Encyclopedia entry on "Fast Multipole Methods."

Per-Gunnar Martinsson, University of Colorado at Boulder, August 2012

Short definition. The Fast Multipole Method (FMM) is an algorithm for rapidly evaluating all pairwise interactions in a system of N electrical charges. While the direct computation requires $O(N^2)$ work, the FMM carries out this task in only $O(N)$ operations. A parameter in the FMM is the prescribed accuracy ε to within which the electrostatic potentials and forces are computed. The choice of ε affects the scaling constant implied by the $O(N)$ notation. A more precise estimate of the time required (in d dimensions) is $O(N \log^{(d-1)}(1/\varepsilon))$ as $\varepsilon \to 0$.

More generally, the term "FMM" refers to a broad class of algorithms with linear or close to linear complexity for evaluating all pairwise interactions between N particles, given some pairwise interaction kernel (e.g. the kernels associated with elasticity, gravitation, wave propagation, etc.). An important application is the evaluation of the matrix-vector product $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ where $\mathbf{A}$ is a dense $N \times N$ matrix arising from the discretization of an integral operator.

The classical FMM and its descendants rely on quad-trees or oct-trees to hierarchically subdivide the computational domain, and are sometimes called "tree code" algorithms. The tree structure enables such schemes to adaptively refine non-uniform charge distributions, and are well-suited for multi-core and parallel computing platforms.

1. **Introduction.** To introduce the concepts supporting fast summation techniques like the FMM, we will in this note describe a bare-bones algorithm for solving the problem addressed in the original work [10] of Greengard and Rokhlin, namely the evaluation of all pairwise interactions between a set of N electrical charges in the plane. The basic technique has since [10] was published been substantially improved and extended. Analogous fast summation techniques have also been developed for related summation problems, most notably those associated with acoustic and electro-magnetic scattering theory. These improvements and extensions are reviewed in Section 11.

2. **Notation.** We let $\{\boldsymbol{x}_i\}_{i=1}^N$ denote the locations of a set of electric charges, and let $\{q_i\}_{i=1}^N$ denote their source strengths. Our task is then to evaluate the potentials

$$u_i = \sum_{j=1}^{N} g(\boldsymbol{x}_i, \boldsymbol{x}_j)\, q_j, \qquad i = 1,\, 2,\, \ldots,\, N, \tag{1}$$

where $g(\boldsymbol{x}, \boldsymbol{y})$ is the interaction potential of electrostatics in the plane

$$g(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \log|\boldsymbol{x} - \boldsymbol{y}| & \boldsymbol{x} \neq \boldsymbol{y}, \\ 0 & \boldsymbol{x} = \boldsymbol{y}. \end{cases} \tag{2}$$

(We omit the common scaling by $-\frac{1}{2\pi}$.) Let Ω denote a square that holds all points, see Fig. 1(a).

It will be convenient to use complex notation. We think of each source location $\boldsymbol{x}_i$ as a point in the complex plane and let G denote the complex interaction potential

$$G(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \log(\boldsymbol{x} - \boldsymbol{y}) & \boldsymbol{x} \neq \boldsymbol{y}, \\ 0 & \boldsymbol{x} = \boldsymbol{y}. \end{cases}$$

We introduce a vector $\mathbf{q} \in \mathbb{C}^N$ and a matrix $\mathbf{A} \in \mathbb{C}^{N \times N}$ via

$$\mathbf{q}(i) = q_i, \qquad \text{and} \qquad \mathbf{A}(i,j) = G(\boldsymbol{x}_i,\, \boldsymbol{x}_j) \qquad i, j = 1,\, 2,\, 3,\, \ldots,\, N.$$

We then seek to evaluate the matrix-vector product

$$\mathbf{u} = \mathbf{A}\,\mathbf{q}. \tag{3}$$

The real potentials u_i defined by (1) are given by real parts of the entries of $\mathbf{u}$.

3. **The general idea.** The key to rapidly evaluating the sum (1) is that the kernel $g(\boldsymbol{x}, \boldsymbol{y})$ defined by (2) is smooth when $\boldsymbol{x}$ and $\boldsymbol{y}$ are not close. To illustrate how this can be exploited, let us first consider a simplified situation where we are given a set of electric sources $\{q_j\}_{j=1}^N$ at locations $\{\boldsymbol{y}_j\}_{j=1}^N$ in one box Ω_σ and seek the potentials these sources induce at some target locations $\{\boldsymbol{x}_i\}_{i=1}^M$ in a different box Ω_τ. In other words, we seek to evaluate the sum

$$u_i = \sum_{j=1}^{N} g(\boldsymbol{x}_i, \boldsymbol{y}_j)\, q_j, \qquad i = 1,\, 2,\, \ldots,\, M. \tag{4}$$

Since g is smooth in this situation, we can approximate it by a short sum of tensor products

$$g(\boldsymbol{x}, \boldsymbol{y}) \approx \sum_{p=0}^{P-1} B_p(\boldsymbol{x})\, C_p(\boldsymbol{y}), \qquad \text{when } \boldsymbol{x} \in \Omega_\tau,\ \boldsymbol{y} \in \Omega_\sigma, \tag{5}$$

where P is a small integer called the *interaction rank*. (How to construct the functions B_p and C_p, and how to choose P will be discussed in Section 4.) As a result, an approximation to the sum (4) can be constructed via the two steps

$$\hat{q}_p = \sum_{j \in I_\sigma} C_p(\boldsymbol{x}_j)\, q_j, \qquad p = 0,\, 1,\, 2,\, \ldots,\, P-1. \tag{6}$$

and

$$u_i \approx \sum_{p=0}^{P-1} B_p(\boldsymbol{x}_i)\, \hat{q}_p, \qquad i = 1,\, 2,\, \ldots,\, M. \tag{7}$$

While evaluating (4) directly requires $M\,N$ operations, evaluating (6) and (7) requires only $P(M+N)$ operations. The power of this observation stems from the fact that high accuracy is achieved even for small P when the regions Ω_σ and Ω_τ are moderately well separated, cf. Section 9.

