A Crash Course on Fast Multipole Method (FMM)

Hanliang Guo
Kanso Biodynamical System Lab

AME 599, 2018 April
What is FMM?

• A fast algorithm to study N body interactions (Gravitational, Coulombic interactions...)

![Galaxies](image-url)
Top 10 algorithms in the 20th century*

- Metropolis Algorithm for Monte Carlo
- Simplex Method for Linear Programming
- Krylov Subspace Iteration Methods
- The Decompositional Approach to Matrix Computations
- The Fortran Optimizing Compiler
- QR Algorithm for Computing Eigenvalues
- Quicksort Algorithm for Sorting
- Fast Fourier transform
- Integer Relation Detection
- Fast Multipole Method

*:chronological order

What is FMM?

- N body interactions (Coulombic, Gravitational interactions…)
- Reduces complexity from $O(N^2)$ to $O(N)$
- An approximation with well defined error tolerance
- Not (too) difficult to implement
- Exists libraries to use directly
Goal for this course

• Learn:
  • Understand the algorithm
    (Why is it accurate… Why is it fast…)
  • Get a hand on my toy FMM code (matlab)
    • How to use existing libraries

• Future:
  • Adaptive mesh
  • Periodic boundary conditions
Outline

• Complexity

• Multipole expansion

• Coarse graining

• $N \log(N)$ algorithm

• Fast multipole method (FMM)

• Application to Stokes flow

• Kernel independent FMM (KIFMM)
Complexity

- How computational cost scales with the problem size
- Does not care the constant factor
- Extremely important for large $N$
- $O(N^2)$ complexity is usually considered infeasible!
- The constant factor is also very important in practice
- A naive implementation of $N$ body interaction problem has $O(N^2)$ complexity
A working example

$N$ source points with charges $q_i$

Want to know:
Potential and electrostatic forces at $N$ target points

Assume target points and source points are the same (not necessary)

Away from source point:
$$\phi_{x_o}(x) = -\log(||x - x_o||)$$
$$E_{x_o}(x) = \frac{x - x_o}{||x - x_o||^2}$$

$N=400$
A working example

$N$ source points with charges $q_i$

Want to know:
Potential and electrostatic forces at $N$ target points

Assume target points and source points are the same (not necessary)

Away from source point:

$$\phi_{x_o}(x) = \text{Re}(- \log(z - z_o))$$

$$E_{x_o}(x) = \nabla \phi_{x_o}(x)$$
Multipole expansion

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

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

where

$$Q = \sum_{i=1}^{m} q_i$$

and

$$a_k = \sum_{i=1}^{m} \frac{-q_i z_i^k}{k}$$

Proof...

Can we truncate the infinite series with a known error tolerance?
Error bound

For any $p \geq 1$,

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

$$\leq \left( \frac{A}{p + 1} \right) \left( \frac{1}{c - 1} \right) \left( \frac{1}{c} \right)^p$$

$c = \left| \frac{z}{r} \right|$, $A = \sum_{i=1}^{m} |q_i|$, $\alpha = \frac{A}{1 - \left| \frac{r}{z} \right|}$

For any tolerance $\epsilon$, choose $p$ s.t.

$$p = - \log_c \epsilon$$
A naive coarse graining method

- Group particles in each box as clusters.
- Form a multipole expansion for each box.
- Compute interactions between distant clusters when possible.
- Add far-field and direct near-field (direct computations) together.

**Complexity is still** $O(N^2)$!

**The constant factor could be reduced.**
$N \log(N)$ algorithm

Divide and conquer!

Consider clusters of particles at successive levels of spatial refinement, and to compute interactions between distant clusters by means of expansions when possible.
$N \log(N)$ algorithm

**Near neighbors:** two boxes are at the same refinement level and share a boundary point.

**Well separated:** two boxes are at the same refinement level and are not near neighbors.

**Interaction list:** box $i$’s interaction list consists of the children of the near neighbors of $i$’s parent which are well separated from box $i$. 
Near neighbors: two boxes are at the same refinement level and share a boundary point

Well separated: two boxes are at same refinement level and are not near neighbors

Interaction list: box $i$’s interaction list consists of the children of the near neighbors of $i$’s parent which are well separated from box $i$.

Maximum size of interaction list is 27
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.

Halt the recursive process after roughly $\log(N)$ levels of refinement.

At the finest level, compute interactions between nearest neighbors.
Fast multipole method (FMM)

• Can we do better than $N \log(N)$?

• The previous algorithm is of $N \log(N)$ complexity because all particles are “accessed” at every level of refinement.

