Introduction to Simulation - Lecture 23

Fast Methods for Integral Equations

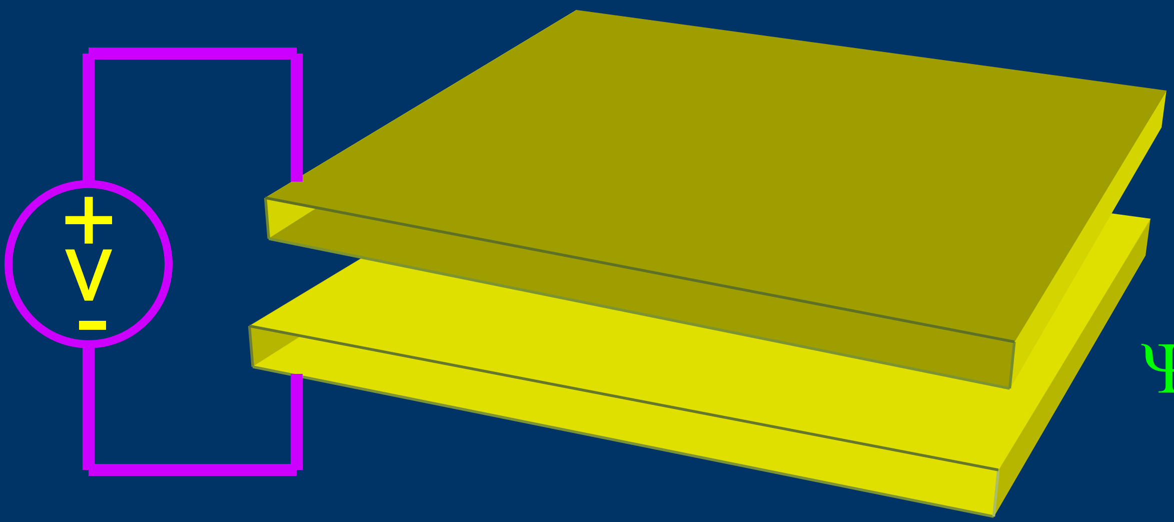
Jacob White

Thanks to Deepak Ramaswamy, Michal Rewienski, and Karen Veroy

Outline

- Solving Discretized Integral Equations
  - Using Krylov Subspace Methods
  - Fast Matrix-Vector Products
- Multipole Algorithms
  - Multipole Representation.
  - Basic Hierarchy
- Algorithmic Improvements
  - Local Expansions
  - Adaptive Algorithms
- Computational Results

Exterior Problem in Electrostatics



potential

$$\nabla^2 \Psi = 0 \quad \text{Outside}$$

Ψ is given on Surface

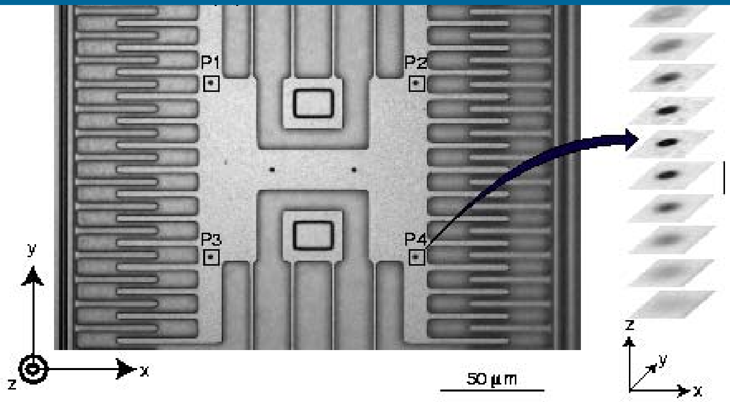
"Dirichelet Problem"

First Kind Integral Equation For Charge:

$$\Psi(x) = \int_{surface} \underbrace{\frac{1}{\left\| x - x' \right\|}}_{\text{Green's Function}} \underbrace{\sigma(x')}_{\text{Charge Density}} dS'$$

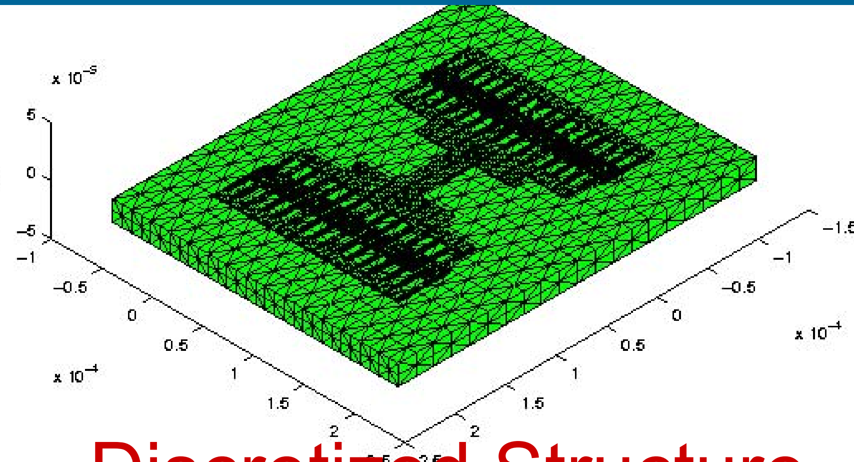
potential

Drag Force in a Microresonator

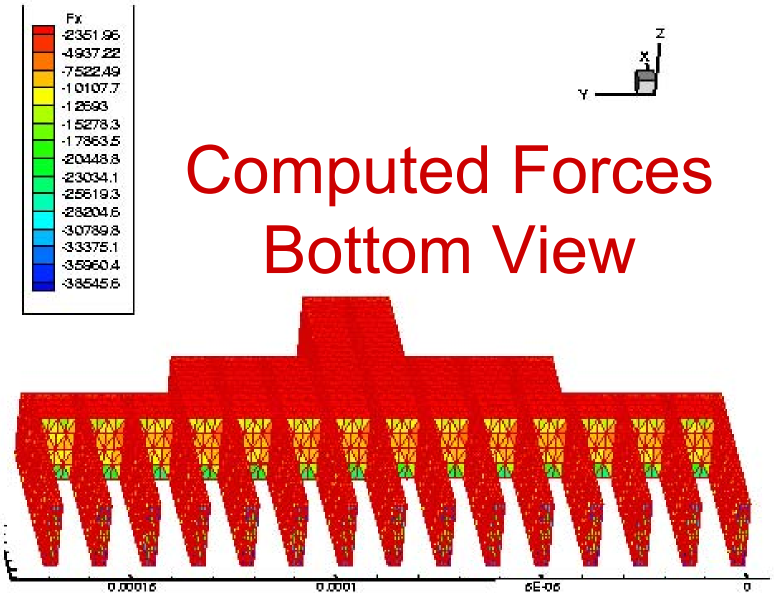


Courtesy of Werner Hemmert, Ph.D. Used with permission.

