FMM and $\mathcal{H}$-matrices:
a short introduction to the basic idea

M. Fenn and G. Steidl

Abstract

The aim of this paper is a short introduction to a fundamental algorithm for the fast multiplication of vectors with fully populated, special matrices arising in various applications. The basic idea is known as fast multipole method, fast multiplication by $\mathcal{H}$–matrices or by mosaic–skeleton matrices. We prefer a linear algebraic approach which may serve as a basis for student seminars in Mathematics, Computer Science or Engineering. Our introduction is accompanied by a broad, but far away from complete, list of references, where the reader may find more sophisticated material.

1 Introduction

This paper gives a short introduction to a fundamental algorithm for the fast multiplication of a vector by a fully populated matrix $\boldsymbol{M} = (m_{jk})_{j,k=1}^{N}$ which of course must have some special properties. Otherwise the straightforward matrix-vector multiplication requires $\mathcal{O}(N^2)$ arithmetic operations. In literature the considered algorithm appears under three names, namely *fast multipole method* (FMM), *fast mosaic-skeleton matrix multiplication* and *fast $\mathcal{H}$-matrix multiplication*. Each of these approaches shows some special features mainly due to the applications the authors had in mind, but the basic ideas coincide.

The *FMM* with arithmetic complexity $\mathcal{O}(N)$ and its slower variant, the hierarchical multipole method with arithmetic complexity $\mathcal{O}(N \log N)$ were designed by L. Greengard and V. Rokhlin [17, 19] for the particle simulation in $\mathbb{R}^d$. Here

$$m_{j,k} = K(\boldsymbol{x}_j - \boldsymbol{x}_k),$$

where K is the radial function (isotropic kernel) $K(\boldsymbol{x}-\boldsymbol{y}) = \log\|\boldsymbol{x}-\boldsymbol{y}\|$ if $d = 2$ and $K(\boldsymbol{x}-\boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|^{-1}$ if $d = 3$. L. Greengard and other authors have also used the method for the fast Gauss transform, where K is the Gaussian [22, 23] and for many other large-scale matrix computations [2, 11, 20, 10, 13, 18, 21, 37]. Further the FMM was adapted to other radial basic functions arising in the approximation of curves and surfaces by R. Beatson, W. Light and co-workers [4, 3, 5, 12].

E. Tyrtyshnikov et al. [39, 40, 16] have designed algorithms for fast $\mathcal{O}(N \log N)$ matrix-vector multiplications from a linear algebraic point of view. E. Tyrtyshnikov calls the idea behind the algorithm 'mosaic-skeleton approximation' of $\boldsymbol{M}$ and refers to [41] for an early appearance of the idea. Here the matrix coefficients are $m_{j,k} = K(\boldsymbol{x}_j, \boldsymbol{x}_k)$, where the kernel has to be a modified asymptotically smooth function [9].

W. Hackbusch et al. [25, 29, 28, 27, 30, 26] have created the concept of $\mathcal{H}$-matrices, where $\mathcal{H}$ abbreviates 'hierarchical'. It includes the concept of *panel clustering* earlier developed by W. Hackbusch and co-workers in order to solve boundary integral equations in an efficient numerical way [31, 24]. The matrix entries arise from a collocation or Galerkin approach and have e. g. the form

$$m_{j,k} = \int_{\Omega_i} \int_{\Omega_j} K(\boldsymbol{x}, \boldsymbol{y})\, \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y},$$

where K is the same kernel as in the particle simulation. The original algorithm is of arithmetic complexity $\mathcal{O}(N \log N)$. In case of $\mathcal{H}^2$-matrices one can develop an $\mathcal{O}(N)$ algorithm if in addition a so-called 'consistency condition' is fulfilled. The idea coincides with those of the FMM. We mention that the whole $\mathcal{H}$-matrix concept is not restricted to fast matrix-vector multiplications but includes also fast $\mathcal{H}$-matrix inversions via Schur complement methods.

Although we restrict our attention to FMM-like algorithms we like to mention the existence of other algorithms for the fast matrix-vector multiplication which don't fit into the FMM/$\mathcal{H}$-matrix/mosaic-skeleton-matrix concept:

- *Wavelet methods* [1, 7, 32] are based on an approximation of $\boldsymbol{M}$ by

$$\boldsymbol{M} \approx \tilde{\boldsymbol{W}} \boldsymbol{S} \boldsymbol{W},$$

  where the vector multiplications with the wavelet transform matrices $\tilde{\boldsymbol{W}}, \boldsymbol{W}$ require only $\mathcal{O}(N)$ arithmetic operations and where $\boldsymbol{S}$ is a sparse matrix containing only $\mathcal{O}(N \log N)$ nonzero elements.

  Note that the wavelet method works without the explicit knowledge of K. For a completely discrete approach see [33].

- Based on the *fast Fourier transform for nonequispaced knots* (NFFT) one can find an approximation of $\boldsymbol{M}$ by

$$\boldsymbol{M} \approx \boldsymbol{B}_y \boldsymbol{T} \boldsymbol{B}_x, \tag{1}$$

  where the vector multiplications with the sparse matrices $\boldsymbol{B}_y$ and $\boldsymbol{B}_x$ require only $\mathcal{O}(N)$ arithmetic operations and where $\boldsymbol{T}$ is a Toeplitz matrix which can be multiplied by a vector with $\mathcal{O}(N \log N)$ arithmetic operations [36, 35].

- G. Beylkin et al. [8] have suggested an algorithm based on two-scale relations of scaling functions arising in wavelet theory or subdivision schemes. This algorithm is closely related to the NFFT based algorithm, in particular it can be written in the form (1), see [36].

