# JUST ANOTHER IMPLEMENTATION OF THE FAST MULTIPOLE METHOD FOR THE ELECTROSTATIC POTENTIAL IN 2D

RODRIGO ARRIETA*

**Abstract.** In this work we describe the Fast Multipole Method (FMM), invented by L. Greengard and V. Rokhlin in 1987 [3] to compute the potential interaction of $N$ particles in $\mathcal{O}(N)$ time, in contrast with the $\mathcal{O}(N^2)$ time that a direct approach would take. We consider a very particular case, where the potential is given by the electrostatic potential in two dimension and the distribution of points is taken as random uniformly distributed. This work follows closely the description of the FMM given by Martinsson [4].

The FMM was coded from scratch in a Julia package [1] and its performances was compared against the theoretical predictions. Our code manages to achieve the theoretical error scaling with respect to the number of particles $N$ and the interaction rank $P$. It also achieves the theoretical time scaling with respect to $N$, but performs better-than-expected when the time is considered with respect to $P$. Our hypothesis for this phenomenon is that the values of $P$ considered are so low that compiler or low-level routines optimizations does not allow us to see the correct scaling.

**1. The multipole expansion for well separated clusters of points.** Consider the electrostatic potential $\{u_i\}_{i=1}^M \subset \mathbb{R}$ at some target points $\{\boldsymbol{x}\}_{i=1}^M \subset \mathbb{R}^2$ generated by a set of point charges $\{q_j\}_{j=1}^N \subset R$ located at source points $\{\boldsymbol{y}_j\}_{j=1}^N$. Mathematically, this is expressed as

$$u(\boldsymbol{x}_i) := u_i = \sum_{j=1}^{N} G(\boldsymbol{x}_i, \boldsymbol{y}_j) q_j, \quad i = 1, 2, \ldots, M, \tag{1.1}$$

where the kernel $G$ is taken as a scaled version of the fundamental solution of the Laplace equation in two dimensions,

$$G(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \log |\boldsymbol{x} - \boldsymbol{y}|, & \text{if } \boldsymbol{x} \neq \boldsymbol{y}, \\ 0, & \text{if } \boldsymbol{x} = \boldsymbol{y}. \end{cases} \tag{1.2}$$

Clearly, a direct evaluation of (1.1) would take $\mathcal{O}(NM)$ operations, which is prohibitively expensive when we have a large number of source and target points $N, M >> 1$. In what follows, we will describe how to compute an arbitrarily close approximation to this sum in a much faster way, subject to the constraint that source points are far away from target points.

Suppose that the source points are clustered in a box $\Omega_\sigma$, with center $\boldsymbol{c}_\sigma$, whereas the target points are clustered in a disjoint box $\Omega_\tau$ of the same size, with center $\boldsymbol{c}_\tau$, as shown in Fig. 1. The separation between boxes is $d := |\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma|$.

For simplicity, let us switch to complex notation, where each point $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^2$ is reinterpreted as a point in the complex plane $\mathbb{C}$, and with abuse of notation we will write the complex kernel $G(\boldsymbol{x}, \boldsymbol{y}) = \log(\boldsymbol{x} - \boldsymbol{y})$, where log is the principal branch of the complex logarithm. The real kernel in (1.2) is retrieved by taking $\operatorname{Re}\{G(\boldsymbol{x}, \boldsymbol{y})\}$.

*Mathematics Department, Massachusetts Institute of Technology, Cambridge, MA 02139, United States (rarrieta@mit.edu).

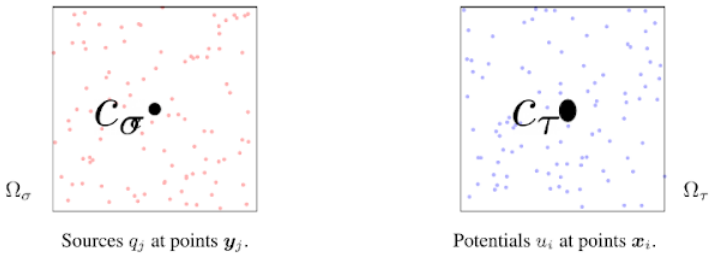



FIGURE 1. *The problem of computing the electrostatic potential between well-separated source and target boxes.*

