded by $\log_{2^d} n + \mu$, the ply of $B(P)$ is bounded by $\log_{2^d} n + \mu$.

First, notice that each particle can be viewed as a box of size 0. To prove that $FM^\beta(P)$ is a subgraph of the α-overlap graph of $B(P)$, we need to show that *for each edge $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ of $FM^\beta(P)$, the 3β dilation of the smaller box touches the larger box.* Notice that $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ is an edge either in the hierarchical tree, in the near-field graph NF^β, or in the far-field graph.

If $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ is an edge of the hierarchical tree or $\boldsymbol{b}_1$ is a particle in $\boldsymbol{b}_2$, then $\boldsymbol{b}_1 \cap \boldsymbol{b}_2 \neq \emptyset$, and hence $(3\beta) \cdot \boldsymbol{b}_1 \cap \boldsymbol{b}_2 \neq \emptyset$. We now prove this statement for the case when $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ is an edge in the near-field graph. There are two cases: (1) if $\boldsymbol{b}_2$ is a geometric near-field neighbor of $\boldsymbol{b}_1$, then we have $size(\boldsymbol{b}_2) \leq size(\boldsymbol{b}_1)$ and $\boldsymbol{b}_2$ is not well separated from $\boldsymbol{b}_1$. Hence there is a particle in $\boldsymbol{b}_1$ that is not β-well-separated from $\boldsymbol{b}_2$, implying that $\beta \cdot \boldsymbol{b}_2$ contains the particle and hence touches $\boldsymbol{b}_1$; (2) otherwise, $\boldsymbol{b}_2$ is a hierarchical near-field neighbor of some boxes in the subtree of $\boldsymbol{b}_1$, $\boldsymbol{b}_1$ has the same size as $\boldsymbol{b}_2$, and further, $\boldsymbol{b}_1$ is not β-well-separated from $\boldsymbol{b}_2$. Therefore, the distance between $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ is at most $\beta \cdot size(\boldsymbol{b}_1)$, implying $\beta \cdot \boldsymbol{b}_1$ touches $\boldsymbol{b}_2$.

We now prove the statement when $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ is an edge of the far-field graph. Assume that $\boldsymbol{b}_1$ is the larger box of the two. There are two cases. (1) If $\boldsymbol{b}_1$ is not a leaf-box, then we have $size(\boldsymbol{b}_2) \geq size(\boldsymbol{b}_1)/2$. Hence $size(\boldsymbol{b}_1) \leq size(parent(\boldsymbol{b}_2)) \leq 2size(\boldsymbol{b}_1)$. Because $parent(\boldsymbol{b}_2)$ and $\boldsymbol{b}_1$ are not β-well-separated, the distance between $parent(\boldsymbol{b}_2)$ and $\boldsymbol{b}_1$ is less than $\beta \cdot size(parent(\boldsymbol{b}_2))$. Hence $\beta parent(\boldsymbol{b}_2) \cap \boldsymbol{b}_1 \neq \emptyset$. Because $parent(\boldsymbol{b}_2)$ is properly contained in $3 \cdot \boldsymbol{b}_2$, we have $(3\beta) \cdot \boldsymbol{b}_2$ intersects $\boldsymbol{b}_1$. (2) If $\boldsymbol{b}_1$ is a leaf-box, recall that the well-separation condition is that all particles of $\boldsymbol{b}_1$ are well-separated from $\boldsymbol{b}_2$ but some are not well separated from $parent(\boldsymbol{b}_2)$. Thus, there exists a particle $\boldsymbol{p}$ in $\boldsymbol{b}_1$ such that the distance between $\boldsymbol{p}$ and $parent(\boldsymbol{b}_2)$ is less than $\beta \cdot size(parent(\boldsymbol{b}_2))$, implying that $\beta \cdot parent(\boldsymbol{b}_2)$ contains $\boldsymbol{p}$ and hence intersects $\boldsymbol{b}_1$. Because $parent(\boldsymbol{b}_2)$ is properly contained in $3 \cdot \boldsymbol{b}_2$, we have $(3\beta) \cdot \boldsymbol{b}_2$ intersects $\boldsymbol{b}_1$. □

LEMMA 4.5. *Let $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ be a point set in $\mathbb{R}^d$ that is μ-nonuniform. Then $BH_P^\beta(P)$ is a subgraph of the 3β-overlap graph of $B(P)$.*

Proof. In our definition of BH_P^β, there is no direct edge from a particle $\boldsymbol{p}$ to a box $\boldsymbol{b}$ that does not contains $\boldsymbol{p}$. If $\boldsymbol{p}$ needs information from $\boldsymbol{b}$, in the combination technique, $\boldsymbol{b}$ sends its information to a box $\boldsymbol{b}_1$ (not necessarily a leaf-box) that contains $\boldsymbol{p}$. The choice of $\boldsymbol{b}_1$ is that $(\boldsymbol{b}, \boldsymbol{b}_1)$ is an edge of FM^β.

