Moore's Law

- Every 18 months, the speed of your computer is doubled
- Every 18 months, the memory on your computer is doubled
- At the same time, the cost of your computer goes down - not quite exponentially, because the box does not become much cheaper!
- A good number to look at

$$R_{1970} = \frac{Cost\ of\ CPU\ time}{Cost\ of\ human\ time}$$

- 1970 is the year
- Different CPUs, different humans, etc.

Observation

- $R_{1945} >> 1000$
- $R_{1960} >> 100$
- $R_{1970} >> 10$
- $R_{1980} \sim 1$
- $R_{2000} << 0.01$
- Unlike men, not all CPUs are created equal! But then, most CPUs do not vote...
- The thing is not slowing down, though eventually . . .
- What should we be doing as applied mathematicians, numerical analysts, etc.?

Consequences

- Ticket reservations
- Phone systems
- Tactical bombing
- Experimental science
- Manufacturing
- . . .

Missing from the list

- Philosophy
- Theater
- Politics
- Dealing with teen-age children
- Mathematics
- Numerical simulation of physical phenomena (???!!!)

Subject of the Talk

- Neither the numerical algorithms nor the paradigms for their application have kept pace with the developments of the computer hardware

- There are identifiable reasons for this, and to some extent, remedies can be devised and implemented

- In several environments, the results have been spectacular

- The usual message of an extremist: we are the future, with us or against us, victor or victim

- A somewhat different message for a mathematician

Structure of the Talk

- Changing paradigm in the numerical use of computers

- Interaction of Moore's law with numerical algorithms

- Characteristics of a modern numerical algorithm

- Example: Gravitational n-body problem

- Pontification

Paradigm as of 1945

- Critical mission (Manhattan project, for example)
- Willingness to expend human time on programming (ouch!), debugging of the numerical scheme, interpretation
- Limited computer resources: only small-scale problems can be solved
- Extremely uncomfortable programming environment
- Air of heroism and desperation
- No difference between theoretical numerical analysts and practitioners
- Numerical approaches appropriate to small-scale problems
- Numerical algorithms usually written from scratch

Paradigm as of 1970

- Mission not necessarily critical (oil exploration, NACA airfoils, more involved air-dynamics, civil and mechanical engineering, rocket fuel stoichiometry, . . .)
- Willingness to expend human time on programming (still pretty uncomfortable), interpretation
- Much improved computer capabilities; CPU time still quite expensive, but the flop rate is much higher; one can try running things at night
- The air much less heroic; most applications in non-desperate environments
- Numerical algorithms appropriate to small-scale problems
- Most numerical codes are written from scratch

Paradigm as of 2000

- Mission usually not critical: computer games, medical imaging, design of fishing rods, Boeing-767's . . .
- Limited willingness to expend human time on programming (could be fun, though!), interpretation. . . and most interpreters are not named Teller, Ulam, or Fermi. . .
- Very much improved computer capabilities; CPU time dirt cheap, and flop rate is about to become gigaflop rate
- Air not heroic at all; lots of applications, and most in non-desperate environments
- Numerical algorithms appropriate to small-scale problems
- Most numerical codes are written from scratch

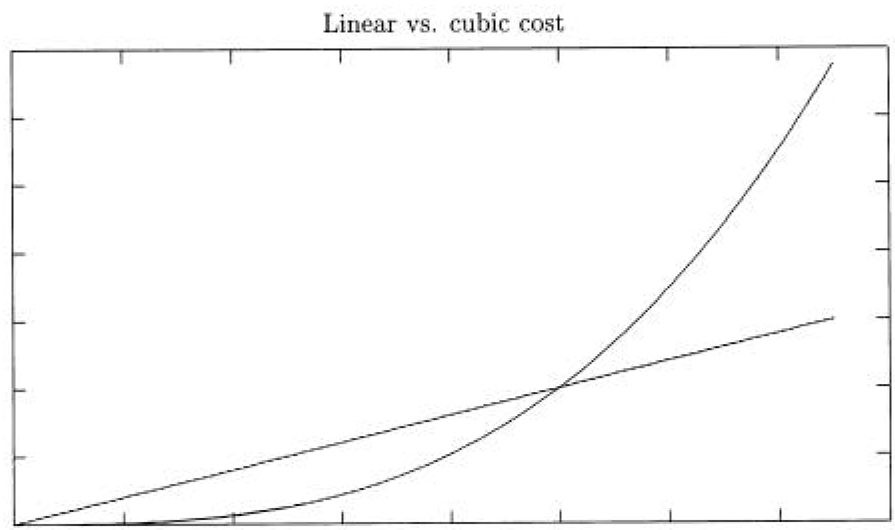
The Purpose of a Modern Numerical Algorithm

- Produce engineering (physical, biochemical, etc.) results with a minimum expenditure of *human* time

- CPU time is irrelevant *as long as it is affordable* (!!!)

- Note to the algorithm designer: torpedoes should not be aimed at the present location of the ship!