We can now formally expand the potential $u_i$ in a Taylor expansion,

$$
\begin{aligned}
u_i &= \sum_{j=1}^{N} \log(\boldsymbol{x}_i - \boldsymbol{y}_j) q_j \\
&= \sum_{j=1}^{N} \log((\boldsymbol{x}_i - \boldsymbol{c}_\sigma) - (\boldsymbol{y}_j - \boldsymbol{c}_\sigma)) q_j \\
&= \sum_{j=1}^{N} \left[ \log(\boldsymbol{x}_i - \boldsymbol{c}_\sigma) + \log\left(1 - \frac{\boldsymbol{y}_j - \boldsymbol{c}_\sigma}{\boldsymbol{x}_i - \boldsymbol{c}_\sigma}\right) \right] q_j \\
&= \sum_{j=1}^{N} \left[ \log(\boldsymbol{x}_i - \boldsymbol{c}_\sigma) - \sum_{p=1}^{\infty} \frac{1}{p} \frac{(\boldsymbol{y}_j - \boldsymbol{c}_\sigma)^p}{(\boldsymbol{x}_i - \boldsymbol{c}_\sigma)^p} \right] q_j \\
&\approx \log(\boldsymbol{x}_i - \boldsymbol{c}_\sigma) \hat{q}_0^\sigma + \sum_{p=1}^{P} \frac{1}{p} \frac{1}{(\boldsymbol{x}_i - \boldsymbol{c}_\sigma)^p} \hat{q}_p^\sigma,
\end{aligned}
\tag{1.3}
$$

where we used the fact that $\log(1-z) = -\sum_p z^p/p$ for $|z| < 1$, the distance $d$ is chosen such that $\left|\frac{\boldsymbol{y}_j - \boldsymbol{c}_\sigma}{\boldsymbol{x}_i - \boldsymbol{c}_\sigma}\right| < 1$ for every pair of target/source points, and we have approximated the infinite sum by $P$ terms, where $P$ is known as the *interaction rank*. The quantities $\hat{\boldsymbol{q}}^\sigma := \{\hat{q}_p^\sigma\}_{p=0}^{P-1}$ given by

$$
\begin{aligned}
\hat{q}_0^\sigma &:= \sum_{j=1}^{N} q_j, \\
\hat{q}_p^\sigma &:= \sum_{j=1}^{N} \frac{-1}{p} (\boldsymbol{y}_j - \boldsymbol{c}_\sigma)^p q_j, \quad p = 1, 2, \ldots, P-1,
\end{aligned}
\tag{1.4}
$$

are the *outgoing expansions* of the source box $\sigma$. On the other hand, the potential $u(x)$ is analytic, thus it can be expanded as a (truncated) Taylor series around the target box center $\boldsymbol{c}_\tau$,

$$
u(\boldsymbol{x}_i) = u_i \approx \sum_{p=0}^{P-1} (\boldsymbol{x}_i - \boldsymbol{c}_\tau)^p \hat{\boldsymbol{v}}_p^\tau,
\tag{1.5}
$$

where $\hat{\boldsymbol{v}}^\tau := \{\hat{v}_p^\tau\}_{p=0}^{P-1}$ are the *incoming expansions* of the target box $\tau$. By expanding (1.3) in Taylor, and again choosing the distance $d$ large enough, it can be deduced

that

(1.6)

$$\hat{v}_0^\tau = \log(\boldsymbol{c}_\tau - \boldsymbol{c}_\sigma)\hat{q}_0^\sigma + \sum_{p=1}^{P-1}(-1)^p \frac{1}{(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^p}\hat{q}_p^\sigma,$$

$$\hat{v}_r^\tau = -\frac{1}{r(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^r}\hat{q}_0^\sigma + \sum_{p=1}^{P-1}(-1)^p \binom{r+p-1}{p-1} \frac{1}{(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{r+p}}\hat{q}_p^\sigma, \quad r = 1, 2, \ldots, P-1.$$

Denote $N_\tau$ and $N_\sigma$ the numbers of target and source points, respectively, $\boldsymbol{q}^\sigma := \{q_j\}_{j=1}^{N_\sigma}$ and $\boldsymbol{u}^\tau := \{u_i\}_{i=1}^{N_\tau}$. Note that we have implicitly defined the linear maps

$$\hat{\boldsymbol{q}}^\sigma = \mathbf{T}_\sigma^{\text{ofs}} \boldsymbol{q}^\sigma, \tag{1.7}$$

$$\hat{\boldsymbol{v}}^\tau = \mathbf{T}_{\tau,\sigma}^{\text{ifo}} \hat{\boldsymbol{q}}^\sigma \quad \text{and} \tag{1.8}$$

$$\boldsymbol{u}^\tau = \mathbf{T}_\tau^{\text{tfi}} \hat{\boldsymbol{v}}^\tau, \tag{1.9}$$

where $\mathbf{T}_\sigma^{\text{ofs}}$ is the $P \times N_\sigma$ *outgoing-from-sources* translation operator defined by (1.4), $\mathbf{T}_\sigma^{\text{ifo}}$ is the $P \times P$ *incoming-from-outgoing* translation operator defined by (1.6) and $\mathbf{T}_\sigma^{\text{tfi}}$ is the $N_\tau \times P$ *target-from-incoming* translation operator defined by (1.5). With this, we have effectively computed a *rank-P* approximation of the potential

$$\boldsymbol{u}^\tau \approx \mathbf{T}_\tau^{\text{tfi}} \left( \mathbf{T}_{\tau,\sigma}^{\text{ifo}} \left( \mathbf{T}_\sigma^{\text{ofs}} \boldsymbol{q}^\sigma \right) \right), \tag{1.10}$$

which takes $\mathcal{O}\left(P(N_\sigma + N_\tau) + P^2\right)$ operations, much less than the $\mathcal{O}\left(N_\sigma N_\tau\right)$ operations of the naive method for (1.1).