In the following we want to describe the basic idea of both the $\mathcal{O}(N \log N)$ algorithm and the $\mathcal{O}(N)$ algorithm in a simple way. Here we mainly profit from [38]. For this we restrict our attention to the fast computation of

$$\boldsymbol{f} = \boldsymbol{M} \boldsymbol{\alpha} \tag{2}$$

where

$$\boldsymbol{M} = \left(K(x_k, y_j) \right)_{j=1, k=1}^{M, N}$$

and where $x_k, y_j, \in [0, 1)$ are one-dimensional knots. We assume that, except for some singular points, the kernel K is sufficiently smooth and satisfies one of the following conditions

$$
\begin{aligned}
|\partial_x^p K(x, y)| &\leq Cp! |x - y|^{-p} & (p \in \mathbb{N}), & \qquad (3) \\
|\partial_x^\beta \partial_y^\gamma K(x, y)| &\leq Cp! |x - y|^{-p} & (\beta + \gamma = p; \, p \in \mathbb{N}). & \qquad (4)
\end{aligned}
$$

As typical example we consider the kernel $K(x, y) = \log |x - y|$ which satisfies (3) and (4) with $C = 1/p \leq 1$. In literature a couple of different conditions on the kernel was considered, see e.g. [9, 39, 7, 33].

Further, we assume for sake of simplicity that both the source knots x_k and the target knots y_j are uniformly distributed and ordered so that $x_1 < \ldots < x_N$ and $y_1 < \ldots < y_M$. Indeed it is sufficient that either source or target points are uniformly distributed. If this is not the case additional adaptation techniques are required [10, 34].

The algorithm is based on

- a hierarchical splitting of $\boldsymbol{M}$ into admissible blocks and

- a low rank approximation of each admissible block.

2 Hierarchical splitting into admissible blocks

The following notation is mainly adapted from W. Hackbusch and co-workers. Although its strength becomes more clear in the multi-dimensional setting we find it also useful in one dimension.

Let $I = \{1, \ldots, N\}$ and $J = \{1, \ldots, M\}$ be index sets and let $X = \{x_i : i \in I\}$ and $Y = \{y_j : j \in J\}$. Let $\mathcal{P}(I)$ be a partition of I, i.e.

$$
I = \overset{.}{\underset{\sigma \in \mathcal{P}(I)}{\bigcup}} \sigma.
$$

For $\sigma \in \mathcal{P}(I)$ and $\tau \in \mathcal{P}(J)$, let

$$
X(\sigma) = \{x_i \in X : i \in \sigma\}, \quad Y(\tau) = \{y_j \in Y : j \in \tau\}.
$$

According to any block of indices $b = \tau \times \sigma$, $\tau \in \mathcal{P}(J)$, $\sigma \in \mathcal{P}(I)$, we can consider the matrix block

$$
\boldsymbol{M}^b = \left(m_{ji} \right)_{j \in \tau, i \in \sigma}.
$$

We are mainly interested in so-called admissible blocks. These will be the blocks which can be approximated by low rank matrices. Let r_σ and r_τ denote the diameters and c_σ and c_τ be the centers of $X(\sigma)$ and $Y(\tau)$, respectively, i.e.,

$$
|x_i - c_\sigma| \leq r_\sigma \quad (i \in \sigma), \quad |y_j - c_\tau| \leq r_\tau \quad (j \in \tau)
$$

and let

$$
\mathrm{dist}(\tau, \sigma) = \min_{j \in \tau, i \in \sigma} |y_j - x_i|
$$

be the distance of two clusters τ and σ. Then a block $b = \tau \times \sigma$ is called *admissible*, if there exists $\eta \in (0, 1]$ so that

$$
\eta \, \mathrm{dist}(\tau, \sigma) \geq r_\tau + r_\sigma. \qquad (5)
$$

In order to split our matrix into admissible blocks we use a hierarchical splitting of the index sets I and J. The tree which corresponds to this hierarchical index splitting is called $\mathcal{H}$-tree by W. Hackbusch. In one dimension we can simply use the following binary splitting to obtain a binary tree:

Let $\mathcal{T}_I(0) = I$. At level ℓ, the vertices of our tree are given by the index sets

$$\sigma = \sigma(\ell, m) = \left\{ k \in I : x_k \in [m/2^\ell, (m+1)/2^\ell) \right\} \qquad (m = 0, \dots, 2^\ell - 1).$$

By $\mathcal{T}_I(\ell)$ we denote the corresponding partition of I. We obtain a similar tree $\mathcal{T}_J$ for J. Since our knots x_k and y_j are uniformly distributed, each $\sigma \in \mathcal{T}_I(\ell)$ has approximately the same number $[N/2^\ell]$ of indices. Here $[a]$ denotes the integer part of a. Note that $r_\sigma \approx 1/2^{\ell+1}$ and $c_\sigma \approx (m + 1/2)/2^\ell$, where both values are smaller than the right-hand sides. We stop our binary partitioning if each index set contains only a small number, say $\leq \nu$, of indices. Let $n = [\log_2(N/\nu)]$ be the number of levels.

By $\mathcal{T}_{J \times I}(\ell) = \mathcal{T}_J(\ell) \times \mathcal{T}_I(\ell)$ we denote the tensor block partition of $J \times I$.