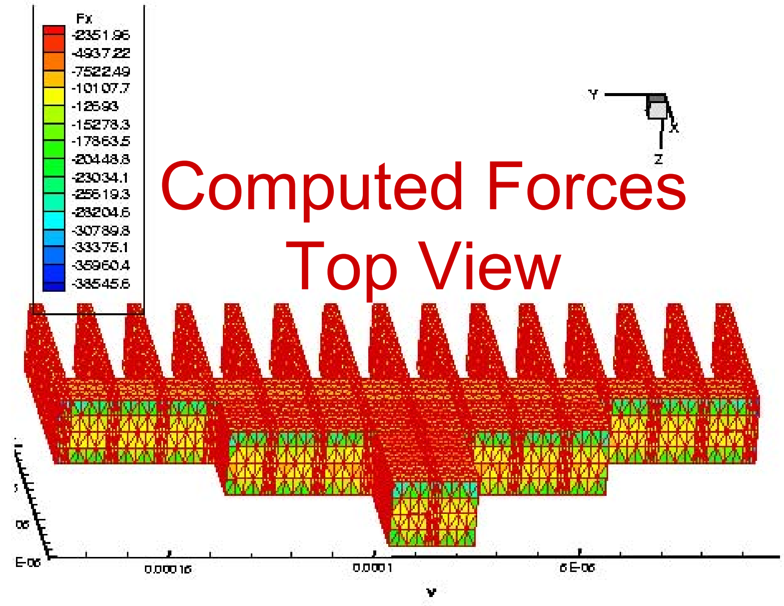
Resonator



Discretized Structure



Computed Forces
Bottom View



Computed Forces
Top View

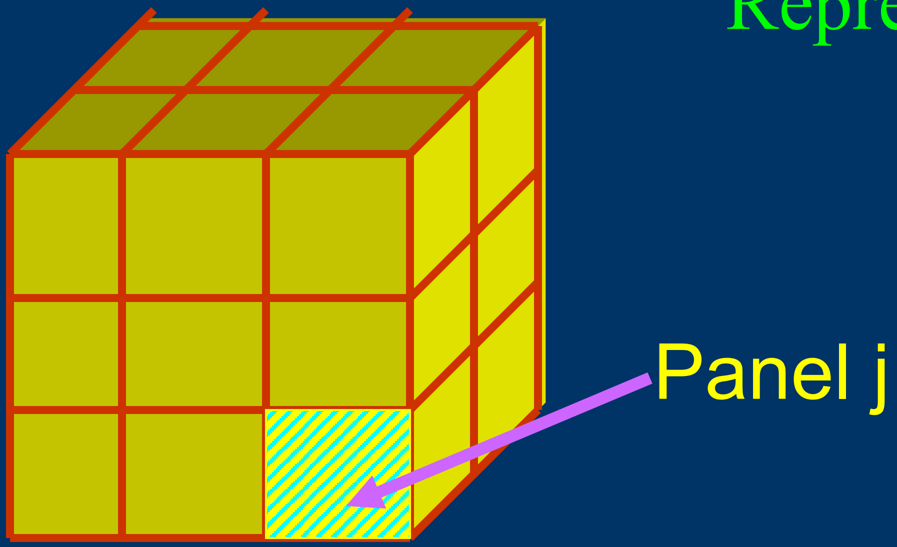
3-D Laplace's Equation

Basis Function Approach

Piecewise Constant Basis

Integral Equation: $\Psi(x) = \int_{surface} \frac{1}{\|x - x'\|} \sigma(x') \, dS'$

Discretize Surface into Panels



Represent $\sigma(x) \approx \sum_{i=1}^{n} \alpha_i \underbrace{\varphi_i(x)}_{\text{Basis Functions}}$

$\varphi_j(x) = 1 \quad \text{if } x \text{ is on panel j}$

$\varphi_j(x) = 0 \quad \text{otherwise}$

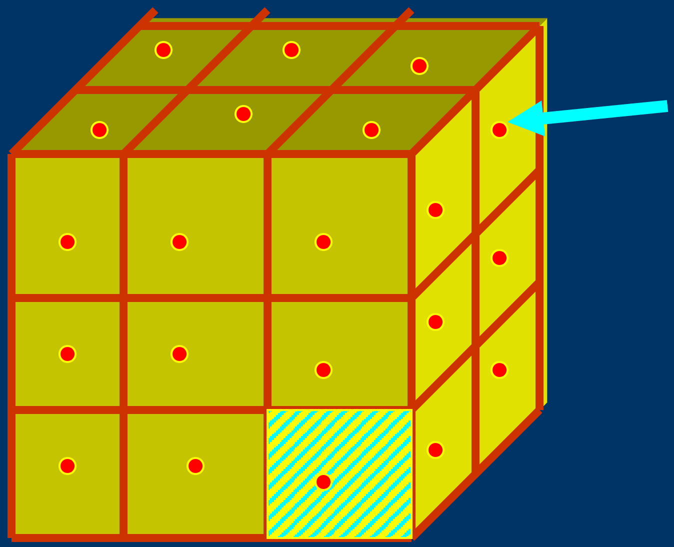
3-D Laplace's Equation

Basis Function Approach

Centroid Collocation

Put collocation points at panel centroids

$$\Psi\left(x_{c_i}\right) = \sum_{j=1}^{n} \alpha_j \underbrace{\int_{panel\ j} G\left(x_{c_i}, x'\right) dS'}_{A_{i,j}}$$

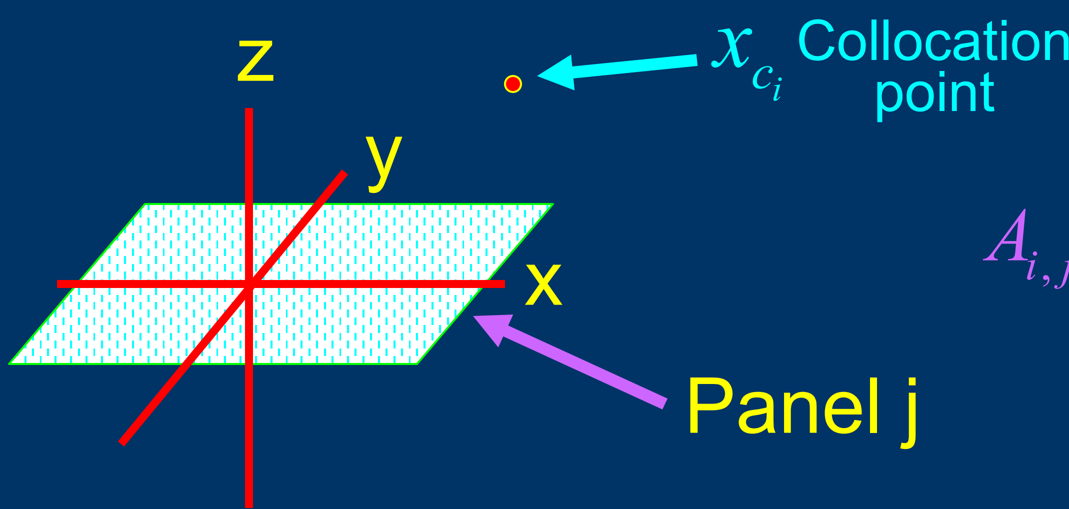


x_{c_i} Collocation point

$$\begin{bmatrix} A_{1,1} & \cdots & \cdots & A_{1,n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ A_{n,1} & \cdots & \cdots & A_{n,n} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} \Psi\left(x_{c_1}\right) \\ \vdots \\ \vdots \\ \Psi\left(x_{c_n}\right) \end{bmatrix}$$

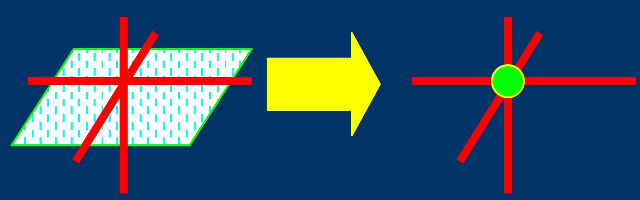
3-D Laplace's Equation

Calculating Matrix Elements



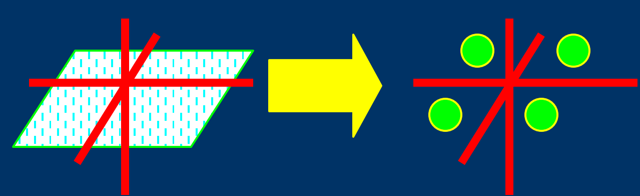
$$A_{i,j} = \int_{panel\ j} \frac{1}{\left\| x_{c_i} - x' \right\|} dS'$$

One point quadrature Approximation



$$A_{i,j} \approx \frac{Panel\ \ Area}{\left\| x_{c_i} - x_{centroid_j} \right\|}$$