The main difference between BH_P^β and FM^β is the edge-weight which we will discuss in section 4.3. Their node partitions are the same. Hence the argument of Lemma 4.5 also applies to this lemma. □

Our main partitioning theorem (Theorem 4.1) then follows directly from Lemmas 4.4 and 4.5 and Theorem 4.3.

4.3. Edge-partitioning of N-body graphs. In an edge-partition of a graph, we remove edges of the graph to divide the graph into two disjoint graphs of roughly equal size. Formally, we use a notion of bisection. A *bisection* of a graph G is a division of its vertices into two disjoint subsets A and B such that $-1 \leq |A| - |B| \leq 1$. The *cost* of a bisection is the total weight of the edges that bridge A and B. Similarly, a *p-way partition* of G is a division of its vertex set into p disjoint subsets of size n/p, where n is the number of vertices in G. The *cost* of a p-way partition is the total weight of the edges whose endpoints are in different subsets.

Bounds on node-partitions also imply bounds on edge-partitions. If a graph has maximum degree D and maximum edge-weight W, then a node-partition of size $f(n)$ can be converted into an edge-partition of cost $DWf(n)$.

In this subsection, we prove the following theorem.

THEOREM 4.6. *Let G be an N-body communication graph (either for BH or FMM) of a set of particles located at $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ in $\mathbb{R}^d$ ($d = 2$ or 3). If P is μ-nonuniform, then G has a bisection of cost $O(n^{1-1/d}(\log n + \mu)^{2+1/d})$.*

The basic idea to prove Theorem 4.6 is to show that the N-body communication graph has degree and edge-weights bounded by $O(\log n + \mu)$. We first give a geometric lemma whose proof can be found in [30].

LEMMA 4.7. *Let $B = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ be a set of (interiorly) disjoint boxes in $\mathbb{R}^d$. Let $\boldsymbol{b}$ be a box in $\mathbb{R}^d$ and $\alpha > 1$ and $\beta > 0$ be two constants. Then the number of boxes of B of size at least $\beta \cdot size(\boldsymbol{b})$ that intersect $\alpha \cdot \boldsymbol{b}$ is bounded by a constant C that depends only on α, β, and d.*

LEMMA 4.8. *If $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ in $\mathbb{R}^d$ is μ-nonuniform, then the degree of $FM^\beta(P)$ is bounded by $O(\log n + \mu)$.*

Proof. Let $\boldsymbol{b}$ be a vertex of FM^β. If $\boldsymbol{b}$ is a particle, then it is connected only to its leaf-box. Hence its degree is 1. If $\boldsymbol{b}$ is a box, then its neighbors are defined in the hierarchical tree (at most one), in the near-field graph, or in the far-field graph. We need to prove that both in- and out-degrees are small. We first prove that a leaf-box can not have too many neighbors in the near-field graph. There are four cases.

1. [*Bound on the hierarchical in-degree*]: Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ be the hierarchical near-field neighbors of $\boldsymbol{b}$. Because $\boldsymbol{b}$ is not well separated from $\boldsymbol{b}_i$, none of $\boldsymbol{b}$'s ancestors is well separated from $\boldsymbol{b}_i$'s. Notice that $\boldsymbol{b}$ has at most $\log_{2^d} n + \mu$ ancestors. By Lemma 4.7, each box can only have a constant number of other boxes of the same size that are not well separated from the box. Because $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ are at least as large as $\boldsymbol{b}$, for each $\boldsymbol{b}_i$, there is an ancestor of $\boldsymbol{b}$ of the same size as $\boldsymbol{b}_i$. Hence the hierarchical in-degree of $\boldsymbol{b}$ is $O(\log_{2^d} n + \mu)$.

