

# A short course on fast multipole methods

**Rick Beatson**

*Department of Mathematics and Statistics, University of Canterbury*

**Leslie Greengard**

*Courant Institute of Mathematical Sciences, New York University*<sup>1</sup>

In this series of lectures, we describe the analytic and computational foundations of fast multipole methods, as well as some of their applications. They are most easily understood, perhaps, in the case of **particle simulations**, where they **reduce the cost of computing all pairwise interactions in a system of  $N$  particles from  $O(N^2)$  to  $O(N)$  or  $O(N \log N)$  operations**. They are equally useful, however, in solving certain partial differential equations by first recasting them as integral equations. We will draw heavily from the existing literature, especially Greengard [23, 24, 25]; Greengard and Rokhlin [29, 32]; Greengard and Strain [34].

## 1 Introduction

### 1.1 Kernels

**Many problems in computational physics require the evaluation of all pairwise interactions in large ensembles of particles**. The  $N$ -body problem of gravitation (or electrostatics), for example, requires the evaluation of

$$\Phi(\mathbf{x}_j) = \sum_{\substack{i=1 \\ i \neq j}}^N \frac{m_i}{r_{ij}}. \quad (1.1)$$

for the gravitational potential and

$$\mathbf{E}(\mathbf{x}_j) = \sum_{\substack{i=1 \\ i \neq j}}^N m_i \frac{\mathbf{x}_j - \mathbf{x}_i}{r_{ij}^3} \quad (1.2)$$

for the gravitational field. Here,  $\mathbf{x}_i$  denotes the location of the  $i$ th particle,  $m_i$  denotes the mass of the  $i$ th particle, and  $r_{ij}$  denotes the Euclidean distance between  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . Problems of electrostatics are governed by

---

<sup>1</sup>The work of Leslie Greengard was supported by the Applied Mathematical Sciences Program of the US Department of Energy, by a NSF Presidential Young Investigator Award and by a Packard Foundation Fellowship.

Coulomb's law, which takes the same form as eqs. (1.1) and (1.2), except that mass is always of one sign, whereas charge is not.

There is no reason, of course, to evaluate the gravitational field at the source locations only. Thus, we will also consider computing

$$\Phi(\mathbf{y}_j) = \sum_{\substack{i=1 \\ i \neq j}}^N \frac{m_i}{\|\mathbf{x}_i - \mathbf{y}_j\|} \quad (1.3)$$

for some large number of target points  $\mathbf{y}_j$ . We will also consider the fields due to continuous distributions of mass,

$$\mathbf{E}(\mathbf{x}) = - \int m(\mathbf{y}) \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|^3} d\mathbf{y}. \quad (1.4)$$

A similar integral arises in magnetostatics, expressing the magnetic induction  $\mathbf{B}$  due to a steady-state current density  $\mathbf{J}$ . This is the Biot–Savart law (Jackson 1975)

$$\mathbf{B}(\mathbf{x}) = \frac{1}{c} \int \mathbf{J}(\mathbf{y}) \times \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|^3} d\mathbf{y}, \quad (1.5)$$

where  $c$  is the speed of light. Acoustic scattering processes give rise to  $N$ -body problems of the form

$$\Phi(\mathbf{x}_j) = \sum_{\substack{i=1 \\ i \neq j}}^N W_i \frac{e^{ikr_{ij}}}{r_{ij}}, \quad (1.6)$$

but they are beyond the scope of these lectures.

For an  $N$ -body problem in diffusion, consider the heat equation

$$u_t = \Delta u$$

with initial temperature distribution

$$u(\mathbf{x}, 0) = w(\mathbf{x}).$$

At any later time  $T > 0$ , the temperature field is well known to be given by

$$u(\mathbf{x}) = \frac{1}{\sqrt{(4\pi T)^{3/2}}} \int e^{-|\mathbf{x} - \mathbf{y}|^2/4T} w(\mathbf{y}) d\mathbf{y}. \quad (1.7)$$

A discrete analog of this is the  $N$ -body problem

$$u(\mathbf{x}_j) = \frac{1}{\sqrt{(4\pi T)^{3/2}}} \sum_{i=1}^N w_i e^{-r_{ij}^2/4T}. \quad (1.8)$$

Each of the preceding examples involves an integral of the form

$$u(\mathbf{x}) = \int K(\mathbf{x}, \mathbf{y}) w(\mathbf{y}) d\mathbf{y} \quad (1.9)$$

or a sum of the form

$$u(\mathbf{x}) = \sum_{i=1}^N w_i K(\mathbf{x}, \mathbf{y}_i). \quad (1.10)$$

Direct evaluation of such sums at  $N$  target points obviously requires  $O(N^2)$  operations, and algorithms which reduce the cost to  $O(N^\alpha)$  with  $1 \leq \alpha < 2$ ,  $O(N \log N)$ ,  $O(N \log^2 N)$ , etc. are referred to as *fast summation methods*. The most well known of these is certainly the Fast Fourier Transform (FFT), which computes

$$u_j = \sum_{k=1}^N e^{2\pi i j k / N} w_k$$

for  $j = 1, \dots, N$  in about  $5N \log N$  operations. From our point of view, the distinguishing features of the FFT are that it is exact, that it is based on considerations of symmetry (*algebra*), and that it is brittle. By the last point, we mean that it requires a uniform spatial grid to be applicable. Fast multipole methods (FMMs) are different. **They are approximate, based on analytic considerations, and robust (insensitive to source distribution).** In this category we include **fast multipole method for the Laplace equation** (Rokhlin [43]; Greengard and Rokhlin [29, 30, 32]; Carrier *et al.* [17]; Greengard [23]), the **fast Gauss transform** (Greengard and Strain [34]), and the **fast multipole method for the Helmholtz equation** (Rokhlin [44, 45]; Coifman *et al.* [18]). **Each one relies on a detailed analysis of the pairwise interaction, and each one permits rigorous *a priori* error bounds.**

There are a number of other, good algorithms for accelerating a variety of  $N$ -body calculations. We do not seek to review them here, and refer the reader to Alpert and Rokhlin [1]; Anderson [2, 3]; Appel [4]; Barnes and Hut [5]; Beylkin *et al.* [11]; Bradie *et al.* [13]; Brandt [14]; Brandt and Lubrecht [15]; Canning [16]; Greengard [25]; Hockney and Eastwood [36]; Odlyzko and Schönhage [42]; Van Dommelen and Rundensteiner [47]. Some of these are very broad in their applicability, while others, like FMMs, are intended for a specific kernel.

## 1.2 Degenerate kernels

Consider now a generic summation problem of the form (1.10), where the **kernel  $K(\mathbf{x}, \mathbf{y})$  can be expressed as a finite series**

$$K(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^p \phi_k(\mathbf{x}) \psi_k(\mathbf{y}). \quad (1.11)$$

Such kernels are called finite rank or degenerate kernels, and  $N$ -body problems governed by them are easily resolved. **First, one computes the moments**

$$A_k = \sum_{i=1}^N w_i \psi_k(\mathbf{y}_i).$$

**Second, one evaluates  $u(\mathbf{x})$  at each desired point** via the formula

$$u(\mathbf{x}) = \sum_{k=1}^p A_k \phi_k(\mathbf{x}).$$

The amount of work required is of the order  $O(Np)$ . While the  $N$ -body problems of mathematical physics are not of this type, the degenerate case serves as a useful model. Note that, independent of the details of the source distribution,  $u(\mathbf{x})$  must always be a linear combination of the functions  $\phi_1(\mathbf{x}), \dots, \phi_p(\mathbf{x})$ . In other words, the *dimension* of the function space where  $u(\mathbf{x})$  lives is much smaller than the  $N$ -dimensional space containing the data  $\{\mathbf{y}_i, w_i\}$  and there is a tremendous amount of compression in the transformation from the latter to the former. The notion of compression is fundamental to the fast Gauss transform and to the FMM for the Laplace equation, although not to the FMM for the Helmholtz equation.

## 2 Hierarchical and fast multipole methods in one dimension

Most **tree codes** and fast multipole methods are based on a judicious combination of the following key features.

- *A specified **acceptable accuracy of computation  $\epsilon$** .*
- *A **hierarchical subdivision of space** into panels, or clusters, of sources.*
- *A **far field expansion** of the “kernel”  $k(\mathbf{x}, \mathbf{y})$  in which the influence of source and evaluation points separates.*
- (Optional) *The conversion of far field expansions into local expansions.*

The first feature is very simple but absolutely crucial. **Once it is admitted that the result of a calculation is needed only to a certain accuracy, then approximations can be used.** This is a key to many fast methods. Actually, analogous remarks about the need to explicitly admit that a certain accuracy is sufficient can be made about other fast methods such as the multigrid method for solving partial differential equations.

Historically, hierarchical and fast multipole schemes were first developed in two and three dimensions, where they are of great practical importance. However, for pedagogical reasons, we will deviate from the historical order of development and first develop a scheme in a one variable setting. In this simpler setting the essential ideas of hierarchical and fast multipole

schemes stand out with great clarity, facilitating an understanding of the more complicated, and much more practically important, two and three dimensional schemes which will be discussed later.

## 2.1 Fast evaluation of multiquadrics

The one dimensional example we shall consider is the fast evaluation of one dimensional multiquadric radial basis functions. Consider the basic function

$$\phi(x) = \sqrt{x^2 + c^2} ,$$

where  $0 < c \leq h$ , and a corresponding spline, or multiquadric radial basis function,

$$s(\cdot) = \sum_{j=1}^N d_j \phi(\cdot - x_j) .$$

In this section we will develop a fast method of evaluating  $s(x)$  when the number of different evaluation points  $x$  is large.

The first step is to develop a series expansion of  $\phi(x-t)$  valid for “large” values of  $|x|$ . To this end consider the function

$$g(u) = (u-t) \sqrt{1 + \frac{c^2}{(u-t)^2}},$$

where  $u \in \mathbf{C}$  and  $\sqrt{\cdot}$  denotes the principal branch of the complex square root.  $g(u)$  coincides with  $\phi(u-t)$  for all real  $u$  greater than  $t$  and with  $-\phi(u-t)$  for all real  $u$  less than  $t$ . Now  $1 + \{c^2/(u-t)^2\}$  is both real and non-positive only for values of  $u$  on the line segment from  $t - ic$  to  $t + ic$ . Hence  $g$  is analytic in the region  $|u| > \sqrt{t^2 + c^2}$  and can be expanded in a Laurent series about zero valid for  $|u| > \sqrt{t^2 + c^2}$ . Computing this Laurent series yields the expansion

$$\begin{aligned} \phi(x-t) &= \sqrt{(x-t)^2 + c^2} \\ &= \text{sign}(x) \left\{ x-t + \frac{1}{2}c^2x^{-1} \right. \\ &\quad \left. + \frac{1}{2}tc^2x^{-2} + \frac{1}{8}(4t^2c^2 - c^4)x^{-3} \right. \\ &\quad \left. + \frac{1}{8}(4t^3c^2 - 3tc^4)x^{-4} + \cdots + q_p(c,t)x^{-p} + \cdots \right\} \end{aligned} \quad (2.1)$$

of  $\phi(x-t)$ . The corresponding truncated expansions are endowed with the error bound

$$\begin{aligned}
& \left| \phi(x-t) - \text{sign}(x) \left\{ x-t + \frac{1}{2}c^2x^{-1} + \cdots + q_p(c,t)x^{-p} \right\} \right| \\
& \leq 2(|t|+c) \left( \frac{\sqrt{t^2+c^2}}{|x|} \right)^{p+1} \frac{1}{1 - \frac{\sqrt{t^2+c^2}}{|x|}} \quad (2.2)
\end{aligned}$$

for  $|x| > \sqrt{t^2+c^2}$ .

