The Fast Multipole Method

Lexing Ying
University of Texas at Austin

July, 2008
Problem statement

Given
- \( \{f_i\} \) a set of charges at \( \{p_i\} \),
- \( G(x, y) \) a smooth kernel,
we want to compute

\[
u_i = \sum_{j=0}^{N-1} G(p_i, p_j)f_j.
\]

Naive algorithm takes \( O(N^2) \). Our goal is to make it \( O(N) \).

Solution: The fast multipole method by Greengard and Rokhlin.
Geometric part

Two sets $A$ and $B$ are well-separated if the distance between $A$ and $B$ are greater than their diameters.

Consider interaction from $B$ to $A$. ($\{x_i\}$ and $\{y_j\}$ are subsets of $\{p_i\}$.)

We use the following approximation. For each $x_i$, its potential $u_i$

$$u_i \approx u(c_A) = \sum_j G(c_A, y_j)f_j \approx G(c_A, c_B)\sum_j f_j.$$  

Do not worry about the accuracy of this approximation for the time being. This is good when $A$ and $B$ are really well-separated.
Three step procedure:

Two representations:

- **Far field representation** \( f_B = \sum_j f_j \).
- **Local field representation** \( u_A = G(c_A, c_B) f_B \).

Interaction is approximately low rank. Here it is a rank-1 approximation.
However, each $p_i$ is both a source and a target. \( \{p_i\} \) are mixed up.

Solution: octree.

- Each leaf box contains a small number \( O(1) \) of points,
- The number of levels of the tree is \( O(\log N) \).
- For each \( B \), near field = adjacent boxes.
- Far field \( F^B \) = all well-separated boxes.
- Interaction list = boxes in \( B \)'s far field but not \( B \)'s parent's far field (i.e., boxes that can be addressed by \( B \) but not by \( B \)'s parent).
Top level, fix a box $B$, we have $O(1)$ well-separated boxes (e.g. $A$).

The interaction between $B$ and $A$ is computed using the previous 3-step procedure.

What about the nearby boxes? Go to the next level.
$B'$ (a child of $B$) has $O(1)$ boxes in its interaction list (e.g. $A'$) that have not been taken care of.

The interaction between $B'$ and $A'$ is computed using the previous 3-step procedure.

For the nearby boxes, go to the next level.
$B''$ (a child of $B'$) has $O(1)$ boxes in its interaction list (e.g. $A''$) that need to be taken care of.

Now $B''$ is also a leaf. The interaction between $B''$ and its neighbors is evaluated directly.
The full algorithm is:

1. At each level, for each box $B$, compute $f_B = \sum_{p_j \in B} f_j$.
2. At each level, for each pair $A$ and $B$ in each other’s interaction list, add $G(c_A, c_B) f_B$ to $u_A$ (F2L translation).
3. At each level, for each box $A$, add $u_A$ to $u_j$ for each $p_j \in A$.
4. At the leaf level, nearby computation.

Complexity analysis:

1. Each point belongs to a box in each of $O(\log N)$ levels. The complexity is $O(N \log N)$.
2. $O(N)$ boxes in total. Each box has $O(1)$ boxes in the interaction list. $O(1)$ operation per F2L translation. The complexity is $O(N)$.
3. Each point belongs to a box in each of $O(\log N)$ levels. The complexity is $O(N \log N)$.
4. $O(N)$ leaf boxes in total. Each one has $O(1)$ points in its neighbors. Direct computation is $O(N)$.

Total complexity is $O(N \log N)$.
Can we do better? Yes. Let us look at a box $B$ and its children $B_1, \cdots, B_4$.

$$f_B = \sum_{p_j \in B} f_j = \sum_{p_j \in B_1} f_j + \sum_{p_j \in B_2} f_j + \sum_{p_j \in B_3} f_j + \sum_{p_j \in B_4} f_j = f_{B_1} + f_{B_2} + f_{B_3} + f_{B_4}.$$

So $f_B$ can be computed from $f_{B_i}$ of its children

- $O(1)$ complexity,
- far field rep of $B_i \Rightarrow$ far field rep of $B$, called F2F translation
- bottom-up traversal of the octree.

Similarly, instead of putting $u_A$ to each of its points, simply do

$$u_{A_i} \leftarrow u_{A_i} + u_A \quad i = 1, 2, 3, 4.$$

What is added to $u_{A_i}$ will eventually be added to the individual points.

- $O(1)$ complexity,
- local field rep $A \Rightarrow$ local field rep of $A_i$, called L2L translation
- top-down traversal of the octree.
The full algorithm is:

1. Bottom up. For each level, each box $B$,
   - if leaf, compute $f_B$ from its points,
   - if non-leaf, compute $f_B$ from its children (F2F).
2. On each level, for each pair $A$ and $B$ in each other’s interaction list, add $G(c_A, c_B)f_B$ to $u_A$ (F2L).
3. Top down. For each level, each box $A$,
   - if leaf, add $u_A$ to $u_j$ for each point $p_j$ in $A$,
   - if non-leaf, add $u_A$ to its children (L2L).
4. At the leaf level, local computation.

