

Accelerating Neighbor Querying in Fast Multipole Method with Morton Coding

Ruoyu Huang, Longfei Shi, Jingzhi Zeng, Junan Lai, Xikui Ma, and Tianyu Dong*

School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China

* Authors to whom correspondence should be addressed. Email:tydong@mail.xjtu.edu.cn

The implementation of fast multipole method (FMM) requires querying neighbor and interaction cells in a hierarchical tree, which can be complicated when the tree is unbalanced. We present a fast and easy-to-implement method to query neighbor and interaction cells in unbalanced trees. By uniformly splitting and indexing the coordinate axes of each layer, the coordinates of the tree nodes are converted into Morton codes, and the querying neighbor querying can be readily realized by bit-shifting the Morton code, which allows fast neighbor and interaction cells querying.

Index Terms—Electrostatics, Fast multipole method (FMM), Nearest neighbor search, Numerical analysis

I. INTRODUCTION

FAST multipole method (FMM) is an efficient algorithm when solving N -body interaction problems, which relies on hierarchical tree structures for spatial decomposition and neighbor queries. When handling the non-uniform distribution of sources, unbalanced tree structures are typically involved, which become expensive for neighbor querying. To address the issue, various methods have been developed, including adjacency queries based on graph theory [1], shortest path queries [2], cell coordinate computation [3], and ray intersection algorithms [4], to name a few, which are expensive and inefficient in FMM. Here, we propose a simple and efficient neighbor construction method based on Morton coding with only bitwise operations, mitigating complicated algorithms.

II. THEORY AND METHODS

A. Fast multipole method and hierarchical tree structure

Without loss of generality, the electrostatic potential can be simply expressed as $\varphi(\mathbf{r}) = \sum_{j=1}^N \varepsilon_0^{-1} q_j G_0(\mathbf{r}, \mathbf{r}'_j)$, where $G_0 = 1/(4\pi R)$ is the Green's function in free space with $R = |\mathbf{r} - \mathbf{r}'_j|$ being the distance between the field point $\mathbf{r}$ and the source point $\mathbf{r}'_j$; q_j is the j -th point charge; and N is the number of sources. When N is large, the computation becomes involved, and the so-called FMM is typically used. The implementation of FMM begins with spatial partitioning, as shown in Fig. 1, which can be treated in terms of hierarchical structures. Within the FMM, the potential at a given point can be contributed from the far and near sources, i.e. $\varphi = \varphi^{\text{near}} + \varphi^{\text{far}}$, where $\varphi^{\text{near}} = \sum_{j=1}^{N^{\text{near}}} \varepsilon_0^{-1} q_j G_0(\mathbf{r}, \mathbf{r}'_j)$ is directly computed and φ^{far} is obtained by virtue of the multipole expansion of the Green's function, which reads as

$$\varphi^{\text{far}}(\mathbf{r}_{k,i}) = \sum_{l=1}^{N^{\text{interaction}}_{\text{cell},k}} \sum_{n=0}^{\infty} \sum_{m=-n}^n L_n^m(\mathbf{r}_{k,i}, \mathbf{r}_k^*) F_n^m(\mathbf{r}'_{l,:}, \mathbf{r}_l^*, \mathbf{r}_k^*), \quad (1)$$

where $\mathbf{r}_{k,i}$ denotes the i -th field point inside the k -th cell; $N^{\text{interaction}}_{\text{cell},k}$ is the number of interaction cells of k -th target cell.

Here, L_n^m and F_n^m represent the local expansion and the far field expansion, respectively, which can be expressed as

$$L_n^m(\mathbf{r}_{k,i}, \mathbf{r}_k^*) = \tilde{r}^n Y_n^m(\tilde{\theta}, \tilde{\phi}), \quad (2a)$$

$$F_n^m(\mathbf{r}'_{l,:}, \mathbf{r}_l^*, \mathbf{r}_k^*) = \sum_{n'=0}^{\infty} \sum_{m'=-n'}^{n'} S_{n,n'}^{m,m'}(\mathbf{r}_k^*, \mathbf{r}_l^*) M_{n'}^{m'}(\mathbf{r}'_{l,:}, \mathbf{r}_l^*), \quad (2b)$$