Four point quadrature Approximation

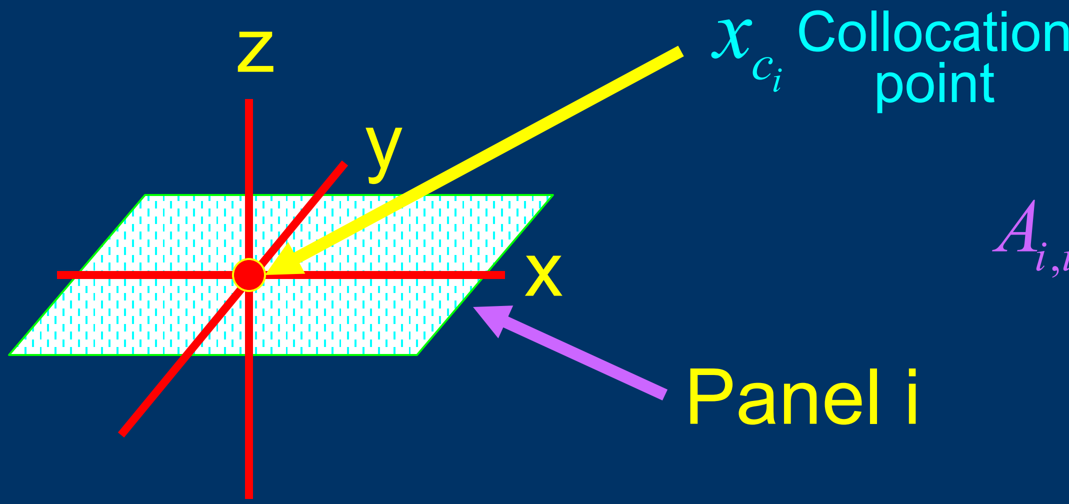


$$A_{i,j} \approx \sum_{j=1}^{4} \frac{0.25 * Area}{\left\| x_{c_i} - x_{point_j} \right\|}$$

3-D Laplace's Equation

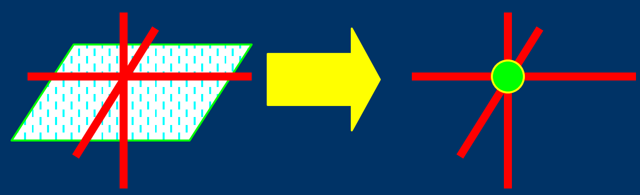
Basis Function Approach

Calculating "Self-Term"



$$A_{i,i} = \int_{panel\ i} \frac{1}{\left\| x_{c_i} - x' \right\|} dS'$$

One point quadrature Approximation



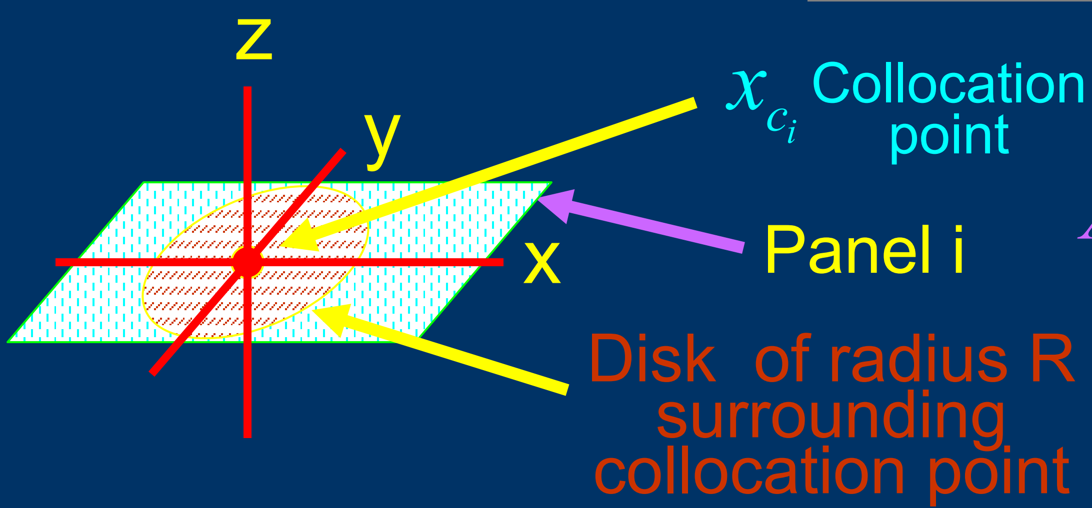
$$A_{i,i} \approx \frac{Panel\ \ Area}{\underbrace{\left\| x_{c_i} - x_{c_i} \right\|}_{0}}$$

$$A_{i,i} = \int_{panel\ i} \frac{1}{\left\| x_{c_i} - x' \right\|} dS' \quad \text{is an integrable singularity}$$

3-D Laplace's Equation

Basis Function Approach

Calculating "Self-Term" Tricks of the trade



$$A_{i,i} = \int_{panel\ i} \frac{1}{\left\| x_{c_i} - x' \right\|} dS'$$

Integrate in two pieces

$$A_{i,i} = \int_{disk} \frac{1}{\left\| x_{c_i} - x' \right\|} dS' + \int_{rest\ of\ panel} \frac{1}{\left\| x_{c_i} - x' \right\|} dS'$$

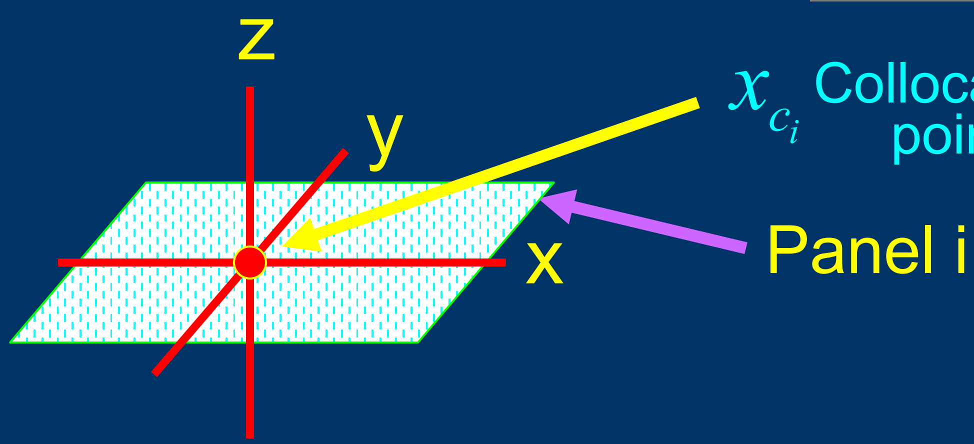
Disk Integral has singularity but has analytic formula

$$\int_{disk} \frac{1}{\left\| x_{c_i} - x' \right\|} dS' = \int_0^R \int_0^{2\pi} \frac{1}{r} r dr d\theta = 2\pi R$$

3-D Laplace's Equation

Basis Function Approach

Calculating "Self-Term"
Other Tricks of the trade



$$A_{i,i} = \int_{panel\ i} \frac{1}{\underbrace{\left\| x_{c_i} - x' \right\|}_{\text{Integrand is singular}}} dS'$$

1) If panel is a flat polygon, analytical formulas exist

2) Curve panels can be handled with projection

3-D Laplace's Equation

Basis Function Approach

Galerkin (test=basis)

$$\underbrace{\int \varphi_i(x)\Psi(x)\,dS}_{b_i} = \sum_{j=1}^{n} \alpha_j \underbrace{\int\int \varphi_i(x) G(x,x')\varphi_j(x')\,dS'dS}_{A_{i,j}}$$

<u>For piecewise constant Basis</u>

$$\underbrace{\int \Psi(x)\,dS'}_{b_i} = \sum_{j=1}^{n} \alpha_j \underbrace{\int_{panel\ i}\int_{panel\ j} \frac{1}{\|x-x'\|}\,dS'dS}_{A_{i,j}}$$

$$\begin{bmatrix} A_{1,1} & \cdots & \cdots & A_{1,n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ A_{n,1} & \cdots & \cdots & A_{n,n} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{bmatrix}$$

Integral Equation Method Generate Huge Dense Matrices

$$\begin{bmatrix} A_{1,1} & \cdots & \cdots & A_{1,n} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ A_{n,1} & \cdots & \cdots & A_{n,n} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} \Psi\left(x_{c_1}\right) \\ \vdots \\ \vdots \\ \Psi\left(x_{c_n}\right) \end{bmatrix}$$

Gaussian Elimination Much Too Slow!