Using matrix notation, the approximation (5) implies that the $M \times N$ matrix $\mathbf{A}$ with entries $\mathbf{A}(i,j) = g(\boldsymbol{x}_i, \boldsymbol{y}_j)$ admits an approximate rank-P factorization $\mathbf{A} \approx \mathbf{B}\,\mathbf{C}$. Then clearly the matrix-vector product $\mathbf{Aq}$ can cheaply be evaluated via $\mathbf{Aq} \approx \mathbf{B}\,(\mathbf{Cq})$.

In the problem (1), the summation problem that we are actually interested in, the sets of target locations and source locations coincide. In this case, no one relation like (5) can hold for all combinations of target and source points. Instead, we are going to cut the domain up into pieces, and use approximations such as (5) to evaluate interactions between distant pieces, and use direct evaluation only for points that are close. Equivalently, one could say that we will evaluate the matrix-vector product (3) by exploiting rank-deficiencies in off-diagonal blocks of $\mathbf{A}$.

The algorithm will be introduced incrementally. Section 4 formalizes the discussion of the case where target and source boxes are separate. Section 5 describes a method based on a single-level tessellation of the domain. Section 6 provides a conceptual description of a multi-level algorithm with $O(N)$ complexity, with details given in Sections 7 and 8.

4. **Multipole expansions.** We start by considering a subproblem of (1) corresponding to the interaction between two disjoint subsets Ω_σ and Ω_τ, as illustrated in Figure 1(b). Specifically, we seek to evaluate the potential at all points in Ω_τ (the "target points") caused by sources in Ω_σ. To formalize, let I_σ and I_τ be index sets pointing to the locations inside each box so that, e.g.,

$$i \in I_\sigma \qquad \Leftrightarrow \qquad \boldsymbol{x}_i \in \Omega_\sigma.$$

Our task is then to evaluate the sums

$$v_i = \sum_{j \in I_\sigma} G(\boldsymbol{x}_i, \boldsymbol{x}_j)\, q_j, \qquad i \in I_\tau. \tag{8}$$

In matrix notation, (8) is equivalent to the matrix vector-product

$$\mathbf{v}^\tau = \mathbf{A}(I_\tau,\, I_\sigma)\, \mathbf{q}(I_\sigma). \tag{9}$$

We will next derive an approximation like (5) for the kernel in (8). To this end, let $\boldsymbol{c}_\sigma$ and $\boldsymbol{c}_\tau$ denote the centers of Ω_σ and Ω_τ, respectively. Then, for $\boldsymbol{y} \in \Omega_\sigma$ and $\boldsymbol{x} \in \Omega_\tau$,

$$\begin{aligned} G(\boldsymbol{x}, \boldsymbol{y}) = \log\big(\boldsymbol{x} - \boldsymbol{y}\big) &= \log\big((\boldsymbol{x} - \boldsymbol{c}_\sigma) - (\boldsymbol{y} - \boldsymbol{c}_\sigma)\big) \\ &= \log\big(\boldsymbol{x} - \boldsymbol{c}_\sigma\big) + \log\left(1 - \frac{\boldsymbol{y} - \boldsymbol{c}_\sigma}{\boldsymbol{x} - \boldsymbol{c}_\sigma}\right) = \log\big(\boldsymbol{x} - \boldsymbol{c}_\sigma\big) - \sum_{p=1}^{\infty} \frac{1}{p} \frac{(\boldsymbol{y} - \boldsymbol{c}_\sigma)^p}{(\boldsymbol{x} - \boldsymbol{c}_\sigma)^p}, \end{aligned} \tag{10}$$

where the series converges whenever $|\boldsymbol{y} - \boldsymbol{c}_\sigma| < |\boldsymbol{x} - \boldsymbol{c}_\sigma|$. Observe that the last expression in (10) is precisely of the form (5) with $C_p(\boldsymbol{y}) = -\frac{1}{p}(\boldsymbol{y} - \boldsymbol{c}_\sigma)^p$ and $B_p(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{c}_\sigma)^{-p}$. When the sum is truncated after $P - 1$ terms, the error incurred is roughly of size $\big(|\boldsymbol{y} - \boldsymbol{c}_\sigma|/|\boldsymbol{x} - \boldsymbol{c}_\sigma|\big)^P$.

We define the *outgoing expansion* of Ω_σ as the vector $\hat{\mathbf{q}}^\sigma = \{\hat{q}_p^\sigma\}_{p=0}^{P-1}$ where

$$\begin{cases} \hat{q}_0^\sigma = \displaystyle\sum_{j \in I_\sigma} q_j \\ \hat{q}_p^\sigma = \displaystyle\sum_{j \in I_\sigma} -\frac{1}{p}\big(\boldsymbol{x}_j - \boldsymbol{c}_\sigma\big)^p\, q_j, \qquad p = 1,\, 2,\, 3,\, \ldots,\, P-1. \end{cases} \tag{11}$$

The vector $\hat{\mathbf{q}}^\sigma$ is a compact representation of the sources in Ω_σ. It contains all information needed to evaluate the field $v(\boldsymbol{x}) = \sum_{j \in I_\sigma} G(\boldsymbol{x}, \boldsymbol{x}_j)\, q_j$ when $\boldsymbol{x}$ is a point "far away" from Ω_σ.

It turns out to be convenient to also define an *incoming expansion* for Ω_τ. The basic idea here is that for $\boldsymbol{x} \in \Omega_\tau$, the potential

$$v(\boldsymbol{x}) = \sum_{j \in I_\sigma} G(\boldsymbol{x}, \boldsymbol{x}_j)\, q_j = \log(\boldsymbol{x} - \boldsymbol{c}_\sigma)\, \hat{q}_0^\sigma + \sum_{p=1}^{\infty} \frac{1}{(\boldsymbol{x} - \boldsymbol{c}_\sigma)^p}\, \hat{q}_p^\sigma \tag{12}$$

is a harmonic function on Ω_τ. In consequence, it has a convergent expansion

$$v(\boldsymbol{x}) = \sum_{p=0}^{\infty} \big(\boldsymbol{x} - \boldsymbol{c}_\tau\big)^p\, \hat{v}_p^\tau.$$

A simple computation shows that the complex numbers $\{\hat{v}_p^\tau\}_{p=0}^{\infty}$ can be obtained from $\{\hat{q}_p\}_{p=0}^{\infty}$ via

$$\begin{cases} \hat{v}_0^\tau = \hat{q}_0^\sigma \log(\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma) + \displaystyle\sum_{j=1}^{\infty} \hat{q}_j^\sigma\, (-1)^j\, \frac{1}{(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^j}, \\ \hat{v}_i^\tau = -\hat{q}_0^\sigma \dfrac{1}{i(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^i} + \displaystyle\sum_{j=1}^{\infty} \hat{q}_j^\sigma (-1)^j \binom{i+j-1}{j-1} \frac{1}{(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{i+j}}. \end{cases} \tag{13}$$

The vector $\hat{\mathbf{v}}^\tau = \{\hat{v}_p^\tau\}_{p=0}^{P-1}$ is the *incoming expansion* for Ω_τ generated by the sources in Ω_σ. It is a compact (approximate) representation of the harmonic field v defined by (12).