Let us introduce the concept of *well separated* boxes, which gives us an estimate of the error committed in (1.10). We will say that a box $\Omega_\tau$ of center $\boldsymbol{c}_\tau$ is *well separated* from a box $\Omega_\sigma$ of center $\boldsymbol{c}_\sigma$ and side length $2a$ if

$$\|\boldsymbol{c}_\tau - \boldsymbol{c}_\tau\| \geq 4a. \tag{1.11}$$

Now, for *well separated* source and target boxes of the same size, $\Omega_\sigma$ and $\Omega_\tau$, the error committed in (1.10) using a *rank-P* approximate potential $\boldsymbol{u}_P^\tau$ is given by [2]

$$\|\boldsymbol{u}^\tau - \boldsymbol{u}_P^\tau\|_\infty \leq \frac{\eta^P}{P(1-\eta)} \|\boldsymbol{q}^\sigma\|_1, \tag{1.12}$$

where $\eta = \sqrt{2}/(4-\sqrt{2}) \approx 0.547$. This bound can be derived by considering the remainders of the Taylor expansions of (1.5) and (1.3), and the definition of *well separated* boxes. It follows that the error decays exponentially with respect to the interaction rank $P$, and we can pick $P = \mathcal{O}(\log(1/\epsilon))$ to achieve an error of $\epsilon$.

**2. The Fast Multipole Method for an uniform cluster of charges.** Consider an uniform cluster of $N$ points $\boldsymbol{x}_i$ with associated charges $q_i$, $i = 1, \ldots, N$, as shown in Figure 2. We wish to compute the electrostatic potential at every point $\boldsymbol{x}_i$, this is

$$u(\boldsymbol{x}_i) := u_i = \sum_{j=1}^{N} G(\boldsymbol{x}_i, \boldsymbol{x}_j) q_j, \quad i = 1, 2, \ldots, N. \tag{2.1}$$

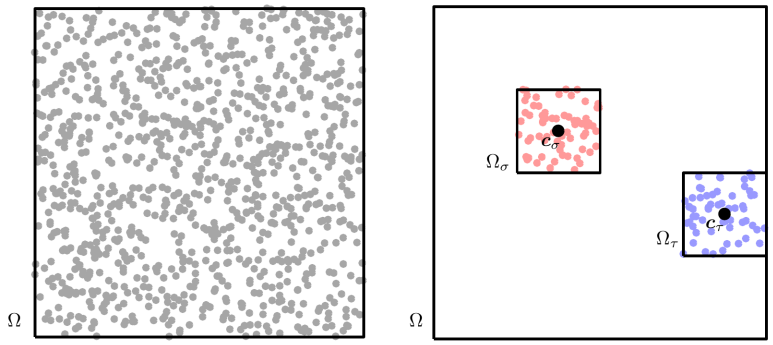



FIGURE 2. *Left: uniform cluster of charges. Right: for well separated boxes inside the domain, we could resort to the fast multipole approach of section 1.*

As before, computing the potentials using the naive formula (2.1) takes $\mathcal{O}(N^2)$ operations. Here, we will describe the Fast Multipole Method (FMM), which allows us to compute all potentials with only $\mathcal{O}(N)$, using the multipole technique presented in section 1.

In the FMM we construct a *quadtree* of boxes, a hierarchical tree data structure, where we start with a *root* box which is divided into four smaller and identical *children* boxes. The children constitute the first level of the tree. To construct the second level, each child is again subdivided into four smaller and identical children. This process gets repeated until we reach a prescribed number of levels $L+1$, where level 0 corresponds only to the root box, whereas level $L$ contains the finer boxes, the *leaves*. An example of a quadtree structure with $L = 3$ is shown in Figure 3.

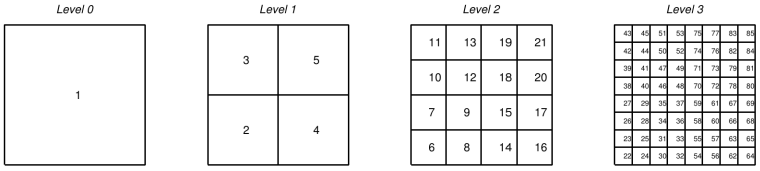



FIGURE 3. *Quadtree of boxes with $L+1$ levels, $L = 3$.*

The idea of this hierarchical data structure is the following. Leaf boxes compute the potentials at their points using multipole expansions for *well separated* boxes and direct evaluations for neighboring boxes and self-interactions. For *closer* well separated boxes we use the multipole expansion shown in section 1. This set of *closer* well separated boxes is known as the *interaction list* $\mathcal{L}_\sigma^{\text{int}}$ of the leaf box $\sigma$, we will provide a precise definition below. Instead, for farther away boxes, it is much more efficient that the leaves' parents compute the multipole interactions (*outgoing* or *incoming* expansions) for the parent's *interaction list*, and then they pass the interaction to their children. Again, if two *well separated* parent boxes are very far away (meaning that they do not belong to each others *interaction list*) then it is more efficient that the parents' parents compute the interaction and so on. This hierarchical transfer of information is the secret behind FMM which allows to achieve a $\mathcal{O}(N)$ complexity. In the end, boxes only care about boxes in their interaction list. If they are farther away, then their ancestry (parents, grandparents, etc.) compute

the interaction. And for leaf boxes, they have to care about their interaction list, neighbors and self interactions to finally compute the potentials at their points.