Illustration: Algorithms with CPU time estimates $O(n^3)$, $O(n \cdot log(n))$

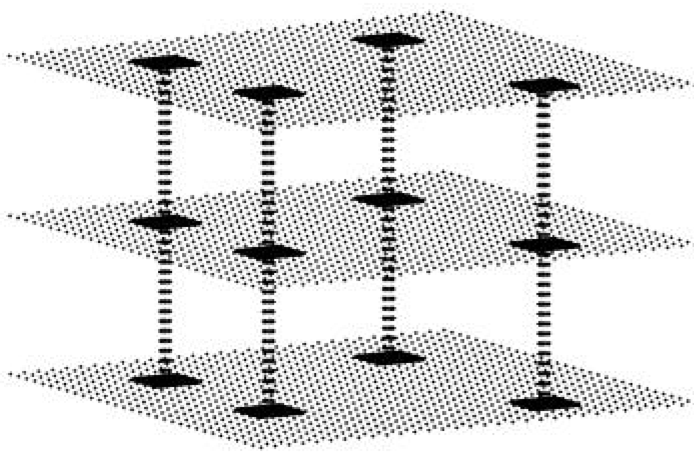


- To a large extent, the choice of the algorithm is determined by the power of one's computer (!!)

What *do* We Want from a Numerical Algorithm?

- Speed, in the asymptotic sense
- Adaptivity
- Robustness
- Rapid convergence and controlled accuracy: fallacy of the "engineering accuracy" argument; high cost of low precision
- Surprise: adaptivity implies controlled condition numbers, implies (more or less) integral vs. differential equations, implies fast algorithms
- Related surprise: in order to be efficient (or even simply useful), certain algorithms have to be fairly complicated (think about modern cars)

Numerical N-Body Problem



The calculation of all pairwise interactions in a system of N particles requires $O(N^2)$ work.

Particle Simulations

- ▷ Molecular Dynamics
- ▷ Fluid Dynamics
- ▷ Plasma Physics
- ▷ Dislocations and Plastic Deformation
- ▷ Astrophysics
- ▷ . . .

Integral Equations

- ▷ Capacitance calculations
- ▷ Dielectric interface problems
- ▷ Electrodeposition
- ▷ Elasticity
- ▷ Potential flow
- ▷ Incompressible Fluid Dynamics
- ▷ . . .

Alternative Approaches

- ▷ Field Methods
  (Based on Fast Solvers, FFT)
- ▷ Hierarchical Methods
  (Based on clustering at varying spatial scales)
- ▷ Wavelet, SVD Methods
  (Based on compression of operators)

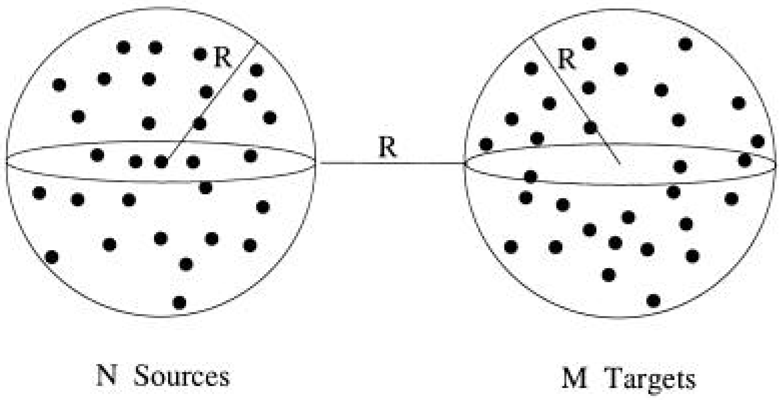
Critical Issues

• speed • adaptivity • ease of use

Overview of the Remainder of the Talk

- Analytic Preliminaries
- A simple $O(N \log N)$ algorithm
- The original FMM
- The modern FMM
- Pontification

A simple example

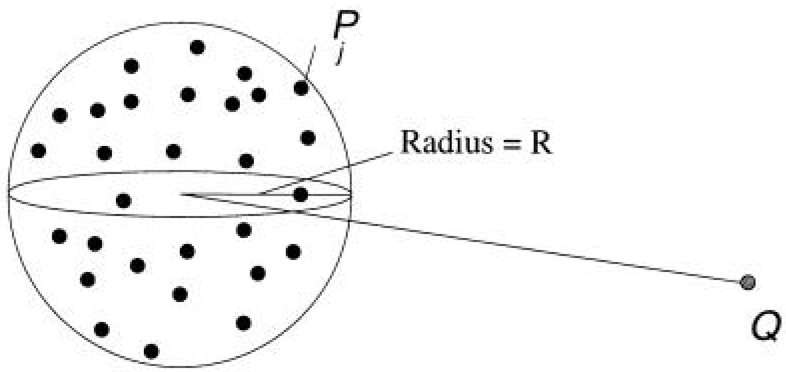


$$V(Q_i) = \sum_{j=1}^{N} \frac{q_j}{\|Q_i - P_j\|}$$

Direct evaluation requires $O(NM)$ work.

Newton knew how to fix it...

Multipole expansion



$$V(Q) = V(r,\theta,\phi) \approx \sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{M_n^m Y_n^m(\theta,\phi)}{r^{n+1}},$$

with multipole moments

$$M_n^m = \sum_{j=1}^{N} q_j Y_n^{-m}(\theta_j,\phi_j) r_j^n, \qquad P_j = (r_j,\theta_j,\phi_j)$$

The error in the multipole approximation decays like $(R/|Q|)^{p+1}$.

For our simple example, $R/|Q| < 1/2$, so that setting $p = \log_2(\frac{1}{\epsilon})$ yields a precision of ϵ.

