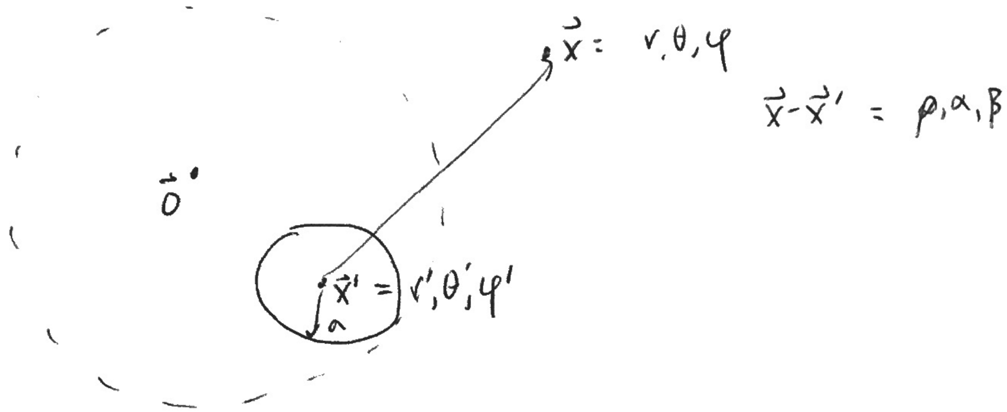
Lecture 4

Last time: - Higher-order tree code
- Multipole-to-multipole translation

M2M translation



Then

$$\phi(\vec{x}) = \sum_{l,m}^{\infty} \frac{M'_{lm} \, Y_l^m(\alpha, \beta)}{\rho^{l+1}}$$

valid outside $\|\vec{x} - \vec{x}'\| > a$

$\Rightarrow$

$$\phi(\vec{x}) = \sum_{l,m}^{\infty} M_{lm} \frac{Y_l^m(\theta, \varphi)}{r^{l+1}} \quad \text{for } \|\vec{x}\| > r' + a$$

and

$$\vec{M}_{lm} = \underset{\mathcal{M}}{} \vec{M}'_{lm}$$

$T^{\ell\ell}$ ← M2M translation operator defined by

$$M_{\ell m} = \sum_{l'=0}^{l} \sum_{m'=-l'}^{l'} M'_{(m-m')(l-l')} \underbrace{\boxed{\frac{J_{(m-m')(m)} A_{l'm'} A_{(m-m')(l-l')} r'^{l'} Y_{l'}^{m'}(\theta',\varphi')}{A_{l'}^{m'}}}}_{T_{lm,l'm'}}$$

<u>Proof</u> substitute in addition formula for $\frac{Y_l^m(\theta\varphi)}{r^{l+1}}$, collect terms. Applying $T^{\ell\ell}$ requires $O(p^4)$ operations.

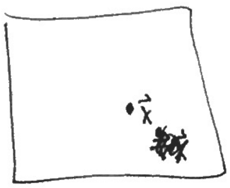
<u>Desc. of algorithm</u>

~~Formatt~~ 0- sort particles, set p.

1- Upward pass: form all p^{th} order multipple expansions for all boxes $O(p^4 n)$.

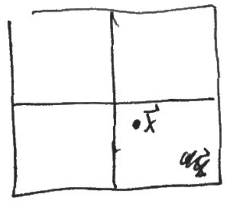
2- Evaluate: ~~For all~~ starting at level 0, evaluate multipole expansions <u>whenever</u> <u>well-separated</u>

Level 0



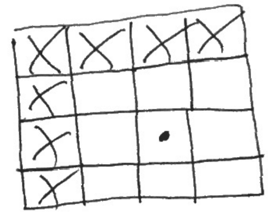
No interactions

Level 1



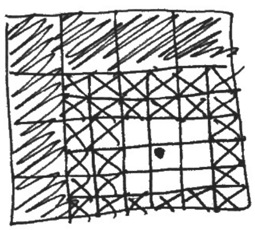
No interactions

Level 2



Evaluate MPs from X's

Level 3



Eval MP's from X's.

⇒ Interaction list

⇒ "List 2"

"well-separated boxes that are children of parents' neighbors."

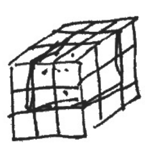
(cf: "List 1" → self, neighbors)

How many boxes are there in list 2?

2D: parent has 8 neighbors, they have 32 children, 5 of which will be neighbors ⇒ $\boxed{27}$

In 3D

Assume uniform tree, homogeneous dist. of points



$3 \times 3 \times 3 = 27$ ~~boxes as neighbors~~.

⇒ 26 neighbors of parent.