Note that each term  $q_j(c,t)x^{-j}$  in the series (2.2) separates into the product of a coefficient depending on the source point alone, namely  $q_j(c,t)$ , and a homogeneous function depending on the evaluation point alone, namely  $x^{-j}$ . Thus the truncated far field expansions corresponding to different sources  $t = x_j$  in a panel  $T$  may be summed. The form of the series stays the same. Having several sources rather than only one merely changes the values of the coefficients of the homogeneous functions  $\{x^{-j}\}$ .

To be more precise we associate with a panel  $T$  that part of the spline  $s$  due to sources lying in the panel by setting

$$s_T(\cdot) = \sum_{j:x_j \in T} d_j \phi(\cdot - x_j).$$

Then, presuming for the moment that the panel is centered about the point  $x = 0$ , we approximate  $s_T$  by a far field expansion also centered at  $x = 0$ ,

$$r_T(x) = \text{sign}(x) \{ a_{-1}x + a_0 + a_1x^{-1} + \cdots + a_px^{-p} \}. \quad (2.3)$$

This series is obtained from the truncated expansion (2.2) for a single source at  $t$  with unit weight by choosing  $t = x_j$ , multiplying by  $d_j$ , and then summing over all sources  $x_j$  in the panel.

If the source panel is  $T = [-h, h)$ , and the geometry of the source and evaluation regions is as in Fig. 1 below, then in the evaluation region

$$\max_{j:x_j \in T} \frac{\sqrt{x_j^2 + c^2}}{|x|} \leq \frac{\sqrt{h^2 + h^2}}{|x|} \leq \frac{\sqrt{2}}{3} = \frac{1}{2.12\dots}.$$

Hence, using the estimate of equation (2.2)

$$|s_T(x) - r_T(x)| \leq D_T 4h \left( \frac{1}{2.12\dots} \right)^{p+1} \frac{1}{1 - \frac{1}{2.12\dots}} \quad (2.4)$$

for all  $x$  in the evaluation region, where  $D_T = \sum_{j:x_j \in T} |d_j|$ .

If the source panel is  $[t-h, t+h)$  centered at  $x = t$  rather than at  $x = 0$  then the appropriate form of the far field expansion is also centered

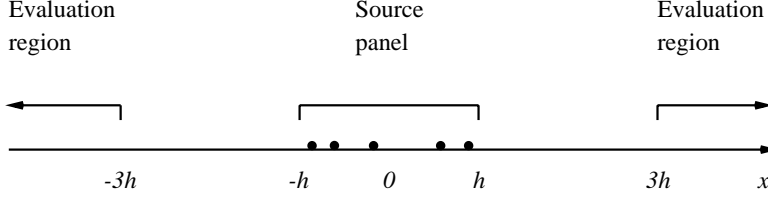


FIG. 1. Source and evaluation regions.

at  $x = t$ , and making the natural change of variable in (2.3) is

$$r_T(x) = \text{sign}(x - t) \left\{ a_{-1}(x - t) + a_0 + a_1(x - t)^{-1} + \cdots a_p(x - t)^{-p} \right\}. \quad (2.5)$$

It enjoys the error bound

$$|s_T(x) - r_T(x)| \leq D_T 4h \left( \frac{1}{2.12\dots} \right)^{p+1} \frac{1}{1 - \frac{1}{2.12\dots}} \quad (2.6)$$

whenever  $|x - t| \geq 3h$ , where  $t$  is the center of the source panel  $T$  and  $h$  is its radius. Thus the far field expansion,  $r_T$ , converges at a conveniently fast rate whenever  $x$  is separated from the panel  $T$  by at least the diameter of  $T$ . Such points  $x$  will be said to be **well separated** from  $T$ . A panel  $U$  will be said to be **well separated** from  $T$  if all of its points are well separated from  $T$ . Approximate evaluation of  $s_T(x)$  via the the far field approximation  $r_T(x)$  will be much quicker than direct evaluation whenever the number of terms in the series is much smaller than the number of sources in the panel  $T$ .

We are now ready to construct a fast evaluation scheme by combining the far field expansions with a hierarchical subdivision of space. The underlying idea is to summarize/approximate the influence of many sources by evaluating a few short series, thus reducing the cost in flops of a single extra evaluation from  $\mathcal{O}(N)$  to  $\mathcal{O}(\log N)$ , or even  $\mathcal{O}(1)$ .

Assume that the problem has been standardized so that all the source and evaluation points lie in the interval  $[0, 1]$ . We subdivide  $[0, 1]$  into a binary tree of panels as illustrated in Fig. 2 below.

At any point  $x$  we will approximate  $s(x)$  by summing the direct influence,  $s_T(x)$ , of the near neighbours of the childless panel containing  $x$ , together with the far field approximation  $r_T(x)$  to the influence of panels,  $T$ , far away from  $x$ . Since the underlying motive is to save floating point operations we always use the largest possible, that is the highest level, source panel for the far field series. For example, suppose the tree is as in Fig. 2 and is a uniform binary tree down to level 3. Then to approximately evaluate  $s(x)$  for  $x$  in the panel  $[0, \frac{1}{8}]$  we use

$$\begin{aligned}
s(x) \approx & s_{[0, \frac{1}{8})}(x) + s_{[\frac{1}{8}, \frac{1}{4})}(x) \\
& + r_{[\frac{1}{4}, \frac{3}{8})}(x) + r_{[\frac{3}{8}, \frac{1}{2})}(x) \\
& + r_{[\frac{1}{2}, \frac{3}{4})}(x) + r_{[\frac{3}{4}, 1]}(x).
\end{aligned}$$

Note in particular the use of the two level 2 approximations  $r_{[\frac{1}{2}, \frac{3}{4})}$  and  $r_{[\frac{3}{4}, 1]}$ , rather than the four level 3 approximations  $r_{[\frac{1}{2}, \frac{5}{8})}, \dots, r_{[\frac{7}{8}, 1]}$ . The use of these larger/higher level panels enables us to approximate the influence of the sources in the interval  $[\frac{1}{2}, 1]$  with two rather than four series evaluations: that is, it halves the flop count for this part of the evaluation task. Focusing on the use of  $r_{[\frac{1}{2}, \frac{3}{4})}(x)$  this is allowable since the panel  $[\frac{1}{2}, \frac{3}{4})$  is well separated from the panel  $[0, \frac{1}{4})$ , the parent of  $[0, \frac{1}{8})$ . Furthermore, the use of  $r_{[\frac{3}{4}, 1]}(x)$  is allowable since  $[\frac{3}{4}, 1]$  is well separated from  $[0, \frac{1}{4})$ .

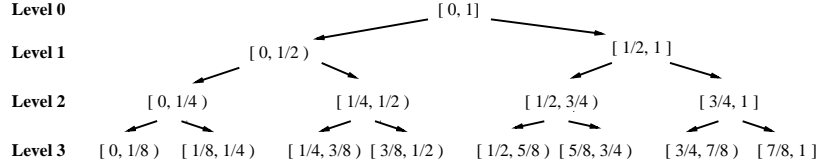


FIG. 2. Binary tree structure induced by a uniform subdivision of the unit interval.

Similarly, to approximately evaluate  $s(x)$  in the panel  $[\frac{1}{2}, \frac{5}{8})$  we would use

$$\begin{aligned}
s(x) \approx & s_{[\frac{3}{8}, \frac{1}{2})}(x) + s_{[\frac{1}{2}, \frac{5}{8})}(x) + s_{[\frac{5}{8}, \frac{3}{4})}(x) \\
& + r_{[\frac{1}{4}, \frac{3}{8})}(x) + r_{[\frac{3}{4}, \frac{7}{8})}(x) + r_{[\frac{7}{8}, 1]}(x) \\
& + r_{[0, \frac{1}{4})}(x).
\end{aligned}$$

In view of the error bound (2.6) the overall error in such a procedure will be bounded by  $\epsilon$  if the error in approximating  $\phi(x - t)$  at every level is bounded by  $\epsilon/||\mathbf{d}||_1$ .

This motivates a simple tree code for evaluating a univariate multi-quadratic.

#### A hierarchical code for evaluating a univariate mutiquadric radial basis function: setup part.

**Input:** the desired accuracy  $\epsilon$ ,  $0 < \epsilon < \frac{1}{2}$ , the source locations and weights  $\{x_j, d_j\}_{j=1}^N$ .

**Step 1:** Choose  $p \approx \lceil \log_{2.12}(\epsilon/||\mathbf{d}||_1) \rceil$ .



This choice guarantees that the desired accuracy in approximating  $s(x)$  will be attained.

**Step 2:**

*Choose the number of levels  $m \approx \log_2 N$ .*

If the data is approximately uniformly distributed this choice guarantees that each childless panel contains a number of sources which is bounded independently of  $N$ .

**Step 3:**

*Work down from the root panel,  $[0, 1]$ , subdividing each panel at levels  $0, \dots, m - 1$  in half, and assigning sources  $x_j$  to the panels containing them.*

The work here is one comparison for every source for each level from 0 through  $m - 1$  in order to assign the sources associated with each parent panel to the correct child panel.

**Step 4:**

*Work up the tree from level  $m$  to level 2, calculating the far field expansions associated with each panel.*

The work here is calculating the  $p + 2$  coefficients  $a_{-1}, a_0, a_1, \dots, a_p$  of the expansion  $r_T(\cdot)$ , for each panel  $T$ .

Once the setup phase has been completed we can use the far field approximations for fast evaluation in the manner of the examples immediately preceding the description of the setup phase. Given  $x$  one proceeds down the levels from level 2 to level  $m$ . At each level one uses the approximations from far away panels (those from which the panel containing  $x$  is well separated) and ignores source panels whose influence has already been incorporated at a higher level. No work is undertaken for near neighbours until the finest level. The method is of a divide and conquer type with near neighbours at level  $k - 1$  being split to yield smaller near neighbour panels and some source panels whose far field approximations can be used at level  $k$ . At the finest level the near neighbours are no longer to be split, and their influence is incorporated directly by adding the “direct” evaluations  $s_T(x)$ . In this way the largest possible source panels are always used for the far field approximations, and thus the flop count is minimized. A typical situation at an intermediate level  $k$  is shown in Fig. 3.