Using multipole expansions

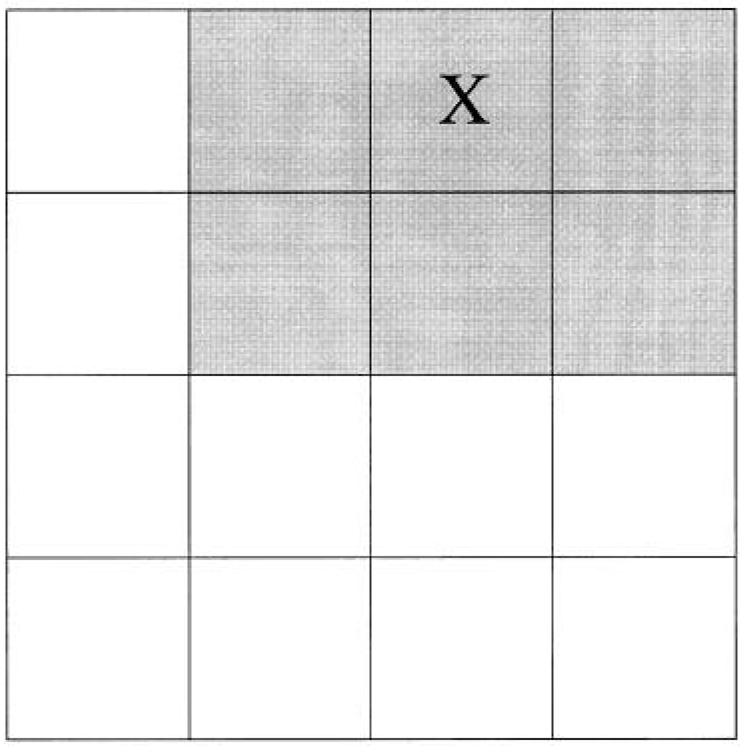
- ▷ Evaluate multipole coefficients M_n^m for $n = 0, \ldots, p$.
- ▷ Evaluate expansion at target points Q_j, for $j = 1, \ldots, M$.
- ▷ Total operation count: $p^2 \cdot (N + M) = (N + M) \cdot \log^2(\frac{1}{\epsilon})$

The Fast Multipole Method (FMM)

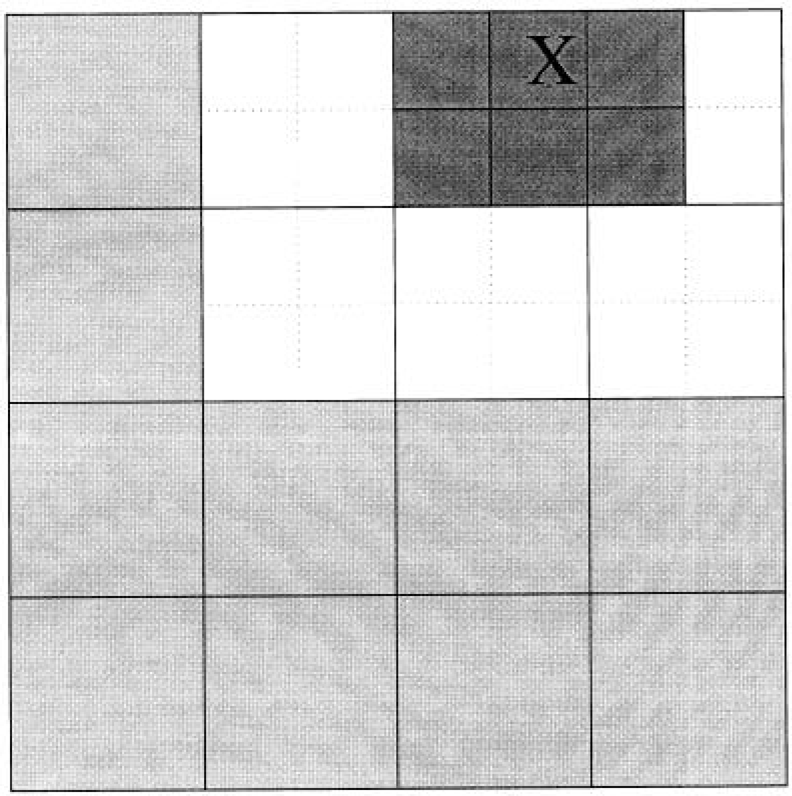
For more general distributions of sources and targets, FMM couples previous analysis with a divide & conquer strategy.

- ▷ Clustering at various spatial length scales
- ▷ Interactions with distant clusters computed by means of multipole expansions
- ▷ Interactions with nearby particles computed directly
- ▷ Fully adaptive algorithm
  Performance essentially independent of particle distribution