2. [*Bound on the hierarchical out-degree*]: Let $\boldsymbol{b}$ be a leaf-box and let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ be the set of boxes such that $\boldsymbol{b}$ is their hierarchical near-field neighbor. The number k can be quite large as we have shown in section 3.4. However, in NF^β, $\boldsymbol{b}$ is not directly connected to $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$. Instead, $\boldsymbol{b}$ is connected to their ancestors of the same size as $\boldsymbol{b}$. These ancestors are interiorly disjoint among themselves and are not well separated from $\boldsymbol{b}$. Hence, they all intersect $\beta \cdot \boldsymbol{b}$. By Lemma 4.7, there are only a constant number of them. Notice also, by the same reason, that each ancestor of $\boldsymbol{b}_i$'s can be connected to only a constant number of leaf-boxes in NF^β.
3. [*Bound on the geometric in-degree*]: Let $\boldsymbol{b}$ be a leaf-box and let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ be the set of geometric near-field neighbors of $\boldsymbol{b}$. We have $size(\boldsymbol{b}_i) \leq size(\boldsymbol{b})$ and $\boldsymbol{b}_i$ is not well separated from $\boldsymbol{b}$. Hence there is a particle in $\boldsymbol{b}$ which is contained in $\beta \cdot \boldsymbol{b}_i$. We can divide $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ into at most $O(\log_{2^d} n + \mu)$ sets according to their size. Notice that for each such set, there are at most a constant number of boxes whose β dilation contains a particular point. Thus, there are at most $O(\log_{2^d} n + \mu)$ leaf-boxes whose β dilation can contain a particular point of $\boldsymbol{b}$. Because there are a constant number of particles in $\boldsymbol{b}$, we have $k \leq O(\log_{2^d} n + \mu)$.
4. [*Bound on the geometric out-degree*]: Let $\boldsymbol{b}$ be a leaf-box and let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ be the set of boxes of whom $\boldsymbol{b}$ is the geometric near-field neighbor. We have $size(\boldsymbol{b}_i) \geq size(\boldsymbol{b})$ and $\beta \cdot \boldsymbol{b}$ intersects $\boldsymbol{b}_i$. By Lemma 4.7, k is bounded by a constant.

We now prove that a box cannot have too many neighbors in the far-field graph.

First we show that a box $\boldsymbol{b}$ is connected with a constant number of boxes in bidirectional interaction-pairs. Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ be the set of boxes such that $(\boldsymbol{b}, \boldsymbol{b}_i)$ is a bidirectional interaction-pair. We have $size(\boldsymbol{b})/2 \leq size(\boldsymbol{b}_i) \leq 2size(\boldsymbol{b})$. Moreover, $parent(\boldsymbol{b}_i)$ is not well separated from $\boldsymbol{b}$. Each box has at most 2^d child-boxes. The set of parent-boxes is interiorly disjointed. By Lemma 4.7, k is bounded by a constant.

Let $\boldsymbol{b}$ be a leaf-box. We now bound the number of boxes $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ such that $(\boldsymbol{b}, \boldsymbol{b}_i)$ is a unidirectional interaction-pair from $\boldsymbol{b}_i$ to $\boldsymbol{b}$. It follows from the analysis of BH [3] that each particle is involved with at most $O(\log n + \mu)$ well-separation boxes. In our new definition of well-separation condition for the case when $\boldsymbol{b}$ is a leaf-box, box $\boldsymbol{b}_i$ is well separated from $\boldsymbol{b}$ only if $\boldsymbol{b}_i$ is well separated from all particles in $\boldsymbol{b}$. Because $\boldsymbol{b}$ has a constant number of particles, there are at most $O(\log n + \mu)$ well-separated boxes that are involved in unidirectional interaction-pairs with a leaf-box $\boldsymbol{b}$.

Finally, for each box $\boldsymbol{b}$ we give a bound on the number of leaf-boxes $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ such that $(\boldsymbol{b}_i, \boldsymbol{b})$ is a unidirectional interaction pair from $\boldsymbol{b}$ to $\boldsymbol{b}_i$. From Procedure Refined Interaction-Pair, we have $size(\boldsymbol{b}) \leq size(\boldsymbol{b}_i)$ and the parent of $\boldsymbol{b}$ is not well separated from $\boldsymbol{b}_i$. Hence $\beta \cdot parent(\boldsymbol{b})$ intersects $\boldsymbol{b}_i$. By Lemma 4.7, k is bounded by a constant.

Therefore, the degree of FM^β and hence BH_P^β is bounded by $O(\log n + \mu)$. ◻

The proof above, in fact, shows that all non-leaf-boxes of FM^β and BH_P^β have in-degree bounded by a constant.

LEMMA 4.9. *If* $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ *in* $\mathbb{R}^d$ *is* μ*-nonuniform, then the edge-weight of* $FM^\beta(P)$ *and* $BH_P^\beta(P)$ *is bounded by* $O(\log n + \mu)$.