The linear maps introduced in this section can advantageously be represented via matrices that we refer to as *translation operators*. Let N_σ and N_τ denote the number of points in Ω_σ and Ω_τ, respectively. The map (11) can then upon truncation be written

$$\hat{\mathbf{q}}^\sigma = \mathbf{T}_\sigma^{\text{ofs}}\, \mathbf{q}(I_\sigma),$$

where $\mathbf{T}_\sigma^{\text{ofs}}$ is a $P \times N_\sigma$ matrix called the *outgoing-from-sources* translation operator with the entries implied by (11). Analogously, (13) can upon truncation be written $\hat{\mathbf{v}}^\tau = \mathbf{T}_{\tau,\sigma}^{\text{ifo}}\, \hat{\mathbf{q}}^\sigma$, where $\mathbf{T}_{\tau,\sigma}^{\text{ifo}}$ is the *incoming-from-outgoing* translation operator. Finally, the *targets-from-incoming* translation

operator is the matrix $\mathbf{T}_\tau^{\text{tfi}}$ such that $\mathbf{v}^\tau = \mathbf{T}_\tau^{\text{tfi}}\,\hat{\mathbf{v}}^\tau$, where $\mathbf{v}^\tau$ is an approximation to the field v defined by (12), in other words $\mathbf{T}_\tau^{\text{tfi}}(i,p) = (\boldsymbol{x}_i - \boldsymbol{c}_\tau)^{p-1}$. These three translation operators are factors in an approximate rank-P factorization

$$\begin{array}{cccccc} \mathbf{A}(I_\tau, I_\sigma) & \approx & \mathbf{T}_\tau^{\text{tfi}} & \mathbf{T}_{\tau,\sigma}^{\text{ifo}} & \mathbf{T}_\sigma^{\text{ofs}}. \\ N_\tau \times N_\sigma & & N_\tau \times P & P \times P & P \times N_\sigma \end{array} \tag{14}$$

A diagram illustrating the factorization (14) is given as Figure 2.

Remark 1. *The terms "outgoing expansion" and "incoming expansion" are slightly non-standard. The corresponding objects were in the original papers called "multipole expansion" and "local expansion," and these terms continue to be commonly used; even in summation schemes where the expansions have nothing to do with multipoles. Correspondingly, what we call the "incoming-from-outgoing" translation operator is often called the "multipole-to-local" or "M2L" operator.*

5. **A single-level method.** Having dealt with the simplified situation where the source points are separated from the target points in Section 4, we now return to the original problem (1) where the two sets of points are the same. In this section, we construct a simplistic method that does not achieve $O(N)$ complexity but will help us introduce some concepts.

Sub-divide the box Ω into a grid of $m \times m$ equisized smaller boxes $\{\Omega_\tau\}_{\tau=1}^{m^2}$ as shown in Figure 3(a). As in Section 4, we let for each box τ the index vector I_τ list the points inside Ω_τ and let $\boldsymbol{c}_\tau$ denote the center of τ. The vector $\hat{\mathbf{q}}^\tau$ denotes the *outgoing expansion* of τ, as defined by (11).

For a box τ, let $\mathcal{L}_\tau^{\text{nei}}$ denote the list of *neighbor boxes*; these are the boxes that directly touch τ (there will be between 3 and 8 of them, depending on where τ is located in the grid). The remaining boxes are collected in the list of *far-field boxes* $\mathcal{L}_\tau^{\text{far}}$. Figure 3(b) illustrates the lists.

The sum (1) can now be approximated via three steps:

(1) *Compute the outgoing expansions:* Loop over all boxes τ. For each box, compute its outgoing expansion $\hat{\mathbf{q}}^\tau$ via the *outgoing-from-sources* translation operator:

$$\hat{\mathbf{q}}^\tau = \mathbf{T}_\tau^{\text{ofs}}\,\mathbf{q}(I_\tau).$$

(2) *Convert outgoing expansions to incoming expansions:* Loop over all boxes τ. For each box, construct a vector $\hat{\mathbf{u}}^\tau$ called the *incoming expansion*. It represents the contribution to the potential in τ from sources in all boxes in the far-field of τ and is given by

$$\hat{\mathbf{u}}^\tau = \sum_{\sigma \in \mathcal{L}_\tau^{\text{far}}} \mathbf{T}_{\tau,\sigma}^{\text{ifo}}\,\hat{\mathbf{q}}^\sigma.$$

(3) *Compute near interactions:* Loop over all boxes τ. Expand the incoming expansion and add the contributions from its neighbors via direct summation:

$$\mathbf{u}(I_\tau) = \mathbf{T}_\tau^{\text{tfi}}\,\hat{\mathbf{u}}^\tau + \mathbf{A}(I_\tau, I_\tau)\,\mathbf{q}(I_\tau) + \sum_{\sigma \in \mathcal{L}_\tau^{\text{nei}}} \mathbf{A}(I_\tau, I_\sigma)\,\mathbf{q}(I_\sigma).$$

The asymptotic complexity of the method as the number of particles N grows depends on how the number m is picked. If the number m^2 of boxes is large, then Steps 1 and 3 are cheap, but Step 2 is expensive. The optimal choice is $m^2 \sim N^{2/3}$, and leads to overall complexity $O(N^{4/3})$.