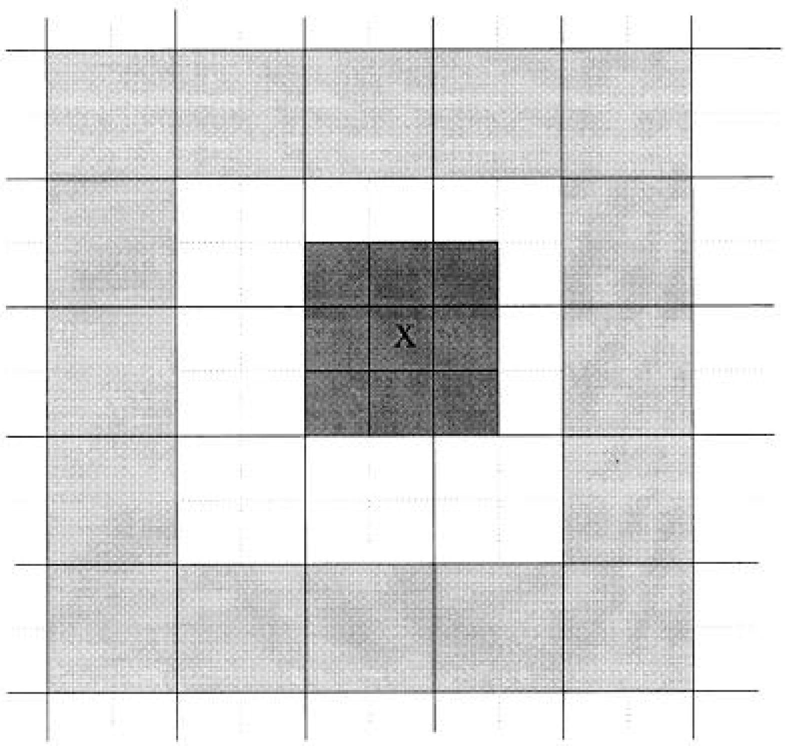
Step 1: $N \log N$ Scheme



Step 2: $N \log N$ Scheme



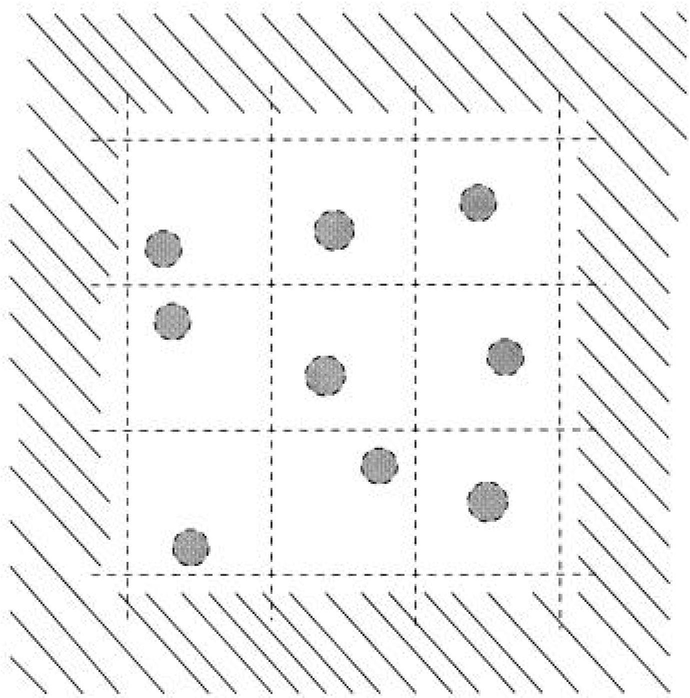
Step M: $N \log N$ Scheme



Final Step: $N \log N$ Scheme

Terminate procedure after $O(\log_8(N))$ steps.

Total operation count: $O(N \cdot \log_8 N \cdot p^2)$,
where $p = \log_c(\frac{1}{\epsilon})$ and $c = 3/\sqrt{3} \approx 1.73$.



Nearest neighbors: $O(N)$ work.

Optimization of constants

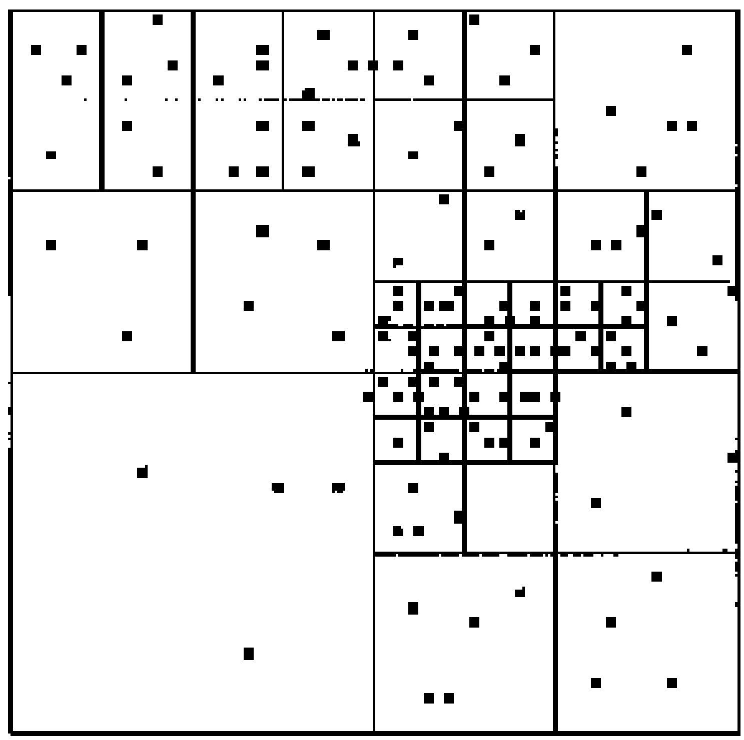
Assume that the distribution is uniform and let s be the number of particles per box at the finest level.

Multipole expansion work $= 189\, N\, p^2 \log_8(N/s)$.
Nearest neighbor work $= O(27Ns)$.

Optimal value for s is

$$s \approx p^2.$$

Adaptive algorithm

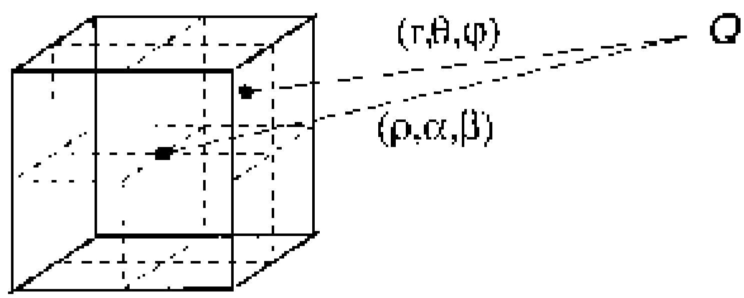


Final stage of subdivision process.