Proof. It follows from the proof of Lemma 4.8 that all non-leaf-boxes of FM^β and BH_P^β have in-degree bounded by a constant. The in-degree gives a bound on the number of message units that are sent to a box during the hierarchical method. A non-leaf-box needs to forward some of these messages down the hierarchical tree.

The lemma follows from the following observation: suppose T is a tree of height D. Each internal node of T needs to send at most C (which is a constant) number of data units down the tree. Then the number of data units that pass any tree edge is bounded by CD. $\square$

Theorem 4.6 then follows directly from Lemmas 4.8 and 4.9 and Theorem 4.1.

5. Load balancing parallel adaptive N-body simulation.

Partitioning the communication graph FM^{β} and BH_P^{β} into a given number of subgraphs of roughly equal size does not necessarily balance the load of computation in parallel N-body simulation because the amount of computation at a particle or a box is different from that at another. Using FM^{β} and BH_P^{β} for modeling a hierarchical method, we need to map computations onto these communication graphs. To model the load at each particle and box, we introduce a weight to each vertex of graphs FM^{β} and BH_P^{β}. The weight is equal to the number of operations that are performed at the vertex. We will show how to compute these weights in the next section.

The load balancing problem for parallel N-body simulation can now be described as follows: given a parallel machine with p processors, partition the weighted FM^{β} or BH_P^{β} into p subgraphs with roughly equally total weights while minimizing the communication overhead.

Load balancing for N-body simulation is more difficult than that for unstructured finite-element computation, partially because the computation load of each vertex in the N-body communication graph is not uniform and the N-body communication graph is not given (explicitly) in advance. We have to either compute such a graph explicitly or devise a partitioning algorithm that does not require the combinatorial structure of the communication graph.

Our partitioning theorem can be extended to weighted N-body graphs. In the next section, we will use the nonweighted partitioning algorithm to efficiently compute the N-body communication graph in parallel. Putting these together, we develop an efficient load balancing scheme for parallel adaptive N-body simulation.

Teng, Pramono, and Ruppert [32, 31, 11] showed that the partitioning theorem of Miller et al. [20] (Theorem 4.3) can be naturally extended to the weighted case. Applying their weighted version result to our main partitioning Theorem 4.1 we have the following weighted partitioning theorem.

THEOREM 5.1. *Let G be a weighted N-body communication graph (for either BH or FMM) of a set of particles at $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ in $\mathbb{R}^d$ ($d = 2$ or 3). If P is μ-nonuniform, then G can be partitioned into two equally weighted subgraphs by removing at most $O(n^{1-1/d}(\log n + \mu)^{1/d})$ nodes, or by removing edges of at most $O(n^{1-1/d}(\log n + \mu)^{2+1/d})$ total edge weights.*

Recursively applying our partitioning theorem, we can analyze the quality of the recursive bisection scheme for p-way partitioning. (See Simon and Teng [27] for unstructured meshes.)

COROLLARY 5.2. *If G is a (weighted) N-body communication graph for particles that are μ-nonuniform, then G can be partitioned into p equally weighted subgraphs such that the total weight of the removed edges is bounded by $O(p^{1/d}n^{1-1/d}(\log n + \mu)^{2+1/d})$.*

Proof. By Lemma 4.1 of [27], the total weight of removed edges of the recursive bisection scheme is

$$\sum_{i}^{\log p - 1} 2^i f(n/2^i),$$

where $f(x) = x^{1-1/d}(\log x + \mu)^{2+1/d}$. Summing up, we can bound the formula by $O(p^{1/d}n^{1-1/d}(\log n+\mu)^{2+1/d})$. See section 4 of [27] for the bound on general recursive bisection schemes. □

Therefore, computation and total communication volume is equal to, respectively, $O(n/p)$ and $O(p^{1/d}n^{1-1/d}(\log n + \mu)^{2+1/d})$ for FMM and, respectively, $O(n \log n/p)$ and $O(p^{1/d}n^{1-1/d}(\log n + \mu)^{2+1/d})$ for BH. These ratios of computation to communication imply that parallel N-body simulation can be made as scalable as computation on regular grids and as efficient as N-body simulations for uniformly distributed particles.

6. Constructing N-body computation graphs in parallel.

The first step of N-body simulation is to construct the hierarchical tree. Theoretically, Bern, Eppstein, and Teng [4] showed that the hierarchical tree can be computed from the set of particles in $O(n/p)$ time using p processors. Several efficient parallel implementations have been developed [15, 16, 23, 24, 34]. In practice, the hierarchical tree construction takes a very small fraction of the total simulation time.