Let us compute the complexity:

1. $O(N)$ boxes. $O(1)$ per F2F. Totally $O(N)$.
2. Same $O(N)$.
3. $O(N)$ boxes. $O(1)$ per L2L. Totally $O(N)$.
4. Same $O(N)$.

Total complexity is $O(N)$. 
Analytic Part

Come back to the question that total mass (charge)

\[ f_B = \sum_{p_j \in B} f_j \]

is not a good approximation.

We can do better. This is the analytic part of the FMM: given a prescribed accuracy \( \varepsilon \), all representations and translations shall have accuracy \( O(\varepsilon) \).

2D case. One considers \( x \) and \( y \) to be complex numbers. Up to a constant,

\[ G(x, y) = \ln |x - y| = \text{Re}(\ln(x - y)). \]

We will regard \( G(x, y) = \ln(x - y) \) and throw away the complex part at the end.
Far field representation

\[ G(x, y) = \ln(x - y) = \ln(x) + \ln(1 - y/x) = \ln(x) + \sum_{k=1}^{\infty} (-1/k)(y/x)^k. \]

\[ u(x) = \sum_j G(x, y_j)f_j = \ln(x)(\sum_j f_j) + \sum_{k=1}^{p} 1/x^k(-1/k \sum_j y_j^k f_j) + O(\varepsilon) \]

where \( p = O(\log(1/\varepsilon)) \) because \(|y_j/x| < \sqrt{2}/3\).

Hence the far field representation is

\[ a_0 = \sum_j f_j, \quad a_k = -1/k \sum_j y_j^k f_j \quad (1 \leq k \leq p). \]

This is called the multipole expansion.
Local field representation

\[ G(x, y) = \ln(x - y) = \ln(-y) + \ln(1 - x/y) = \ln(y) + \sum_{k=1}^{\infty} (-1/k)(x/y)^k. \]

\[ u(x) = \sum_{j} G(x, y_j)f_j = \sum_{j} \ln(-y_j)f_j + \sum_{k=1}^{p} x^k (-1/k \sum_{j} 1/y_j^k f_j) + O(\varepsilon) \]

where \( p = O(\log(1/\varepsilon)) \) because \( |x/y_j| < \sqrt{2}/3 \).

Hence the local field representation is

\[ a_0 = \sum_{j} \ln(-y_j)f_j, \quad a_k = -1/k \sum_{j} y_j^k f_j \quad (1 \leq k \leq p). \]
If the multipole expansion of child $B'$ is \{a_k\}, i.e.,

$$u(z) = a_0 \ln(z - z_0) + \sum_{k=1}^{p} a_k / (z - z_0)^k + O(\varepsilon)$$

then the multipole expansion of the parent $B$ is \{b_l\} with

$$b_0 = a_0, \quad b_l = -a_0 \frac{z_0^l}{l} + \sum_{k=1}^{l} a_k \binom{l-1}{k-1} z_0^{l-k} \quad (1 \leq l \leq p).$$

$$u(z) = b_0 \ln(z) + \sum_{l=1}^{p} b_l / z^l + O(\varepsilon).$$

The complexity of F2F is $O(p^2)$. 
F2L (far rep of $B$ to local rep of $A$)

If the multipole representation at $B$ is $\{a_k\}$, i.e.,

$$u(z) = a_0 \ln(z - z_0) + \sum_{k=1}^{p} a_k/(z - z_0)^k + O(\varepsilon)$$

then the local representation at $A$ is $\{b_l\}$ with

$$b_0 = a_0 \ln(-z_0) + \sum_{k=1}^{p} a_k/(-z_0)^k \quad b_l = -a_0/lz_0^l + 1/z_0^l \sum_{k=1}^{p} a_k(-z_0)^{-k} \left(\binom{l+k-1}{k-1}\right).$$

$$u(z) = \sum_{l=0}^{p} b_l z^l + O(\varepsilon)$$

The complexity of F2L is $O(p^2)$. 
L2L (local rep of A to local rep of $A'$)

If the local representation at $A$ is $\{a_k\}$, i.e.,

$$u(z) = \sum_{k=0}^{p} a_k (z-z_0)^k + O(\varepsilon)$$

then the local representation at $A'$ is $\{b_l\}$ with

$$b_l = \sum_{k=l}^{p} a_k \binom{k}{l} (-z_0)^{k-l}.$$  

$$u(z) = \sum_{l=0}^{p} b_l (z)^l + O(\varepsilon).$$

The complexity of L2L is $O(p^2)$. 
For a fixed $\varepsilon$,

- both representations are of size $O(p) = O(\log(1/\varepsilon))$,
- all translations are of complexity $O(p^2) = O(\log^2(1/\varepsilon))$,
- the FMM algorithm with these representations and translations still has complexity $O(N)$. 