**A hierarchical code for evaluating a univariate multiquadric radial basis function: Evaluation part.**

The evaluation procedure can be expressed in recursive and non-recursive forms, both of which generalize easily to nonuniform partitions of the problem region.

**Recursive form of the evaluation part**

Given the root node  $R$  of the binary tree of panels and the point  $x$  at which

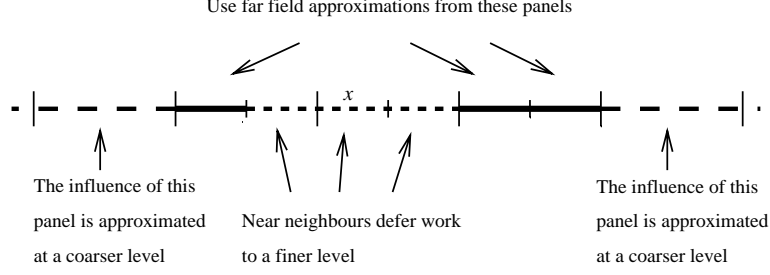


FIG. 3. A typical configuration of source panels at level  $k$  when evaluating at  $x$ .

$s(x)$  is to be evaluated, a call to routine *eval\_recur* specified below with the root node and  $x$  returns an approximation to  $s(x)$ .

In the pseudo-code to follow a C++ like notation is used. Thus the two children of a panel  $T$  are denoted by  $T->child[0]$  and  $T->child[1]$ , and remarks are indicated by a leading double slash, *//*.

```
eval_recur( $T, x$ ) {
  if ( $T$  is far away from  $x$ ) then
    // Approximate evaluation gives sufficient accuracy.
    // Perform it and terminate descent of this branch.
    return(  $r_T(x)$  )
  else if ( $T$  is childless)
    // Cannot descend further on this branch.
    // Evaluate the influence of  $T$  directly.
    return (  $s_T(x)$  )
  else
    // Descend the branch to a finer level in the
    // tree where approximations may be allowable.
    return ( eval_recur(  $T->child[0]$ ,  $x$  )
              + eval_recur(  $T->child[1]$ ,  $x$  ) )
  end if
}
```

We give an equivalent non-recursive expression of the evaluation phase below. This may be more appealing to some readers, and could give rise to more efficient implementations in some environments.

#### Nonrecursive form of the evaluation part

Given the root node  $R$  of the binary tree of panels and the point  $x$  at which  $s(x)$  is to be evaluated the following pseudo-code returns an approximation to the value of  $s(x)$  in the variable  $a$ .

##### Step1:

Initialize  $a$  to zero.

**Step 2:**

For  $k$  from 2 to  $m$ , add to  $a$  the far field approximation  $r_T(x)$  to the influence of each panel  $T$  well separated from  $x$ , which is not a subset of a panel already used at a coarser level (see Fig. 3).

**Step 3:**

Let  $Q$  be the bottom level (level  $m$ ) panel containing  $x$ . Add to  $a$  the direct evaluation  $s_T(x)$  of the influence of each near neighbour  $T$  of  $Q$ .

Note that in the case of a uniform subdivision deciding which panel a point lies in requires only a single multiplication by  $1/h$ . Also, if we work from level  $m$  backwards up to level 2 then the panel at level  $k-1$  containing  $x$  is the parent of the panel at level  $k$  containing  $x$ , so that even this single multiplication is avoidable except at the bottom-most level.

We are now ready to **estimate the computational cost of a single extra evaluation**, ignoring the cost of setup. The far field part of the work involves the evaluation of at most 3 far field expansions of  $p+2$  terms at each level from 2 to  $m$ . Hence the flop count arising from the evaluation of far field expansions is  $\mathcal{O}(mp)$ . The direct part of the evaluation is the direct evaluation of the influence of at most 3 bottom level panels. Since each bottom level panel contains only  $\mathcal{O}(1)$  sources the flop count arising from the direct part of the evaluation is  $\mathcal{O}(1)$ . Hence the incremental cost in flops of an additional evaluation is

$$\mathcal{O}(mp) \approx \mathcal{O}(|\log(\epsilon/\|\mathbf{d}\|_1)| \log N),$$

which is an order of magnitude faster than the  $\mathcal{O}(N)$  flop count for a single direct evaluation of  $s(x)$ .

**2.2 Enhancements**

The algorithm outlined above can be improved in several ways.

**First, it is clearly inefficient to split uniformly down to level  $m$  irrespective of the distribution of sources.** The panel splitting should be done in a more adaptive manner. For example subdividing a panel only when it contains a certain minimum number of sources. Also, once the sources have been allocated to children there is no reason to keep the child panels of uniform size. They can be shrunk to the minimum size necessary to contain all their sources. In this way the tree would be more refined in regions where there is a high density of sources, and finish after relatively few levels in a region where the density of sources is low.

**Second, the far field expansions are of the form**

$$r_T(x) = \text{sign}(x-t) \left( a_{-1}(x-t) + a_0 + a_1(x-t)^{-1} + \cdots + a_p(x-t)^{-p} \right).$$

Such a function is smooth away from  $t$ . Hence **it can be well approximated by a truncated Taylor series in any panel well separated from  $t$ .** Employing this idea instead of summing the values of Laurent series one can approx-

imate them with Taylor polynomials, and sum these polynomials, i.e. add corresponding coefficients. The result in the end is a polynomial associated with each panel,  $T$ , from level 2 to the bottom level, which approximates the influence of all sources in panels far away from  $T$ . Then approximate evaluation of  $s(x)$  involves the evaluation of one polynomial of order approximately  $p$ , plus the calculation of the direct influence of at most three panels each containing  $\mathcal{O}(1)$  sources. Therefore the incremental cost of a single extra evaluation is reduced from  $\mathcal{O}(|\log(\epsilon/\|\mathbf{d}\|_1)| \log N)$  flops to  $\mathcal{O}(|\log(\epsilon/\|\mathbf{d}\|_1)|)$  flops. There is a price to pay of course. This price is the extra work of converting Laurent expansions into local polynomial expansions, summing such polynomial expansion coefficients, and shifting such expansions to be centered at the center of the child panel. However, even for relatively moderate values of  $N$ , the fast multipole scheme described in this paragraph is quicker than the hierarchical scheme previously described. A variety of hierarchical and fast multipole methods for evaluating radial basis functions are described in the papers [6, 7, 8, 9].

### 3 The fast Gauss transform

Given a set of points  $\mathbf{x}_i = (x_i, y_i)$  and a set of source strengths  $w_i$ , the function

$$U(\mathbf{x}) = \sum_{i=1}^N w_i e^{-|\mathbf{x}-\mathbf{x}_i|^2/4T} \quad (3.1)$$

will be referred to as the (discrete) Gauss transform. This is an infinitely differentiable function and the derivatives decay rapidly in space for any  $T > 0$ . To be a bit more precise, we introduce the Hermite functions  $h_n(x)$ , defined by

$$h_n(x) = (-1)^n \frac{d^n}{dx^n} (e^{-x^2}).$$

Consider now a two-dimensional problem, where a source  $\mathbf{x}_i$  is located in a square with center  $\mathbf{c} = (c_1, c_2)$  and side length  $\sqrt{T}$ . Then the heat kernel

$$e^{-|\mathbf{x}-\mathbf{x}_i|^2/4T}$$

can be expressed as an Hermite series

$$e^{-|\mathbf{x}-\mathbf{x}_i|^2/4T} = \sum_{n_1, n_2=0}^{\infty} \Phi_{n_1 n_2}(\mathbf{x} - \mathbf{c}) \Psi_{n_1 n_2}(\mathbf{x}_i - \mathbf{c}), \quad (3.2)$$

where, for  $\mathbf{x} = (x, y)$ ,

$$\Psi_{n_1 n_2}(\mathbf{x}) = \frac{1}{n_1! n_2!} \left( \frac{x}{\sqrt{4T}} \right)^{n_1} \left( \frac{y}{\sqrt{4T}} \right)^{n_2},$$

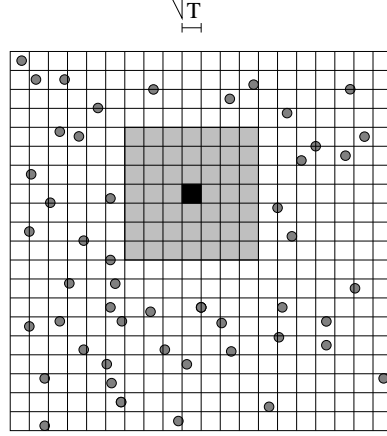


FIG. 4. The fast Gauss transform mesh.

and

$$\Phi_{n_1 n_2}(\mathbf{x}) = h_{n_1} \left( \frac{x}{\sqrt{4T}} \right) h_{n_2} \left( \frac{y}{\sqrt{4T}} \right).$$

One can also derive a precise estimate of the error in truncating the series (3.2) after a finite number of terms [34]:

$$\left| e^{-|\mathbf{x}-\mathbf{x}_i|^2/4T} - \sum_{n_1, n_2=0}^p \Phi_{n_1 n_2}(\mathbf{x}) \Psi_{n_1 n_2}(\mathbf{x}_i) \right| \leq \left( \frac{1}{p!} \right) \left( \frac{1}{8} \right)^p. \quad (3.3)$$

Such a rapidly decaying error makes the heat kernel very nearly degenerate. Suppose, for example, that one is satisfied with four digits of accuracy. Then  $p = 4$  is sufficient to guarantee it. For eight digits, choose  $p = 6$ ; for fourteen digits, choose  $p = 10$ ; and so on. **It turns out that (3.2) together with its error bound (3.3) is the only analytical tool needed to construct a fast Gauss transform. It remains only to organize the computation so that the finite series can be used effectively.**

We begin by assuming that all sources and targets lie in a square  $B_0$  of unit area on which we superimpose a uniform mesh which subdivides  $B_0$  into finer squares of side length  $\sqrt{T}$  (Fig. 4). [If  $T \geq 1$ ,  $B_0$  needs no further refinement.]

---

### A simple fast Gauss transform

#### Step 1:

Sort the  $N$  sources into the fine squares of the uniform mesh.

#### Step 2:

Choose  $p$  sufficiently large that the error estimate (3.3) is less than the

desired precision  $\epsilon$ .

**Step 3:**

For each fine square  $B$  with center  $c$ , compute the moments

$$A_{n_1 n_2} = \sum_{\mathbf{x}_j \in B} w_j \Psi_{n_1 n_2}(\mathbf{x}_j), \quad \text{for } n_1, n_2 \leq p.$$

*Operation count:*

Each source contributes to exactly one expansion, so that the amount of work required to form the moments for all nonempty boxes is proportional to  $Np^2$ .