Given the set of particles and its hierarchical tree, we need to efficiently construct the N-body communication graphs FM^{β} or BH_P^{β} and determine the computational load at each particle and box. The key is to compute such a communication graph without going through the comparison phase. In this section, we show that our partitioning algorithm and theorem enable us to use the separator-based geometric divide-and-conquer scheme of Frieze, Miller, and Teng [10] to construct FM^{β} and BH_P^{β} efficiently in parallel.

The basic idea is divide-and-conquer. We exploit the following important property of our partitioning scheme: because FM^{β} and BH_P^{β} are subgraphs of a 3β-overlap graph of a $(\log_{2^d} n+\mu)$-ply box system, we can efficiently find a provably good partition of FM^{β} and BH_P^{β} directly from particles and their hierarchical tree. See Appendix A. The partitioning algorithm does not directly use the combinatorial structure of FM^{β} and BH_P^{β}. Hence, we can apply our partitioning scheme first to divide FM^{β} or BH_P^{β} evenly into a given number of subgraphs of roughly equal size and recursively construct each subgraph in parallel.

6.1. The construction algorithm.

Let B be the set of boxes and particles in the hierarchical tree of P that is μ-nonuniform in $\mathbb{R}^d$.

1. Find a top level separating sphere S that divides B into three sets B_I, B_E, and B_O, where B_I, B_E, and B_O are the set of boxes in B that, respectively, are in the interior, are in the exterior, and are the vertex partition of FM^{β} induced by S. (See Appendix A for the construction of B_O.) Our partition theorem implies that $|B_I|$ and $|B_E|$ are a constant fraction $(1/2 < \gamma < 1)$ of $|B|$ and $|B_O| \leq O(n^{1-1/d}(\log n + \mu)^{1/d})$.
2. Recursively (in parallel) construct the N-body communication graphs G_I for $B_I \cup B_O$ and G_E for $B_E \cup B_O$.
3. Merge G_I and G_E (by deleting multiple edges) to obtain the communication graph for the N-body simulation.

This class of divide-and-conquer schemes has been carefully proven and analyzed in [10, 21, 30]. We refer interested readers to these references. We conclude this subsection with the following theorem, which follows directly from the results of [10, 21, 30].

THEOREM 6.1. *Let P be a μ-nonuniform point set in $\mathbb{R}^d$. Given the hierarchical tree of P, FM^{β} and BH_P^{β} can be constructed in random parallel $O(n \log p/p)$ time, using p processors.*

Proof. A detailed proof of this class of separator-based divide-and-conquer algorithms can be found in [10] for a parallel implementation and [21, 30] for a sequential implementation. Here we just give a high level argument. Let $T(N, Q)$ be the time used by the above algorithm to construct FM^β for N boxes using Q processors. To eliminate the trivial case when N is a very small constant, we assume that N is large enough so that $N^{1-1/d} \leq \epsilon N$ for $\epsilon \ll 1 - \gamma$. Then, recursively,

$$(1) \qquad T(N, Q) \leq \begin{cases} O(N) & \text{if } Q = 1, \\ T(\gamma N + N^{1-1/d}, \gamma Q) + \log Q & \text{otherwise.} \end{cases}$$

Because $N^{1-1/d} \leq \epsilon N \ll \gamma N$, as shown in [10, 21], $T(\gamma N + N^{1-1/d}, \gamma Q) + \log Q = O(T(\gamma N, \gamma Q) + \log Q) = O(N \log Q/Q)$. □

Given FM^β, we can determine the set of all interaction-pairs and the pattern for the combination technique. From interaction-pairs, we can determine the number of operations that a box or a particle needs to perform. Therefore, we can efficiently compute the weights for nodes in FM^β and BH_P^β.

7. Final remarks and open questions. In this paper, we develop an efficient partitioning and load balancing algorithm for parallel adaptive N-body simulation. We provide a rigorous theoretical analysis of its quality. Our theoretical result provides a "quality standard" for practical partitioning and load balancing schemes for adaptive N-body problems. We show how to apply the idea of BH to improve parallel adaptive FMM and how to use FMM communication patterns to support parallel BH implementation. These connections bring together the two "competing" methods and provide a new understanding of N-body problems. In addition, our results can be directly extended to Appel's adaptive N-body solver.