Before describing the FMM algorithm, we will explain how children/parents boxes transfer their *outgoing/incoming expansions* to each other. First, a parent box $\tau$ needs to compute its *outgoing expansion* $\hat{\boldsymbol{q}}^\tau$. In fact, this can be done exactly (without any approximation) using the *outgoing expansions* of their children, as shown next. Fix a child box $\sigma$ of $\tau$. Denote $I_\sigma = \{i : \boldsymbol{x}_i \in \Omega_\sigma\}$. Its outgoing expansion of order $p$ is

$$\hat{q}_p^\sigma = \sum_{j \in I_\sigma} \frac{-1}{p} (\boldsymbol{x}_j - \boldsymbol{c}_\sigma)^p q_j, \tag{2.2}$$

where $p = 1, 2, \ldots, P-1$. On the other hand, we can write the order $k = 1, 2, \ldots, P-1$ *outgoing expansion* of $\tau$ given by the charges at $\Omega_\sigma$ as

$$\hat{q}_k^{\tau,\sigma} = \sum_{j \in I_\sigma} \frac{-1}{k} (\boldsymbol{y}_j - \boldsymbol{c}_\tau)^k q_j \tag{2.3}$$

$$= \sum_{j \in I_\sigma} \frac{-1}{k} ((\boldsymbol{y}_j - \boldsymbol{c}_\sigma) - (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau))^k q_j \tag{2.4}$$

$$= \sum_{j \in I_\sigma} \frac{-1}{k} \sum_{p=0}^{k} \binom{k}{p} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{k-p} (\boldsymbol{y}_j - \boldsymbol{c}_\sigma)^p q_j \tag{2.5}$$

$$= \frac{-1}{k} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^k \hat{q}_0^\sigma + \sum_{p=1}^{k} \frac{-1}{k} \binom{k}{p} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{k-p} \sum_{j \in I_\sigma} (\boldsymbol{y}_j - \boldsymbol{c}_\sigma)^p \tag{2.6}$$

$$= \frac{-1}{k} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^k \hat{q}_0^\sigma + \sum_{p=1}^{k} \frac{p}{k} \binom{k}{p} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{k-p} \sum_{j \in I_\sigma} \frac{-1}{p} (\boldsymbol{y}_j - \boldsymbol{c}_\sigma)^p \tag{2.7}$$

$$= \frac{-1}{k} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^k \hat{q}_0^\sigma + \sum_{p=1}^{k} \frac{p}{k} \binom{k}{p} (\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{k-p} \hat{q}_p^\sigma. \tag{2.8}$$

This defines a linear operator $\mathbf{T}^{\text{ofo}}_{\tau,\sigma}$, known as the $P \times P$ *outgoing-from-outgoing* translation operator, which maps the *outgoing expansion* $\hat{\boldsymbol{q}}^\sigma$ of the child $\sigma$ into the parent $\tau$ expansion $\hat{\boldsymbol{q}}^\tau$. This definition of $\mathbf{T}^{\text{ofo}}_{\tau,\sigma}$ is slightly different to the one shown in [4]. Finally, the *total outgoing expansion* of $\tau$ is given by the sum of the contributions of their children, this is

$$\hat{\boldsymbol{q}}^\tau = \sum_{\sigma \in \mathcal{L}^{\text{child}}_\tau} \mathbf{T}^{\text{ofo}}_{\tau,\sigma} \hat{\boldsymbol{q}}^\sigma, \tag{2.9}$$

where $\mathcal{L}^{\text{child}}_\tau$ is the set of children of $\tau$.

