# Parallel Fast Multipole

Paul Beckman, Mariya Savinov

NYU Courant

April 26, 2022

# Motivating Problem

# Motivating Problem

- Consider a collection of interacting particles with a potential

# Motivating Problem

- Consider a collection of interacting particles with a potential
- **Total potential** at a point $x$ due to **particles** $x_j$ with **charges** $q_j$ is

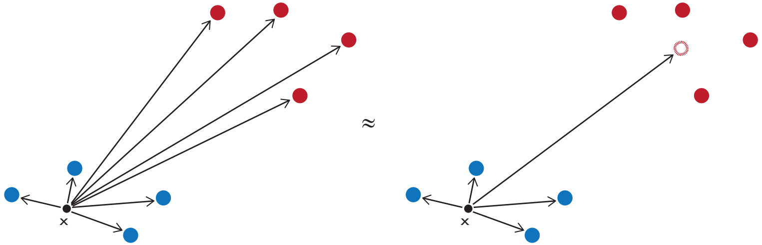
$$u(x) = \sum_{j=1}^{n} \frac{q_j}{|x - x_j|} = \sum_{j=1}^{n} q_j \phi(x - x_j)$$

# Motivating Problem

- Consider a collection of interacting particles with a potential
- **Total potential** at a point $x$ due to **particles** $x_j$ with **charges** $q_j$ is

$$u(x) = \sum_{j=1}^{n} \frac{q_j}{|x - x_j|} = \sum_{j=1}^{n} q_j \phi(x - x_j)$$

- Separate sum into **near-field** and **far-field** contributions
  - Take *aggregate* effect of far-field charges



# Approximating far-field

# Approximating far-field

- Use **Taylor series expansion** of $\phi(x - x_j)$ for small $\delta = \dfrac{x_j - x^*}{x^* - x}$:

$$
\begin{aligned}
\phi(x_j - x) &= \phi\left((x^* - x)\left(1 + \delta\right)\right) = \phi(x^* - x)\phi(1 + \delta) \\
&\approx \phi(x^* - x)\left[\sum_{k=0}^{p} \frac{\phi^{(k)}(1)}{k!}\delta^k + O(\delta^{p+1})\right] \\
&= \sum_{k=0}^{p} a_k(x_j - x^*) S_k(x^* - x) + O(\delta^{p+1})
\end{aligned}
$$

# Approximating far-field

- Use **Taylor series expansion** of $\phi(x - x_j)$ for small $\delta = \dfrac{x_j - x^*}{x^* - x}$:

$$
\begin{aligned}
\phi(x_j - x) &= \phi\left((x^* - x)(1 + \delta)\right) = \phi(x^* - x)\phi(1 + \delta) \\
&\approx \phi(x^* - x)\left[\sum_{k=0}^{p} \frac{\phi^{(k)}(1)}{k!}\delta^k + O(\delta^{p+1})\right] \\
&= \sum_{k=0}^{p} a_k(x_j - x^*)S_k(x^* - x) + O(\delta^{p+1})
\end{aligned}
$$

- Then potential from $x_j \in$ far-field is

$$
\sum_{j \in \text{far-field}} q_j \phi(x - x_j) \approx \sum_{k=0}^{p} \left[\sum_{j \in \text{far-field}} q_j a_k(x_j - x^*)\right] S_k(x^* - x) + O(\delta^{p+1})
$$

# Approximating far-field

- Use **Taylor series expansion** of $\phi(x - x_j)$ for small $\delta = \dfrac{x_j - x^*}{x^* - x}$:

$$
\begin{aligned}
\phi(x_j - x) &= \phi\left((x^* - x)\left(1 + \delta\right)\right) = \phi(x^* - x)\phi(1 + \delta) \\
&\approx \phi(x^* - x)\left[\sum_{k=0}^{p} \frac{\phi^{(k)}(1)}{k!}\delta^k + O(\delta^{p+1})\right] \\
&= \sum_{k=0}^{p} a_k(x_j - x^*)S_k(x^* - x) + O(\delta^{p+1})
\end{aligned}
$$

- Then potential from $x_j \in$ far-field is

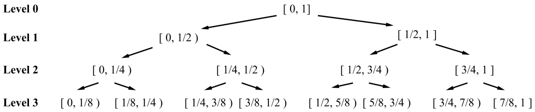
$$
\sum_{j \in \text{far-field}} q_j \phi(x - x_j) \approx \sum_{k=0}^{p}\left[\sum_{j \in \text{far-field}} q_j a_k(x_j - x^*)\right] S_k(x^* - x) + O(\delta^{p+1})
$$