We demonstrate how methods and structures from computational geometry can be used to solve scientific computing problems. Our theoretical result has a potential impact on practical N-body computations. On one hand, it guides the development of provably good load balancing algorithms and software for dealing with adaptive and nonuniform N-body problems. On the other hand, practitioners who have been developing the state-of-the-art N-body codes can use our theoretical result as a quality-and-performance standard and a "theoretical benchmark" for experimentally evaluating their schemes and software. Even if a practical scheme or a code has not been analyzed theoretically, our result provides a strong theoretical confidence for the quality and applicability of the code for a large class of natural practical particle distributions. We also hope that our work will motivate new investigations that help close the gap between numerical experiments and a theoretical guarantee for untested data. Such an investigation, we hope and believe, will ultimately advance our understanding of this very important problem and produce more efficient and robust parallel N-body solvers.

We now conclude the paper with some discussions of future projects and open questions.

- Implement and experiment with our partitioning and load balancing algorithm and incorporate it into existing N-body solvers.[3]
- Apply this load balancing algorithm to dynamic N-body simulation.
- Apply our results and techniques to the other load balancing schemes (such as k-d trees and recursive coordinate bisection) and show that they or their

[3] We are currently implementing this geometry-based partitioning algorithm for N-body simulation. Experimental results of this and related partitioning algorithms will appear in a subsequent paper [18].

variants also give provably good partitioning. Some extensions will be given in [18].

- Investigate new applications of our parallel N-body-communication-graph construction algorithm.
- Theoretical investigation of the sequential BH_S^β. We conjecture it does not have a sublinear-sized partition. Such a result will illustrate the importance of our definition of BH_P^β for parallel implementation.
- Theoretical improvement of the partition theorem of this paper. We conjecture that FM^β can be partitioned into two subgraphs by removing only $O(n^{1-1/d})$ nodes $(d = 1, 2)$.

Appendix A. Geometric partitioning: algorithm and theory. We now briefly review Miller et al. geometric graph partitioning technique. For details and proofs, see their papers [20, 19]. To relate their approach to our paper, we illustrate their work in the context of box-systems.

The partitioning algorithm maps the d-dimensional box-system into a $(d+1)$-dimensional space.

The central theorem about overlap graphs is that they have good *separators*, that is, small sets of vertices whose removal divides them approximately in half. A regular cubic grid in d-space, with n vertices in an array $n^{1/d}$ on a side, can be divided in half by removing the $n^{(d-1)/d}$ vertices on a $(d-1)$-dimensional slice through the middle of the array. Up to a constant factor that depends on α, k, and d, an overlap graph in d dimensions has as good a separator as the cubic grid.

THEOREM A.1 (geometric separators [19]). *Let G be an n-vertex α-overlap graph of a k-ply box-system in d dimensions. Then the vertices of G can be partitioned into three sets A, B, and C, such that*

- *no edge joins A and B,*
- *A and B each have at most $(d+1)/(d+2)$ vertices,*
- *C has only $O(\alpha k^{1/d} n^{(d-1)/d})$ vertices.*

Miller et al. gave a randomized algorithm to find the separator in the theorem, which runs in linear time on a sequential machine or in parallel $O(n/p)$ time with p processors. The separator is defined by a sphere in $\mathbb{R}^d$. The algorithm chooses the separating sphere at random, from a distribution that is carefully constructed so that the separator will satisfy the conclusions of Theorem A.1 with high probability. The distribution is described in terms of a stereographic projection and conformal mapping on the surface of a sphere one dimension higher, in $\mathbb{R}^{d+1}$.

Here is an outline of the algorithm.

- *Project up.* Project the centers of the input boxes stereographically from $\mathbb{R}^d$ to the unit sphere centered at the origin in $\mathbb{R}^{d+1}$. Point $\boldsymbol{p} \in \mathbb{R}^d$ is projected to the sphere along the line through $\boldsymbol{p}$ and the "north pole" $(0, \ldots, 0, 1)$.
- *Find centerpoint.* Compute a *centerpoint* of the projected points in $\mathbb{R}^{d+1}$. This is a special point in the interior of the unit sphere, as described below.
- *Conformal map: Rotate and dilate.* Move the projected points in $\mathbb{R}^{d+1}$ on the surface of the unit sphere in two steps. First, rotate the projected points about the origin in $\mathbb{R}^{d+1}$ so that the centerpoint becomes a point $(0, \ldots, 0, r)$ on the $(d+1)$st axis. Second, dilate the points on the surface of the sphere so that the centerpoint becomes the origin. The dilation can be described as a scaling in $\mathbb{R}^d$: project the rotated points stereographically down to $\mathbb{R}^d$; scale the points in $\mathbb{R}^d$ by a factor of $\sqrt{(1-r)/(1+r)}$; and project the scaled points up to the unit sphere in $\mathbb{R}^{d+1}$ again.