Solving Discretized Integral Equations

The Generalized Conjugate Residual Algorithm

The kth step of GCR

compute Ap_k

For discretized Integral equations, A is dense

$$\alpha_k = \frac{\left(r^k\right)^T \left(Ap_k\right)}{\left(Ap_k\right)^T \left(Ap_k\right)}$$

Determine optimal stepsize in kth search direction

$$x^{k+1} = x^k + \alpha_k p_k$$
$$r^{k+1} = r^k - \alpha_k Ap_k$$

Update the solution and the residual

$$p_{k+1} = r^{k+1} - \sum_{j=0}^{k} \frac{\left(Ar^{k+1}\right)^T \left(Ap_j\right)}{\left(Ap_j\right)^T \left(Ap_j\right)} p_j$$

Compute the new orthogonalized search direction

Complexity of GCR

compute Ap_k

Dense Matrix-vector product costs $O(n^2)$

$$\alpha_k = \frac{(r^k)^T (Ap_k)}{(Ap_k)^T (Ap_k)}$$

Vector inner products, O(n)

$$x^{k+1} = x^k + \alpha_k p_k$$
$$r^{k+1} = r^k - \alpha_k Ap_k$$

Vector Adds, O(n)

$$p_{k+1} = r^{k+1} - \sum_{j=0}^{k} \frac{(Ar^{k+1})^T (Ap_j)}{(Ap_j)^T (Ap_j)} p_j$$

O(k) inner products, total cost O(nk)

Algorithm is $O(n^2)$ for Integral Equations even though # iters (k) is small!

Fast Matrix Vector Products

exactly compute Ap_k

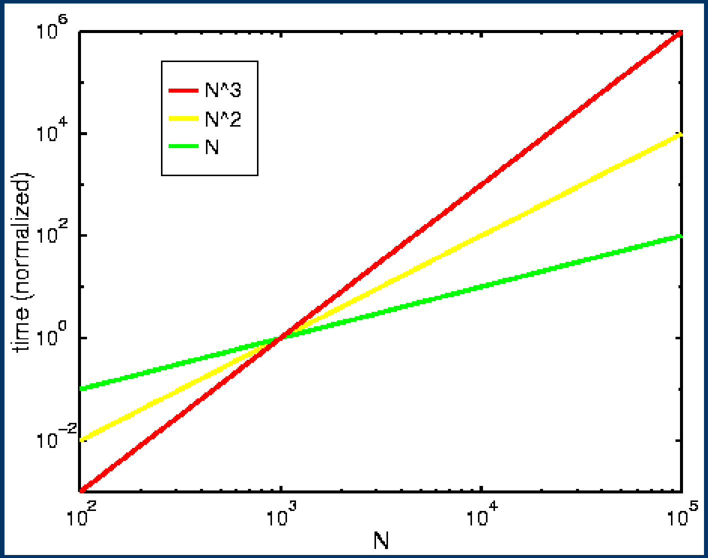
Dense Matrix-vector product costs $O(n^2)$

approximately compute Ap_k

Reduces Matrix-vector product costs to O(n) or O(nlogn)

Fast Solvers

Computational Costs



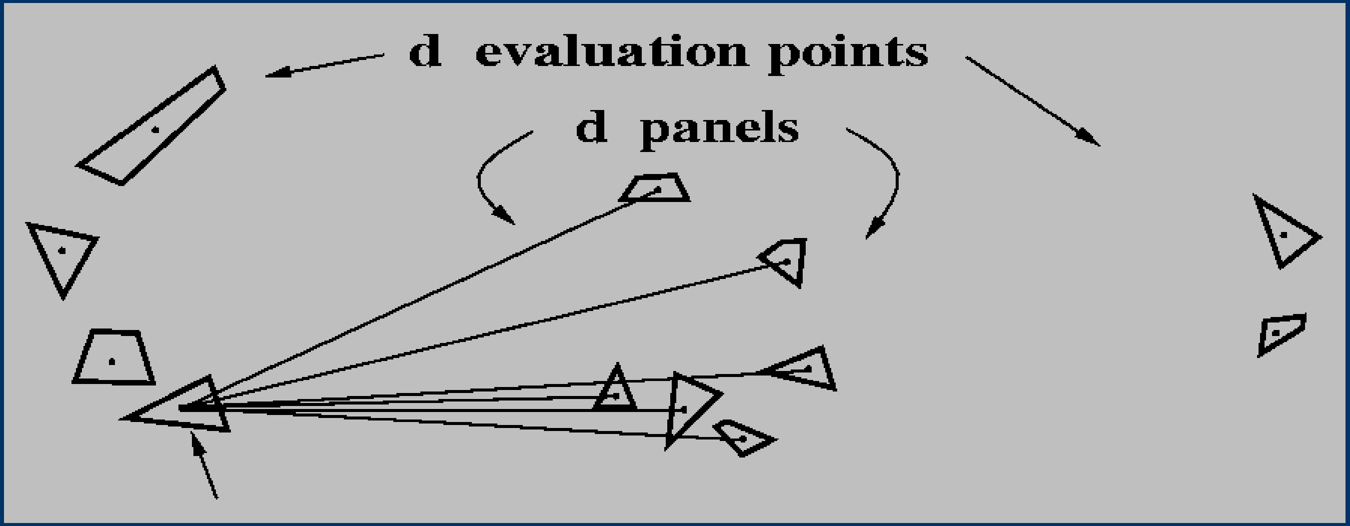
DEC 21164-333		
N	Gaussian Elim	"Fast" $O(N)$
	300 MFLOPS	30 MFLOPS
5e4	3 days,20GB	80sec, 130M
1e5	25 days,80GB	2.5min, 300M
5e5	8.8yrs,2TB	15min, 1.5GB

- Gaussian Elimination: $O(n^3)$ time, $O(n^2)$ memory
- GCR with direct M-V: $O(n^2)$ time, $O(n^2)$ memory
- Fast Methods: $O(n)$ time, $O(n)$ memory

Basic Multipole Concepts

Multipole Representation

Direct Potential Evaluation



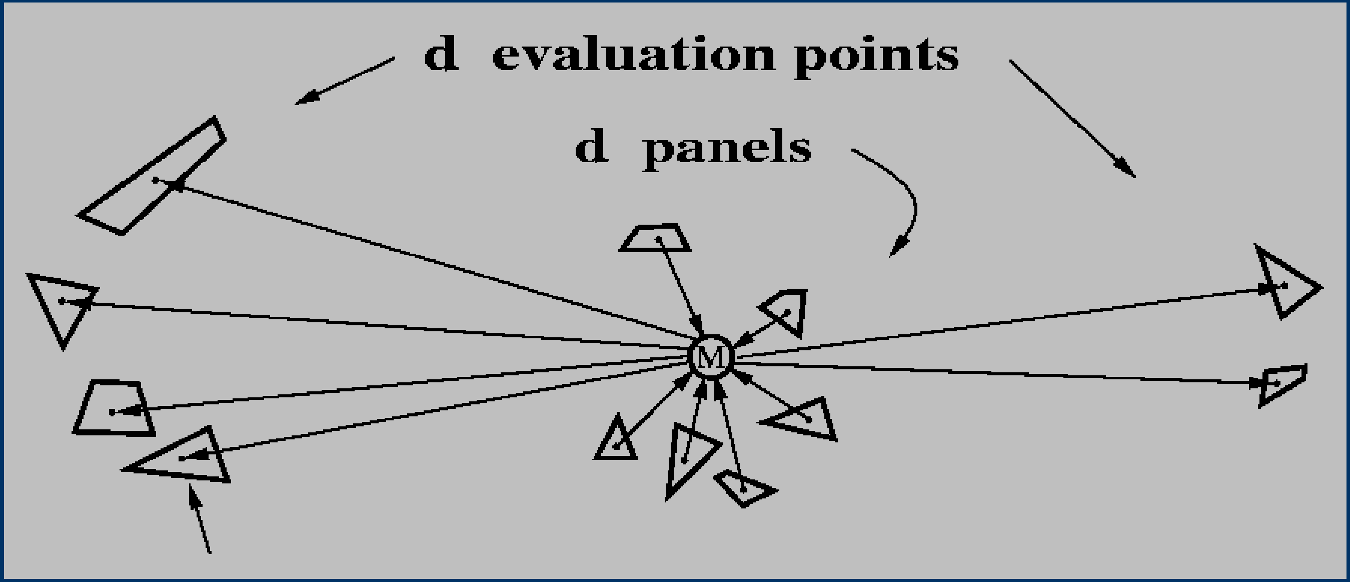
N1

- Potential at point i: $v_i(r_i, \phi_i, \theta_i) = \sum_{j=1}^{d} q_j P_{ij}$.
- Complete evaluation at d points costs d^2 operations.