where $\mathbf{r}_k^*$ denotes the expansion center in the k -th target cell; $Y_n^m(\tilde{\theta}, \tilde{\phi}) = \sqrt{(n-|m|)!/(n+|m|)!} P_n^{|m|}(\cos \tilde{\theta}) e^{im\tilde{\phi}}$ is the spherical harmonic function in terms of the associated Legendre function $P_n^{|m|}$ with $(\tilde{r}, \tilde{\theta}, \tilde{\phi})$ denoting the spherical coordinates associated with $\mathbf{r}_{k,i} - \mathbf{r}_k^*$. In addition, $\mathbf{r}'_{l,:}$ and $\mathbf{r}_l^*$ denote all source points and the expansion center in the l -th interaction cell, respectively, as indicated in Fig. 1(a); $S_{n,n'}^{m,m'} = J_{n,m}^{n',m'} Y_{n'-m}^{n+n'}(\alpha, \beta) / \rho^{n+n'+1}$ is the conversion of the multipole expansion into the local expansion, with $J_{n,m}^{n',m'} = i^{|m-m'|} i^{|m|-|m'|} A_n^m A_{n'}^{m'} / [(-1)^{n'} A_{n+n'}^{m'-m}]$ and (ρ, α, β) being the spherical coordinates associated with $\mathbf{r}_l^* - \mathbf{r}_k^*$; $A_n^m = (-1)^n / \sqrt{(n-m)!(n+m)!}$. Moreover, $M_{n'}^{m'}(\mathbf{r}'_{l,:}, \mathbf{r}_l^*)$ is the multipole expansion, which reads as $M_{n'}^{m'}(\mathbf{r}'_{l,:}, \mathbf{r}_l^*) = \sum_{j=1}^{N_l} q_j \tilde{r}_j^{n'} Y_{n'}^{-m'}(\tilde{\theta}'_j, \tilde{\phi}'_j)$, where $(\tilde{r}'_j, \tilde{\theta}'_j, \tilde{\phi}'_j)$ denotes the polar coordinate associated with the vector $\mathbf{r}'_{l,j} - \mathbf{r}_l^*$; here, $\mathbf{r}'_{l,j}$ and N_l are the j -th source point and number of source points in l -th interaction cell, respectively.

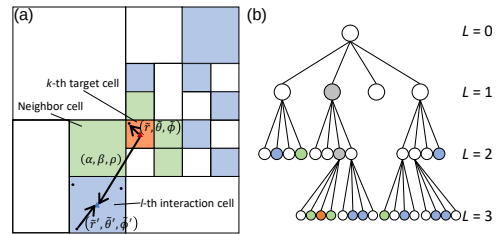


Fig. 1. Illustration of spatial partitioning and hierarchical tree structure in the FMM. (a) Spatial division at the third layer for a non-uniform distribution, demonstrating the neighboring cells (green) and interaction cells (blue) of the target cell (orange). The red and blue triangles represent the expansion centers of the target and interacting cells, respectively. (b) The corresponding unbalanced hierarchical tree structure.

One of the fundamental tasks in FMM is determining the neighbor and interaction cells of a target cell. For balanced trees, neighbors are at the same level and can be readily identified. However, traditional approaches often require traversing the unbalanced trees to find the least common ancestor, which reduces query efficiency, which can be improved by the proposed algorithm based on Morton code.

B. Neighbor querying with Morton coding

Fig. 2 illustrates the proposed Morton coding procedure in each layer. In general, the higher-order bits of the Morton code of the child cells for the same parent node are identical (except for the root cell). For example, a cell with the Morton code 11_2 in the first layer will have the same high two bits for all of its child cells in the second layer. As a result, the relationship between cells at different levels can be determined.

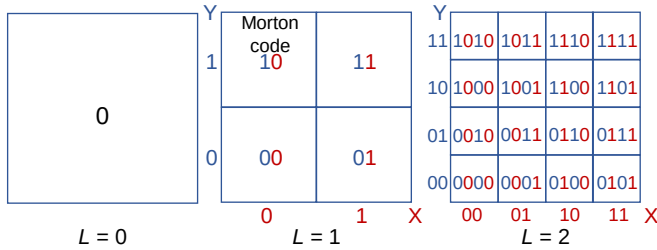


Fig. 2. **Illustration of the Morton encoding process.** The coordinate axes are divided into 2^L identical parts in the L -th layer and are numbered. The coordinates of the cell are bit-interleaved and transformed into a one-dimensional Morton code, which is represented in a binary code (alternating red and blue).