- *Find great circle.* Choose a random great circle (i.e., d-dimensional unit sphere) on the unit sphere in $\mathbb{R}^{d+1}$.
- *Unmap and project down.* Transform the great circle to a circle in $\mathbb{R}^d$ by undoing the dilation, rotation, and stereographic projection.
- *Convert circle to separator.* The vertex separator C is the vertices whose boxes (in $\mathbb{R}^d$) either (i) intersect the separating circle, or (ii) are smaller than the separating circle and would intersect it if magnified by a factor of α. The two sets A and B are the remaining vertices whose boxes lie inside and outside the circle, respectively.

A *centerpoint* of a given set of points is a point (not necessarily one of the given points) such that every (hyper)plane through the centerpoint divides the given points approximately evenly (in the ratio $d:1$ or better, in $\mathbb{R}^d$). Every finite point set in $\mathbb{R}^d$ has a centerpoint, which can be found efficiently in random constant time [8]. After the projection and conformal mapping, the origin of $\mathbb{R}^{d+1}$ is a centerpoint for the mesh points. Therefore the mapped points are divided approximately evenly by every plane through the origin, that is, by every great circle on the unit sphere in $\mathbb{R}^{d+1}$.

Every great circle determines a separator C that satisfies all the conclusions of Theorem A.1 except the last one, on the size of the separator. Miller et al. show that the average size of the separators determined by all the great circles is as stated in the theorem, and therefore that a randomly chosen great circle probably gives a separator within a constant factor of the desired size.

Acknowledgments. I would like to thank John Gilbert, Gary Miller, Dafna Talmor, William Thurston, and Steve Vavasis for their invaluable contributions to our joint work on unstructured mesh computations, which motivated and greatly helped this work. I would like to thank Alan Edelman for helpful discussions on N-body simulation and parallel scientific computing in general and Yu Hu and Lennart Johnsson for the many helpful discussions we had in Cambridge, MA, and the inspiration I received from their CM5 data-parallel N-body implementation. I would like to express my appreciation to the 1995 SIAM Conference on Parallel Processing for Scientific Computing for bringing together experts on N-body simulation. Those talks and discussions, especially with Alan Edelman, J. Salmon, M. Warren, and Feng Zhao, provided a much clearer picture on the state-of-the-art of parallel N-body simulation, from both theoretical and practical viewpoints. I would like to thank Edmond Chow, Ananth Grama, and Vivek Sarin for helpful discussions and their suggestions to earlier drafts. Last, but not least, I would like to thank both referees for their very helpful comments.

REFERENCES

[1] C. R. Anderson, *An implementation of the fast multipole method without multipoles*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 932–947.

[2] A. W. Appel, *An efficient program for many-body simulation*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 85–103.

[3] J. Barnes and P. Hut, *A hierarchical $O(n \log n)$ force calculation algorithm*, Nature, 324 (1986), pp. 446–449.

[4] M. Bern, D. Eppstein, and S.-H. Teng, *Parallel construction of quadtrees and quality triangulations*, in Workshop on Algorithms and Data Structures, Lecture Notes in Computer Science 709, Springer-Verlag, 1993, pp. 188–199.

[5] J. A. Board, Z. S. Hakura, W. D. Elliott, and W. T. Ranklin, *Scalable variants of multipole-based algorithms for molecular dynamic applications*, in Parallel Processing for Scientific Computing, SIAM, Philadelphia, PA, 1995, pp. 295–300.

[6] P. B. Callahan and S. R. Kosaraju, *A decomposition of multidimensional point sets with applications to* k*-nearest-neighbors and* n*-body potential fields*, J. Assoc. Comput. Mach., 42 (1995), pp. 67–90.

[7] J. Carrier, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 669–686.

[8] K. Clarkson, D. Eppstein, G. L. Miller, C. Sturtivant, and S.-H. Teng, *Approximating center points with and without linear programming*, in Proceedings of 9th ACM Symposium on Computational Geometry, ACM, New York, 1993, pp. 91–98.

[9] K. Esselink, *The order of Appel's algorithm*, Inform. Process. Lett., 41 (1992), pp. 141–147.

[10] A. M. Frieze, G. L. Miller, and S.-H. Teng, *Separator based divide and conquer in computational geometry*, in Proceedings of the 1992 ACM Symposium on Parallel Algorithms and Architectures, ACM, New York, 1992, pp. 420–430.