6. **Conceptual description of a multi-level algorithm.** To achieve linear complexity in evaluating (1), the FMM uses a multi-level technique in which the computational domain Ω is split into a tree of boxes, cf. Figure 4. It evaluates the sum (1) in two passes over the tree, one going upwards (from smaller boxes to larger) and one going downwards:

The upwards pass: In the upwards pass, the outgoing expansion is computed for all boxes. For a leaf box τ, the straight-forward approach described in Section 4 is used. For a box τ that has children, the outgoing expansion is computed not directly from the sources located in the box, but from the outgoing expansions of its children, which are already available.

The downwards pass: In the downwards pass, the incoming expansion is computed for all boxes. This is done by converting the outgoing expansions constructed in the upwards pass to incoming expansions via the formula (13). The trick is to organize the computation so that each conversion happens at its appropriate length-scale. Some further machinery is required to describe exactly how this is done, but the end result is that the FMM computes the incoming expansion for a leaf box τ from the outgoing expansions of a set of $O(\log N)$ boxes that are sufficiently well-separated from the target that the expansions are all accurate, cf. Figure 6

Once the upwards and downwards passes have been completed, the incoming expansion is known for all leaf boxes. All that remains is then to expand the incoming expansion into potentials and adding the contributions from sources in the near-field via direct computations.

In order to formally describe the upwards and downwards passes, we need to introduce two new translation operators (in addition to the three introduced in Section 4). Let Ω_τ be a box containing a smaller box Ω_σ which in turn contains a set of sources. Let $\hat{\mathbf{q}}^\sigma$ denote the outgoing expansion of these sources around the center $\boldsymbol{c}_\sigma$ of σ. These sources could also be represented via an outgoing expansion $\hat{\mathbf{q}}^\tau$ around the center $\boldsymbol{c}_\tau$ of τ. One can show that

$$\begin{cases} \hat{q}^\tau_0 = \hat{q}^\sigma_0, \\ \hat{q}^\tau_i = -\hat{q}^\sigma_0 \dfrac{1}{i}(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^i + \displaystyle\sum_{j=1}^{i} \hat{q}^\sigma_j \binom{i-1}{j-1} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{i-j}. \end{cases} \tag{15}$$

Analogously, now suppose that a set of sources that are distant to Ω_τ give rise to a potential v in τ represented by an incoming expansion $\hat{\mathbf{u}}^\tau$ centered around $\boldsymbol{c}_\tau$. Then the corresponding incoming representation $\hat{\mathbf{u}}^\sigma$ of v centered around $\boldsymbol{c}_\sigma$ is given by

$$\hat{u}^\sigma_i = \sum_{j=i}^{\infty} \hat{u}^\tau_j \binom{j}{i} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{j-i}. \tag{16}$$

Upon truncating the series in (15) and (16) to the first P terms, we write (15) and (16) in matrix form using the *outgoing-from-outgoing* translation operator $\mathbf{T}^{\text{ofo}}_{\tau,\sigma}$ and the *incoming-from-incoming* translation operator $\mathbf{T}^{\text{ifi}}_{\sigma,\tau}$,

$$\hat{\mathbf{q}}^\tau = \mathbf{T}^{\text{ofo}}_{\tau,\sigma}\, \hat{\mathbf{q}}^\sigma \qquad \text{and} \qquad \hat{\mathbf{u}}^\sigma = \mathbf{T}^{\text{ifi}}_{\sigma,\tau}\, \hat{\mathbf{u}}^\tau.$$

Both $\mathbf{T}^{\text{ofo}}_{\tau,\sigma}$ and $\mathbf{T}^{\text{ifi}}_{\sigma,\tau}$ are matrices of size $P \times P$.

7. **A tree of boxes.** Split the square Ω into 4^L equisized smaller boxes, where the integer L is chosen to be large enough that each box holds only a small number of points. (The optimal number of points to keep in a box depends on many factors, but having about 100 points per box is often reasonable.) These 4^L equisized small boxes form the *leaf boxes* of the tree. We merge the leaves by sets of four to form 4^{L-1} boxes of twice the side-length, and then continue merging by sets of four until we recover the original box Ω, which we call the *root*.

The set consisting of all boxes of the same size forms what we call a *level*. We label the levels using the integers $\ell = 0, 1, 2, \ldots, L$, with $\ell = 0$ denoting the root, and $\ell = L$ denoting the leaves.

Given a box τ in the hierarchical tree, we next define some index lists, cf. Figure 5:

- The parent of τ is the box on the next coarser level that contains τ.
- The children of τ is the set $\mathcal{L}_\tau^{\text{child}}$ of boxes whose parent is τ.
- The neighbors of τ is the set $\mathcal{L}_\tau^{\text{nei}}$ of boxes on the same level that directly touch τ.
- The interaction list of τ is the set $\mathcal{L}_\tau^{\text{int}}$ of all boxes σ such that (1) σ and τ are on the same level, (2) σ and τ do not touch, and (3) the parents of σ and τ do touch.

8. **The classical Fast Multipole Method.** We now have all tools required to describe the classical FMM in detail.

Given a set of sources $\{q_i\}_{i=1}^N$ with associated locations $\{\boldsymbol{x}_i\}_{i=1}^N$, the first step is to find a minimal square Ω that holds all points. Next, subdivide Ω into a hierarchy of smaller boxes as described in Section 7. Then fix an integer P that determines the accuracy (a larger P gives higher accuracy, but also higher cost, cf. Section 9). The algorithm then proceeds in five steps as follows:

(1) *Compute the outgoing expansions on the leaves:* Loop over all leaf boxes τ. For each box, compute its outgoing expansion $\hat{\mathbf{q}}^\tau$ via

$$\hat{\mathbf{q}}^\tau = \mathbf{T}_\tau^{\text{ofs}}\, \mathbf{q}(I_\tau).$$

(2) *Compute the outgoing expansions on all parent boxes:* Loop over all parent boxes τ; proceed from finer to coarser levels so that when a box is processed, the outgoing expansions for its children are already available. The outgoing expansion $\hat{\mathbf{q}}^\tau$ is then computed via

$$\hat{\mathbf{q}}^\tau = \sum_{\sigma \in \mathcal{L}_\tau^{\text{child}}} \mathbf{T}_{\tau,\sigma}^{\text{ofo}}\, \hat{\mathbf{q}}^\sigma.$$