Once a parent $\tau$ computes its *incoming expansion* $\hat{\boldsymbol{v}}^\tau$ using the *outgoing expansions* of the boxes in its interaction list, it needs to transfer this *incoming expansion* to each of its four children. Fix a children $\sigma$ of $\tau$. Its *incoming expansion* is given by

$$\hat{\boldsymbol{v}}^\sigma = \mathbf{T}^{\text{ifi}}_{\tau,\sigma} \hat{\boldsymbol{v}}^\tau + \sum_{\nu \in \mathcal{L}^{\text{int}}_\sigma} \mathbf{T}^{\text{ifo}}_{\sigma,\nu} \hat{\boldsymbol{q}}^\nu, \tag{2.10}$$

where the first term is the contribution from the parent (this includes the contribution of boxes farther away than the interaction list $\mathcal{L}^{\text{int}}_\sigma$ of $\sigma$), and the second term correspond to the contribution of the interaction list $\mathcal{L}^{\text{int}}_\sigma$ of $\sigma$. The linear operator $\mathbf{T}^{\text{ifi}}_{\sigma,\tau}$

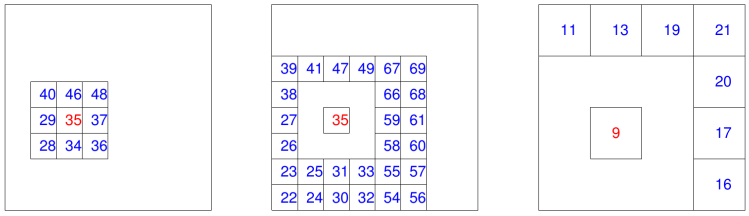
is the $P \times P$ *incoming-from-incoming* translation operator, which maps the *incoming expansion* $\hat{\boldsymbol{v}}^\tau$ of the parent $\tau$ into the child $\sigma$ expansion $\hat{\boldsymbol{v}}^\sigma$. Using the expression (1.5) and the same binomial expansion tricks as used in (2.8), it can be derived that

$$\mathbf{T}^{\text{ifi}}_{\sigma,\tau} = \begin{cases} \binom{p}{r}(\boldsymbol{c}_\sigma - \boldsymbol{c}_\tau)^{p-r}, & \text{for } r \le p, \\ 0, & \text{for } r > p, \end{cases} \tag{2.11}$$

for $p, r = 0, 1, \ldots, P-1$.

For a box $\tau$ let us define the following:

- The parent of $\tau$ is the box on the next coarsest level that contains $\tau$.
- The children list of $\tau$ is the set $\mathcal{L}^{\text{child}}_\tau$ of boxes whose parent is $\tau$.
- The neighbor list of $\tau$ is the set $\mathcal{L}^{\text{nei}}_\tau$ of boxes on the same level that directly touch $\tau$.
- The interaction list of $\tau$ is the set $\mathcal{L}^{\text{int}}_\tau$ of all boxes $\sigma$ such that (1) $\sigma$ and $\tau$ are on the same level, (2) $\sigma$ and $\tau$ do not touch, and (3) the parents of $\sigma$ and $\tau$ do touch.
- Denote $A(I_\tau, I_\sigma)$ as the (dense) operator that maps the charges $\boldsymbol{q}^\sigma$ at box $\sigma$ into the potentials $\boldsymbol{u}^\tau$ at box $\tau$, i.e., $\boldsymbol{u}^\tau = A(I_\tau, I_\sigma)\boldsymbol{q}^\sigma$.




For $\tau = 35$ (red), the neighbor list $\mathcal{L}^{\text{nei}}_\tau$ is shown in blue.

For $\tau = 35$ (red), the interaction list $\mathcal{L}^{\text{int}}_\tau$ is shown in blue.

For $\tau = 9$ (red), the interaction list $\mathcal{L}^{\text{int}}_\tau$ is shown in blue.

FIGURE 4. *Examples of interaction and neighbor lists.*

A diagram that shows an example of interaction and neighbor lists is presented in Figure 4. Now we have all the tools to describe the FMM algorithm. In the precomputation step we first assemble the quadtree, the boxes lists and precompute all of the translation operators. Now, to compute the potentials $\boldsymbol{u}$ for a given set of charges $\boldsymbol{q}$ we follow three steps:

1. Upward pass: compute the *outgoing expansions* of all boxes. Start at the finest level computing the expansions directly from the charges. For parent boxes, compute their expansions using their children's expansions.
2. Downward pass: compute the *incoming expansions* of all boxes. From level $l = 2$ to the finest, compute the *incoming expansions* using the parent's contribution and the interaction list's contribution.
3. Compute potentials: Each leaf computes the potentials at its points, using the incoming expansions and direct evaluations of the potentials for neighboring boxes and self-interactions.

The FMM algorithm is detailed in Figure 5, Figure 6 and Figure 7.

**3. Error estimation.** The error for FMM type of methods is usually estimated using a single translation operation, which includes the processes that maps charges into *outgoing expansions*, into *incoming expansions* and finally into potentials. As shown in section 1, this error $\epsilon$ decays exponentially fast with respect to the interaction

**loop** over levels $\ell$, from the finest to the coarsest
  **loop** over all boxes $\tau$ on level $\ell$
    **if** ($\tau$ is a leaf)
      $\hat{\mathbf{q}}_\tau = \mathbf{T}^{\mathrm{ofs}}_\tau \, \mathbf{q}(I_\tau)$
    **else**
      $\hat{\mathbf{q}}_\tau = \sum_{\sigma \in \mathcal{L}^{\mathrm{child}}_\tau} \mathbf{T}^{\mathrm{ofo}}_{\tau,\sigma} \, \hat{\mathbf{q}}_\sigma$
    **end if**
  **end loop**
**end loop**