Basic Multipole Concepts

Multipole Representation

Multipole Potential Evaluation



N2

- Approximate potential at point i:

$$v_i(r_i, \phi_i, \theta_i) \approx \sum_{j=0}^{order} \sum_{k=-j}^{j} \frac{M_j^k}{r_i^{j+1}} Y_j^k(\phi_i, \theta_i).$$

Basic Multipole Concepts

Multipole Representation

...Multipole Potential Evaluation

- Multipole coefficients function of panel charges:

$$M_j^k \triangleq \sum_{i=1}^{d} \frac{q_i}{A_i} \int_{\text{panel } i} \rho^j Y_j^{-k}(\alpha, \beta) dA.$$

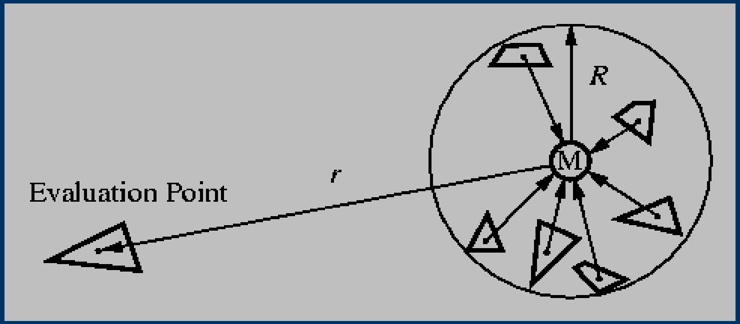
- Computing Multipole expansions costs order d operations.
- Each approximate potential evaluation costs order 1 operations.

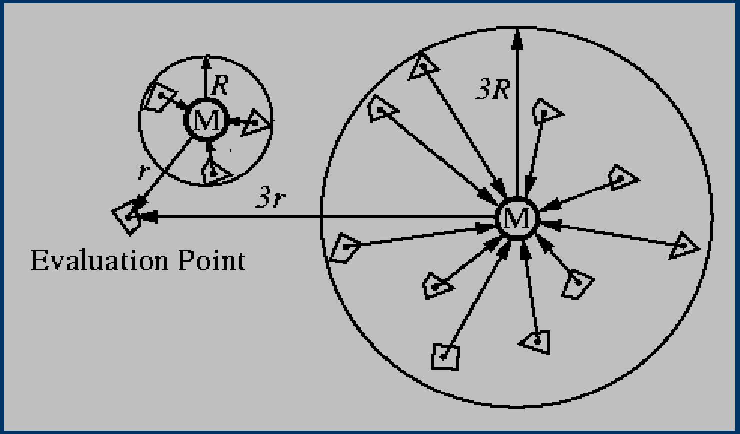
d potential evaluation due to d panels in order d operations

Basic Multipole Concepts

Multipole Representation

Scale Invariance of Error





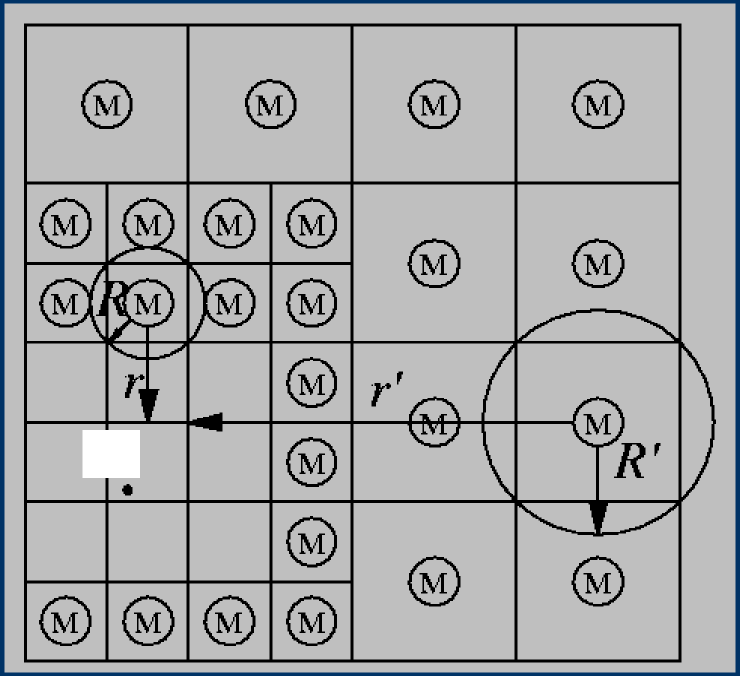
$$\text{Error} \leq K \left(\frac{R}{r}\right)^{order+1}$$

$$\text{Error} \leq K \left(\frac{3R}{3r}\right)^{order+1}$$

Basic Multipole Concepts

Multipole Representation

Multipole Algorithm Hierarchy



Hierarchy guarantees:

- Bounded error:

$$\text{Error} \leq K \left(\frac{R}{r}\right)^{order+1}$$

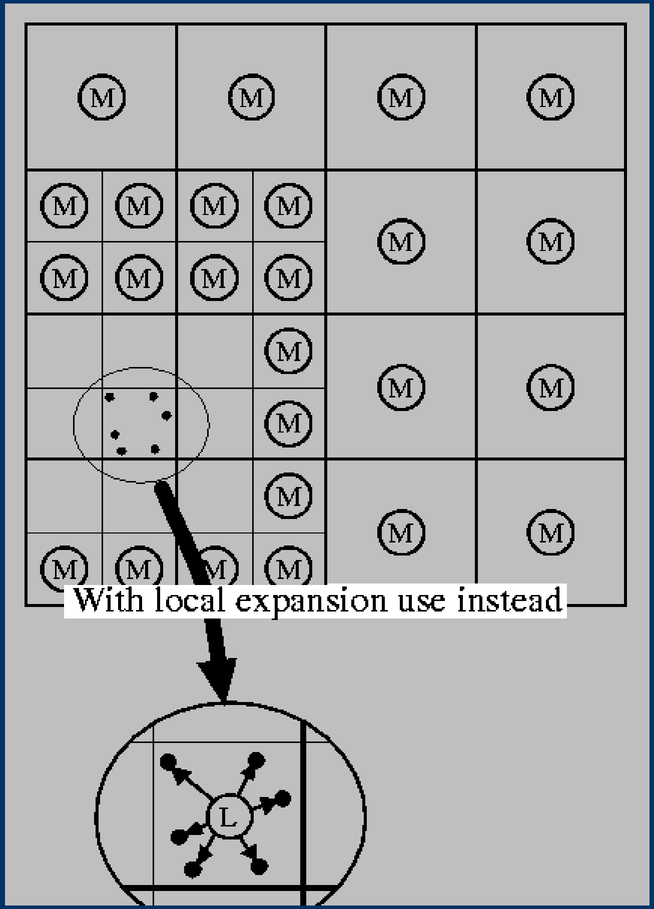
$$\leq K \left(\frac{1}{2}\right)^{order+1}$$

$order = 2$ yields one percent accuracy.