**Step 4:** (Repeat for each target  $\mathbf{x}$ )

Find the box  $B$  in which point  $\mathbf{x}$  lies. Since the heat kernel decays exponentially fast in space, we need only consider the influence of certain near neighbours in our decomposition of  $B_0$  (the shaded region in Fig. 4). To be more precise, ignoring all but the nearest  $(2n+1)^2$  boxes incurs an error of the order  $e^{-n^2/4}$ . For  $n=6$ , this is approximately  $10^{-4}$ . The influence of the sources contained in each of these nearby boxes can be obtained by evaluating the expression

$$\sum_{n_1, n_2=0}^p A_{n_1 n_2}^j \Phi_{n_1 n_2}(\mathbf{x} - \mathbf{c}_j),$$

where  $\{A_{n_1 n_2}^j\}$  are the precomputed moments for neighbour  $j$  and  $\mathbf{c}_j$  is its center.

*Operation count:*

The amount of work required is of the order  $(2n+1)^2 p^2$  for each target, which is a constant. The total amount of work, therefore, is proportional to  $N+M$  where  $N$  is the number of sources and  $M$  is the number of targets.

It is worth considering two extreme regimes. For very small times, say  $T = 10^{-10}$ , the effect of each heat source is vanishingly small beyond a distance of approximately  $10^{-5}$ . Assuming the sources and targets are widely distributed, the question of rapidly computing sums of the form (3.1) becomes substantially one of sorting and finding near neighbours. For large times, say  $T = 1$ , then all sources have nonnegligible interactions, but only one box is constructed, one set of moments is computed, and one expansion is evaluated for each target. As time marches forward, there is less and less information content in the temperature field  $U(\mathbf{x})$  as would be expected from a diffusion process.

For a more complex but more efficient version of the method, we refer the reader to the original paper by Greengard and Strain [34]. For 100,000 sources and targets, the algorithm is about three orders of magnitude faster

than direct summation. Finally, note that the algorithm performs *worst* when sources are uniformly distributed. That case provides an upper bound on the number of expansions which need to be evaluated at each target position. In other words, the more clustering present in the source distribution, the better. This behavior is atypical of standard numerical methods based, for example, on finite difference equation solvers.

## 4 The FMM in two dimensions

The evaluation of gravitational or Coulombic interactions is somewhat more complicated than the evaluation of interactions governed by the heat kernel. The force is long-ranged and nonsmooth (at least locally). It requires a new set of tools for organizing the computation. Appel [4], Barnes and Hut [5], and others working in the astrophysics community developed what have come to be known as “tree codes” in order to overcome the computational obstacle presented by this  $N$ -body problem. They are based on the observation that at some distance from the sources, the gravitational field is smooth and should be representable in some compressed form.

This lecture begins with the two-dimensional case, since both the analytic and computational aspects of the problem are easier to understand. We will use the language of electrostatics, and assume it known that a two-dimensional point charge located at  $\mathbf{x}_0 = (x_0, y_0) \in \mathbf{R}^2$  gives rise to potential and electrostatic fields at any location  $\mathbf{x} = (x, y) \neq \mathbf{x}_0$  of the form

$$\phi_{\mathbf{x}_0}(x, y) = -\log(\|\mathbf{x} - \mathbf{x}_0\|) \quad (4.1)$$

and

$$E_{\mathbf{x}_0}(x, y) = \frac{(\mathbf{x} - \mathbf{x}_0)}{\|\mathbf{x} - \mathbf{x}_0\|^2} \quad (4.2)$$

respectively. Away from source points, the potential  $\phi$  is harmonic, that is, it satisfies the Laplace equation

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0. \quad (4.3)$$

Since for every harmonic function  $u$ , there exists an analytic function  $w$  for which  $u = \text{Re}(w)$ , we will use complex analysis to simplify notation. Equating  $(x, y)$  with the complex point  $z$ , we note that

$$\phi_{\mathbf{x}_0}(\mathbf{x}) = \text{Re}(-\log(z - z_0)), \quad (4.4)$$

and will refer to the analytic function  $\log(z)$  as the potential due to a charge. We will continue to use analytic expressions for the potentials due to more complicated charge distributions, even though we are only interested in their real part. For future reference, we note that we can extract the electrostatic field from the complex potential.

**Lemma 4.1** *If  $u = \operatorname{Re}(w)$  describes the potential field at  $(x, y)$ , then the corresponding force field is given by*

$$\nabla u = (u_x, u_y) = (\operatorname{Re}(w'), -\operatorname{Im}(w')), \quad (4.5)$$

where  $w'$  is the derivative of  $w$ .

Consider now a point charge of strength  $q$ , located at  $z_0$ . A straightforward calculation shows that for any  $z$  with  $|z| > |z_0|$ ,

$$\phi_{z_0}(z) = q \log(z - z_0) = q \left( \log(z) - \sum_{k=1}^{\infty} \frac{1}{k} \left( \frac{z_0}{z} \right)^k \right). \quad (4.6)$$

This provides us with a means of computing the multipole expansion due to a collection of charges (Rokhlin [43]; Greengard and Rokhlin [29]).

**Lemma 4.2 (Multipole expansion)** *Suppose that  $m$  charges of strengths  $\{q_i, i = 1, \dots, m\}$  are located at points  $\{z_i, i = 1, \dots, m\}$ , with  $|z_i| < r$ . Then for any  $z$  with  $|z| > r$ , the potential  $\phi(z)$  induced by the charges is given by*

$$\phi(z) = Q \log(z) + \sum_{k=1}^{\infty} \frac{a_k}{z^k}, \quad (4.7)$$

where

$$Q = \sum_{i=1}^m q_i \quad \text{and} \quad a_k = \sum_{i=1}^m \frac{-q_i z_i^k}{k}. \quad (4.8)$$

Furthermore, for any  $p \geq 1$ ,

$$\begin{aligned} \left| \phi(z) - Q \log(z) - \sum_{k=1}^p \frac{a_k}{z^k} \right| &\leq \frac{1}{p+1} \alpha \left| \frac{r}{z} \right|^{p+1} \\ &\leq \left( \frac{A}{p+1} \right) \left( \frac{1}{c-1} \right) \left( \frac{1}{c} \right)^p, \end{aligned} \quad (4.9)$$

where

$$c = \left| \frac{z}{r} \right|, \quad A = \sum_{i=1}^m |q_i|, \quad \text{and} \quad \alpha = \frac{A}{1 - \left| \frac{r}{z} \right|}. \quad (4.10)$$

**Proof** The form of the multipole expansion follows immediately from the lemma above by summing over the  $m$  expansions corresponding to the  $m$  sources  $z_i$ . To see the error bound observe that

$$\left| \phi(z) - Q \log(z) - \sum_{k=1}^p \frac{a_k}{z^k} \right| \leq \left| \sum_{k=p+1}^{\infty} \frac{a_k}{z^k} \right|.$$



Substituting expression (4.8) for  $a_k$  into the above

$$\begin{aligned} \left| \sum_{k=p+1}^{\infty} \frac{a_k}{z^k} \right| &\leq A \sum_{k=p+1}^{\infty} \frac{r^k}{k|z|^k} \leq \frac{A}{p+1} \sum_{k=p+1}^{\infty} \left| \frac{r}{z} \right|^k \\ &= \frac{\alpha}{p+1} \left| \frac{r}{z} \right|^{p+1} = \left( \frac{A}{p+1} \right) \left( \frac{1}{c-1} \right) \left( \frac{1}{c} \right)^p. \end{aligned}$$

which is the required result.  $\square$

Note that if  $c \geq 2$ , then

$$\left| \phi(z) - Q \log(z) - \sum_{k=1}^p \frac{a_k}{z^k} \right| \leq A \left( \frac{1}{2} \right)^p. \quad (4.11)$$

The reader may recall that in the fast Gauss transform, the error bound (3.3) was independent of target location. Here, the amount of *compression* achieved depends both on the desired precision and the distance of the target from the sources.

The ability to form multipole expansions and compute error bounds via (4.9) is all that is required for an  $O(N \log N)$  fast summation algorithm. It must be combined, however, with a recursive “divide and conquer” strategy.

#### 4.1 The $N \log N$ algorithm

To simplify the number of issues addressed, let us assume for the moment that the particles are fairly homogeneously distributed in a square. In order to make systematic use of multipole expansions, we introduce a hierarchy of boxes which refine the computational domain into smaller and smaller regions. At refinement level 0, we have the entire computational domain. Refinement level  $l+1$  is obtained recursively from level  $l$  by subdivision of each box into four equal parts. This yields a natural tree structure, where the four boxes at level  $l+1$  obtained by subdivision of a box at level  $l$  are considered its children.

**Definition 4.3** Two boxes are said to be **near neighbours** if they are at the same refinement level and share a boundary point. (A box is a near neighbour of itself.)

**Definition 4.4** Two boxes are said to be **well separated** if they are at the same refinement level and are not near neighbours.

**Definition 4.5** With each box  $i$  is associated an **interaction list**, consisting of the children of the near neighbours of  $i$ 's parent which are well separated from box  $i$  (Fig. 4).

The basic idea is to consider clusters of particles at successive levels of spatial refinement, and to compute interactions between distant clusters by means of multipole expansions when possible. It is clear that at levels

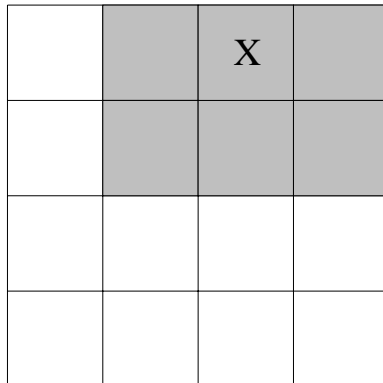


FIG. 5. The first step of the algorithm. Interactions between particles in box X and the white boxes can be computed via multipole expansions. Interactions with near neighbours (grey) are not computed.

0 and 1, there are no pairs of boxes which are well separated. At level 2, on the other hand, sixteen boxes have been created and there are a number of well separated pairs. Multipole expansions can then be used to compute interactions between these well separated pairs (Fig. 5) with rigorous bounds on the error. **In fact, the bound (4.11) is valid so that given a precision  $\epsilon$ , we need to use  $p = \log_2(1/\epsilon)$  terms.**

It remains to compute the interactions between particles contained in each box with those contained in the box's near neighbours, and this is where recursion enters the picture. After each level 2 box is refined to create level 3, we seek to determine which other boxes can be interacted with by means of multipole expansions. But notice that those boxes outside the region of the parent's nearest neighbours are already accounted for (at level 2) and that interactions with current near neighbours cannot accurately be computed by means of an expansion. The remaining boxes correspond exactly to the interaction list defined above (Fig. 6).

The nature of the recursion is now clear. At every level, the multipole expansion is formed for each box due to the particles it contains. The resulting expansion is then evaluated for each particle in the region covered by its interaction list (Fig. 4).

We halt the recursive process after roughly  $\log N$  levels of refinement. The amount of work done at each level is of the order  $O(N)$ . To see this, note first that approximately  $Np$  operations are needed to create all expansions, since each particle contributes to  $p$  expansion coefficients. Secondly, from the point of view of a single particle, there are at most 27 boxes (the maximum size of the interaction list) whose expansions are computed, so that  $27Np$  operations are needed for all evaluations.