Now we can produce a hierarchical splitting of our coefficient matrix $\boldsymbol{M}$ into admissible blocks. We start at level 2. We split $\boldsymbol{M}$ with respect to the blocks $b = \tau \times \sigma \in \mathcal{T}_{J \times I}(2)$ and sort admissible and nonadmissible blocks:

$$\boldsymbol{M} = \boldsymbol{M}_2 + \boldsymbol{N}_2,$$

where $\boldsymbol{M}_2$ consists of the admissible blocks of $\mathcal{T}_{J \times I}(2)$ and $\boldsymbol{N}_2$ of the other ones. We proceed with $\boldsymbol{N}_2$, i.e.

$$\boldsymbol{N}_2 = \boldsymbol{M}_3 + \boldsymbol{N}_3,$$

where $\boldsymbol{M}_3$ consists of the admissible blocks of $\mathcal{T}_{J \times I}(3)$ contained in $\boldsymbol{N}_2$ and $\boldsymbol{N}_3$ of the other ones. Repeating this procedure up to level n we obtain the final additive splitting

$$\boldsymbol{M} = \sum_{\ell=2}^{n} \boldsymbol{M}_\ell + \boldsymbol{N}_n \tag{6}$$

of $\boldsymbol{M}$ into admissible blocks contained in the matrices $\boldsymbol{M}_\ell$ and into a 'near-field matrix' $\boldsymbol{N}_n$.

It is easy to check that there is only a small number $\leq \gamma$ of non-zero blocks in each row/column of $\boldsymbol{M}_\ell$. In particular, if $\eta = 2^{-r}$ ($r \in \mathbb{N}$ small) then $\gamma = [2/\eta] + 1$. Therefore, $\boldsymbol{M}_\ell$ consists of no more than $2^\ell \gamma$ non-zero blocks. The same holds for $\boldsymbol{N}_\ell$. Figure 1 shows the non-zero blocks of $\boldsymbol{M}_\ell$ (thick lines) for $\ell = 2, 3, 4$ in the cases $\eta = 1$ and $\eta = 1/2$. Indeed for the upper figure η can be chosen smaller than 1.

3 Low rank approximation of admissible blocks

Next we will see how admissible blocks can be approximated by low rank matrices. Of course, supposed that a 'good' low rank approximation exists, it is easy to find, if the singular value decomposition (SVD) of the admissible blocks is accessible. But the SVD is computationally very expensive, so that approximations based on the SVD cannot lead to fast algorithms. In this context E. Tyrtyshnikov et al. have proposed a $\boldsymbol{CGR}$ decomposition of admissible blocks [39, 40], M. Bebendorf an iterative approximation scheme [6] and W. Hackbusch et al. Taylor expansion [25, 29, 27] and polynomial interpolation [26].

$$\ell = 2 \qquad\qquad \ell = 3 \qquad\qquad \ell = 4$$

$$\eta = 1$$

$$\eta = \tfrac{1}{2}$$
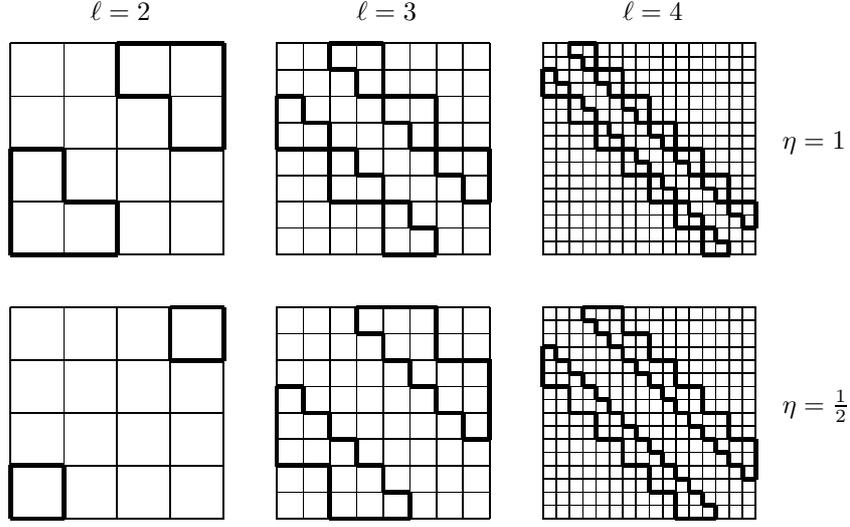
Figure 1: Non-zero blocks of $\boldsymbol{M}_\ell$

In this paper, we consider only the simplest case that K is known and satisfies one of the properties (3) or (4).

Let $b = \tau \times \sigma$ be an admissible block and let $x \in X(\sigma)$ and $y \in Y(\tau)$. If K satisfies (3), then we obtain by Taylor expansion at c_σ with respect to x

$$
\begin{aligned}
K(x,y) &= \sum_{\ell=0}^{p-1} \frac{1}{\ell!} (x - c_\sigma)^\ell \, \partial_x^\ell K(c_\sigma, y) \; + \; R_p(x,y) \\
&= \sum_{\ell=0}^{p-1} \frac{1}{\ell!} \varphi_\ell^\sigma(x) \psi_\ell^{\tau,\sigma}(y) \; + \; R_p(x,y),
\end{aligned}
$$

where

$$\varphi_\ell^\sigma(x) = (x - c_\sigma)^\ell \quad \text{and} \quad \psi_\ell^{\tau,\sigma}(y) = \partial_x^\ell K(c_\sigma, y).$$

For the approximation error we have by (3) that $|\tilde{x} - y| \geq (r_\tau + r_\sigma)/\eta$ and consequently

$$|R_p(x,y)| = \frac{1}{p!}|x - c_\sigma|^p \, |\partial_x^p K(\tilde{x}, y)| \leq C \frac{|x - c_\sigma|^p}{|\tilde{x} - y|^p},$$

where $\tilde{x} = c_\sigma + \theta(x - c_\sigma)$ $(\theta \in (0,1))$, and by the admissibility condition (5) that

$$|R_p(x,y)| \leq C\eta^p \left(\frac{r_\sigma}{r_\tau + r_\sigma} \right)^p.$$

Thus, if $\eta \leq 1$, then $\boldsymbol{M}^b = \big(K(x_k, y_j)\big)_{j \in \tau, k \in \sigma}$ can be approximated with a small error by

$$\boldsymbol{M}^b \approx \tilde{\boldsymbol{M}}^b = (\boldsymbol{\Psi}^{\tau,\sigma})^T \boldsymbol{D}\, \boldsymbol{\Phi}^\sigma \tag{7}$$

where $\boldsymbol{D} = \operatorname{diag}(1/\ell!)_{\ell \in P}$ with index set $P = \{0, \ldots, p-1\}$ and

$$\boldsymbol{\Phi}^\sigma = \big(\varphi_\ell^\sigma(x_k)\big)_{\ell \in P, k \in \sigma} \in \mathbb{R}^{p, \#\sigma}, \quad \boldsymbol{\Psi}^{\tau,\sigma} = \big(\psi_\ell^{\tau,\sigma}(y_j)\big)_{\ell \in P, j \in \tau} \in \mathbb{R}^{p, \#\tau}.$$