The order $O(N)$ algorithm

- Several analytical prerequisites
- Richer structure
- Many possible variants

Translation of multipole expansion



$$\sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{M_n^m Y_n^m(\theta,\phi)}{r^{n+1}} \rightarrow \sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{N_n^m Y_n^m(\alpha,\beta)}{\rho^{n+1}}$$

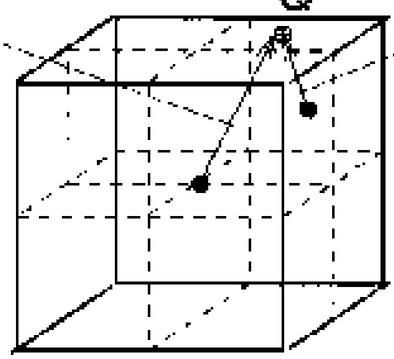
Cost: $O(p^4)$ work

Construction of local expansion

$$\sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{M_n^m Y_n^m(\theta,\phi)}{r^{n+1}} \rightarrow \sum_{n=0}^{p} \sum_{m=-n}^{n} L_n^m Y_n^m(\alpha,\beta)\, \rho^n$$

Cost: $O(p^4)$ work

Translation of local expansion



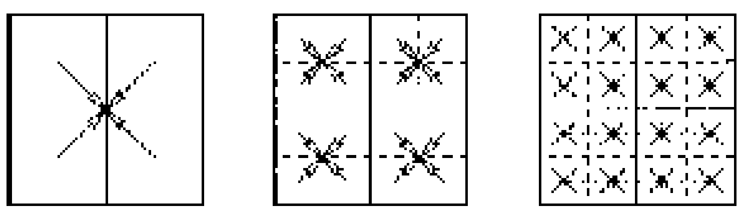
$$\sum_{n=0}^{p}\sum_{m=-n}^{n} L_n^m Y_n^m(\theta,\phi)\, r^n \rightarrow \sum_{n=0}^{p}\sum_{m=-n}^{n} O_n^m Y_n^{-m}(\alpha,\beta)\, \rho^n$$

Cost: $O(p^4)$ work

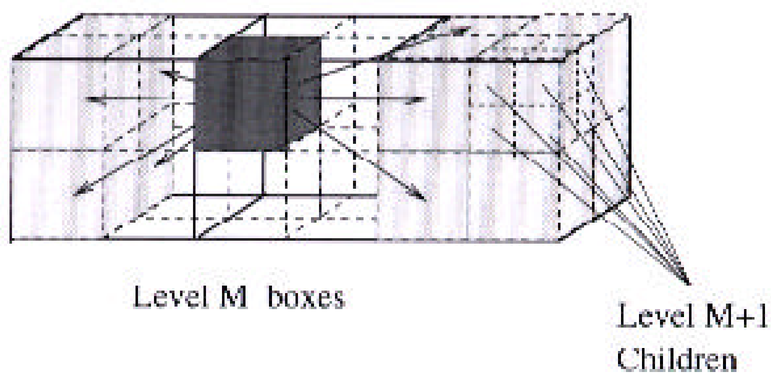
Complexity analysis

Why $N \log N$?

- Forming multipole expansions.
- Evaluating multipole expansions.



Capture far field in local expansions



- ▷ Use multipole to local translations
- ▷ Use translation of local expansion to transmit information to children

The order N algorithm

Upward Pass

- ▷ Form multipole expansions at finest level (from source positions and strengths)
- ▷ Form multipole expansions at coarser levels by merging

Downward Pass

- ▷ Account for interactions at each level by conversion lemma
- ▷ Transmit information to finer levels by shifting lemma

Total operation count

$$189 \frac{N}{s} p^4 + 2\, N\, p^2 + 54 N\, s$$

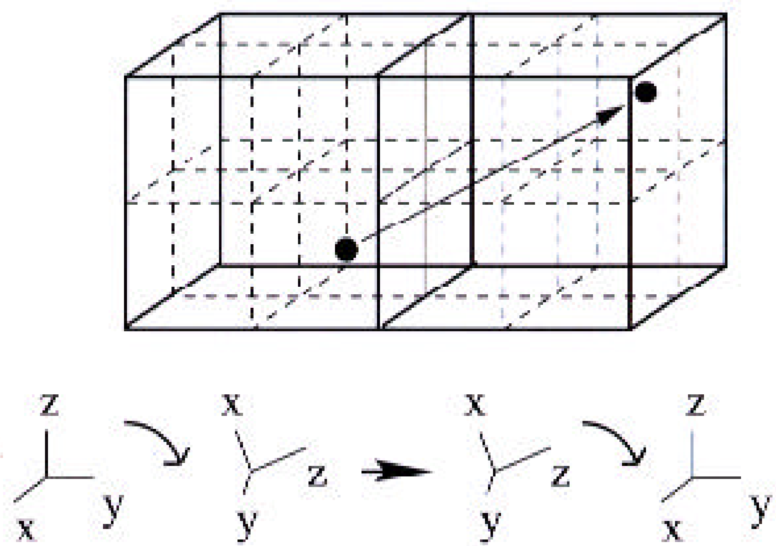
Setting $s = 1.5\, p^2$, the total operation count is

$$200\, N\, p^2.$$

Recall that the optimal $N \log N$ scheme required

$$189\, N\, p^2 (1 + \log \tfrac{N}{7p^2}) \quad \text{operations.}$$

Fast translations I: Rotation



$3p^3$ work is required for each shift, so the total operation count is

$$189 \frac{N}{s} 3p^3 + 2\,N\,p^2 + 54N\,s.$$

Setting $s = 3\,p^{3/2}$, the total operation count is

$$351\,N\,p^{3/2} + 2N\,p^2.$$

Diagonal translation: (G & R, 1988)