• We can avoid this by some more analytical manipulations.

• In particular…

  *Translation of a multipole expansion (M2M)*
  *Conversion of a multipole expansion into a local expansion (M2L)*
  *Translation of a local expansion (L2L)*
Translation of a multipole expansion (M2M)

Suppose that \( \phi(z) = a_o \log(z - z_o) + \sum_{k=1}^{\infty} \frac{a_k}{(z - z_o)^k} \) is a multipole expansion of the potential due to a set of \( m \) charges of strengths \( q_1, q_2, \ldots, q_m \), all of which are located inside the circle \( D \) of radius \( R \) with center at \( z_o \).

Then for \( z \) outside the circle \( D_1 \) of radius \( (R + |z_o|) \) and center at the origin,

\[
\phi(z) = a_o \log(z) + \sum_{l=1}^{\infty} \frac{b_l}{z^l}
\]

where

\[
b_l = -\frac{a_o z_o^l}{l} + \sum_{k=1}^{l} a_k z_o^{l-k} \left( \begin{array}{c} l - 1 \\ k - 1 \end{array} \right)
\]

Proof…
Error bound of M2M

For any $p \geq 1$,

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

$$
A = \sum_{i=1}^{m} |q_i|
$$

Similar to the multipole expansion...

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

$$
\leq \left( \frac{A}{p+1} \right) \left( \frac{1}{c - 1} \right) \left( \frac{1}{c} \right)^{p}
$$
Conversion of a multipole expansion into a local expansion (M2L)

Suppose that m charges of strengths $q_1, q_2, \ldots, q_m$ are located inside the circle $D_1$ with radius $R$ and center $z_o$, and that $|z_o| = (c+1)R$ with $c > 1$. Then the corresponding multipole expansion

$$\phi(z) = a_o \log(z - z_o) + \sum_{k=1}^{\infty} \frac{a_k}{(z - z_o)^k}$$

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:

$$\phi(z) = \sum_{l=0}^{\infty} b_l z^l$$

$$b_o = a_o \log(-z_o) + \sum_{k=1}^{\infty} \frac{a_k}{z_o^k} (-1)^k$$

$$b_l = -\frac{a_o}{l z_o^l} + \frac{1}{z_o^l} \sum_{k=1}^{\infty} \frac{a_k}{z_o^k} \left( \begin{array}{c} l + k - 1 \\ k - 1 \end{array} \right) (-1)^k$$
Error bounds of M2L

\[
\left| \phi(z) - \sum_{l=0}^{p} b_l z^l \right| < \frac{A(4e(p + c)(c + 1) + c^2)}{c(c - 1)} \left( \frac{1}{c} \right)^{p+1}
\]

\( (c > 1) \)

Proof…
See Greengard & Rokhlin JCP 1987…
Translation of a local expansion (L2L)

For any complex \( z_o, z, \) and \( \{a_k\}, k=0,1,2,\ldots,n, \)

\[
\sum_{k=0}^{n} a_k(z - z_o)^k = \sum_{l=0}^{n} \left( \sum_{k=l}^{n} a_k \binom{k}{l} (-z_o)^{k-l} \right) z^l
\]

Expand series close to the origin.
Immediate results of Maclaurin series.
No error bound needed!
More generally… (in case you want to code)

Multipole expansion about \( z_C \):

\[
\phi(z) = a_o \log(z - z_C) + \sum_{k=1}^{p} \frac{a_k}{(z - z_C)^k} + \mathcal{O}\left(\frac{r^p}{R^p}\right)
\]

\[
a_o = \sum_{j=1}^{m} q_j, \quad a_k = \sum_{j=1}^{m} -\frac{q_j(z_j - z_C)^k}{k}
\]

M2M from \( z_C \) to \( z_M \):

\[
\phi(z) = b_o \log(z - z_M) + \sum_{l=1}^{p} \frac{b_l}{(z - z_M)^l} + \mathcal{O}(\epsilon)
\]

\[
b_o = a_o, \quad b_l = -\frac{a_o(z_C - z_M)^l}{l} + \sum_{k=1}^{l} a_k(z_M - z_C)^{l-k} \left( \frac{l-1}{k-1} \right)
\]

M2L from \( z_M \) to \( z_L \):

\[
\phi(z) = \sum_{l=0}^{p} c_l(z - z_L)^l + \mathcal{O}(\epsilon)
\]

\[
c_o = b_o \log(z_L - z_M) + \sum_{k=1}^{p} \frac{(-1)^k b_k}{(z_M - z_L)^k}, \quad c_l = -\frac{b_o}{l(z_M - z_L)^l} + \frac{1}{(z_M - z_L)^l} \sum_{k=1}^{p} \frac{(-1)^k b_k}{(z_M - z_L)^k} \left( \frac{l+k-1}{k-1} \right)
\]

L2L from \( z_L \) to \( z_T \):

\[
\phi(z) = \sum_{l=0}^{p} d_l(z - z_T)^l
\]

\[
d_l = \sum_{k=1}^{p} c_k \binom{k}{l} (z_T - z_L)^{k-l}
\]

Ying et al. JCP (2004)
The *fast* part!