The error decays exponentially with increasing p. Since $\tilde{\boldsymbol{M}}^b$ is a matrix of rank $\leq p$ its multiplication with a vector requires only $\mathcal{O}(p(\#\sigma + \#\tau))$ arithmetic operations. Note that E. Tyrtyshnikov calls the rank-1 matrices $\left(\psi_\ell^{\tau,\sigma}(y_j)\right)_{j\in\tau}\left(\varphi_\ell^\sigma(x_k)\right)_{k\in\sigma}^T$ *skeletons*.

If K satisfies (4), then we obtain by bivariate Taylor expansion at (c_σ, c_τ)

$$
\begin{aligned}
K(x,y) &= \sum_{\ell=0}^{p-1} \frac{1}{\ell!}\left((x-c_\sigma)\partial_x + (y-c_\tau)\partial_y\right)^\ell K(c_\sigma, c_\tau) \;+\; R_p(x,y) \\
&= \sum_{0\leq\ell+m\leq p-1} \frac{1}{\ell!m!}\partial_x^\ell\partial_y^m K(c_\sigma,c_\tau)\,(x-c_\sigma)^\ell(y-c_\tau)^m \;+\; R_p(x,y) \\
&= \sum_{0\leq\ell+m\leq p-1} \frac{1}{\ell!m!}\partial_x^\ell\partial_y^m K(c_\sigma,c_\tau)\,\varphi_\ell^\sigma(x)\psi_m^\tau(y) \;+\; R_p(x,y),
\end{aligned}
$$

where

$$\varphi_\ell^\sigma(x) = (x-c_\sigma)^\ell \quad \text{and} \quad \psi_m^\tau(y) = (y-c_\tau)^m.$$

For the approximation error we have by (4) that

$$
\begin{aligned}
|R_p(x,y)| &= \frac{1}{p!}\left|(x-c_\sigma)\partial_x + (y-c_\tau)\partial_y\right|^p K(\tilde{x},\tilde{y}) \\
&\leq C\frac{\left(|x-c_\sigma| + |y-c_\tau|\right)^p}{|\tilde{x}-\tilde{y}|^p}
\end{aligned}
$$

where $\tilde{x} = c_\sigma + \theta(x-c_\sigma)$, $\tilde{y} = c_\tau + \theta(y-c_\tau)$ ($\theta \in (0,1)$), and by the admissibility condition (5) that

$$|R_p(x,y)| \leq C\eta^p.$$

Thus, if $\eta < 1$, then $\boldsymbol{M}^b = \left(K(x_k, y_j)\right)_{j\in\tau, k\in\sigma}$ can be approximated with small error by

$$\boldsymbol{M}^b \approx \tilde{\boldsymbol{M}}^b = (\boldsymbol{\Psi}^\tau)^T \boldsymbol{A}^{\tau,\sigma} \boldsymbol{\Phi}^\sigma \tag{8}$$

where

$$\boldsymbol{\Phi}^\sigma = \left(\varphi_\ell^\sigma(x_k)\right)_{\ell\in P, k\in\sigma} \in \mathbb{R}^{p,\#\sigma}, \quad \boldsymbol{\Psi}^\tau = \left(\psi_m^\tau(y_j)\right)_{\ell\in P, m\in\tau} \in \mathbb{R}^{p,\#\tau}$$

and $\boldsymbol{A}^{\tau,\sigma} = \left(a_{\ell,m}^{\tau,\sigma}\right)_{\ell,m\in P} \in \mathbb{R}^{p,p}$ with

$$a_{\ell,m}^{\tau,\sigma} = \frac{1}{\ell!m!}\partial_x^\ell\partial_y^m K(c_\sigma,c_\tau) \quad \text{if} \quad 0\leq\ell+m\leq p-1$$

and $a_{\ell,m}^{\tau,\sigma} = 0$ otherwise. Again the error decreases exponentially with increasing p. Since $\tilde{\boldsymbol{M}}^b$ is a matrix of rank $\leq p(p+1)/2$ its multiplication with a vector requires only $\mathcal{O}(p(\#\sigma+(p+1)/2+\#\tau))$ arithmetic operations. Of course we can also use a Taylor expansion of K such that $\boldsymbol{A}^{\tau,\sigma}$ is a fully populated $p\times p$ matrix.