(3) *Convert outgoing expansions to incoming expansions:* Loop over all boxes τ. For each box, collect contributions to its incoming expansion $\hat{\mathbf{u}}^\tau$ from cells in its interaction list,

$$\hat{\mathbf{u}}^\tau = \sum_{\sigma \in \mathcal{L}_\tau^{\text{int}}} \mathbf{T}_{\tau,\sigma}^{\text{ifo}}\, \hat{\mathbf{q}}^\sigma.$$

(4) *Complete the construction of the incoming expansion for each box:* Loop over all boxes τ; proceed from coarser to finer levels so that when a box τ is processed, the incoming expansion for its parent σ is available. The incoming expansion $\hat{\mathbf{u}}^\tau$ is then constructed via

$$\hat{\mathbf{u}}^\tau = \hat{\mathbf{u}}^\tau + \mathbf{T}_{\tau,\sigma}^{\text{ifi}}\, \hat{\mathbf{u}}^\sigma.$$

(5) *Construct the potentials on all leaf boxes:* Loop over all leaf boxes τ. For each box compute the potentials at the target points by expanding the incoming expansion and adding the contributions from the near field via direct computation,

$$\mathbf{u}(I_\tau) = \mathbf{T}_\tau^{\text{tfi}}\, \hat{\mathbf{u}}^\tau + \mathbf{A}(I_\tau, I_\tau)\, \mathbf{q}(I_\tau) + \sum_{\sigma \in \mathcal{L}_\tau^{\text{nei}}} \mathbf{A}(I_\tau, I_\sigma)\, \mathbf{q}(I_\sigma).$$

Observe that the translation operators $\mathbf{T}_{\tau,\sigma}^{\text{ofo}}$, $\mathbf{T}_{\tau,\sigma}^{\text{ifo}}$, and $\mathbf{T}_{\tau,\sigma}^{\text{ifi}}$ can all be pre-computed since they depend only on P and on the vectors $\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma$. The tree structure of the boxes ensures that only a small number of values of $\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma$ are encountered.

Remark: *To describe how the FMM computes the potentials at the target points in a given leaf box τ (cf. Figure 6), we first partition the computational box: $\Omega = \Omega_\tau^{\text{near}} \cup \Omega_\tau^{\text{far}}$. Interactions with sources in the near-field $\Omega_\tau^{\text{near}} = \Omega_\tau + \bigcup_{\sigma \in \mathcal{L}_\tau^{\text{nei}}} \Omega_\sigma$ are evaluated via direct computations. To define*

the far-field, we first define the "list of ancestors" $\mathcal{L}_\tau^{\text{anc}}$ as the list holding the parent, grand-parent, great-grandparent, etc, of τ. Then $\Omega_\tau^{\text{far}} = \bigcup_{\nu \in \mathcal{L}_\tau^{\text{anc}}} \bigcup_{\sigma \in \mathcal{L}_\nu^{\text{int}}} \Omega_\sigma$. Interactions with sources in the far-field are evaluated via the outgoing expansions of the boxes in the list $\bigcup_{\nu \in \mathcal{L}_\tau^{\text{anc}}} \bigcup_{\sigma \in \mathcal{L}_\nu^{\text{int}}}$. These are the boxes marked "$\otimes$" in Figure 6.

9. **Error analysis.** The potentials computed by the FMM are not exact since all expansions have been truncated to P terms. An analysis of how such errors could propagate through the transformations across all levels is technically complicated, and should seek to estimate both the worst case error, and the statistically expected error [5]. As it happens, the global error is in most cases similar to the (worst case) local truncation error, which means that it scales roughly as α^P, where $\alpha = \sqrt{2}/(4-\sqrt{2}) = 0.5469\cdots$. As a rough estimate, we see that in order to achieve a given tolerance ε, we need to pick

$$P \approx \log(\varepsilon)/\log(\alpha).$$

As P increases, the asymptotic complexity of the 2D FMM is $O(P\,N)$ (if one enforces that each leaf node holds $O(P)$ sources). In consequence, the overall complexity can be said to scale as $\log(1/\varepsilon)\,N$ as $\varepsilon \to 0$ and $N \to \infty$.

10. **Adaptive trees for non-uniform distributions of particles.** For simplicity, the presentation in this brief article has been restricted to the case of relatively uniform particle distributions for which a a fully populated tree (as described in Section 7) is appropriate. When the particle distribution is non-uniform, locally adaptive trees perform much better. The basic FMM can readily be adapted to operate on non-uniform trees. The only modification required to the method described in Section 8 is that some outgoing expansions need to be broadcast directly to target points, and some incoming expansions must receive direct contributions from source points in certain boxes [2].

Note: In situations where the sources are distributed uniformly in a box, the FMM faces competition from techniques such as P^3M (particle-particle/particle-mesh). These are somewhat easier to implement, and can be very fast since they leverage the remarkable speed of FFT-accelerated Poisson solvers. However, the FMM has few competitors for non-uniform point distributions such as, e.g., the distributions arising from the discretization of a boundary integral equations.

11. **Extensions, accelerations, and generalizations.**

11.1. *Extension to $\mathbb{R}^3$:* In principle, the FMM described for problems in the plane can readily be extended to problems in $\mathbb{R}^3$; simply replace $\log|\boldsymbol{x}-\boldsymbol{y}|$ by $1/|\boldsymbol{x}-\boldsymbol{y}|$, replace the McLaurin expansions by expansions in spherical harmonics, and replace the quad-tree by an oct-tree. However, the resulting algorithm performs quite poorly (especially at high accuracies) for two reasons: (1) The typical number of elements in an "interaction list" grows from 27 in 2D to 189 in 3D. (2) The number of terms required in an outgoing or incoming expansion to achieve accuracy ε grows from $O(\log(1/\varepsilon))$ in 2D to $O(\log(1/\varepsilon)^2)$ in 3D. Fortunately, accelerated techniques that use more sophisticated machinery for converting outgoing to incoming expansions have been developed [11].