- If we know the multipole expansion for all children boxes, we can get the multipole expansion for the parent box by doing M2M four times. \((4p^2\) operations)\)

- If we know the multipole expansion for any box, we can get the local expansion for any well separated box by doing M2L one time. \((p^2\) operations)\)

- If we know the local expansion for the parent box, we can get the local expansion for all the children boxes by doing L2L four times. \((4p^2\) operations)\)
FMM algorithm

- Initialization: Choose a number of levels so that there are, on average, $s$ particles per box at the finest level.

- Upward Pass: Begin at the finest level, create multipole expansions from the source positions and strengths. The expansions for all boxes at all higher levels are the formed by the merging procedure (M2M).

- Downward Pass: We convert the multipole expansion into a local expansion about the center of all boxes in $b$’s interaction list (M2L). Shift local expansions from parent boxes to children boxes (L2L). Add far field and near field interactions at the finest level.

- Note: Within all steps, you only save the coefficients of the expansions.
FMM algorithm

1. **Upward pass** computes multipoles for all cells: M-to-M translation

2. **Downward pass** translates multipoles to local terms for all cells
   - Constant (189 in 3D) interactive (cousin) cells per destination cell contribute to M-to-L translation
   - Inheritance from the parent cell: L-to-L translation (& delegation)

3. **Direct interactions** for the nearest-neighbor leaf cells

Adapted from A. Nakano’s lecture note
Algorithm - initialization & upward pass

- Initialization
  - Choose a level of refinement $n \approx \log_4 N$, a precision $\epsilon$, and set $p \approx -\log_2(\epsilon)$

- Upward pass
  1. Form multipole expansions of potential field due to particles in each box about the box center at the finest mesh level
  2. Form multipole expansions about the centers of all boxes at all coarser mesh levels, each expansion representing the potential field due to all particles contained in one box.
Algorithm - downward pass

• Downward pass

3. Form a local expansion about the center of each box at each mesh level \( l \leq n-1 \). This local expansion describes the field due to all particles in the system that are not contained in the current box or its nearest neighbors. Once the local expansion is obtained for a given box, it is shifted, in the second inner loop to the centers of the box’s children, forming the initial expansion for the boxes at the next level.

4. Compute interactions at finest mesh level.

5. Evaluate local expansions at particle positions.

6. Compute potential (or force) due to nearest neighbor directly.

7. Add direct and far-field terms together.
Algorithm - complexity

<table>
<thead>
<tr>
<th>Step</th>
<th>Operation count</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>order $Np$</td>
<td>Each particle contributes to one expansion at the finest level.</td>
</tr>
<tr>
<td>2</td>
<td>order $Np^2$</td>
<td>At the $l$th level, $4^l$ shifts involving order $p^2$ work per shift must be performed.</td>
</tr>
<tr>
<td>3</td>
<td>order $\leq 28Np^2$</td>
<td>There are at most 27 entries in the interaction list for each box at each level. An extra order $Np^2$ work is required for the second loop.</td>
</tr>
<tr>
<td>4</td>
<td>order $\leq 27Np^2$</td>
<td>Again, there are at most 27 entries in the interaction list for each box and $\approx N$ boxes.</td>
</tr>
<tr>
<td>5</td>
<td>order $\leq 27Np^2$</td>
<td>One $p$-term expansion is evaluated for each particle.</td>
</tr>
<tr>
<td>6</td>
<td>order $\approx Nk_n$</td>
<td>Let $k_n$ be a bound on the number of particles per box at the finest mesh level. Interactions must be computed within the box and its eight nearest neighbors, but using Newton’s third law, we need only compute half of the pairwise interactions.</td>
</tr>
<tr>
<td>7</td>
<td>order $N$</td>
<td>Adding two terms for each particle.</td>
</tr>
</tbody>
</table>