Example. Let $K(x,y) = \log|x-y|$. Then

$$
a_{\ell,m}^{\tau,\sigma} = \begin{cases}
\log|c_\tau - c_\sigma| & \text{for } \ell = m = 0, \\
-\frac{(-1)^\ell}{\ell+m}(c_\tau - c_\sigma)^{-\ell-m}\binom{\ell+m}{\ell} & \text{for } \ell+m\leq p-1, \\
0 & \text{otherwise.}
\end{cases}
$$

In the following, we assume that each admissible block $\boldsymbol{M}^b$ can be approximated with only small error by a matrix $\tilde{\boldsymbol{M}}^b$ of one of the following forms

$$
\begin{aligned}
\tilde{\boldsymbol{M}}^b &= (\boldsymbol{\Psi}^{\tau,\sigma})^T \, \boldsymbol{D}^{\tau,\sigma} \, \boldsymbol{\Phi}^{\tau,\sigma}, & (9) \\
\tilde{\boldsymbol{M}}^b &= (\boldsymbol{\Psi}^{\tau})^T \, \boldsymbol{A}^{\tau,\sigma} \, \boldsymbol{\Phi}^{\sigma}, & (10)
\end{aligned}
$$

where

$$
\boldsymbol{\Phi}^{\bullet} \in \mathbb{R}^{p \times \#\sigma}, \ \boldsymbol{\Psi}^{\bullet} \in \mathbb{R}^{p \times \#\tau}, \ \boldsymbol{A}^{\bullet} \in \mathbb{R}^{p \times p}
$$

and $\boldsymbol{D}^{\bullet} \in \mathbb{R}^{p \times p}$ is a diagonal matrix. The first representation (9) may be simply obtained from an SVD, while (10) is of the form (8). The approximation (7) corresponds to a mixture of both forms and a fast matrix-vector multiplication algorithm follows straightforward if we have algorithms for (9) and (10).

Note that one can use level-dependent approximations of admissible blocks $\boldsymbol{M}^b$ where the rank of the approximating matrix $\tilde{\boldsymbol{M}}^b$ depends on the decomposition level ℓ of the $\mathcal{H}$-tree, see [27]. This is beyond the scope of this paper.

Now (6) can be approximated by

$$
\boldsymbol{M} \approx \sum_{\ell=2}^{n} \tilde{\boldsymbol{M}}_\ell \, + \, \boldsymbol{N}_n, \tag{11}
$$

where the blocks in $\tilde{\boldsymbol{M}}_\ell$ are low rank approximations of the form (9) or (10) of the admissible blocks in $\boldsymbol{M}_\ell$. W. Hackbusch calls the matrix on the right-hand side of (11) an $\mathcal{H}$-matrix (in case of (10) an *uniform* $\mathcal{H}$-matrix) and E. Tyrtyshnikov a *mosaic-skeleton approximation* of $\boldsymbol{M}$.

If we have an approximation of type (10) then $\tilde{\boldsymbol{M}}_\ell$ can be further rewritten as

$$
\tilde{\boldsymbol{M}}_\ell = \text{blockdiag}(\boldsymbol{\Psi}^{\tau})^T_{\tau \in \mathcal{T}_J(\ell)} \, \boldsymbol{A}_\ell \, \text{blockdiag}(\boldsymbol{\Phi}^{\sigma})_{\sigma \in \mathcal{T}_I(\ell)}, \tag{12}
$$

where $\boldsymbol{A}_\ell \in \mathbb{R}^{p2^\ell, p2^\ell}$ has the non-zero blocks $\boldsymbol{A}^{\tau,\sigma} \in \mathbb{R}^{p,p}$ at the 'position' of the non-zero blocks of $\boldsymbol{M}_\ell$.

4 The hierarchical $\mathcal{O}((N + M) \log N)-$Algorithm

Assume that the non-zero blocks of $\boldsymbol{M}_\ell$ are of the form (9). Using (11) the matrix–vector multiplication (2) can be computed approximately by

$$
\boldsymbol{f} = \boldsymbol{M}\boldsymbol{\alpha} \approx \underbrace{\sum_{\ell=2}^{n} \tilde{\boldsymbol{M}}_\ell \boldsymbol{\alpha}}_{\text{far-field}} \, + \, \underbrace{\boldsymbol{N}_n \boldsymbol{\alpha}}_{\text{near-field}} \, = \, f_F + f_N \, .
$$

We call the computation of the first $n-1$ matrix–vector products on the right-hand side 'far-field computation' and the last matrix-vector multiplication 'near-field computation'.

Since multiplication with a block $\tilde{\boldsymbol{M}}^b$ requires $\mathcal{O}(p(\#\sigma + \#\tau))$ arithmetic operations, where $\#\tau \leq M/2^\ell$, $\#\sigma \leq N/2^\ell$, and there are no more than $2^\ell \gamma$ such blocks in $\tilde{\boldsymbol{M}}_\ell$, the computation of $\tilde{\boldsymbol{M}}_\ell \boldsymbol{\alpha}$ requires $\mathcal{O}(p(M + N))$ arithmetic

operations. Adding this up over all levels, we get an arithmetic complexity of $\mathcal{O}(p(N+M)\log N)$ for the far-field computation. Note that the approximation error becomes smaller with increasing p.

Since $\boldsymbol{N}_n$ has at most γ non-zero blocks per row each with $\leq \nu$ columns, the near-field correction requires $\mathcal{O}(M\nu\gamma)$ arithmetic operations.

Since ν, γ and p are constants, the whole algorithm requires $\mathcal{O}((N+M)\log N)$ arithmetic operations.

5   The fast $\mathcal{O}(N+M)$-Algorithm

In this section we introduce a fast algorithm of arithmetic complexity $\mathcal{O}(N+M)$.