FIGURE 5. *FMM upward pass.*

Set $\hat{\mathbf{u}}_\tau = \mathbf{0}$ for every box $\tau$ on level 1.
**loop** over levels $\ell$, from level $\ell = 2$ to the finest
  **loop** over all boxes $\tau$ on level $\ell$
    Let $\nu$ denote the parent of $\tau$.
    $\hat{\mathbf{u}}_\tau = \mathbf{T}^{\mathrm{ifi}}_{\tau,\nu} \, \hat{\mathbf{u}}_\nu + \sum_{\sigma \in \mathcal{L}^{\mathrm{int}}_\tau} \mathbf{T}^{\mathrm{ifo}}_{\tau,\sigma} \, \hat{\mathbf{q}}_\sigma$.
  **end loop**
**end loop**

FIGURE 6. *FMM downward pass.*

**loop** over every leaf box $\tau$
  $\mathbf{u}(I_\tau) = \mathbf{A}^{\mathrm{tfi}}_\tau \, \hat{\mathbf{u}}_\tau + \mathbf{A}(I_\tau, I_\tau) \, \mathbf{q}(I_\tau) + \sum_{\sigma \in \mathcal{L}^{\mathrm{nei}}_\tau} \mathbf{A}(I_\tau, I_\sigma) \, \mathbf{q}(I_\sigma)$
**end loop**

FIGURE 7. *FMM compute potentials step.*

rank $P$, i.e., $P = \mathcal{O}(\log(1/\epsilon))$. A more careful analysis and a sharper bound is given in [5].

**4. Cost estimation.** Let $N$ be the total number of charges, $L$ the number of levels ($L + 1$ levels in total) and $b$ the average number of charges in a leaf box ($b \approx N/4^L$, where $4^L$ is the number of leaf boxes, recall that the distribution of charges is uniform). Let us compute the cost of applying each of the translation operators:

- The *outgoing-from-sources* $\mathbf{T}^{\mathrm{ofs}}_\sigma$ operator is of size $P \times b$ and is applied once for every leaf box. There are $N/b$ leaf boxes, so $\mathbf{T}^{\mathrm{ofs}}_\sigma \sim Pb \times N/b = PN$.
- The *outgoing-from-outgoing* $\mathbf{T}^{\mathrm{ofo}}_{\tau,\sigma}$ operator is of size $P \times P$ and is applied once for every box. The total number of boxes is bounded by twice the number of leaf boxes, hence $\mathbf{T}^{\mathrm{ofo}}_{\tau,\sigma} \sim P^2 \times 2N/b \sim P^2 N/b$.
- The *outgoing-from-incoming* $\mathbf{T}^{\mathrm{ofi}}_{\tau,\sigma}$ operator and the *incoming-from-incoming* $\mathbf{T}^{\mathrm{ifi}}_{\sigma,\tau}$ operator follow a similar bound as the *outgoing-from-outgoing* $\mathbf{T}^{\mathrm{ofo}}_{\tau,\sigma}$ operator, therefore $\mathbf{T}^{\mathrm{ifo}}_{\tau,\sigma} \sim \mathbf{T}^{\mathrm{ifi}}_{\tau,\sigma} \sim \mathbf{T}^{\mathrm{ofo}}_{\tau,\sigma} \sim P^2 N/b$.
- The cost of the *target-from-incoming* $\mathbf{T}^{\mathrm{tfi}}_\tau$ operator is the same as the *outgoing-from-outgoing* $\mathbf{T}^{\mathrm{ofo}}_{\tau,\sigma}$ operator, so $\mathbf{T}^{\mathrm{tfi}}_\tau \sim \mathbf{T}^{\mathrm{ofo}}_{\tau,\sigma} \sim P^2 N/b$.
- For the near interactions, i.e., neighbor and self-interactions, each particle has to compute at most $9b$ interactions, since there are at most 9 neighboring boxes. Thus, $\mathbf{T}^{\mathrm{near}} \sim Nb$.

Adding up all costs, we end up with a total cost of $\mathbf{T}^{\mathrm{FMM}} \sim P^2 N/b + PN + Nb$. Since we are free to choose $b$ by tuning the numbers of levels $L$, we can choose $b \sim P$

so that the cost is minimized,

$$\mathbf{T}^{\mathrm{FMM}} \sim PN \sim N \log(1/\epsilon), \tag{4.1}$$

where we used the fact $P \sim \log(1/\epsilon)$, with $\epsilon$ being the error.