Multipole Optimizations

Local Expansions

Cost Reduction

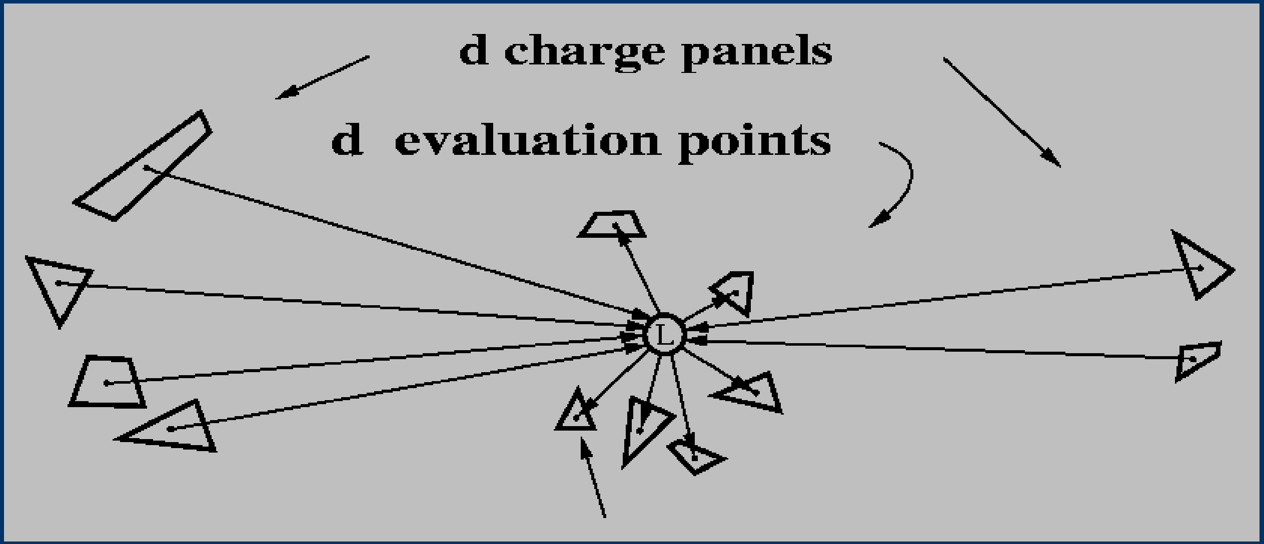


- Construct a local expansion to represent distant charge potentials.
- Evaluate a single local expansion, rather than many multipole expansions, at each evaluation point.

Multipole Optimizations

Local Expansions

Clustered Evaluations



N3

- Local expansion summarizes the influence of distant charge for clusters of evaluation points.

Multipole Optimizations

Local Expansions

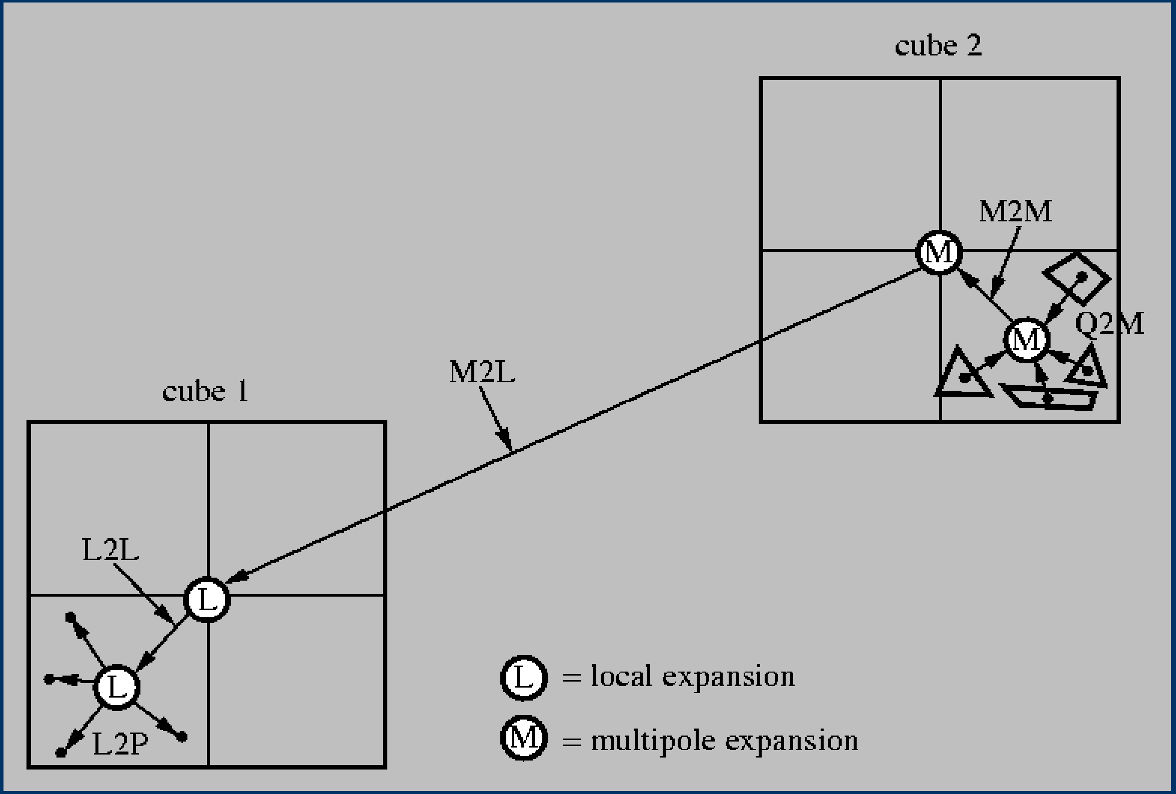
...Clustered Evaluations

- Gives $O(n)$ potential evaluation when combined with coalescing of charge done by multipole expansions.
- Approximate potential at point i:
  $v_i(r_i, \phi_i, \theta_i) \approx \sum_{j=0}^{order} \sum_{k=-j}^{j} L_j^k Y_j^k(\phi_i, \theta_i) r_i^j.$

Multipole Optimizations

Local Expansions

Summary of Operations



N4

Multipole Optimizations

Local Expansions

...Summary of Operations

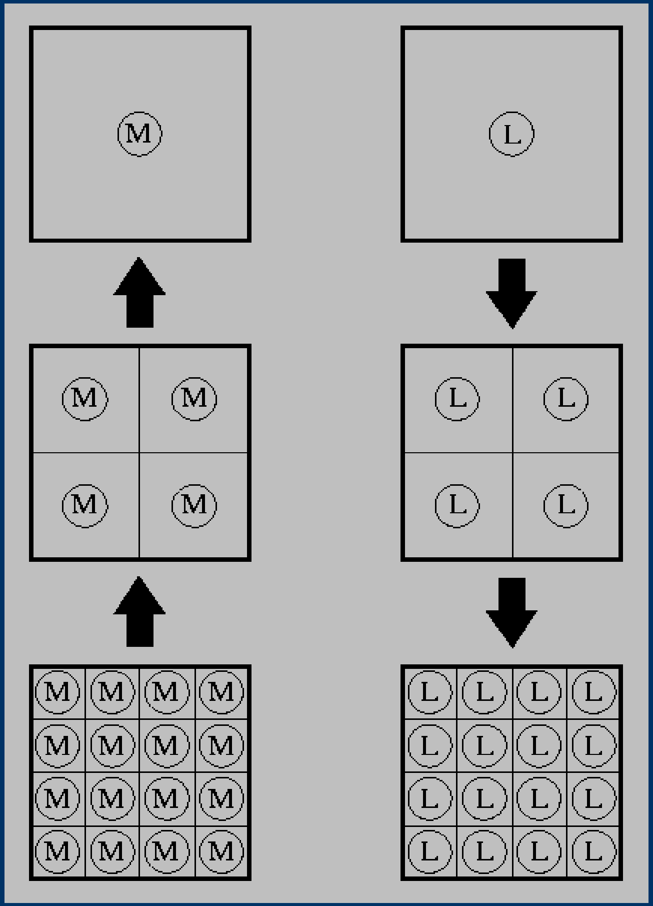
- Multipole and local expansions are built using complementary hierarchies.
- Complete calculation consists of:

1. Build multipoles (Upward Pass).
2. Build locals (Downward Pass).
3. Evaluate local expansions and nearby charge potential (Evaluation Pass).