11.2. *The Helmholtz equation.* One of the most important applications of the FMM is the solution of scattering problems via boundary integral equations techniques. For such tasks a sum like (1) needs to be evaluated for a kernel associated with the Helmholtz equation or the closely related time-harmonic version of the Maxwell equations. When the computational domain is not large compared to the wave-length (say at most a few dozen wave-lengths), then an FMM can be constructed by simple modifications to the basic scheme described here. However, when the domain becomes large compared to the scattering wave-length the paradigm outlined here breaks

down. The problem is that the interaction ranks in this case depend on the size of the boxes involved and get prohibitively large at the higher levels of the tree. The (remarkable) fact that fast summation is possible even in the short wave-length regime was established in 1992 [18]. The new high-frequency FMM relies on similar data structures to the basic scheme described here, but the interaction mechanisms between boxes are quite different. A version of the high-frequency FMM that is stable in all regimes was described in [3]. See also [4]. It was shown in [6] that close to linear complexity can be attained while relying on rank-deficiencies alone, provided that different tessellations of the domain are implemented.

11.3. *Other interaction potentials (elasticity, Stokes, etc):* Variations of the FMM have been constructed for most of the kernels associated with the elliptic PDEs of mathematical physics such as the equations of elasticity [7], the Stokes and unsteady Stokes equations [9], the modified Helmholtz (a.k.a. Yukawa) equations [12], and many more.

11.4. *Kernel free FMMs.* While FMMs can be developed for a broad range of kernels (cf. Section 11.3), it is quite labor intense to re-derive and re-implement the various translation operators required for each special case. The so called *kernel free FMMs* [19, 8] overcome this difficulty by setting up a common framework that works for a broad range of kernels.

11.5. *Matrix operations beyond the matrix-vector product.* The FMM performs a matrix-vector multiply $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ involving certain dense $N \times N$ matrices in $O(N)$ operations. It achieves the lower complexity by exploiting rank-deficiencies in the off-diagonal blocks of the matrix $\mathbf{A}$. It turns out that such rank deficiencies can also be exploited to perform other matrix operations, such as matrix inversions, construction of Schur complements, LU factorizations, etc, in close to linear time. The so called $\mathcal{H}$-matrix methods [13] provide a general framework that can be applied in many contexts. Higher efficiency can be attained by designing direct solvers specifically for the linear systems arising upon the discretization of certain boundary integral equations [16].

12. **Practical notes and further reading.** We have provided only the briefest of introductions to the vast topic of Fast Multipole Methods. A fuller treatment can be found in numerous tutorials (e.g. [1, 15]), survey papers (e.g. [17]), and full length text books (e.g. [4, 14]).

Let us close with a practical note. While it is not that daunting of an endeavor to implement an FMM with linear or close to linear asymptotic scaling, it is another matter entirely to write a code that actually achieves high *practical* performance — especially in a scattering environment. This would be an argument against using FMMs were it not for the fact that the algorithms are very well suited for black box implementation. Some such codes are available publicly, and more are expected to become available in the next several years. Before developing a new code from scratch, it is usually worth-while to first look to see if a high-quality code may already be available.

References

[1] L. Beatson and L. Greengard, *A short course on fast multipole methods*, 1997.

[2] J Carrier, L Greengard, and V Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. SCI. STAT. COMPUT. **9** (1988), no. 4, 669–686.

[3] Hongwei Cheng, William Y. Crutchfield, Zydrunas Gimbutas, Leslie F. Greengard, J. Frank Ethridge, Jingfang Huang, Vladimir Rokhlin, Norman Yarvin, and Junsheng Zhao, *A wideband fast multipole method for the Helmholtz equation in three dimensions*, J. Comput. Phys. **216** (2006), no. 1, 300–325.

[4] W.C. Chew, J.-M. Jin, E. Michielssen, and J. Song, *Fast and efficient algorithms in computational electromagnetics*, Artech House, 2001.

[5] E. Darve, *The fast multipole method i: Error analysis and asymptotic complexity*, SIAM Journal on Numerical Analysis **38** (2000), no. 1, 98–128.

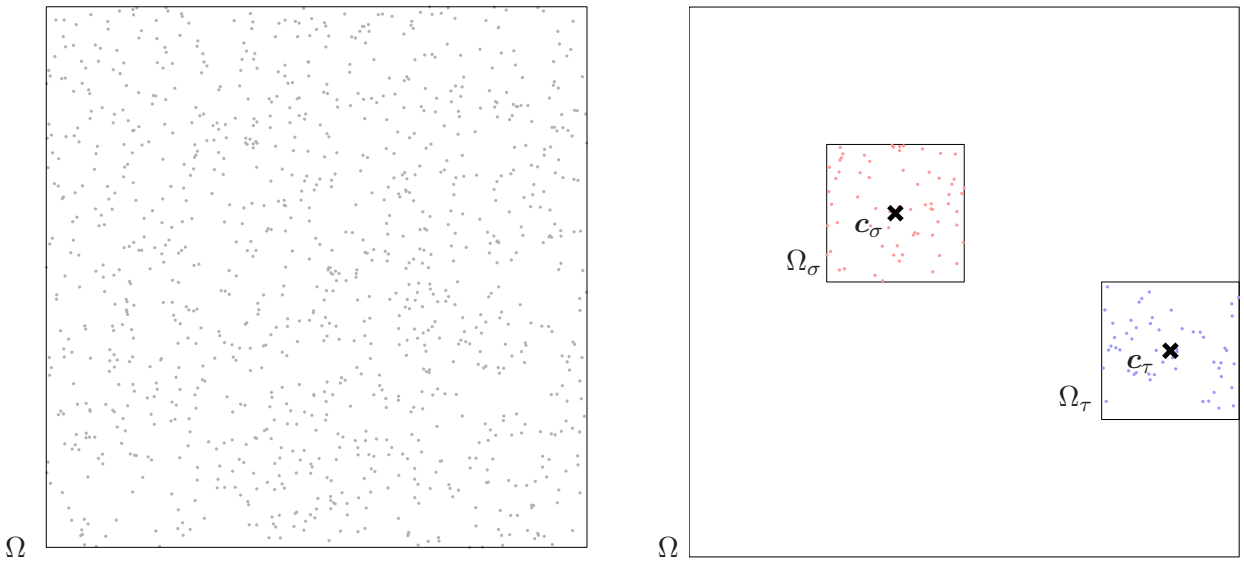


FIGURE 1. (a) Geometry of the full N-body problem. The domain Ω is drawn in black and the points $\boldsymbol{x}_i$ are gray. (b) The geometry described in Sections 3 and 4. The box Ω_σ contains source locations (red) and Ω_τ contains target locations (blue).