⇒ $8 \times 26 = 208$ children

How many of the 208 will be neighbors?

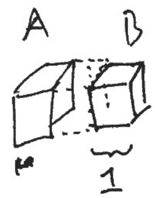
⇒ of 26 neighbors, 7 have same parent,

⇒ $26 - 7 = 19$ to have other parent

⇒ $208 - 19 = \boxed{189}$

Error analysis $\boxed{\text{No theta}}$ vs. Barnes-Hut

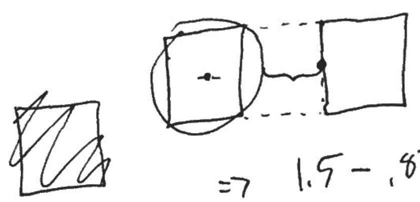
MP expansions are evaluated only in boxes that are well-separated:



In 3D, for box of width 1, ~~&~~ the smallest sphere enclosing the box has radius $r = \sqrt{\frac{1}{2^2}+\frac{1}{2^2}+\frac{1}{2^2}} = \sqrt{\frac{3}{4}} = \frac{\sqrt{3}}{2} \approx .87 = a$

The closest point in box B is in the middle of the face:

$\frac{3}{2} - \frac{\sqrt{3}}{2} = \frac{3-\sqrt{3}}{2} \approx .63$



$\Rightarrow 1.5 - .87 = .63$

$\frac{\sqrt{3}/2}{3/2} = \frac{\sqrt{3}}{3}$

$\frac{a}{r}$ in multipole estimate

$$\Rightarrow \left| \phi(\vec{x}) - \sum_{l=0}^{p} \sum_{m=-l}^{l} \frac{M_{lm}}{r^{l+1}} Y_l^m(\theta,\varphi) \right| \le \frac{\sum |q_j|}{r-a} \left(\frac{a}{r}\right)^{p+1}$$

$$\le \frac{\sum |q_j|}{.63} \left(\frac{1}{\sqrt{3}}\right)^{p+1} \le 2\left(\sum |q_j|\right)(.57)^{p+1} \le \epsilon$$

$p+1 \gtrsim \log \frac{1}{\epsilon}$

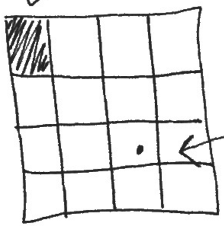
If precision ϵ is desired in the calculation, set $p \gtrsim O\left(\log_{\sqrt{3}} \frac{1}{\epsilon}\right)$. No theta, p controls the accuracy.

Reducing $O(n \log n)$ to $O(n)$.

The "expansion" part of the earlier scheme are

- translation $O(p^4)$
- $\log n$ ~~eval~~ MP evals per target.

This multipole



expansion is evaluated for every target in

~~there~~ many targets may exist on finer levels.

Introduce Local expansions: The dual of a multipole expansion:

Recall:
$$\frac{1}{\|\vec{x}-\vec{x}'\|} = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} \frac{r'^l}{r^{l+1}} Y_l^{-m}(\theta',\varphi')\, Y_l^m(\theta,\varphi)$$

target ($\vec{x}$), source ($\vec{x}'$)

$$= \sum_{l=0}^{\infty}\sum_{m=-l}^{l} \left(r'^l Y_l^{-m}(\theta',\varphi') \right) \frac{Y_l^m(\theta,\varphi)}{r^{l+1}}$$

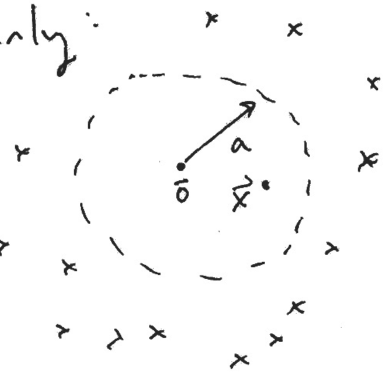
Just flip the source & target:

Likewise, at $\vec{x}'$, for a unit source located at $\vec{x}$, $\|x\| > \|\vec{x}'\|$

we have
$$\frac{1}{\|\vec{x}-\vec{x}'\|} = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} \underbrace{\left(\frac{Y_l^{-m}(\theta,\varphi)}{r^{l+1}} \right)}_{L_{lm}} r'^l\, Y_l^{m}(\theta',\varphi')$$

Local Expansion

Similarly:



For sources outside radius a, at targets inside radius a,

$$\phi(\vec{x}) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} L_{lm}\, r^{l}\, Y_l^{m}(\theta,\varphi)$$

with (by superposition), $L_{lm} = \sum_{j=1}^{N} q_j \frac{1}{r_j^{l+1}} Y^{-m}(\theta_j,\varphi_j)$