- Based on observation that translations are nearly convolutional
- Diagonalized by Fourier Transform
- Numerically unstable
- Can be stabilized by substructuring (Board et al. 1995)

Operation count

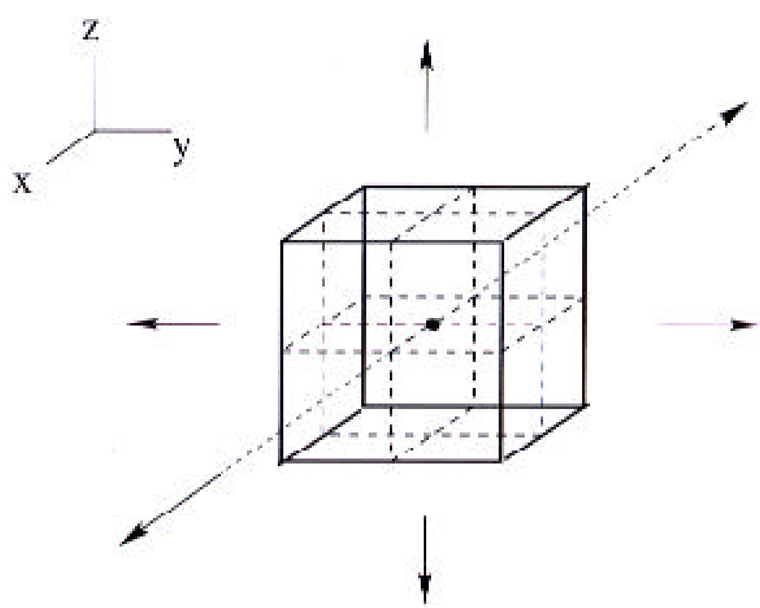
$$189\,\frac{N}{s}\,(2p)^2 + 2\,N\,p^2 + 54N\,s + \frac{N}{s}\,p^2\log p.$$

Setting $s = 1.5\,p$, the total operation count is

$$550\,N\,p + 2N\,p^2 + \frac{2}{3}N\,p\log p.$$

The new FMM

- 2D scheme : Hrycak and Rokhlin (1995)
- Based on expansion in plane waves
- Requires additional analytical machinery



Exponential representation (+ z)

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

- Discretization of α integral: trapezoidal rule
- Discretization of λ integral: Laguerre or generalized Gaussian quadrature (Yarvin & Rokhlin, 1996)

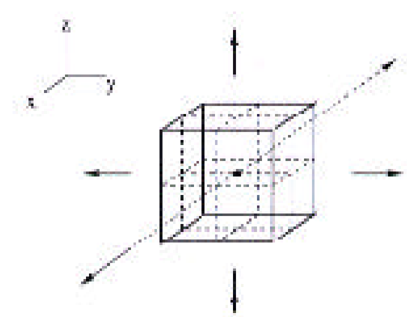
$$\sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{M_n^m Y_n^m(\theta, \phi)}{r^{n+1}} \approx$$

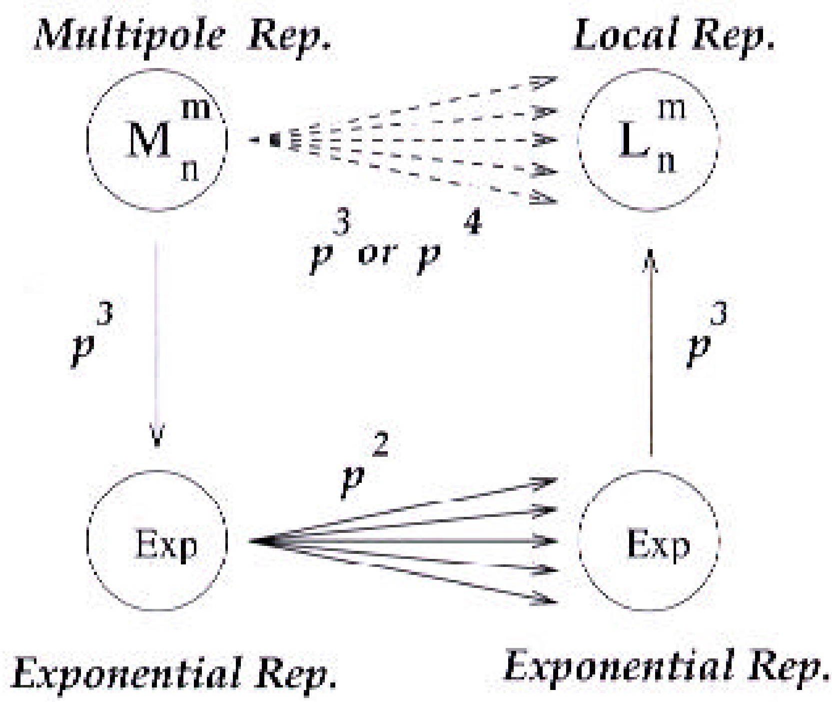
$$\sum_{j=1}^{P_l} \sum_{k=1}^{K_j} e^{-\lambda_j (z - ix\cos\theta_k - iy\theta_k)} S(j, k)$$