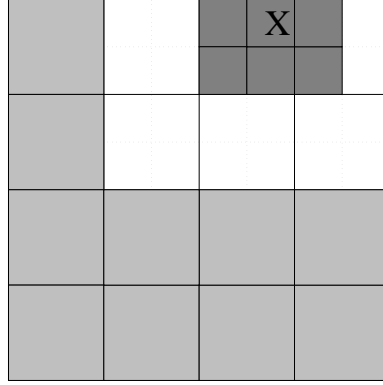


FIG. 6. The second step of the algorithm. After refinement, note that the particles in the box marked X have already interacted with the most distant particles (light grey). They are now well separated from the particles in the white boxes, so that these interactions can be computed via multipole expansions. The near neighbour interactions (dark grey) are not computed.

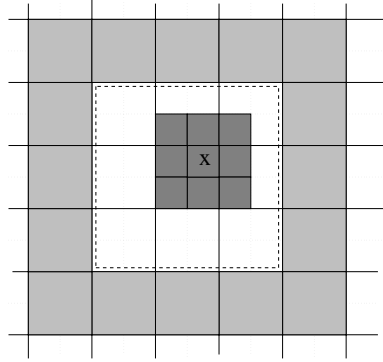


FIG. 7. Subsequent steps of the algorithm. The interaction list for box X is indicated in white.

At the finest level, we have created roughly  $4^{\log_4 N} = N$  boxes and it remains only to compute interactions between nearest neighbours. By the assumption of homogeneity, there are  $O(1)$  particles per box, so that this last step requires about  $8N$  operations (Fig. 5). The total cost is approximately

$$28Np \log N + 8N. \quad (4.12)$$

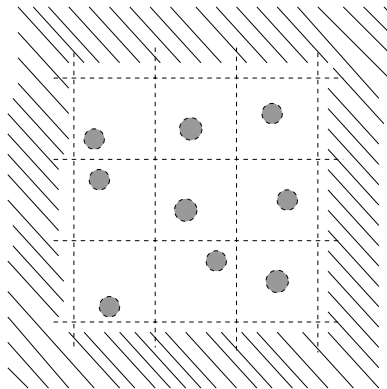


FIG. 8. At the finest level, interactions with near neighbours are computed directly.

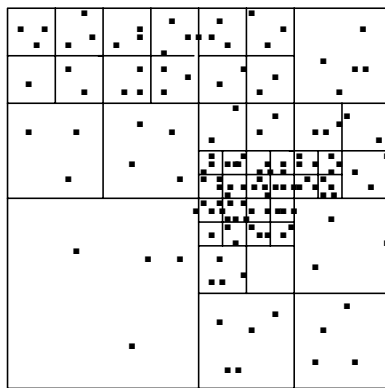


FIG. 9. An adaptive data structure

#### 4.2 The adaptive algorithm

When the distribution of particles is nonuniform, a somewhat different strategy must be employed. One option is that during the refinement process, each box is examined to determine whether it contains any particles. If so, it is subdivided further. If not, it is pruned from the tree structure and ignored at subsequent levels (Fig. 9). The complexity of this adaptive algorithm is harder to state precisely, since it depends on the total number of refinement levels which is not determined a priori. For distributions of practical interest, this turns out to be proportional to  $\log N$ . Note that if the interparticle spacing collapses as  $N^{-p}$ , where  $p$  is independent of  $N$ , then  $p \cdot \log N$  levels are needed. It is, therefore, quite reasonable to refer to the adaptive algorithm as also being of the order  $N \log N$ .

The algorithm just described is similar to that of Van Dommelen and

Rundensteiner [47].

### 4.3 The FMM

In order to develop an  $O(N)$  method, we need several further analytic results concerning multipole expansions. Lemma 4.6 provides a formula for shifting the center of a multipole expansion, Lemma 4.7 describes how to convert such an expansion into a local (Taylor) expansion in a circular region of analyticity, and Lemma 4.8 furnishes a mechanism for shifting the center of a Taylor expansion within a region of analyticity. We state the second result without complete proof. A complete proof can be found in Greengard and Rokhlin [29] and Greengard [23].

**Lemma 4.6 (Translation of a multipole expansion)** *Suppose that*

$$\phi(z) = a_0 \log(z - z_0) + \sum_{k=1}^{\infty} \frac{a_k}{(z - z_0)^k} \quad (4.13)$$

*is a multipole expansion of the potential due to a set of  $m$  charges of strengths  $q_1, q_2, \dots, q_m$ , all of which are located inside the circle  $D$  of radius  $R$  with center at  $z_0$ . Then for  $z$  outside the circle  $D_1$  of radius  $(R + |z_0|)$  and center at the origin,*

$$\phi(z) = a_0 \log(z) + \sum_{l=1}^{\infty} \frac{b_l}{z^l}, \quad (4.14)$$

where

$$b_l = -\frac{a_0 z_0^l}{l} + \sum_{k=1}^l a_k z_0^{l-k} \binom{l-1}{k-1}, \quad (4.15)$$

with  $\binom{l}{k}$  the binomial coefficients. Furthermore, for any  $p \geq 1$ ,

$$\left| \phi(z) - a_0 \log(z) - \sum_{l=1}^p \frac{b_l}{z^l} \right| \leq \left( \frac{A}{1 - \frac{|z_0|+R}{|z|}} \right) \left| \frac{|z_0|+R}{z} \right|^{p+1} \quad (4.16)$$

with  $A$  defined in (4.10).

**Proof** The form of the translated multipole expansion follows from the two equations

$$\log(z - z_0) = \log\left(z \left(1 - \frac{z_0}{z}\right)\right) = \log(z) - \sum_{\ell=1}^{\infty} \frac{1}{\ell} \left(\frac{z_0}{z}\right)^{\ell},$$

and

$$(z - z_0)^{-k} = \sum_{\ell=k}^{\infty} \binom{\ell-1}{k-1} \frac{z_0^{\ell-k}}{z^{\ell}},$$

both valid for  $|z| > |z_0|$ , by substituting, summing, and truncating. The error bound follows from the uniqueness of the multipole expansion, which implies that the expansion about the origin obtained indirectly above must be identical with the one obtained directly. Therefore its truncations enjoy the error bound of the direct expansion, namely (4.9).  $\square$

**Lemma 4.7 (Conversion of a multipole expansion into a local expansion)** *Suppose that  $m$  charges of strengths  $q_1, q_2, \dots, q_m$  are located inside the circle  $D_1$  with radius  $R$  and center at  $z_0$ , and that  $|z_0| > (c+1)R$  with  $c > 1$ . Then the corresponding multipole expansion (4.13) converges inside the circle  $D_2$  of radius  $R$  centered about the origin. Inside  $D_2$ , the potential due to the charges is described by a power series:*

$$\phi(z) = \sum_{l=0}^{\infty} b_l \cdot z^l, \quad (4.17)$$

where

$$b_0 = a_0 \log(-z_0) + \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} (-1)^k, \quad (4.18)$$

and

$$b_l = -\frac{a_0}{l \cdot z_0^l} + \frac{1}{z_0^l} \sum_{k=1}^{\infty} \frac{a_k}{z_0^k} \binom{l+k-1}{k-1} (-1)^k, \quad \text{for } l \geq 1. \quad (4.19)$$

Furthermore, an error bound for the truncated series is given by

$$\left| \phi(z) - \sum_{l=0}^p b_l \cdot z^l \right| < \frac{A(4e(p+c)(c+1) + c^2)}{c(c-1)} \left( \frac{1}{c} \right)^{p+1}, \quad (4.20)$$

where  $A$  is defined in (4.10) and  $e$  is the base of natural logarithms.

The form of the local series follows from substituting the expressions

$$\begin{aligned} \log(z - z_0) &= \log \left( -z_0 \left( 1 - \frac{z}{z_0} \right) \right) \\ &= \log(-z_0) - \sum_{\ell=1}^{\infty} \frac{1}{\ell} \left( \frac{z}{z_0} \right)^{\ell}, \end{aligned}$$

and

$$\begin{aligned} (z - z_0)^{-k} &= \left( \frac{1}{-z_0} \right)^k \left( \frac{1}{1 - \frac{z}{z_0}} \right)^k \\ &= \left( \frac{1}{-z_0} \right)^k \sum_{\ell=0}^{\infty} \binom{\ell+k-1}{k-1} \left( \frac{z}{z_0} \right)^{\ell}, \end{aligned}$$

both valid for  $|z| \leq R$ , into the multipole expansion (4.13). The proof of the error bound may be found in Greengard and Rokhlin [29] or Greengard [23].

**Lemma 4.8 (Translation of a local expansion)** *Translation of a complex polynomial centered about  $z_0$*

$$\sum_{k=0}^p a_k (z - z_0)^k, \quad (4.21)$$

*into a complex polynomial centered about 0*

$$\sum_{k=0}^n b_k z^k, \quad (4.22)$$

*can be achieved by the complete Horner scheme*

```

for j from 0 to p - 1 do
  for k from p - j - 1 to p - 1 do
     $a_k := a_k - z_0 a_{k+1}$ 
  end
end
```

*which given the vector of coefficients  $\mathbf{a}$  overwrites it with  $\mathbf{b}$ .*

**Proof** This scheme is essentially nested multiplication and can be derived from the relation

$$\begin{aligned} & (a_m z^j + a_{m-1} z^{j-1} + \cdots + a_{m-j}) (z - z_0) + a_{m-j-1} \\ = & a_m z^{j+1} + \tilde{a}_{m-1} z^j + \cdots + \tilde{a}_{m-j-1}, \end{aligned}$$

which corresponds to an intermediate stage in converting expression (4.21) into expression (4.22) by performing nested multiplication.  $\square$

**Observation 1:** Suppose that we have created the multipole expansion for each of the four children of some box in the mesh hierarchy. We would like to form a single expansion for the parent box without re-examining each particle. Lemma 4.6 provides just such a mechanism at the cost of  $4p^2$  operations, since each shift requires  $p^2$  work.

**Observation 2:** Suppose that we have a local expansion for a box  $b$  which describes the field induced by all particles outside  $b$ 's nearest neighbours. We would like to transmit this information to  $b$ 's children. Lemma 4.8 provides just such a mechanism at the cost of  $4p^2$  operations, since each shift requires  $p^2$  work.

Note now that the  $N \log N$  scheme was of that complexity because all particles were “accessed” at every level of refinement. The analytic ma-

nipulations just described allow us to avoid this, and their incorporation into the  $N \log N$  scheme results in the FMM. Here, we simply indicate the modifications which lead to its implementation.

### Initialization

Choose a number of levels so that there are, on average,  $s$  particles per box at the finest level. (The number of boxes is then approximately  $N/s$ .)