**5. Numerical results.** The FMM algorithm described in section 2 was coded in a Julia package [1] from scratch.

In our first numerical experiment we considered $N$ uniformly distributed particles in $[0,1]^2$ with associated normally distributed charges $\boldsymbol{q}$ (the charges were normalized, so that $\|\boldsymbol{q}\|_1 = 1$). We computed the whole potential using the FMM and compared against the exact potential (2.1) for 256 randomly chosen particles. The number of particles was taken between $N = 2^{12}$ and $N = 2^{22}$, the interaction rank was fixed at $P = 5$ and the number of levels $L$ was chosen so that the average number of particles per leaf box was $b = 128$. In Figure 8 we show the time taken by both the FMM and the direct evaluation method to compute the full potential. For the direct evaluation method we estimated the total time using the time taken in computing the potential for the 256 particles. It is observed that the direct method scales as the expected $\mathcal{O}(N^2)$, whereas our FMM implementation achieves the $\mathcal{O}(N)$ scaling predicted in (4.1). For $N = 2^{22}$, while the direct method would take more than 100 days in computing the full potential, our FMM implementation only takes a bit more than 100 seconds. In Figure 9 we show the relative error (measured with respect to $\|\cdot\|_\infty$) committed by the FMM at the 256 randomly chosen particles. It is observed that the error remains stable and below 0.1% for all values of $N$.

Our second numerical experiment consider a similar setting as before, but with $N = 2^{18}$, the interaction rank varying from $P = 2$ to $P = 20$, and $b = 64P$. The relative error committed by the FMM as a function of $P$ is shown in Figure 10. It is observed that the error decays exponentially with respect to $P$, as pointed out in section 3. The time to compute the full potential as a function of $P$ is shown in Figure 11. It is observed that the time remains constant for most values of $P$, but then goes up for $P = 18$ and $P = 20$. This behavior does not agree with (4.1), which assures that the time scaling should be $\mathcal{O}(P)$ for $N$ fixed. Our hypotheses for this phenomenon is that, although we are increasing $P$, the cost of applying all the translation operators is not increasing, because the operators' size is very small ($20 \times 20$ at most) and it is possible that the low-level linear algebra routine that performs the matrix-vector products do some optimization which reduces the true cost $P^2$ cost of applying the operators. A comparison between the relative error and the time to compute the full potential is shown in Figure 12.

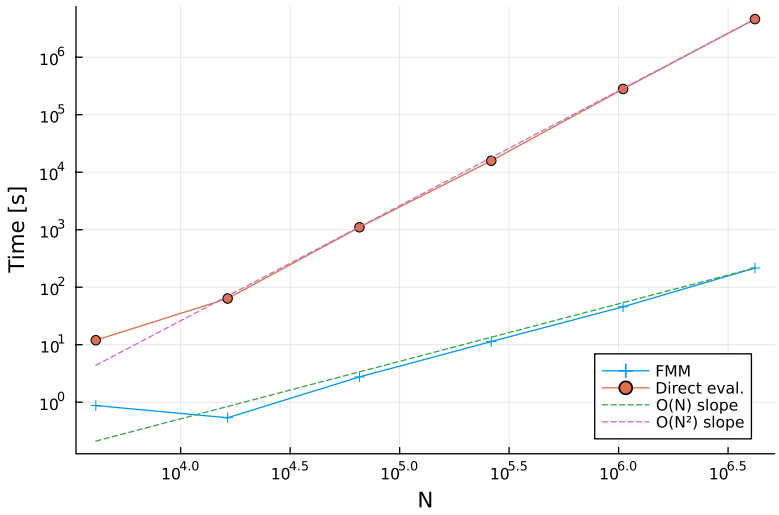



FIGURE 8. *A comparison of the time to compute the full potential between the FMM and the direct method, first numerical experiment.*

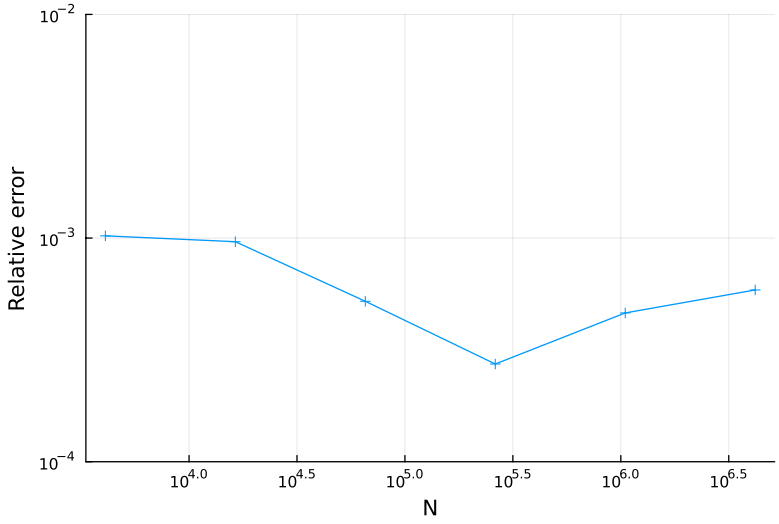



FIGURE 9. *Relative error committed by the FMM as a function of $N$, first numerical experiment.*