Fig. 3 displays an example of an unbalanced quad-tree with non-uniform distribution. The determination of neighbor and interaction cells is based on two conditions: (i) the neighbors of the target cell must exist within the child cells of its parent at the same level (excluding itself), the parent cell's neighboring cells, and the child cells of the parent's neighbors; and (ii) the parent node of a neighbor at the same level must be a neighbor of the target node.

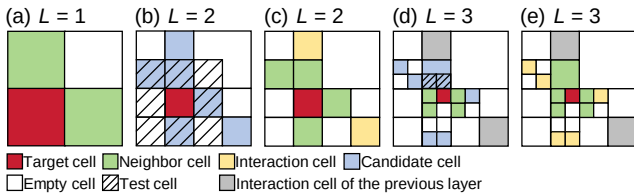


Fig. 3. **Querying procedure for the neighbor and interaction cells of the target node in each layer.**

Without loss of generality, we use Fig. 3 as an example to explain the querying method for identifying neighbor cells (green) and interaction cells (yellow) of target nodes across different layers. At the first layer (Fig. 3(a)), the root cell encompassing all sources is divided into smaller cells. At

this level, all neighbor cells are at the same layer, and there are no interaction cells, allowing neighbors to be directly identified. In the second layer (Fig. 3(b)), spatial division produces candidate cells (blue) satisfying the aforementioned condition. The Morton code of the target node is offset to compute the Morton codes of surrounding test units, which are then compared with candidate cells. Matches are classified as neighbors, while remaining candidates, having no lower-level counterparts, are classified as interaction cells (Fig. 3(c)). At the third layer (Fig. 3(d)), a similar procedure is followed. Candidate cells are matched with Morton codes of test cells to identify neighbors. Additionally, test cells belonging to child cells of candidates lead to the classification of those candidates as neighbors. The remaining candidates are considered interaction cells (Fig. 3(e)). Algorithm 1 summarizes the proposed Morton code-based query method for the neighbor and interaction cells. We shall stress that the proposed method supports only the structured hierarchical trees such as quad-trees and octrees, and quick querying methods for unstructured and unbalanced trees require further investigation.

Algorithm 1 Neighbor and interaction querying.

Require: I - target cell;
 L - layer of target cell;
 C - candidate cell determined by the condition (i).

- 1: neighbor, interaction = [] // initialize as an empty set
- 2: **if** $L = 0$ **then**
- 3: **return** // the root cell has no neighbors
- 4: **end if**
- 5: **if** $L = 1$ **then**
- 6: /* cells with same parent are neighbors */
- 7: **return** neighbor = C
- 8: **end if**
- 9: /* find neighbors at the same and lower layer */
- 10: neighbor += findSameLayerNeighbor(I, L, C)
- 11: neighbor += findLowLayerNeighbor(I, L, C)
- 12: **return** neighbor, interaction = C - neighbor

ACKNOWLEDGMENT

The authors acknowledge the financial support from the State Grid Corporation of China under grant no. 5500-202155584A-0-5-SF.

REFERENCES

- [1] H. Samet, "Neighbor finding in images represented by octrees," *Comput. Vis. Graph. Image Process.*, vol. 46, no. 3, pp. 367–386, 1989.
- [2] M. Shneier, "Path-length distances for quadtrees," *Inf. Sci.*, vol. 23, no. 1, pp. 49–67, 1981.
- [3] J. A. Fernández-Fernández, L. Westhofen, F. Löschner, S. R. Jeske, A. Longva, and J. Bender, "Fast octree neighborhood search for SPH simulations," *ACM Trans. Graph.*, vol. 41, no. 6, pp. 1–13, 2022.
- [4] Y. Li, Q. Pan, X. Liu, and J. Xiong, "Monte Carlo CAD geometric transport acceleration method based on octree neighbor searching," *Nucl. Eng. Des.*, vol. 425, p. 113323, 2024.