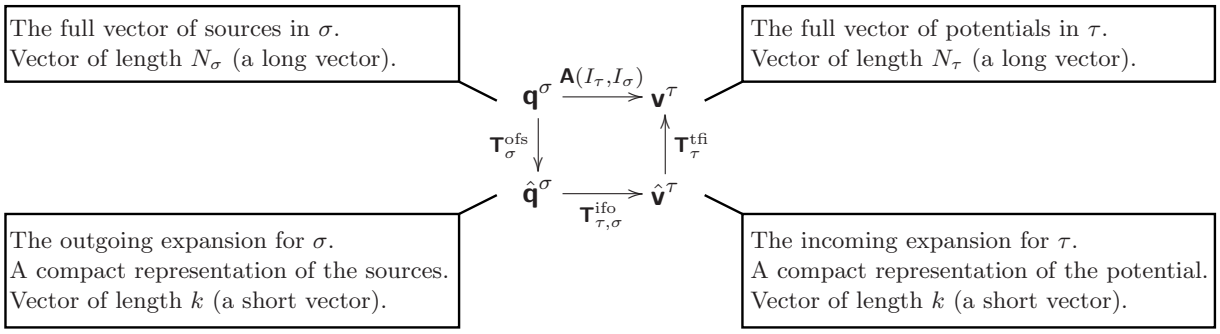


FIGURE 2. The outgoing and incoming expansions introduced in Section 4 are compact representations of the sources and potentials in the source and target boxes, respectively. The diagram commutes to high precision since $\mathbf{A}(I_\tau, I_\sigma) \approx \mathbf{T}_\tau^{\text{tfi}} \mathbf{T}_{\tau,\sigma}^{\text{ifo}} \mathbf{T}_\sigma^{\text{ofs}}$, cf. (14).

[6] B. Engquist and L. Ying, *A fast directional algorithm for high frequency acoustic scattering in two dimensions*, Commun. Math. Sci. **7** (2009), no. 2, 327–345.

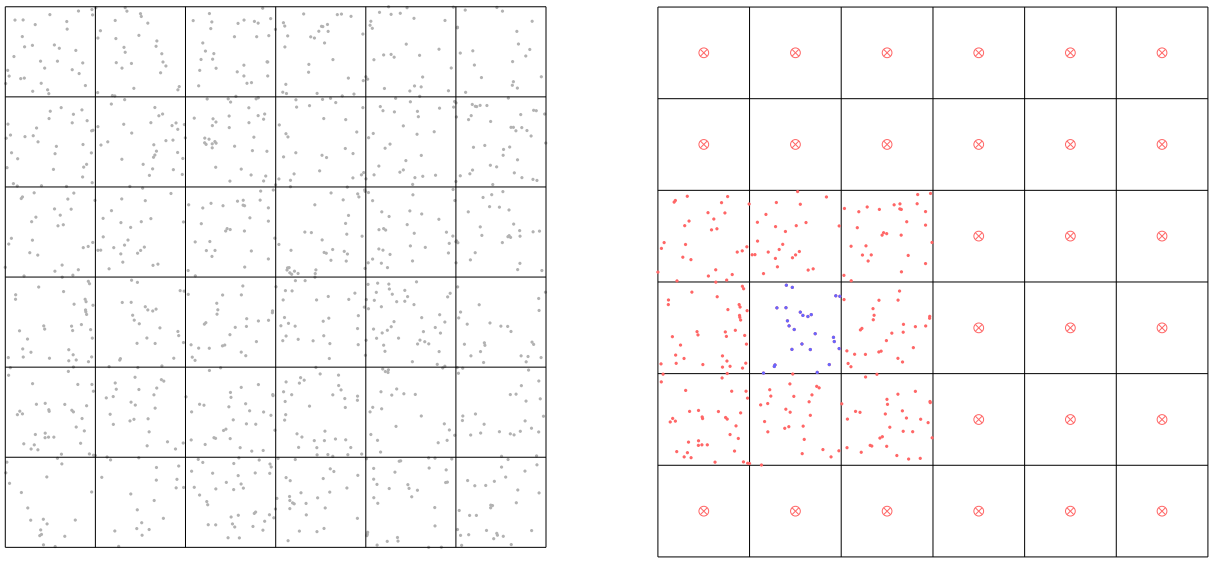
[7] Y. Fu, K. J. Klimkowski, G. J. Rodin, E. Berger, J. C. Browne, J. K. Singer, R. A. Van De Geijn, and K. S. Vemaganti, *A fast solution method for three-dimensional many-particle problems of linear elasticity*, International Journal for Numerical Methods in Engineering **42** (1998), no. 7, 1215–1229.

[8] Zydrunas Gimbutas and Vladimir Rokhlin, *A generalized fast multipole method for nonoscillatory kernels*, SIAM J. Sci. Comput. **24** (2002), no. 3, 796–817 (electronic). MR MR1950512 (2004a:65176)

[9] L. Greengard and M. C. Kropinski, *An integral equation approach to the incompressible navier-stokes equations in two dimensions*, SIAM J. Sci. Comput. **20** (1998), no. 1, 318–336.