- Accuracy depends on choice of $x^*$ and thus size of $\delta = \dfrac{x_j - x^*}{x^* - x}$

# Tree Algorithm by Barnes and Hut, complexity $O(N \log N)$

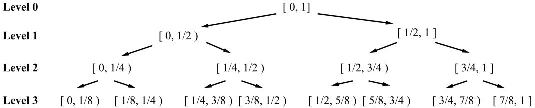
# Tree Algorithm by Barnes and Hut, complexity $O(N \log N)$

- For $N$ particles, partition $[0, 1]$ uniformly at $O(\log N)$ levels:



# Tree Algorithm by Barnes and Hut, complexity $O(N \log N)$

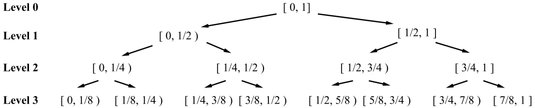
- For $N$ particles, partition $[0, 1]$ uniformly at $O(\log N)$ levels:




- Let $T_{\ell,m}$ be the cell at level $\ell$ with index $m = 1 : 2^\ell$ with center $x^*_{\ell,m}$.

# Tree Algorithm by Barnes and Hut, complexity $O(N \log N)$

- For $N$ particles, partition $[0, 1]$ uniformly at $O(\log N)$ levels:

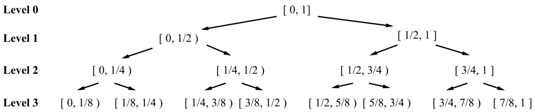



- Let $T_{\ell,m}$ be the cell at level $\ell$ with index $m = 1 : 2^\ell$ with center $x^*_{\ell,m}$.
- Compute weight at each cell, total cost $O(N \log N)$

$$w_{\ell,m,k} = \sum_{x_j \in T_{\ell,m}} q_j a_k(x_j - x^*_{\ell,m})$$

# Tree Algorithm by Barnes and Hut, complexity $O(N \log N)$

- For $N$ particles, partition $[0, 1]$ uniformly at $O(\log N)$ levels:




- Let $T_{\ell,m}$ be the cell at level $\ell$ with index $m = 1 : 2^\ell$ with center $x^*_{\ell,m}$.
- Compute weight at each cell, total cost $O(N \log N)$

$$w_{\ell,m,k} = \sum_{x_j \in T_{\ell,m}} q_j a_k (x_j - x^*_{\ell,m})$$

- For a point $x$: far-field components added at increasingly *coarse* levels

$$u(x) = \sum_{\ell=1}^{O(\log N)} \sum_{k=0}^{p} w_{\ell,m(\ell),k} S_k \left( x^*_{\ell,m(\ell)} - x \right)$$

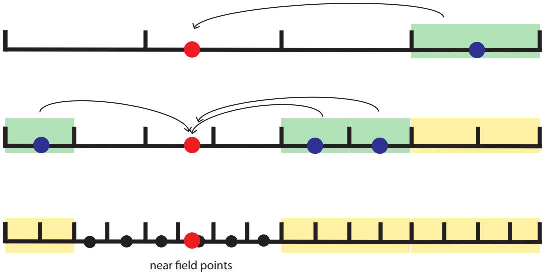
# Interaction List

# Interaction List

- At each level, only 'interact' with up to 4 far-field cells *not already included at higher levels*

# Interaction List

- At each level, only 'interact' with up to 4 far-field cells *not already included at higher levels*



# Parallelism

# Parallelism

- Parallelize with OpenMP

# Parallelism

- Parallelize with OpenMP
  - *Challenge:*

# Parallelism

- Parallelize with OpenMP
  - *Challenge:* Recursive tree code may prevent significant speedup

# Parallelism

- Parallelize with OpenMP
  - *Challenge:* Recursive tree code may prevent significant speedup
- Partition the memory and work with MPI

# Parallelism

- Parallelize with OpenMP
  - *Challenge:* Recursive tree code may prevent significant speedup
- Partition the memory and work with MPI
  - *Challenge:*

# Parallelism

- Parallelize with OpenMP
    - *Challenge:* Recursive tree code may prevent significant speedup
- Partition the memory and work with MPI
    - *Challenge:* Minimizing communication between cores

# Results

# Results

- Speed-up

# Results

- Speed-up
- How we measured speed-up

# Results

- Speed-up
- How we measured speed-up
- What did we expect? Is there some unavoidable overhead?