### Upward Pass

In the  $N \log N$  scheme, we proceeded from the coarsest to the finest level, forming multipole expansions for every box. In the FMM, we begin at the finest level, and create multipole expansions from the source positions and strengths. The expansions for all boxes at all higher levels are then formed by the merging procedure delineated in Observation 1.

### Downward Pass

In the  $N \log N$  scheme, whenever a box  $b$  was under consideration, we used its multipole expansion to compute interactions with all particles contained in the boxes of  $b$ 's interaction list. In the FMM, we *convert* the multipole expansion into a local expansion about the centers of all boxes in  $b$ 's interaction list, using Lemma 4.7.

After these calculations are completed, we are left with a local expansion in each box at each level. Beginning at the coarsest level, these local expansions are *shifted* to the children's level and added to the children's local expansions, as described in Observation 2. After this recursive process reaches the finest refinement level, a local expansion will have been created for each box which describes the field due to all particles outside the box's near neighbours. It is only this expansion which is evaluated. The near neighbour interactions, as before, are computed directly.

The total operation count is approximately

$$Np + 29 \left( \frac{N}{s} \right) p^2 + Np + 9Ns.$$

The terms correspond to formation of the multipole expansions, shifting the expansions, evaluation of the local expansions and computation of the near neighbour interactions.

Choosing  $s = p$ , this yields

$$40Np,$$



which compares favorably with the estimate (4.12) even for modest  $N$ . This is not the end of the story, however. The richer analytic structure of the FMM permits a large number of modifications and optimizations which are not available to other hierarchical schemes. These have to do with the use of symmetry relations to reduce the number of shifts (Greengard and Rokhlin [30]; Wang and LeSar [49]) as well as schemes which reduce the cost of translation itself (Greengard and Rokhlin [31]; Hrycak and Rokhlin [37]; Elliott and Board [21]). We will discuss these variants in greater detail when we come to the three-dimensional algorithm.

As for the adaptive FMM, a rudimentary complexity analysis is given by Carrier *et al.* [17]. A more complete analysis is given by Nabors *et al.* [41].

Typical timings for high precision FMM calculations are shown in the following table, taken from Hrycak and Rokhlin [37].

$N$	$T_{HR}$	$T_{CGR}$	$T_{dir}$	$E_{HR}$	$E_{CGR}$	$E_{dir}$
400	0.5	1.4	1.0	$3.5 \cdot 10^{-13}$	$3.5 \cdot 10^{-13}$	$3.0 \cdot 10^{-13}$
1600	2.3	5.2	17.1	$1.1 \cdot 10^{-12}$	$1.1 \cdot 10^{-12}$	$1.1 \cdot 10^{-12}$
6400	8.7	25.8	285.2	$4.9 \cdot 10^{-13}$	$4.9 \cdot 10^{-13}$	$2.7 \cdot 10^{-12}$
25600	34.7	110.6	(4400)	$2.5 \cdot 10^{-12}$	$2.5 \cdot 10^{-12}$	—

Here  $T_{HR}$  denotes the timing for the Hrycak and Rokhlin method,  $T_{CGR}$  denotes the timing for the Carrier *et al.* method and  $T_{dir}$  the timing for direct evaluation; the  $E$ 's are the corresponding errors.

## 5 The FMM in three dimensions

In three dimensions, as in two dimensions, functions which satisfy the Laplace equation

$$\nabla^2 \Phi = \frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} + \frac{\partial^2 \Phi}{\partial z^2} = 0 \quad (5.1)$$

are referred to as harmonic functions. The theory of such functions is generally called potential theory, and is beautifully described in the classic text by Kellogg [39]. A shorter description is available in the text by Wallace [48] and the first part of this lecture is abstracted from Greengard [23].

Assume now that the task at hand is to compute the Coulomb interactions in a large particle system. If a point charge of strength  $q$  is located at  $P_0 = (x_0, y_0, z_0)$ , then the potential and electrostatic field due to this charge at a distinct point  $P = (x, y, z)$  are given by

$$\Phi = \frac{1}{\tilde{r}} \quad (5.2)$$

and

$$\vec{E} = -\nabla \Phi = \left( \frac{x - x_0}{\tilde{r}^3}, \frac{y - y_0}{\tilde{r}^3}, \frac{z - z_0}{\tilde{r}^3} \right), \quad (5.3)$$

respectively, where  $\tilde{r}$  denotes the distance between points  $P_0$  and  $P$ .

We would like to derive a series expansion for the potential at  $P$  in terms of its distance  $r$  from the origin. For this, let the spherical coordinates of  $P$  be  $(r, \theta, \phi)$  and of  $P_0$  be  $(\rho, \alpha, \beta)$ . Letting  $\gamma$  be the angle between the vectors  $P$  and  $P_0$ , we have from the law of cosines

$$\tilde{r}^2 = r^2 + \rho^2 - 2r\rho \cos \gamma, \quad (5.4)$$

with

$$\cos \gamma = \cos \theta \cos \alpha + \sin \theta \sin \alpha \cos(\phi - \beta). \quad (5.5)$$

Thus,

$$\frac{1}{\tilde{r}} = \frac{1}{r\sqrt{1 - 2\frac{\rho}{r}\cos\gamma + \frac{\rho^2}{r^2}}} = \frac{1}{r\sqrt{1 - 2u\mu + \mu^2}}, \quad (5.6)$$

having set

$$\mu = \frac{\rho}{r} \quad \text{and} \quad u = \cos \gamma. \quad (5.7)$$

For  $\mu < 1$ , we may expand the inverse square root in powers of  $\mu$ , resulting in a series of the form

$$\frac{1}{\sqrt{1 - 2u\mu + \mu^2}} = \sum_{n=0}^{\infty} P_n(u)\mu^n \quad (5.8)$$

where

$$P_0(u) = 1, \quad P_1(u) = u, \quad P_2(u) = \frac{3}{2}(u^2 - \frac{1}{3}), \dots \quad (5.9)$$

and, in general,  $P_n(u)$  is the Legendre polynomial of degree  $n$ . Our expression for the field now takes the form

$$\frac{1}{\tilde{r}} = \sum_{n=0}^{\infty} \frac{\rho^n}{r^{n+1}} P_n(u). \quad (5.10)$$

The formula (5.10), unfortunately, does not meet our needs. The parameter  $u$  depends on both the source and the target locations. We cannot use it to expand the field due to a large number of sources. A more general representation will require the introduction of spherical harmonics.

For this, we first write the Laplace equation in spherical coordinates:

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial \Phi}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial \Phi}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 \Phi}{\partial \phi^2} = 0. \quad (5.11)$$

The standard solution of this equation by separation of variables results in an expression for the field as a series, the terms of which are known as spherical harmonics:

$$\Phi = \sum_{n=0}^{\infty} \sum_{m=-n}^n \left( L_n^m r^n + \frac{M_n^m}{r^{n+1}} \right) Y_n^m(\theta, \phi). \quad (5.12)$$

The terms  $Y_n^m(\theta, \phi)r^n$  are referred to as spherical harmonics of degree  $n$ , the terms  $Y_n^m(\theta, \phi)/r^{n+1}$  are called spherical harmonics of degree  $-n-1$ , and the coefficients  $L_n^m$  and  $M_n^m$  are known as the moments of the expansion.

The spherical harmonics can be expressed in terms of partial derivatives of  $1/r$ , but we will simply define them via the relation

$$Y_n^m(\theta, \phi) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} \cdot P_n^{|m|}(\cos \theta) e^{im\phi}, \quad (5.13)$$

omitting a normalization factor of  $\sqrt{(2n+1)/4\pi}$ .

The special functions  $P_n^m$  are called associated Legendre functions and can be defined by the Rodrigues' formula

$$P_n^m(x) = (-1)^m (1-x^2)^{m/2} \frac{d^m}{dx^m} P_n(x).$$

### 5.1 The multipole expansion

We will need a well known result from the theory of spherical harmonics.

**Theorem 5.1 (Addition theorem for Legendre polynomials)** *Let  $P$  and  $Q$  be points with spherical coordinates  $(r, \theta, \phi)$  and  $(\rho, \alpha, \beta)$ , respectively, and let  $\gamma$  be the angle subtended between them. Then*

$$P_n(\cos \gamma) = \sum_{m=-n}^n Y_n^{-m}(\alpha, \beta) \cdot Y_n^m(\theta, \phi). \quad (5.14)$$

**Theorem 5.2 (Multipole expansion)** *Suppose that  $k$  charges of strengths  $\{q_i, i = 1, \dots, k\}$  are located at the points  $\{Q_i = (\rho_i, \alpha_i, \beta_i), i = 1, \dots, k\}$ , with  $|\rho_i| < a$ . Then for any  $P = (r, \theta, \phi) \in \mathbf{R}^3$  with  $r > a$ , the potential  $\phi(P)$  is given by*

$$\phi(P) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \frac{M_n^m}{r^{n+1}} \cdot Y_n^m(\theta, \phi), \quad (5.15)$$

where

$$M_n^m = \sum_{i=1}^k q_i \cdot \rho_i^n \cdot Y_n^{-m}(\alpha_i, \beta_i). \quad (5.16)$$

Furthermore, for any  $p \geq 1$ ,

$$\left| \phi(P) - \sum_{n=0}^p \sum_{m=-n}^n \frac{M_n^m}{r^{n+1}} \cdot Y_n^m(\theta, \phi) \right| \leq \frac{A}{r-a} \left( \frac{a}{r} \right)^{p+1}, \quad (5.17)$$

where

$$A = \sum_{i=1}^k |q_i|. \quad (5.18)$$

## 5.2 The $N \log N$ scheme

As in the two-dimensional case, Theorem 5.2 is all that is required to construct an  $N \log N$  scheme of arbitrary precision. Since we discussed that case in some detail in a previous lecture, we simply indicate the principal differences.

- The particles are contained within a cube which is subdivided in a recursive manner. In the nonadaptive case, each box has *eight* children, 27 nearest neighbours and an interaction list of dimension 189.
- A multipole expansion is made up of  $p^2$  terms rather than  $p$  terms. Furthermore, a simple geometric calculation shows that for a target in a well-separated cube, the rate of decay of the error with  $p$  looks like  $(\sqrt{3}/3)^p$  rather than  $(1/2)^p$  or  $(\sqrt{2}/3)^p$ .

Thus, the cost of the  $N \log N$  scheme is approximately

$$189N p^2 \log_8 N + 27N, \quad (5.19)$$

where  $p = \log_{\sqrt{3}}(1/\epsilon)$ .

## 5.3 The $O(N)$ scheme

In order to accelerate the preceding scheme, we need some additional analytic machinery to shift multipole expansions and obtain local representations.