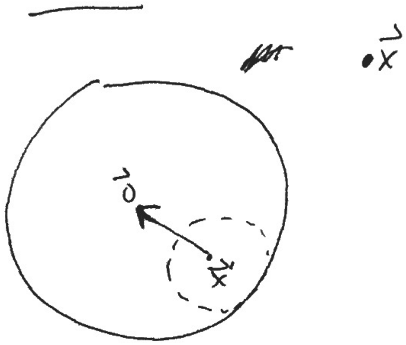
and we have the estimate:

$$\left| \phi(\vec{x}) - \sum_{l=0}^{p} \sum_{m=-l}^{l} L_{lm}\, r^{l}\, Y_l^{m}(\theta,\varphi) \right|$$

$$\le \frac{\sum |q_j|}{a-r} \left(\frac{r}{a}\right)^{p+1}$$

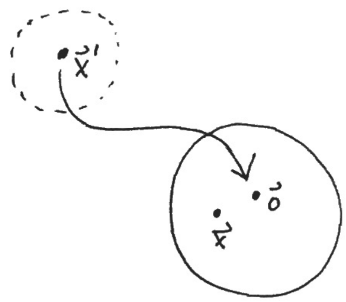
We will <u>not</u> define the local-to-local translation operators, <u>nor</u> multipole to ~~m~~ local translation operators, but rather only describe.

<u>M2~~L~~M:</u>



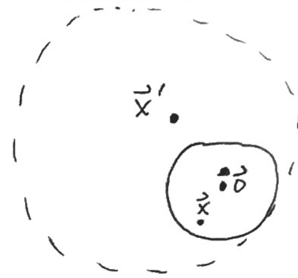
$$\sum_{lm} M'_{lm}\, O_l^m(\rho,\alpha,\beta) \to \sum_{l,m} M_{lm}\, I_l^m(r,\theta,\varphi)$$

<u>M2L</u>



$$\sum M'_{lm}\, O_l^m(\rho,\alpha,\beta) \to \sum L_{lm}\, I_l^m(r,\theta,\varphi)$$

<u>L2L</u>



$$\sum L'_{lm}\, I_l^m(\rho,\alpha,\beta) \to \sum L_{lm}\, I_l^m(r,\theta,\varphi)$$

where $I_l^m(r,\theta,\varphi) = r^l\, Y_l^m(\theta,\varphi)\cdot C_l^m$

$O_l^m(r,\theta,\varphi) = \dfrac{Y_l^m(\theta,\varphi)}{r^{l+1}}\, D_l^m$

so that formulas work better (~~E~~Epton, Dembart)

I_l^m ~~=~~ inner solid harmonics (incoming)

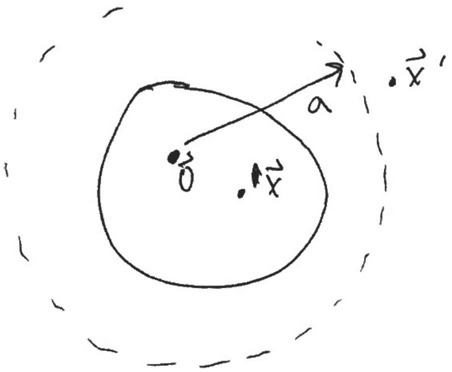
O_l^m = outer solid harmonics (outgoing)

Addition form now:

$$\frac{1}{|\vec{x}-\vec{x}'|} = \sum_{l,m} (-1)^l\, I_l^{-m}\, O_l^m$$

other addition forms are much simpler.

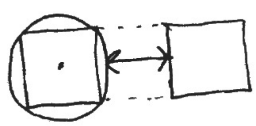
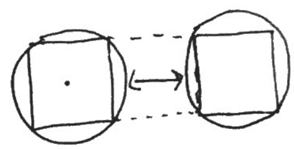
Local expansions satisfy similar error estimates as do multipole estimates: $O\left(\left(\frac{r}{a}\right)^{p+1}\right)$



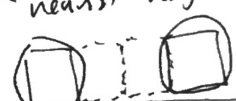
~~Now for the FMM~~

~~2 passes: Upward & Downward~~

Error estimate for a M2L is <u>worse</u> than for an MP Eval:

MPEval  M2L: 

the separation distance is less.