Exponential representation

Precision	p	Exp. Basis Fns.
10^{-3}	10	52
10^{-6}	19	258
10^{-9}	29	670

Exponential Translation





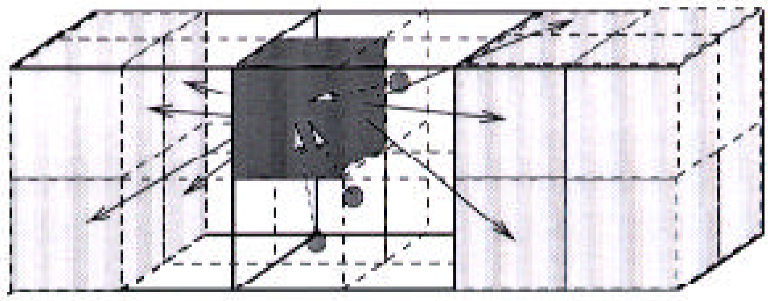
Operation Count

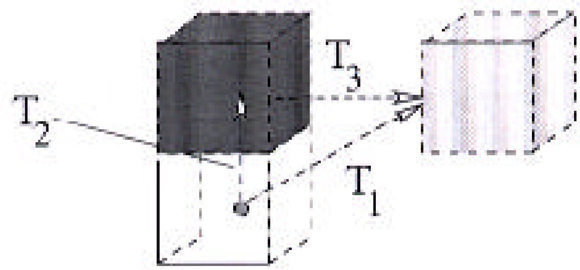
$$189\,\frac{N}{s}\,p^2 + 2\,N\,p^2 + 54N\,s + 6\frac{N}{s}\,p^3.$$

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

$$200\,N\,p + 5N\,p^2.$$

Reducing the Interaction List

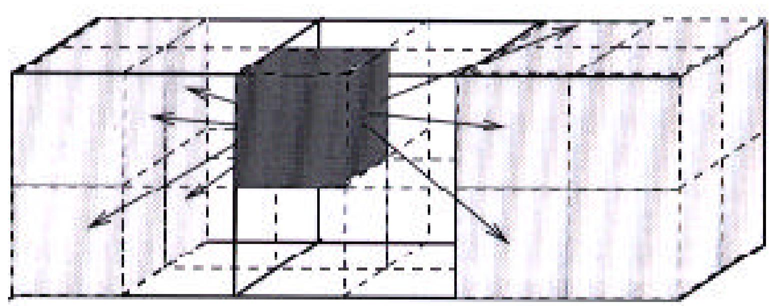




- Diagonal operators commute
- $T_1 = T_3 \cdot T_2$
- Merge before translation
- Reduces number of interactions per box to ≤ 40

Sweeping Under the Rug:

- Numerical compression of translation operators
- Harmonics exterior to a truncated cylinder, harmonics interior to a truncated cylinder, harmonics exterior to union of two truncated cones, etc.
- Nasty formulae, fairly simple numerical schemes
- A lot of fuss for a factor of two or so



Operation Count

$$40\,\frac{N}{s}\,p^2 + 2\,N\,p^2 + 54N\,s + 6\frac{N}{s}\,p^3.$$

Setting $s = 1.5\,p$, the total operation count is $\approx$

$$100\,N\,p + 6N\,p^2.$$

Random Distribution Inside a Cube

3-digit accuracy, times in seconds on UltraSPARC 1, 167 Mhz; calculations performed in single precision

N	Levels	T_{fmm}	T_{dir}	Error
20000	4	13.3	233	$7.9 \cdot 10^{-4}$
50000	4	24.7	1483	$5.2 \cdot 10^{-4}$
200000	5	158	24330	$8.4 \cdot 10^{-4}$
500000	5	268	138380	$7.0 \cdot 10^{-4}$
1000000	6	655	563900	$7.1 \cdot 10^{-4}$

Random Distribution Inside a Cube

6-digit accuracy, calculations performed in single precision

N	Levels	T_{fmm}	T_{dir}	Error
20000	3	15.9	233	$5.1 \cdot 10^{-7}$
50000	4	69	1483	$2.8 \cdot 10^{-7}$
200000	4	198	24330	$4.9 \cdot 10^{-7}$
500000	5	586	138380	$4.4 \cdot 10^{-7}$
1000000	5	1245	563900	$4.4 \cdot 10^{-7}$

Random Distribution Inside a Cube

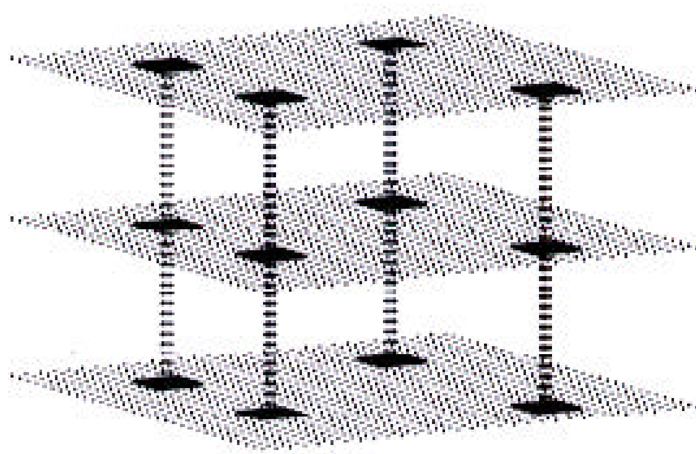
9-digit accuracy, calculations performed in double precision