Greengard & Rokhlin, JCP, 1987
A toy problem in matlab

- [www-scf.usc.edu/~hanliang/fmm_working.m](www-scf.usc.edu/~hanliang/fmm_working.m)
- [www-scf.usc.edu/~hanliang/fmm.m](www-scf.usc.edu/~hanliang/fmm.m) (solution)
Outline

• Complexity
• Multipole expansion
• Coarse graining
• $N \log(N)$ algorithm
• Fast multipole method (FMM)
• Application to Stokes flow
• Kernel independent FMM (KIFMM)
Application to Stokes flow

One fundamental solution to Stokes flow is the *Stokeslet*

\[-\nabla^2 \mathbf{u}(\mathbf{x}) + \nabla p(\mathbf{x}) = \mathbf{f}(\mathbf{y}) \delta(\mathbf{x} - \mathbf{y})\]

\[u_i(\mathbf{x}) = S_{ij}(\mathbf{x}, \mathbf{y}) f_j(\mathbf{y}), \quad i, j = 1, 2, 3\]

\[S_{ij}(\mathbf{x}, \mathbf{y}) = \frac{\delta_{ij}}{|\mathbf{x} - \mathbf{y}|} + \frac{(x_i - y_i)(x_j - y_j)}{|\mathbf{x} - \mathbf{y}|^3}\]
Application to Stokes flow

For a set of $N$ Stokeslets, velocity induced by the other Stokeslets at the $m$-th Stokeslet:

$$u^m_i = \sum_{n=1, n \neq m}^{N} \sum_{j=1}^{3} S_{ij}(x^m, x^n) f_j^n, \quad m = 1, \cdots, N$$

Direct evaluation of velocities at all $N$ Stokeslets leads to $O(N^2)$ complexity

We seek a fast approach that can make use of the FMM to reduce the complexity to $O(N)$

3D Laplace FMM:
$$\phi(x, y) = \frac{1}{|x - y|}, \quad E_i(x, y) = \frac{x_i - y_i}{|x - y|^3}$$

3D Stokes kernel:
$$S_{ij}(x, y) = \frac{\delta_{ij}}{|x - y|} + \frac{(x_i - y_i)(x_j - y_j)}{|x - y|^3}$$
Application to Stokes flow

For a set of $N$ Stokeslets, velocity induced by the other Stokeslets at the $m$-th Stokeslet:

$$ u^m_i = \sum_{n=1, n \neq m}^{N} \sum_{j=1}^{3} S_{ij}(x^m, x^n) f^m_j, \quad m = 1, \ldots, N $$

Direct evaluation of velocities at all $N$ Stokeslets leads to $O(N^2)$ complexity.
We seek a fast approach that can make use of the FMM to reduce the complexity to $O(N)$

3D Laplace FMM:
$$ \phi(x, y) = \frac{1}{|x - y|}, \quad E_i(x, y) = -\frac{\partial}{\partial x_i} \frac{1}{|x - y|} $$

3D Stokes kernel:
$$ S_{ij}(x, y) = \left( \delta_{ij} - (x_j - y_j) \frac{\partial}{\partial x_i} \right) \frac{1}{|x - y|} $$
Application to Stokes flow

For a set of $N$ Stokeslets, velocity induced by the other Stokeslets at the m-th Stokeslet:

$$u^m_i = \sum_{n=1, n\neq m}^N \sum_{j=1}^3 S_{ij}(x^m, x^n)f^n_j, \quad m = 1, \cdots, N$$

$$u^m_i = \sum_{n=1, n\neq m}^N \sum_{j=1}^3 \left( \delta_{ij} - (x^m_j - x^n_j) \frac{\partial}{\partial x^m_i} \right) \frac{f^n_j}{|x^m - x^n|}$$

$$u^m_i = \sum_{j=1}^3 \left[ \left( \delta_{ij} - x^m_j \frac{\partial}{\partial x^m_i} \right) \sum_{n=1, n\neq m}^N \frac{f^n_j}{r_{nm}} \right] + \frac{\partial}{\partial x^m_i} \sum_{n=1, n\neq m}^N \frac{x^n \cdot f^n}{r_{nm}}$$

Four FMM calls!
Kernel independent FMM (KIFMM)

• The traditional FMM works well as long as the multipole expansions for the underlying kernel $K$ are known.

$$
\phi(x) = \int_{\Gamma} K(x, y)q(y)ds \\
\phi(x_i) = \sum_j K(x_i, y_j)q(y_j)
$$

• Sometimes it is difficult/impossible to expand a kernel (e.g. the regularized Stokeslet)

• What can we do?
KIFMM - Basic idea

- To use a distribution of source points with known kernel (typically Laplace kernel) outside the box to represent the far-field/near-field potential generated by the source points inside/outside the box.
KIFMM - Uniqueness theorem