Multipole Optimizations

Local Expansions

Hierarchy Construction

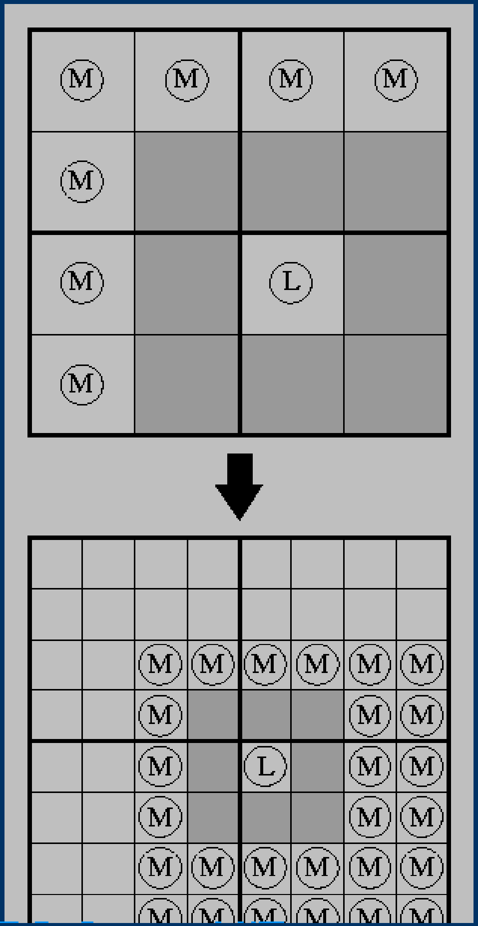


- First build the multipole expansions moving upward from child to parent.
- Then build the local expansions by moving downward from parent to child.
- Computation has a tree structure.

Multipole Optimizations

Local Expansions

Construction Details



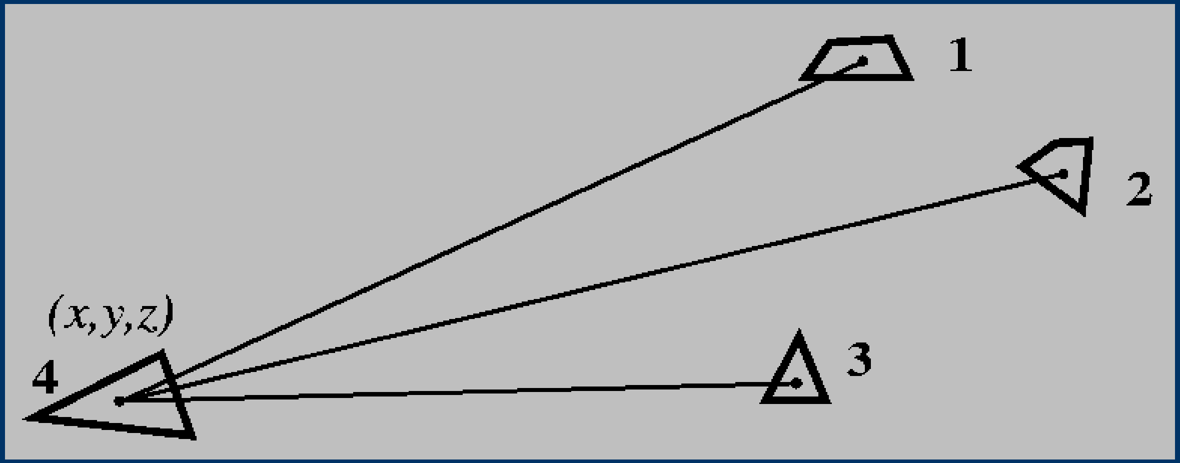
- Conversion of multipole expansions to local expansions.
- A child's local expansion is its parents local expansion plus conversions of multipole expansions in child's interaction range.

Multipole Optimizations

Adaptive Algorithm

Multipole Inefficiency

Direct Evaluation



N5

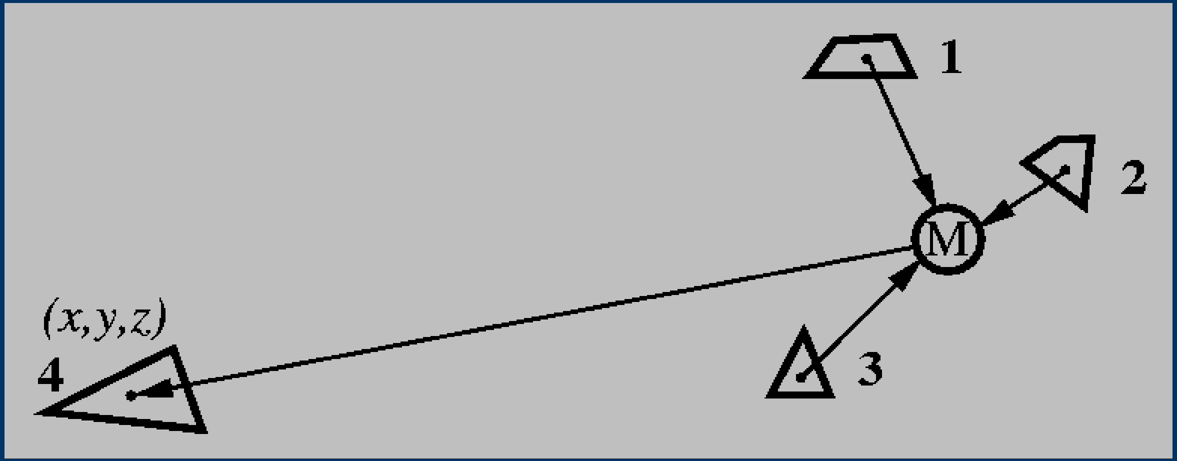
$$v_4(x, y, z) = q_1 P_{41} + q_2 P_{42} + q_3 P_{43}$$

Multipole Optimizations

Adaptive Algorithm

...Multipole Inefficiency

Multipole Evaluation



N6

$$v_4(x,y,z) \approx \bar{M}_0^0 \frac{1}{r} + \bar{M}_1^0 \frac{z}{r^3} - \bar{M}_1^1 \frac{x}{2r^3} - \tilde{M}_1^1 \frac{y}{2r^3}$$

Using Multipole MORE expensive than Direct.

Simple Adaptive Scheme

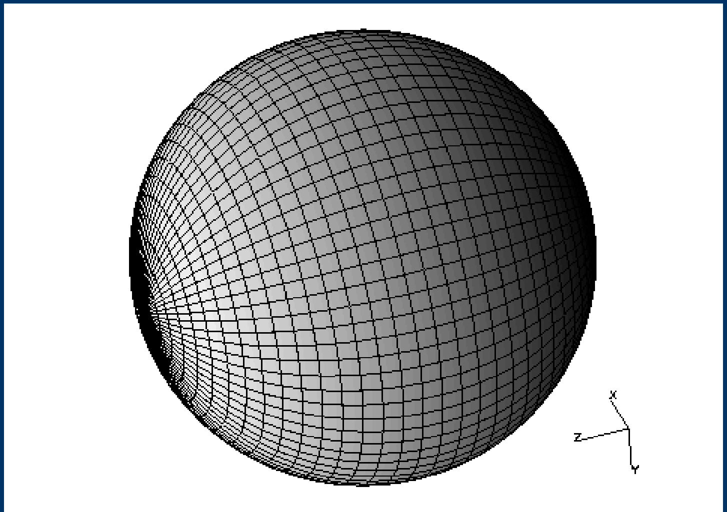
If there are fewer panels than multipole coefficients, calculate the panels' influence directly.

- Similarly, local expansions are not used if there are fewer evaluation points than local expansion coefficients.
- Retains $O(mn)$ complexity for nonuniform panel distributions.