**Theorem 5.3 (Translation of a multipole expansion)** *Suppose that  $l$  charges of strengths  $q_1, q_2, \dots, q_l$  are located inside the sphere  $D$  of radius  $a$  with center at  $Q = (\rho, \alpha, \beta)$ , and that for points  $P = (r, \theta, \phi)$  outside  $D$ , the potential due to these charges is given by the multipole expansion*

$$\Phi(P) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \frac{O_n^m}{r'^{n+1}} \cdot Y_n^m(\theta', \phi'), \quad (5.20)$$

where  $P - Q = (r', \theta', \phi')$ . Then for any point  $P = (r, \theta, \phi)$  outside the

sphere  $D_1$  of radius  $(a + \rho)$ ,

$$\Phi(P) = \sum_{j=0}^{\infty} \sum_{k=-j}^j \frac{M_j^k}{r^{j+1}} \cdot Y_j^k(\theta, \phi) , \quad (5.21)$$

where

$$M_j^k = \sum_{n=0}^j \sum_{m=-n}^n \frac{O_{j-n}^{k-m} \cdot i^{|k|-|m|-|k-m|} \cdot A_n^m \cdot A_{j-n}^{k-m} \cdot \rho^n \cdot Y_n^{-m}(\alpha, \beta)}{A_j^k} , \quad (5.22)$$

with  $A_n^m$  defined by

$$A_n^m = \frac{(-1)^n}{\sqrt{(n-m)!(n+m)!}} . \quad (5.23)$$

Furthermore, for any  $p \geq 1$ ,

$$\left| \Phi(P) - \sum_{j=0}^p \sum_{k=-j}^j \frac{M_j^k}{r^{j+1}} \cdot Y_j^k(\theta, \phi) \right| \leq \left( \frac{\sum_{i=1}^l |q_i|}{r - (a + \rho)} \right) \left( \frac{a + \rho}{r} \right)^{p+1} . \quad (5.24)$$

**Theorem 5.4 (Conversion of a multipole expansion into a local expansion)** Suppose that  $l$  charges of strengths  $q_1, q_2, \dots, q_l$  are located inside the sphere  $D_Q$  of radius  $a$  with center at  $Q = (\rho, \alpha, \beta)$ , and that  $\rho > (c+1)a$  with  $c > 1$ . Then the corresponding multipole expansion (5.20) converges inside the sphere  $D_0$  of radius  $a$  centered at the origin. Inside  $D_0$ , the potential due to the charges  $q_1, q_2, \dots, q_l$  is described by a local expansion:

$$\Phi(P) = \sum_{j=0}^{\infty} \sum_{k=-j}^j L_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j , \quad (5.25)$$

where

$$L_j^k = \sum_{n=0}^{\infty} \sum_{m=-n}^n \frac{O_n^m \cdot i^{|k-m|-|k|-|m|} \cdot A_n^m \cdot A_j^k \cdot Y_{j+n}^{m-k}(\alpha, \beta)}{(-1)^n A_{j+n}^{m-k} \cdot \rho^{j+n+1}} , \quad (5.26)$$

with  $A_r^s$  defined by (5.23). Furthermore, for any  $p \geq 1$ ,

$$\left| \Phi(P) - \sum_{j=0}^p \sum_{k=-j}^j L_j^k \cdot Y_j^k(\theta, \phi) \cdot r^{j+1} \right| \leq \left( \frac{\sum_{i=1}^l |q_i|}{ca - a} \right) \left( \frac{1}{c} \right)^{p+1} . \quad (5.27)$$

**Theorem 5.5 (Translation of a local expansion)** Let  $Q = (\rho, \alpha, \beta)$

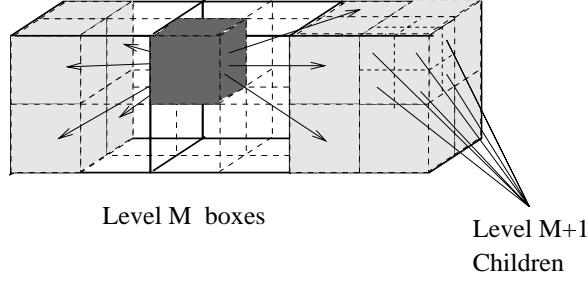


FIG. 10. In the FMM, a box transmits its far field expansion to the boxes in its interaction list. Those boxes then pass the information down to their children.

be the origin of a local expansion

$$\Phi(P) = \sum_{n=0}^p \sum_{m=-n}^n O_n^m \cdot Y_n^m(\theta', \phi') \cdot r'^n, \quad (5.28)$$

where  $P = (r, \theta, \phi)$  and  $P - Q = (r', \theta', \phi')$ . Then

$$\Phi(P) = \sum_{j=0}^p \sum_{k=-j}^j L_j^k \cdot Y_j^k(\theta, \phi) \cdot r^j, \quad (5.29)$$

where

$$L_j^k = \sum_{n=j}^p \sum_{m=-n}^n \frac{O_n^m \cdot i^{|m|-|m-k|-|k|} \cdot A_{n-j}^{m-k} \cdot A_j^k \cdot Y_{n-j}^{m-k}(\alpha, \beta) \cdot \rho^{n-j}}{(-1)^{n+j} \cdot A_n^m}, \quad (5.30)$$

with  $A_r^s$  defined by eqn. (5.23).

We can now construct the FMM by analogy with the two-dimensional case.

---

#### Upward Pass

- Form multipole expansions at finest level (from source positions and strengths).
- Form multipole expansions at coarser levels by merging, according to Theorem 5.3.

#### Downward Pass

- Account for interactions at each level by Theorem 5.4.
  - Transmit information to finer levels by Theorem 5.5.
- 

The total operation count is

$$189 \left( \frac{N}{s} \right) p^4 + 2Np^2 + 27Ns.$$

Choosing  $s = 2p^2$ , the operation count becomes approximately

$$150N p^2. \quad (5.31)$$

This would appear to beat the estimate (5.19) for any  $N$ , but there is a catch. The number of terms  $p$  needed for a fixed precision in the  $N \log N$  scheme is smaller than the number of terms needed in the FMM described above. To see why, consider two interacting cubes  $A$  and  $B$  of unit volume, with sources in  $A$  and targets in  $B$ . The worst-case multipole error decays like  $(\sqrt{3}/3)^p$ , since  $\sqrt{3}/2$  is the radius of the smallest sphere enclosing cube  $A$  and  $3/2$  is the shortest distance to a target in  $B$ . The conversion of a multipole expansion in  $A$  to a local expansion in  $B$ , however, satisfies an error bound which depends on the sphere enclosing  $B$  as well as the sphere enclosing  $A$  and has a worst case error estimate of the order  $(3/4)^p$ .

In the original FMM (Greengard and Rokhlin [30]; Greengard [23]), it was suggested that one redefine the nearest neighbour list to include “second nearest neighbours.” The error can then be shown to decay approximately like  $(0.4)^p$ . However, the number of near neighbours increases to 125 and the size of the interaction list increases to 875. In the latest generation of FMMs (Hrycak and Rokhlin [37]; Greengard and Rokhlin [33]), it turns out that one can recover the same accuracy as the  $N \log N$  schemes without this modification. The relevant ideas will be sketched below.

Returning now to the formula (5.31), it is clear that the major obstacle to achieving reasonable efficiency at high precision is the cost of the multipole to local translations ( $189p^4$  operations per box).

There are a number of schemes for reducing that cost. The first we describe is based on rotating the coordinate system (Fig. 11). By inspection of Theorem 5.4, it is easily shown that translation in the  $z$  direction only requires  $p^3$  operations. A fast translation scheme could then be obtained as follows.

- (1) Rotate the coordinate system ( $p^3$  operations) so that the vector connecting the source box and the target box lies along the  $z$  axis.
- (2) Shift the expansion along the  $z$  axis ( $p^3$  operations).
- (3) Rotate back to the original coordinate system ( $p^3$  operations).

The total operation count becomes

$$189 \left( \frac{N}{s} \right) 3p^3 + 2Np^2 + 27Ns.$$

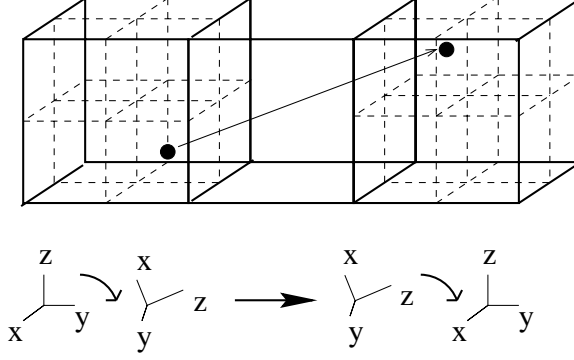


FIG. 11. Fast translation via rotation.

Choosing  $s = 3p^{3/2}$ , the operation count becomes approximately

$$270N p^{3/2} + 2N p^2. \quad (5.32)$$

Over the last few years, a number of diagonal translation schemes were developed, requiring  $O(p^2)$  work (Greengard and Rokhlin [31]; Berman [10]; Elliott and Board [21]). Unfortunately, these schemes are all subject to certain numerical instabilities. They can be overcome, but at additional cost.

#### 5.4 Exponential expansions

The latest generation of FMMs is based on combining multipole expansions with plane wave expansions. The two-dimensional theory is described in Hrycak and Rokhlin [37] and we will just sketch the three-dimensional version here (Greengard and Rokhlin [33]). The starting point is the integral representation

$$\frac{1}{r} = \frac{1}{2\pi} \int_0^\infty e^{-\lambda z} \int_0^{2\pi} e^{i\lambda(x \cos \alpha + y \sin \alpha)} d\alpha d\lambda.$$

To get a discrete representation, one must use some quadrature formula. The inner integral, with respect to  $\alpha$ , is easily handled by the trapezoidal rule (which achieves spectral accuracy for periodic functions), but the outer integral requires more care. Laguerre quadrature is an appropriate choice here, but even better performance can be obtained using generalized Gaussian quadrature rules (Yarvin [50]). The central idea, however, is that the multipole expansion can be converted to an expansion in exponentials.

$$\sum_{n=0}^p \sum_{m=-n}^n \frac{M_n^m Y_n^m(\theta, \phi)}{r^{n+1}} \approx \sum_{j=1}^{P_l} \sum_{k=1}^{K_j} e^{-\lambda_j(z - ix \cos \theta_k - iy \sin \theta_k)} S(j, k).$$



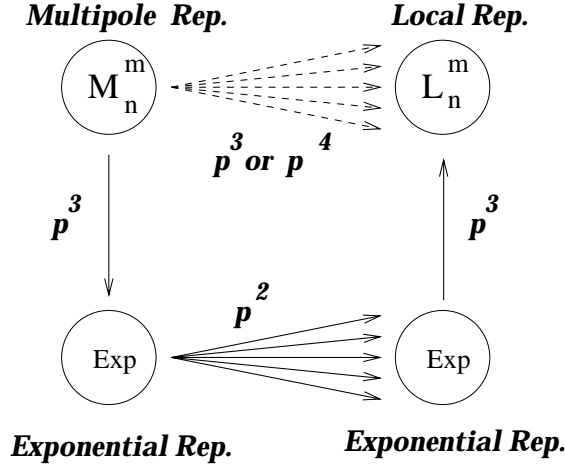


FIG. 12. Multipole translation can be replaced by exponential translation.