The algorithm is only practicable if the admissible blocks of the matrix can be approximated by an expression of the form (10). In addition, the matrices $\boldsymbol{\Phi}^\sigma$ and $\boldsymbol{\Psi}^\tau$ have to be 'nested', i. e., fulfil the following *consistency conditions*: let $\sigma', \sigma'' \in \mathcal{T}_I(\ell+1)$ be the sons of $\sigma \in \mathcal{T}_I(\ell)$ and let $\tau', \tau'' \in \mathcal{T}_J(\ell+1)$ be the sons of $\tau \in \mathcal{T}_J(\ell)$. Then they have to fulfil

$$\boldsymbol{\Phi}^\sigma = [\boldsymbol{C}^{\sigma,\sigma'}\ \boldsymbol{C}^{\sigma,\sigma''}] \begin{pmatrix} \boldsymbol{\Phi}^{\sigma'} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}^{\sigma''} \end{pmatrix} = \boldsymbol{C}^{\sigma,\sigma'}\boldsymbol{\Phi}^{\sigma'} + \boldsymbol{C}^{\sigma,\sigma''}\boldsymbol{\Phi}^{\sigma''}, \quad (13)$$

$$\boldsymbol{\Psi}^\tau = [\boldsymbol{C}^{\tau,\tau'}\ \boldsymbol{C}^{\tau,\tau''}] \begin{pmatrix} \boldsymbol{\Psi}^{\tau'} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Psi}^{\tau''} \end{pmatrix} = \boldsymbol{C}^{\tau,\tau'}\boldsymbol{\Psi}^{\tau'} + \boldsymbol{C}^{\tau,\tau''}\boldsymbol{\Psi}^{\tau''}. \quad (14)$$

Then the matrices in (10) are called $\mathcal{H}^2$-matrices and the corresponding algorithm either FMM or fast $\mathcal{H}^2$-matrix multiplication.

For $\boldsymbol{\Phi}^\sigma \in \mathbb{R}^{p \times \#\sigma}$ and $\boldsymbol{\Psi}^\tau \in \mathbb{R}^{p \times \#\tau}$ arising from Taylor expansions as in (8) the consistency conditions are clearly fulfilled: since

$$(x-c_\sigma)^\ell = \big((x-c_{\sigma'})-(c_\sigma-c_{\sigma'})\big)^\ell$$
$$= \sum_{m=0}^\ell \binom{\ell}{m}(c_{\sigma'}-c_\sigma)^{\ell-m}(x-c_{\sigma'})^m$$

for all $\ell = 0,\ldots,p-1$, we obtain

$$\big((x_k-c_\sigma)^\ell\big)_{k\in\sigma'}^T = \sum_{m=0}^\ell C_{\ell,m}^{\sigma,\sigma'} \big((x_k-c_{\sigma'})^m\big)_{k\in\sigma'}^T,$$

$$\big((x_k-c_\sigma)^\ell\big)_{k\in\sigma''}^T = \sum_{m=0}^\ell C_{\ell,m}^{\sigma,\sigma''} \big((x_k-c_{\sigma''})^m\big)_{k\in\sigma''}^T.$$

Thus (13) is fulfilled with the lower triangular matrix $\boldsymbol{C}^{\sigma,\sigma'} = \left(C_{\ell,m}^{\sigma,\sigma'}\right)_{\ell,m\in P}$, where

$$C_{\ell,m}^{\sigma,\sigma'} = \begin{cases} 0 & \text{for } \ell < m, \\ \binom{\ell}{m}(c_\sigma-c_{\sigma'})^{\ell-m} & \text{for } \ell \geq m. \end{cases}$$

Note that it is often also sufficient if the consistency conditions (13) and (14) are satisfied only approximately, i.e., up to a small error, see [15, 14].

For $\ell = 2, \ldots, n-1$, let

$$\boldsymbol{D}^{\Phi}_{\ell,\ell+1} = \text{blockdiag}\left(\left[\boldsymbol{C}^{\sigma,\sigma'}\,\boldsymbol{C}^{\sigma,\sigma''}\right]\right)_{\sigma \in \mathcal{T}_I(\ell)} \in \mathbb{R}^{p,2p}$$

denote the transform matrices arising from the consistency conditions for all $\sigma \in \mathcal{T}_I(\ell)$. Then the consistency condition at level ℓ reads as

$$\text{blockdiag}\left(\boldsymbol{\Phi}^\sigma\right)_{\sigma \in \mathcal{T}_I(\ell)} = \boldsymbol{D}^{\Phi}_{\ell,\ell+1}\,\text{blockdiag}\left(\boldsymbol{\Phi}^\sigma\right)_{\sigma \in \mathcal{T}_I(\ell+1)}.$$

Now successive application of the consistency condition leads to

$$\begin{aligned}
\tilde{\boldsymbol{M}}_\ell &= \text{blockdiag}\left(\boldsymbol{\Psi}^\tau\right)^T_{\tau \in \mathcal{T}_J(\ell+1)} \left(\boldsymbol{D}^{\Psi}_{\ell,\ell+1}\right)^T \, \boldsymbol{A}_\ell \, \boldsymbol{D}^{\Phi}_{\ell,\ell+1}\,\text{blockdiag}\left(\boldsymbol{\Phi}^\sigma\right)_{\sigma \in \mathcal{T}_I(\ell+1)} \\
&= \cdots \\
&= \text{blockdiag}\left(\boldsymbol{\Psi}^\tau\right)^T_{\tau \in \mathcal{T}_J(n)}\left(\boldsymbol{D}^{\Psi}_{n-1,n}\right)^T \cdots \left(\boldsymbol{D}^{\Psi}_{\ell,\ell+1}\right)^T \, \boldsymbol{A}_\ell \,\times \\
&\quad \times \boldsymbol{D}^{\Phi}_{\ell,\ell+1} \cdots \boldsymbol{D}^{\Phi}_{n-1,n}\text{blockdiag}\left(\boldsymbol{\Phi}^\sigma\right)_{\sigma \in \mathcal{T}_I(n)}.
\end{aligned} \tag{15}$$

The important observation is that the factors $\text{blockdiag}(\boldsymbol{\Phi}^\sigma)_{\sigma \in \mathcal{T}_I(n)}$ and $\text{blockdiag}(\boldsymbol{\Psi}^\tau)_{\sigma \in \mathcal{T}_I(n)}$ appear in all matrices $\tilde{\boldsymbol{M}}_\ell$ $(\ell = 2, \ldots, n)$ and that the factors $\boldsymbol{D}^{\Phi}_{i,i+1}$ and $\boldsymbol{D}^{\Psi}_{i,i+1}$ appear in all matrices $\tilde{\boldsymbol{M}}_\ell$ with $\ell \leq i$.