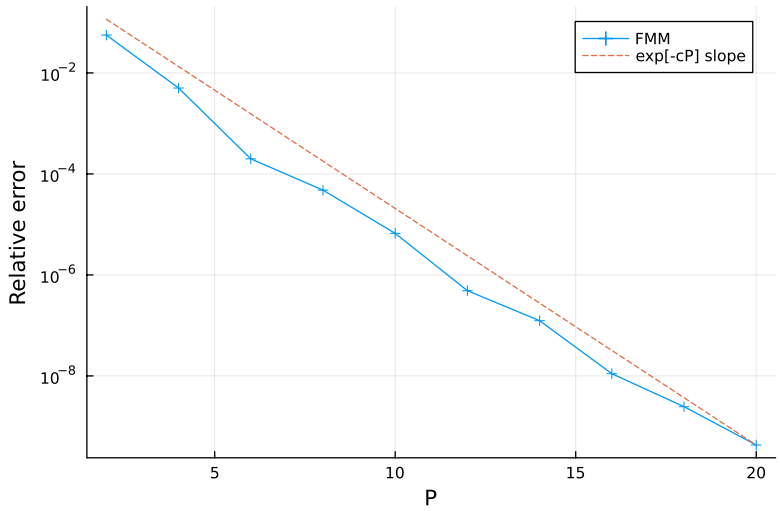



FIGURE 10. *Relative error committed by the FMM as a function of $P$, second numerical experiment.*

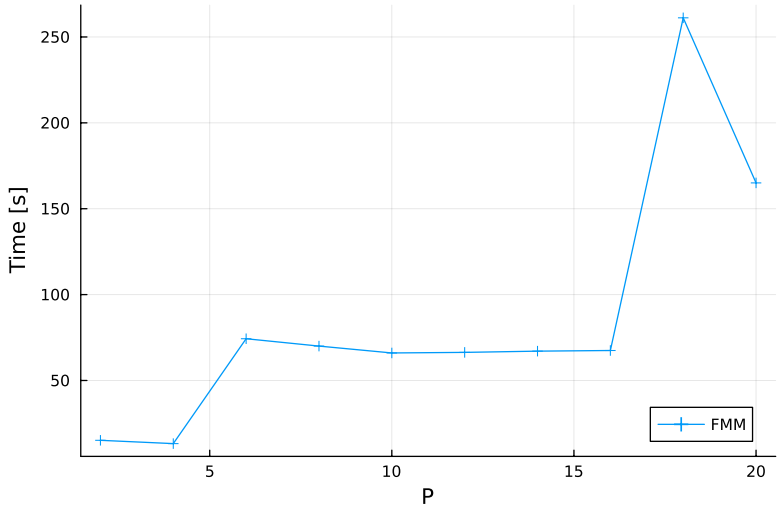



FIGURE 11. *Time to compute the full potential as a function of $P$, second numerical experiment.*

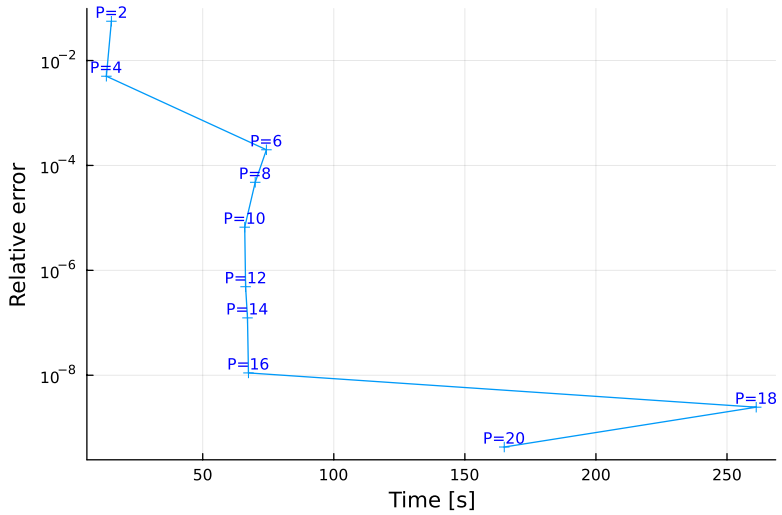



FIGURE 12. *Comparison between relative error and time to compute the full potential, second numerical experiment.*

## REFERENCES


[1] R. ARRIETA, *Just another implementation of the fast multipole method for the electrostatic potential in 2d*, 2022, https://github.com/Riarrieta/FMMProject.

[2] L. GREENGARD, *The rapid evaluation of potential fields in particle systems*, MIT press, 1988.

[3] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, Journal of computational physics, 73 (1987), pp. 325–348.

[4] P.-G. MARTINSSON, *Fast direct solvers for elliptic PDEs*, SIAM, 2019.

[5] H. G. PETERSEN, D. SOELVASON, J. W. PERRAM, AND E. SMITH, *Error estimates for the fast multipole method. i. the two-dimensional case*, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences, 448 (1995), pp. 389–400.