Unlike multipole expansions, however, these plane wave expansions are direction dependent. The one written above is for the  $+z$  direction, and a *different* one is needed for emanation through each of the six faces of the cube.

The reason for preferring exponentials is that translation corresponds to multiplication and requires only  $p^2$  work. Conversion from the multipole representation to the six outgoing exponentials requires  $3p^3$  work. Thus, we can replace  $189p^4$  or  $189p^3$  operations per box with  $3p^3 + 189p^2$  (Fig. 12).

The total operation count is now

$$189 \frac{N}{s} p^2 + 2N p^2 + 27N s + 3 \frac{N}{s} p^3.$$

Setting  $s = 2p$ , the total operation count is about

$$200 N p + 3.5 N p^2.$$

The new FMM has only been tested on relatively uniform distributions using a nonadaptive implementation. For three digit accuracy,  $p = 8$  and on a Sun SPARCStation 2, we have

$N$	Levels	$T_{Rot}$	$T_{Fourier}$	$T_{Exp}$	$T_{dir}$
1000	3	13	12	8	14
8000	4	160	140	67	900
64000	5	1700	1400	610	(58000)

$T_{Rot}$  is the time for the FMM using rotations,  $T_{Fourier}$  is the time required for the diagonal translation scheme based on the convolution structure, and

$T_{Exp}$  is the time required for the plane wave scheme sketched above.  
At ten digits of accuracy,  $p = 30$  and

$N$	Levels	$T_{Rot}$	$T_{Fourier}$	$T_{Exp}$	$T_{dir}$
5000	3	1030	—	230	350
40000	4	12300	—	1930	(22,400)

## 6 Conclusions

FMMs are efficient schemes for computing  $N$ -body interactions. Applications to boundary value problems for the Laplace and biharmonic equations can be found, for example, in Greenbaum *et al.* [22]; Greengard *et al.* [26]; Greengard and Moura [28]; Nabors and White [40]; Rokhlin [43]. Applications to the Poisson equation can be found in Greengard and Lee [27]; Russo and Strain [46]. For applications to molecular dynamics, see Board *et al.* [12]; Ding *et al.* [20].

## Bibliography

1. B. K. Alpert and V. Rokhlin. *A Fast Algorithm for the Evaluation of Legendre Expansions*. SIAM J. Sci. Stat. Comput. **12**, 158–179 (1991).
2. C. R. Anderson. *A Method of Local Corrections for Computing the Velocity Field Due to a Distribution of Vortex Blobs*. J. Comput. Phys. **62**, 111–123 (1986).
3. C. R. Anderson. *An Implementation of the Fast Multipole Method Without Multipoles*. SIAM J. Sci. Stat. Comput. **13**, 923–947 (1992).
4. A. W. Appel. *An Efficient Program for Many-body Simulation*. Siam. J. Sci. Stat. Comput. **6**, 85–103 (1985).
5. J. Barnes and P. Hut. *A Hierarchical  $O(N \log N)$  Force-Calculation Algorithm*. Nature **324**, 446–449 (1986).
6. R. K. Beatson and W. A. Light (1996). *Fast evaluation of Radial Basis Functions: Methods for 2-dimensional Polyharmonic Splines*. To appear in IMA J. Numer. Anal.
7. R. K. Beatson and G. N. Newsam. *Fast Evaluation of Radial Basis Functions, I*. Computers Math. Applic. **24**, 7–19 (1992).
8. R. K. Beatson and G. N. Newsam (1996). *Fast Evaluation of Radial Basis functions: Moment Based Methods*. To appear in SIAM J. Sci. Comput.
9. R. K. Beatson, M. J. D. Powell and A. M. Tan (1996). *Fast Evaluation of Radial Basis Functions: Methods for 3-dimensional Polyharmonic Splines*. In preparation.
10. L. Berman. *Grid-Multipole Calculations*. SIAM J. Sci. Comput. **16**, 1082–1091 (1995).
11. G. Beylkin, R. Coifman and V. Rokhlin. *Fast Wavelet Transforms and*

- Numerical Algorithms I*. Comm. Pure and Appl. Math. **44**, 141–183 (1991).
12. J.A. Board, J. W. Causey, J. F. Leathrum, A. Windemuth, and K. Schulten. *Accelerated Molecular Dynamics Simulation with the Parallel Fast Multipole Method*. Chem. Phys. Let. **198**, 89–94 (1992).
  13. B. Bradie, R. Coifman, and A. Grossmann. *Fast Numerical Computations of Oscillatory Integrals Related to Acoustic Scattering, I*. Appl. and Comput. Harmonic Analysis **1**, 94–99 (1993).
  14. A. Brandt. *Multilevel Computations of Integral Transforms and Particle Interactions with Oscillatory Kernels*. Comp. Phys. Comm. **65**, 24–38 (1991).
  15. A. Brandt and A. A. Lubrecht. *Multilevel Matrix Multiplication and Fast Solution of Integral Equations*. J. Comput. Phys. **90**, 348–370 (1990).
  16. F. X. Canning. *Transformations that Produce a Sparse Moment Method Matrix*. J. Electromagnetic Waves and Appl. **4**, 893–913 (1990).
  17. J. Carrier, L. Greengard, and V. Rokhlin. *A Fast Adaptive Multipole Algorithm for Particle Simulations*. SIAM J. Sci. and Stat. Comput. **9**, 669–686 (1988).
  18. R. Coifman, V. Rokhlin, and S. Wandzura. *The Fast Multipole Method for the Wave Equation: A Pedestrian Prescription*. IEEE Antennas and Propagation Magazine **35**, 7–12 (1993).
  19. J. Crank. *The Mathematics of Diffusion*. Clarendon Press, Oxford (1964).
  20. H.-Q. Ding, N. Karasawa, and W. A. Goddard III. *Atomic Level Simulations on a Million Particles: The Cell Multipole Method for Coulomb and London Nonbond Interactions*. J. Chem. Phys. **97**, 4309–4315 (1992).
  21. W. D. Elliott and J.A. Board. *Fast Fourier Transform Accelerated Fast Multipole Algorithm*. SIAM J. Sci. Comput. **17**, 398–415 (1996).
  22. A. Greenbaum, L. Greengard, and G. B. McFadden. *Laplace's Equation and the Dirichlet–Neumann Map in Multiply Connected Domains*. J. Comp. Phys. **105**, 267–278 (1993).
  23. L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems*. MIT Press, Cambridge (1988).
  24. L. Greengard. *The Numerical Solution of the N-body Problem*. Computers in Physics **4**, p. 142–152 (1990).
  25. L. Greengard. *Fast Algorithms for Classical Physics*. Science **265**, 909–914 (1994).
  26. L. Greengard, M. C. Kropinski and A. Mayo. *Integral Equation Meth-*

- ods for Stokes Flow and Isotropic Elasticity*. J. Comput. Phys. **125**, 403–414 (1996).
27. L. Greengard and J.-Y. Lee. *A Direct Adaptive Poisson Solver of Arbitrary Order Accuracy*. J. Comput. Phys. **125**, 415–424 (1996).
  28. L. Greengard and M. Moura. *On the Numerical Evaluation of Electrostatic Fields in Composite Materials*. Acta Numerica 379–410 (1994).
  29. L. Greengard and V. Rokhlin. *A Fast Algorithm for Particle Simulations*. J. Comput. Phys. **73**, 325–348 (1987).
  30. L. Greengard and V. Rokhlin. *Rapid Evaluation of Potential Fields in Three Dimensions*. Vortex Methods, C. Anderson and C. Greengard eds. Lecture Notes in Mathematics, vol. 1360, Springer-Verlag (1988).
  31. L. Greengard and V. Rokhlin. *On the Efficient Implementation of the Fast Multipole Algorithm*. Department of Computer Science Research Report 602, Yale University (1988).
  32. L. Greengard and V. Rokhlin. *On the Evaluation of Electrostatic Interactions in Molecular Modeling*. Chemica Scripta **29A**, 139–144 (1989).
  33. L. Greengard and V. Rokhlin (1996). *An Improved Fast Multipole Algorithm in Three Dimensions*, to appear.
  34. L. Greengard and J. Strain. *The Fast Gauss Transform*. SIAM J. Sci. Stat. Comput. **12**, 79–94 (1991).
  35. M. Gu and S. C. Eisenstat. *A Divide-And-Conquer Algorithm for the Symmetric Tridiagonal Eigenproblem*. Department of Computer Science Research Report 932, Yale University (1992).
  36. R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. McGraw-Hill, New York (1981).
  37. T. Hrycak and V. Rokhlin. *An Improved Fast Multipole Algorithm for Potential Fields*. Department of Computer Science Research Report 1089, Yale University (1995).
  38. J. D. Jackson. *Classical Electrodynamics*. Wiley, New York (1975).
  39. O. D. Kellogg. *Foundations of Potential Theory*. Dover, New York (1953).
  40. K. Nabors and J. White. *FastCap: A Multipole-Accelerated 3-D Capacitance Extraction Program*. IEEE Transactions on Computer-Aided Design **10**, 1447–1459 (1991).
  41. K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White. *Preconditioned, Adaptive, Multipole-Accelerated Iterative Methods for Three-Dimensional First-Kind Integral Equations of Potential Theory*. SIAM J. Sci. Stat. Comput. **15**, 713–735 (1994).
  42. A. M. Odlyzko and A. Schönhage. *Fast Algorithms for Multiple Evaluations of the Riemann Zeta Function*. Trans. Am. Math. Soc. **309**, 797–809 (1988).

- 43. V. Rokhlin. *Rapid Solution of Integral Equations of Classical Potential Theory*. J. Comput. Phys. **60**, 187–207 (1985).
- 44. V. Rokhlin. *Rapid Solution of Integral Equations of Scattering Theory in Two Dimensions*. J. Comput. Phys. **86**, 414–439 (1990).
- 45. V. Rokhlin. *Diagonal Forms of Translation Operators for the Helmholtz Equation in Three Dimensions*. Appl. and Comput. Harmonic Analysis **1**, 82–93 (1993).
- 46. G. Russo and J.A. Strain. *Fast Triangulated Vortex Methods for the 2D Euler Equations*. J. Comput. Phys. **111**, 291–323 (1994).
- 47. L. Van Dommelen and E. A. Rundensteiner. *Fast, Adaptive Summation of Point Forces in the Two-Dimensional Poisson Equation*. J. Comput. Phys. **83**, 126–147 (1989).
- 48. P. R. Wallace. *Mathematical Analysis of Physical Problems*. Dover, New York (1984).
- 49. H. Y. Wang and R. LeSar. *An Efficient Fast-multipole Algorithm based on an Expansion in the Solid Harmonics*. J. Chem. Phys. **104**, 4173–417 (1996).
- 50. N. Yarvin (1996). *Generalized Gaussian Quadratures and Fast Summation Schemes*, Ph.D. Dissertation, Yale University, in preparation.