• If $u$ is real and its first and second partial derivatives are continuous in a region $V$ and on its boundary $S$, and
  \[ \nabla^2 u = \rho \text{ in } V, \quad u = f \text{ or } \partial u / \partial n = g \]
  where $\rho$, $f$ and $g$ are prescribed functions, then $u$ is unique.
  (or at least up to a constant)
• Match the potential generated by the sources and the equivalent density at the check surface outside the equivalent surface.
• Two potentials are the same on and outside the check surface. (uniqueness of the 2nd order PDE’s solution for outer Dirichlet problem)

1. Evaluate induced potential of the sources on the check surface
   \[ \phi(x_i^{(c)}) = \sum_j K^*(x_i^{(c)}, y_j^{(s)}) q^*(y_j^{(s)}) \]

2. Solve the upward equivalent density
   \[ \phi(x_i^{(c)}) = \sum_j K(x_i^{(c)}, y_j^{(e)}) q(y_j^{(e)}) \]

   \[ R^{(e)} = (\sqrt{2} + d)r, \quad R^{(c)} = (4 - \sqrt{2} - 2d)r \]

   \[ 0 \leq d \leq \frac{4 - \sqrt{2}}{3}, \quad 2r \text{ is the box size} \]
KIFMM - Local expansion

- Match the potential generated by the sources and the equivalent density at the check surface inside the equivalent surface.
- Two potentials are the same on and inside the check surface. (uniqueness of the 2nd order PDE’s solution for inner Dirichlet problem)

1. Evaluate induced potential of the sources on the check surface
   \[ \phi(x_i^{(c)}) = \sum_j K^*(x_i^{(c)}, y_j^{(s)})q^*(y_j^{(s)}) \]

2. Solve the downward equivalent density
   \[ \phi(x_i^{(c)}) = \sum_j K(x_i^{(c)}, y_j^{(e)})q(y_j^{(e)}) \]
   \[ R^{(e)} = (4 - \sqrt{2} - 2d)r, \quad R^{(c)} = (\sqrt{2} + d)r \]
   \[ 0 \leq d \leq \frac{4 - \sqrt{2}}{3}, \quad 2r \text{ is the box size} \]
KIFMM - M2M, M2L, L2L

1. Evaluate induced potential of the equivalent density 1 on the check surface

\[ \phi(x_i^{(c)}) = \sum_j K(x_i^{(c)}, y_j^{(e1)}) q(y_j^{(e1)}) \]

2. Solve for the equivalent density 2

\[ \phi(x_i^{(c)}) = \sum_j K(x_i^{(c)}, y_j^{(e2)}) q(y_j^{(e2)}) \]

Ying et al. JCP (2004)
• Algorithm stays exactly* the same as the traditional FMM.

• No need to expand any kernels!

• Still need to evaluate induced potential for the new kernel at the leaf boxes (bottom level)

• Equivalent densities are found by solving \( p \) algebraic equations (\( p \) is the number of discretization points on the surface, usually small)
Algorithm - initialization & upward pass

• Initialization
  
  • Choose a level of refinement $n \approx \log_4 N$, a precision $\epsilon$, and set $p \approx -\log_2(\epsilon)$

• Upward pass

  1. Form multipole expansions of potential field due to particles in each box about the box center at the finest mesh level

  2. Form multipole expansions about the centers of all boxes at all coarser mesh levels, each expansion representing the potential field due to all particles contained in one box.
Algorithm - downward pass

- Downward pass

3. Form a local expansion about the center of each box at each mesh level \( l \leq n-1 \). This local expansion describes the field due to all particles in the system that are not contained in the current box or its nearest neighbors. Once the local expansion is obtained for a given box, it is shifted, in the second inner loop to the centers of the box’s children, forming the initial expansion for the boxes at the next level.

4. Compute interactions at finest mesh level.

5. Evaluate local expansions at particle positions.

6. Compute potential (or force) due to nearest neighbor directly.

7. Add direct and far-field terms together.
KIFMM - A toy problem in matlab

- www-scf.usc.edu/~hanliang/kifmm_working.m
- www-scf.usc.edu/~hanliang/kifmm.m (solution)
References


- L.X. Ying et al., A kernel-independent adaptive fast multipole algorithm in two and three dimensions, *Journal of Computational Physics*, 2004


Libraries

- Greengard: https://cims.nyu.edu/cmcl/software.html

- PVFMM: https://github.com/dmalhotra/pvfmm