Using (15) and (11) we can formulate the whole algorithm now. (For readers familiar with the FMM we have written the FMM notation of the algorithm in brackets, where FFE stands for far-field extension and LFE for near-field extension.)

Algorithm

1. **Forward Transformation (FFE → FFE)**

   Initialization:
   $$\boldsymbol{x}_n = \text{blockdiag}\left(\boldsymbol{\Phi}^\sigma\right)_{\sigma \in \mathcal{T}_I(n)}\boldsymbol{\alpha} \in \mathbb{R}^{p2^n}$$

   Arithmetic complexity: $\mathcal{O}(pN)$

   For $\ell = n-1, \ldots, 2$ compute
   $$\boldsymbol{x}_\ell = \boldsymbol{D}^{\Phi}_{\ell,\ell+1}\boldsymbol{x}_{\ell+1}.$$

   Arithmetic complexity:
   Since $\boldsymbol{D}^{\Phi}_{\ell,\ell+1}$ consists of 2^ℓ non-zero blocks of the form $[\boldsymbol{C}^{\sigma,\sigma'}\,\boldsymbol{C}^{\sigma,\sigma''}] \in \mathbb{R}^{p \times 2p}$ we have an amount of $\leq 2p^2\,2^\ell$ arithmetic operations in step ℓ. This adds up over all levels to $\mathcal{O}(2p^2\frac{N}{\nu})$ arithmetic operations.

2. **Multiplication Phase (FFE → LFE)**

   For $\ell = 2, \ldots, n$ compute
   $$\boldsymbol{y}_\ell = \boldsymbol{A}_\ell\,\boldsymbol{x}_\ell.$$

   Arithmetic complexity:
   There are at most $2^\ell\gamma$ non-zero blocks on level ℓ and each block in A_ℓ is of size $p \times p$. Thus the computation of $\boldsymbol{A}_\ell\boldsymbol{x}_\ell$ requires $\mathcal{O}(p^2 2^\ell)$ arithmetic operations which adds up to $\mathcal{O}(p^2 N/\nu)$ arithmetic operations over all levels.

3. **Backward Transform (LFE → LFE)**

   In the far–field it remains to compute

   $$\boldsymbol{f}_F = \sum_{\ell=2}^{n-1} \text{blockdiag}\big(\boldsymbol{\Psi}^\tau\big)^T_{\tau \in \mathcal{T}_J(n)} \, (\boldsymbol{D}_{n-1,n}^{\Psi})^T \cdots (\boldsymbol{D}_{\ell,\ell+1}^{\Psi})^T \boldsymbol{y}_\ell.$$

   We apply Horner's rule. Set

   $$\boldsymbol{z}_2 = \boldsymbol{y}_2$$

   and compute for $\ell = 3, \ldots, n$ the vectors

   $$\boldsymbol{z}_\ell = (\boldsymbol{D}_{\ell-1,\ell}^{\Psi})^T \boldsymbol{z}_{\ell-1} + \boldsymbol{y}_\ell.$$

   Arithmetic complexity:
   Multiplication with $(\boldsymbol{D}_{\ell,\ell+1}^{\Psi})^T$ requires as in Step 1 only $\mathcal{O}(2p^2 2^\ell)$ operations such that we have a total of $\mathcal{O}(p^2 N/\nu)$ arithmetic operations.

   Final multiplication:

   $$\boldsymbol{f}_F = \text{blockdiag}\big(\boldsymbol{\Psi}^\tau\big)^T_{\tau \in \mathcal{T}_J(n)} \, \boldsymbol{z}_n.$$

   Arithmetic complexity: $\mathcal{O}(pM)$

4. **Near-Field Correction**: Compute $\boldsymbol{f}_N = \boldsymbol{N}_n \, \boldsymbol{\alpha}$ directly and add $\boldsymbol{f}_F$.

   Arithmetic complexity:
   $\mathcal{O}(M\nu\gamma)$ as in the hierarchical algorithm.

   Choosing $\nu = p$ the arithmetic complexity of the whole algorithm is

   $$\mathcal{O}(p(N + M)) = \mathcal{O}(N + M),$$

where p is a constant which regulates the approximation error.

References

[1] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin. Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM J. Sci. Comput.*, 14:159–184, 1993.

[2] C. R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM J. Sci. Statist. Comput.*, 13:923–947, 1992.

[3] R. K. Beatson and W. A. Light. Fast evaluation of radial basis functions: methods for two-dimensional polyharmonic splines. *IMA J. Numer. Anal.*, 17:343–372, 1997.

[4] R. K. Beatson and G. N. Newsam. Fast evaluation of radial basis functions. I. *Comput. Math. Appl.*, 24:7–19, 1992.

[5] R. K. Beatson and G. N. Newsam. Fast evaluation of radial basis functions: moment-based methods. *SIAM J. Sci. Comput.*, 19:1428–1449, 1998.

[6] M. Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86:565–589, 2000.

[7] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. I. *Comm. Pure Appl. Math.*, 44:141–183, 1991.

[8] G. Beylkin and R. Cramer. A multiresolution approach to regularization of singular operators and fast summation. *SIAM J. Sci. Comput.*, 24:81–117, 2002.

[9] A. Brandt. Multilevel computations of integral transforms and particle interactions with oscillatory kernels. *Comput. Phys. Comm.*, 65:24–38, 1991.

[10] J. Carrier, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations. *SIAM J. Sci. Stat. Comput.*, 9:669–686, 1988.