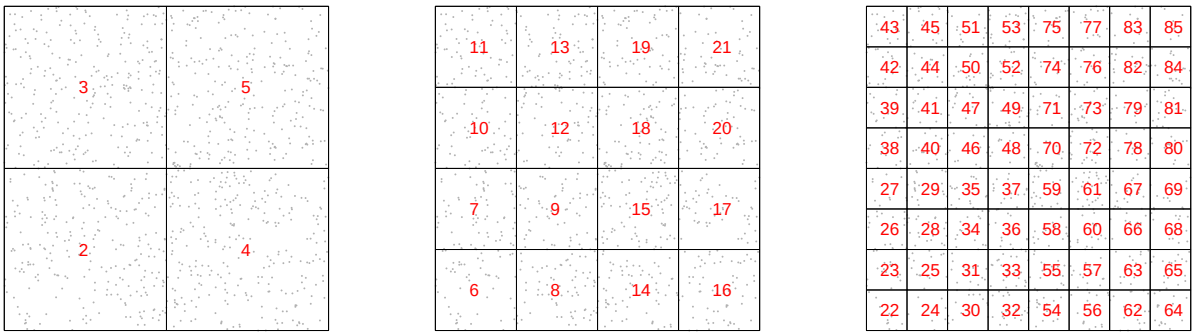
[10] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys. **73** (1987), no. 2, 325–348.

[11] ———, *A new version of the fast multipole method for the Laplace equation in three dimensions*, Acta numerica, 1997, Acta Numer., vol. 6, Cambridge Univ. Press, Cambridge, 1997, pp. 229–269.



(a) (b)

FIGURE 3. (a) A tessellation of Ω into $m \times m$ smaller boxes, cf. Section 5. (b) Evaluation of the potential in a box τ. The target points in τ are marked with blue dots, the source points in the neighbor boxes in $\mathcal{L}_\tau^{\text{nei}}$ are marked with red dots, and the centers of the outgoing expansions in the far-field boxes $\mathcal{L}_\tau^{\text{far}}$ are marked $\otimes$.



Level 1 *Level 2* *Level 3*

FIGURE 4. A tree of boxes on Ω with $L = 3$ levels. The enumeration of boxes shown is simply one of many possible ones.

[12] L. Greengard and X. Sun, *A new version of the fast gauss transform*, Doc. Math. Extra volume ICM 1998 **III** (1998), 575 – 584.

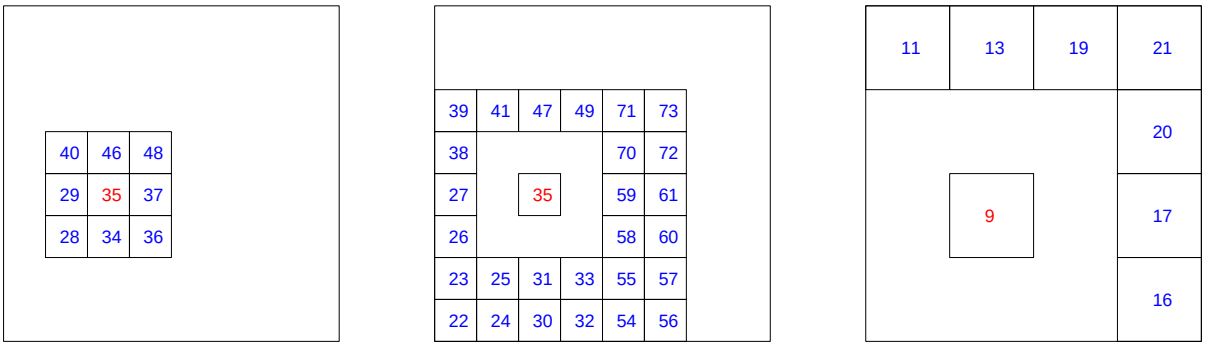
[13] Wolfgang Hackbusch, *A sparse matrix arithmetic based on H-matrices; Part I: Introduction to H-matrices*, Computing **62** (1999), 89–108.

[14] Y. J. Liu, *Fast multipole boundary element method: Theory and applications in engineering*, Cambridge University Press, 2009.

[15] Y.J. Liu and N. Nishimura, *The fast multipole boundary element method for potential problems: A tutorial*, Engineering Analysis with Boundary Elements **30** (2006), no. 5, 371 – 381.

[16] P.G. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, J. Comp. Phys. **205** (2005), no. 1, 1–23.

[17] N. Nishimura, *Fast multipole accelerated boundary integral equation methods*, Applied Mechanics Reviews **55** (2002), no. 4, 299 – 324.



Neighbor list of $\tau = 35$ *Interaction list of* $\tau = 35$ *Interaction list of* $\tau = 9$

FIGURE 5. Illustration of some index vectors called "lists" that were introduced in Section 7. For instance, the left-most figure illustrates that $\mathcal{L}_{35}^{\text{nei}} = \{28, 29, 34, 36, 37, 40, 46, 48\}$. (Boxes are numbered as in Figure 4.)

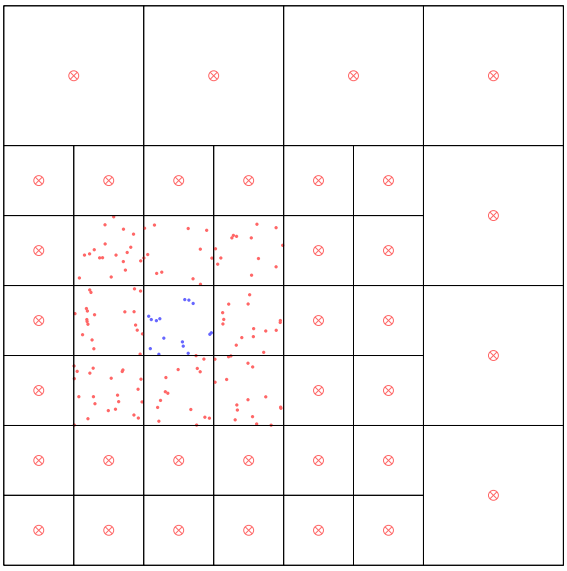


FIGURE 6. Illustration of how the FMM evaluates the potentials in a leaf box τ marked by its blue target points. Contributions to the potential caused by sources in τ itself (blue dots), or in its immediate neighbors (red dots) are computed via direct evaluation. The contributions from more distant sources are computed via the outgoing expansions centered on the $\otimes$ marks in the figure.

[18] V. Rokhlin, *Diagonal forms of translation operators for the helmholtz equation in three dimensions*, Applied and Computational Harmonic Analysis **1** (1993), no. 1, 82 – 93.

[19] L. Ying, G. Biros, and D. Zorin, *A kernel-independent adaptive fast multipole algorithm in two and three dimensions*, J. Comput. Phys. **196** (2004), no. 2, 591–626.