Fine, small detail... To improve accuracy, "2nd nearest neighbor" 

Discuss: Computational Tradeoff

The fast multipole method:

Upward pass: For levels $L, L-1, \ldots, 0$, form the multipole expansion for each box, by translating & merging children expansions.

$$O(8p^4 n)$$

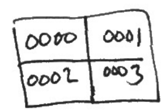
Downward pass: ~~Step 1.~~ Step 1. From levels $0, \ldots, L$, for each box, ~~translate~~ perform multipole to local translation for boxes in List 2. List 2 has 189 boxes, $\Rightarrow \mathcal{O}(189 p^4 n)$

Step 2 From levels $0, \ldots, L$, for each ~~place~~ box, perform L2L from parent (self) to children, add to existing local expansion. $\mathcal{O}(8 p^4 n)$

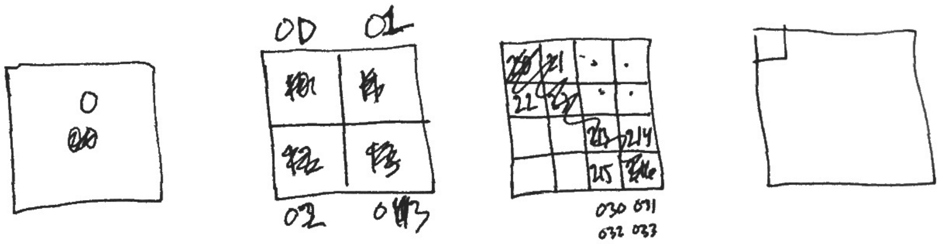
Eval: For each box on ~~place 2 from~~ Level L, evaluate the local expansion and potential due to all neighbors. $\mathcal{O}(p^2 n)$

The L=3

000 001
002 003



(Levels 0,1 are never needed).



~~Form MP: 20, 21, ..., 216~~

~~M2M: 10, ..., from 20~~

(20...23) → 10

⋮

(213...216) → 13

(10...13) → 00

Form MP: 0000 → 0333 } Level 3

M2M: 0000...0003 → 000 } Lev 3→2
0010...0013 → 001

⋮

M2L: 000 → 010...033 } Lev 2

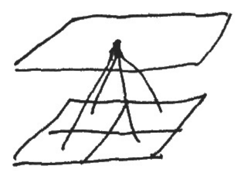
⋮

L2L } Lev 2→3

<u>Note</u>: L2L or M2M or M2L:

adding multipole/local expansion means <u>adding</u> the coefficients in the expansions:

M2M



$$\phi = \sum_{lm} M_{lm} O_l^m = \sum_{lm} \left(\bar{M}^1_{lm} + \dots + \bar{M}^4_{lm}\right) O_l^m$$

$$\phi_1 = \sum_{lm} M^1_{lm} O_l^{1m} \rightarrow \phi_1 = \sum_{lm} \bar{M}^1_{lm} O_l^m$$

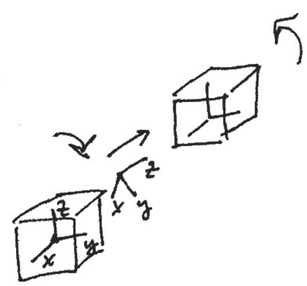
$$\vdots$$

$$\phi_4 = \sum_{lm} M^4_{lm} O_l^{4m} \rightarrow \phi_4 = \sum_{lm} \bar{M}^4_{lm} O_l^m$$

<u>That's the FMM</u>.

Acceleration: The <u>dominant</u> cost of the FMM is the application of the translation operators. $O(p^4 n)$

<u>Method 1</u>: Point and shoot : <u>$p^4 \rightarrow p^3$</u>



- aim z-axis at target (rotate)
- translate
- rotate back

<u>Procedurally</u>: We want to construct an operator which maps: (*)

$$\sum_{0}^{p} \sum_{m=-l}^{l} \alpha_{lm} O_{lm}(r,\theta,\varphi) = \sum_{0}^{p} \sum_{m=-l}^{l} \beta_{lm} \theta_{lm}(r,\theta_1,\varphi_1)$$

Why? If shifting along the z-axis (translating), then

$$\underline{Ex}: \quad M_{lm} = \sum_{l'=m}^{\infty\,(p)} C_{lm}^{l'} \cdot M'_{l'm}$$

(i.e., the m-modes don't mix)

since $\underline{\varphi \text{ old}} = \underline{\varphi \text{ new}}$.

Don't derive $C_{lm}^{l'}$ ... this means that

for each lm, we do $\not{\varepsilon}\ O(p)$ work $\Rightarrow$ $O(p^3)$ to

translate.

Cost for rotation: Since ⊛ must hold for all r,

we need that $\sum_{m=-l}^{l} \alpha_{lm} Y_l^m(\theta,\varphi) = \sum_{m=l}^{l} \beta_{lm} Y_l^m(\theta',\varphi')$

<u>Algorithm 1</u> (directly, quantum mechanics formulae)