[11] H. Cheng and L. Greengard. A method of images for the evaluation of electrostatic fields in systems of closely spaced conducting cylinders. *SIAM J. Appl. Math.*, 58:122–141, 1998.

[12] J. B. Cherrie, R. K. Beatson, and G. N. Newsam. Fast evaluation of radial basis functions: methods for generalized multiquadrics in $\mathbb{R}^n$. *SIAM J. Sci. Comput.*, 23:1549–1571, 2002.

[13] W. C. Chew, J. M. Jin, C. C. Lu, E. Michielssen, and J. M. Song. Fast solution methods in electromagnetics. *IEEE Trans. Antennas Propagation*, 45:533–543, 1997.

[14] A. Dutt, M. Gu, and V. Rokhlin. Fast algorithms for polynomial interpolation, integration, and differentiation. *SIAM J. Numer. Anal.*, 33:1689–1711, 1996.

[15] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. II. *Appl. Comput. Harmon. Anal.*, 2:85–100, 1995.

[16] S. A. Goreinov, E. E. Tyrtyshnikov, and A. Y. Yeremin. Matrix-free iterative solution strategies for large dense linear systems. *Numer. Linear Algebra Appl.*, 4:273–294, 1997.

[17] L. Greengard. *The rapid evaluation of potential fields in particle systems.* MIT Press, Cambridge, MA, 1988.

[18] L. Greengard and J. Helsing. On the numerical evaluation of elastostatic fields in locally isotropic two-dimensional composites. *J. Mech. Phys. Solids*, 46(8):1441–1462, 1998.

[19] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[20] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 135:280–292, 1997.

[21] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica*, 6:229–269, 1997.

[22] L. Greengard and J. Strain. The fast Gauss transform. *SIAM J. Sci. Statist. Comput.*, 12:79–94, 1991.

[23] L. Greengard and X. Sun. A new version of the fast Gauss transform. *Doc. Math.*, pages 575–584, 1998.

[24] W. Hackbusch. The panel clustering algorithm. In J. R. Whiteman, editor, *MAFELAP 1990*, pages 339–348. Academic Press, London, 1991.

[25] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. I. Introduction to $\mathcal{H}$-matrices. *Computing*, 62:89–108, 1999.

[26] W. Hackbusch, L. Grasedyck, and S. Börm. An introduction to hierarchical matrices. *Math. Bohem.*, 127:229–241, 2002.

[27] W. Hackbusch, B. Khoromskij, and S. A. Sauter. On $\mathcal{H}^2$-matrices. In H.-J. Bungartz, R. H. W. Hoppe, and C. Zenger, editors, *Lectures on Applied Mathematics (Munich, 1999)*, pages 9–29, Berlin, 2000. Springer.

[28] W. Hackbusch and B. N. Khoromskij. A sparse $\mathcal{H}$-matrix arithmetic: general complexity estimates. *J. Comput. Appl. Math.*, 125:479–501, 2000.

[29] W. Hackbusch and B. N. Khoromskij. A sparse $\mathcal{H}$-matrix arithmetic. II. Application to multi-dimensional problems. *Computing*, 64:21–47, 2000.

[30] W. Hackbusch and B. N. Khoromskij. Towards $\mathcal{H}$-matrix approximation of linear complexity. In J. Elschner, I. Gohberg, and B. Silbermann, editors, *Problems and Methods in Mathematical Physics. The Siegfried Prössdorf Memorial Volume (Chemnitz, 1990)*, pages 194–220, Basel, 2001. Birkhäuser.

[31] W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54:463–491, 1989.

[32] A. Harten and I. Yad-Shalom. Fast multiresolution algorithms for matrix-vector multiplication. *SIAM J. Numer. Anal.*, 31:1191–1218, 1994.

[33] M. Konik, R. Schneider, and G. Steidl. Matrix compression by discrete wavelet transform. In Shumaker et al., editors, *Approximation Theory VIII, Vol. 2 (College Station, TX, 1995)*, pages 225–234. World Sci. Publishing, River Edge, NJ, 1995.

[34] S. Krishnan and L. V. Kale. A parallel adaptive fast multipole algorithm for n-body problems. In K. Gallivan, editor, *Prooceedings of the International Conference on Parallel Processing*, volume 3, pages 46–50, 1995.

[35] A. Nieslony, D. Potts, and G. Steidl. Rapid evaluation of radial functions by fast Fourier transforms ot nonequispaced knots. Preprint, MU-Luebeck A02-11, May 2002.

[36] D. Potts and G. Steidl. Fast summation at nonequispaced knots by NFFTs. Preprint, MU-Luebeck, A02-02, April 2002.

[37] B. Shanker, S.-K. Han, and W. C. Chew. A fast multipole approach to computing scattering from an inhomogeneous bianisotropic cylindrical object using Beltrami fields. In *IEEE Antennas and Propagation Society International Symposium*, pages 43–51, 1997.

[38] X. Sun and N. P. Pitsianis. A matrix version of the fast multipole method. *SIAM Rev.*, 43:289–300, 2001.

[39] E. E. Tyrtyshnikov. Mosaic-skeleton approximations. *Calcolo*, 33:47–57, 1996.

[40] E. E. Tyrtyshnikov. Mosaic ranks and skeletons. In L. G. Vulkov, J. Wasniewski, and P. Y. Yalamov, editors, *Numerical Analysis and its Applications (Rousse, 1996)*, volume 1196 of *Lecture Notes in Computer Science*, pages 505–516. Springer, Berlin, 1997.

[41] V. V. Voevodin. On a method of reducing the matrix order while solving integral equations. In V. V. Voevodin, editor, *Numerical Analysis on FORTRAN*, volume 17, pages 21–26. Moscow University Press, Moscow, 1979.