[11] J. R. Gilbert, G. L. Miller, and S.-H. Teng, *Geometric mesh partitioning: Implementation and experiments*, SIAM J. Sci. Comput., to appear.

[12] A. Y. Grama, V. Kumar, and A. Sameh, *n-body simulation using message passing parallel computers*, in Parallel Processing for Scientific Computing, SIAM, Philadelphia, PA, 1995, pp. 355–360.

[13] L. Greengard and V. Rokhlin, *A fast algorithm for particles simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

[14] R. W. Hackney and J. W. Eastwood, Computer Simulation Using Particles, McGraw-Hill, New York, 1981.

[15] Y. Hu and S. L. Johnsson, *A Data Parallel Implementation of Hierarchical* N*-Body Methods*, Tech. report TR-26-94, Harvard University, Cambridge, MA, 1994.

[16] P.-F. Liu, *The Parallel Implementation of N-Body Algorithms*, Ph.D. thesis, Yale University, New Haven, CT, 1994.

[17] J. Makino, M. Taiji, T. Ebisuzaki, and D. Sugimoto, *Grape*-4: *A special-purpose computer for gravitational* N*-body problems*, in Parallel Processing for Scientific Computing, SIAM, Philadelphia, PA, 1995, pp. 355–360.

[18] O. Meirhaeghe and S.-H. Teng, *Random* k-d *Trees for Quality Load Balancing in Parallel* N*-Body Simulation*, Department of Computer Science, University of Minnesota, Minneapolis, MN, 1996.

[19] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis, *Automatic mesh partitioning*, in Sparse Matrix Computations: Graph Theory Issues and Algorithms, A. George, J. Gilbert, and J. Liu, eds., IMA Volumes in Mathematics and Its Applications, Springer-Verlag, New York, 1993.

[20] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis, *Geometric separators for finite element meshes*, SIAM J. Sci. Comput., to appear.

[21] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis, *Separators for sphere-packings and nearest neighborhood graphs*, J. Assoc. Comput. Mach., 44 (1997), pp. 1–29.

[22] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White, *Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-find integral equations of potential theory*, SIAM J. Sci. Comput., 15 (1994), pp. 713–735.

[23] J. K. Salmon, *Parallel Hierarchical N-body Methods*, Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1990.

[24] J. K. Salmon, M. S. Warren, and G. S. Winckelmans, *Fast parallel tree codes for gravitational and fluid dynamical* N*-body problems*, Internat. J. Supercomput. Appl., 8 (1994), pp. 129–142.

[25] H. Samet, *The quadtree and related hierarchical data structures*, ACM Comput. Surv., 16 (1984), pp. 187–260.

[26] K. E. Schmidt and M. A. Lee, *Implementing the fast multipole method in three dimensions*, J. Statist. Phys., 63 (1991), pp. 1223–1235.

[27] H. D. Simon and S.-H. Teng, *How good is recursive bisection*? SIAM J. Sci. Comput., 18 (1997), pp. 1436–1445.

[28] J. P. Singh, C. Holt, T. Ttsuka, A. Gupta, and J. L. Hennessey, *Load Balancing and Data Locality in Hierarchical* N*-Body Methods*, Tech. report CSL-TR-92-505, Stanford University, Stanford, CA, 1992.

[29] G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice–Hall, Englewood Cliffs, NJ, 1973.

[30] S.-H. Teng, *Points, Spheres, and Separators: A Unified Geometric Approach to Graph Partitioning*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 1991.

[31] S.-H. Teng, *A Geometric Approach to Parallel Hierarchical and Adaptive Computing on Unstructured Meshes*, in Fifth SIAM Conference on Applied Linear Algebra, SIAM, Philadelphia, PA, 1994, pp. 51–57.

[32] S.-H. Teng, E. Pramono, and J. Ruppert, *Partitioning Weighted Meshes for Parallel Processing*, Tech. report, NASA Ames Research Center, Ames, IA, 1994.

[33] F. Zhao, *An $O(n)$ Algorithm for Three-Dimensional* N-*Body Simulation*, Tech. report TR AI Memo 995, MIT, Cambridge, MA, 1987.

[34] F. Zhao and S. L. Johnsson, *The parallel multipole method on the connection machines*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 1420–1437.

[35] P. M. Vaidya, *An $O(n \log n)$ algorithm for the all-nearest-neighbors problem*, Discrete & Comput. Geom., 4 (1989), pp. 101–115.