N	Levels	T_{fmm}	T_{dir}	Error
20000	3	34	296	$2.8 \cdot 10^{-10}$
50000	3	96	1920	$1.6 \cdot 10^{-10}$
200000	4	385	30800	$1.6 \cdot 10^{-10}$
500000	4	1219	192600	$1.2 \cdot 10^{-10}$

Distribution On a Complicated Surface

3-digit accuracy, calculations performed in single precision

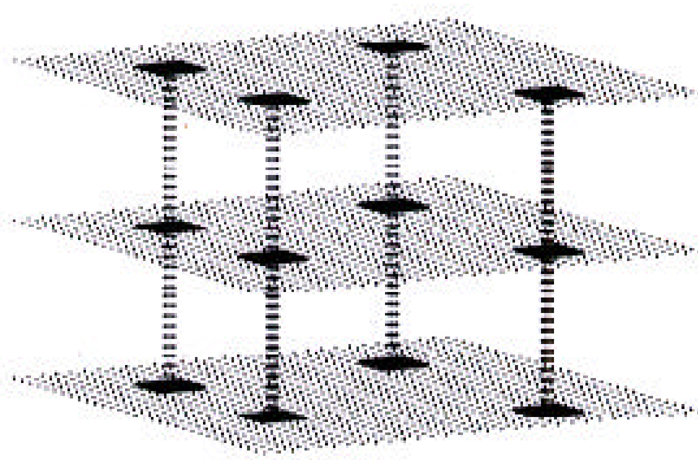
N	Levels	T_{fmm}	T_{dir}	Error
20880	7	6.7	243	$2.2 \cdot 10^{-4}$
51900	8	17	1539	$2.7 \cdot 10^{-4}$
203280	9	60	24730	$3.4 \cdot 10^{-4}$
503775	10	164	141060	$3.3 \cdot 10^{-4}$
1007655	10	282	568090	$2.9 \cdot 10^{-4}$



Distribution On a Complicated Surface

6-digit accuracy, calculations performed in single precision

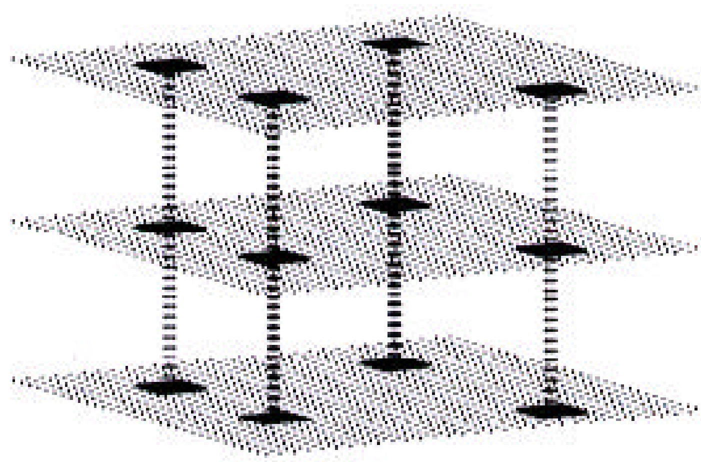
N	Levels	T_{fmm}	T_{dir}	Error
20880	7	17	243	$1.3 \cdot 10^{-7}$
51900	8	40	1539	$9.8 \cdot 10^{-8}$
203280	9	149	24730	$1.2 \cdot 10^{-7}$
503775	9	323	141060	$2.6 \cdot 10^{-7}$
1007655	10	714	568090	$2.0 \cdot 10^{-7}$



Distribution On a Complicated Surface

9-digit accuracy, calculations performed in double precision

N	Levels	T_{fmm}	T_{dir}	Error
20880	6	46	309	$3.6 \cdot 10^{-12}$
51900	7	101	2020	$1.1 \cdot 10^{-10}$
203280	8	342	32050	$6.5 \cdot 10^{-12}$
503775	9	896	193900	$1.0 \cdot 10^{-11}$



Observations

- For uniform structures (worst case):
- Breakeven point less than 1000 for 3 to 4 digit accuracy
- Breakeven point around 2000 for 6 digit accuracy
- Breakeven point around 3000 for 9 digit accuracy
- No loss of accuracy due to adaptivity
- A black box, as per original plan
- Large-scale problems manageable on desktop computers

Post-Mortem

- A simple formulation (gravitational n-body problem, integral equations of classical potential theory)
- Fairly simple incantational solution (early FMM schemes)
- The scheme becomes somewhat involved technically before becoming useful for anything
- Combination of a little mathematics and a fair amount of engineering
- Temptation to be a crook
- We were lucky

Now What ?

- ▷ A different set of bottlenecks and trade-offs: discretization, convergence, etc.
- ▷ Other potentials: Helmholtz, Yukawa, Hea Wave Equation,...
- ▷ Helmholtz potentials: at low frequencies similar to Laplace; at high frequencies quite different. In all regimes not quite as simple as Laplace
- ▷ Different types of equations: parabolic, hyperbolic, etc.
- ▷ Black boxes all
- ▷ Applications, modifications, etc.
- ▷ There are still some freebies left