Computational Examples

Translating Sphere

Potential Distribution

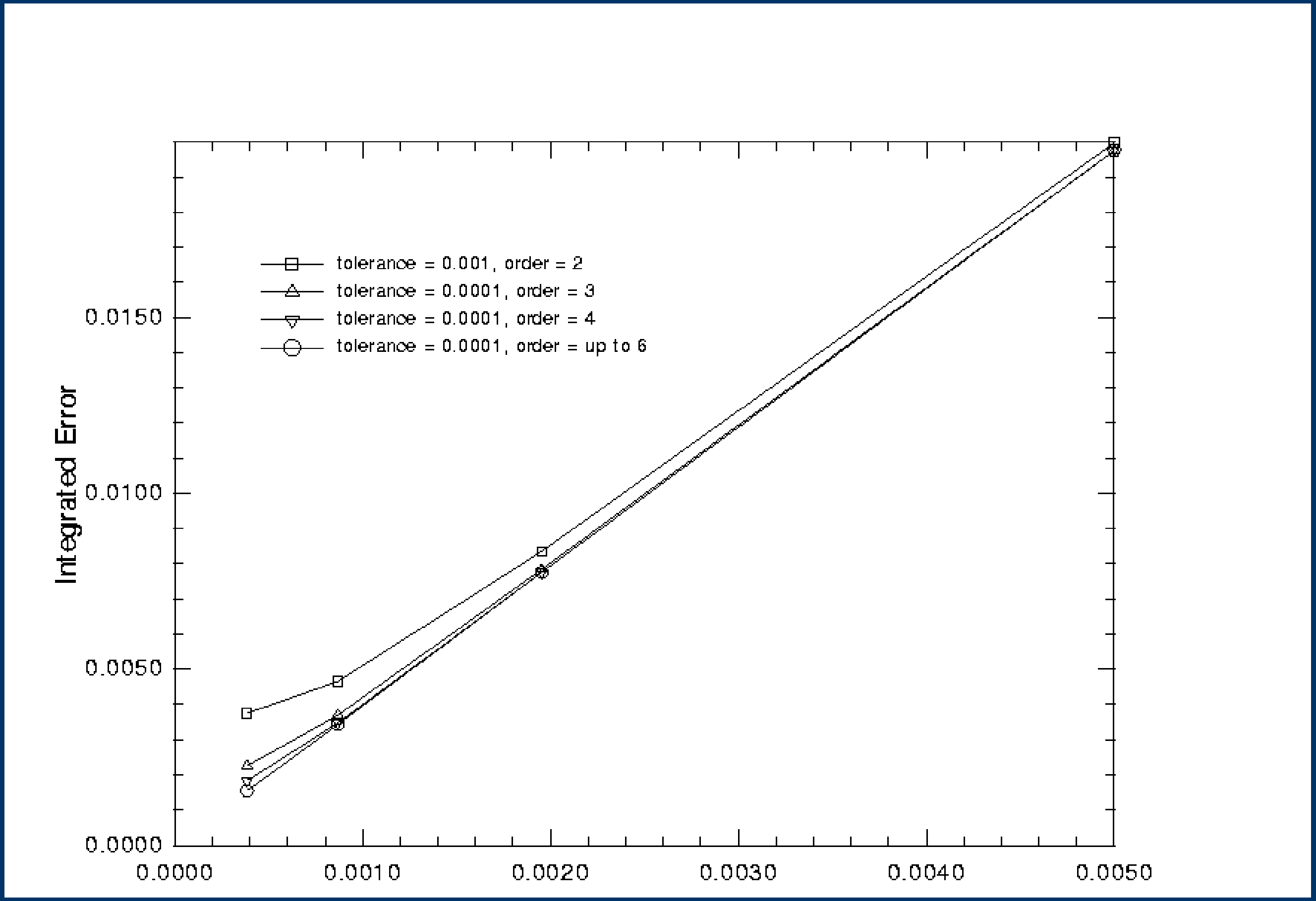


- Potential given by $\psi(x) = -\frac{x_3}{2\|x\|^3}$.
- Charge given by $\sigma(x) = \frac{-3}{8\pi} x_3$.

Computational Examples

Translating Sphere

Discretization Convergence



1/(# panels)

Computational Examples

Translating Sphere

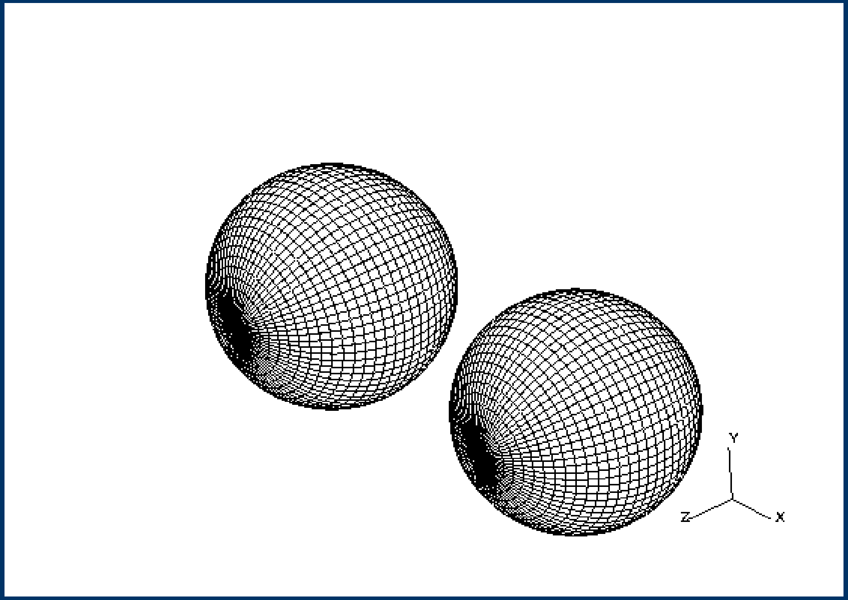
...Discretization Convergence

- Error should decay like $\frac{1}{n}$.
- Multipole approximations eventually interfere.
- Higher-order multipole expansions needed for higher accuracy.

Computational Examples

Two Sphere Example

Potential Distribution

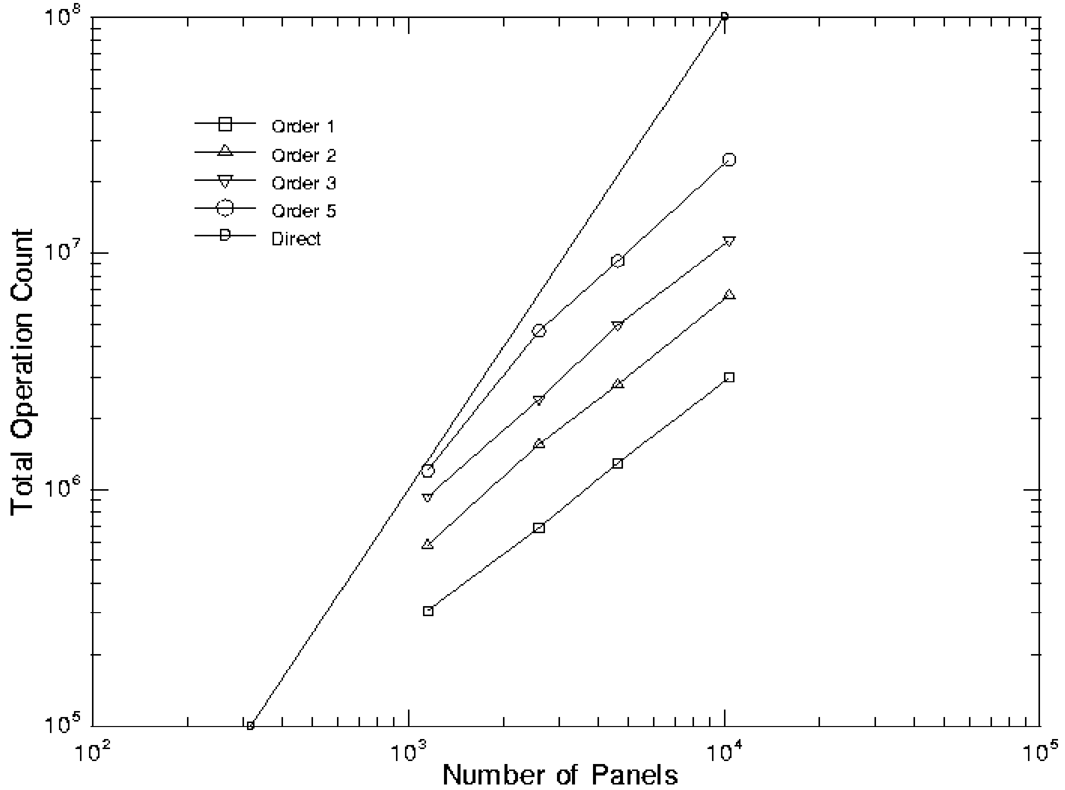


- Potential on each sphere: $\psi(x) = -\frac{x_3}{2\|x\|^3}$.
- Does not correspond to a simple physical problem.

Computational Examples

Two Sphere Example

Matrix-Vector Product Cost



Computational Examples

Two Sphere Example

...Matrix-Vector Product Cost

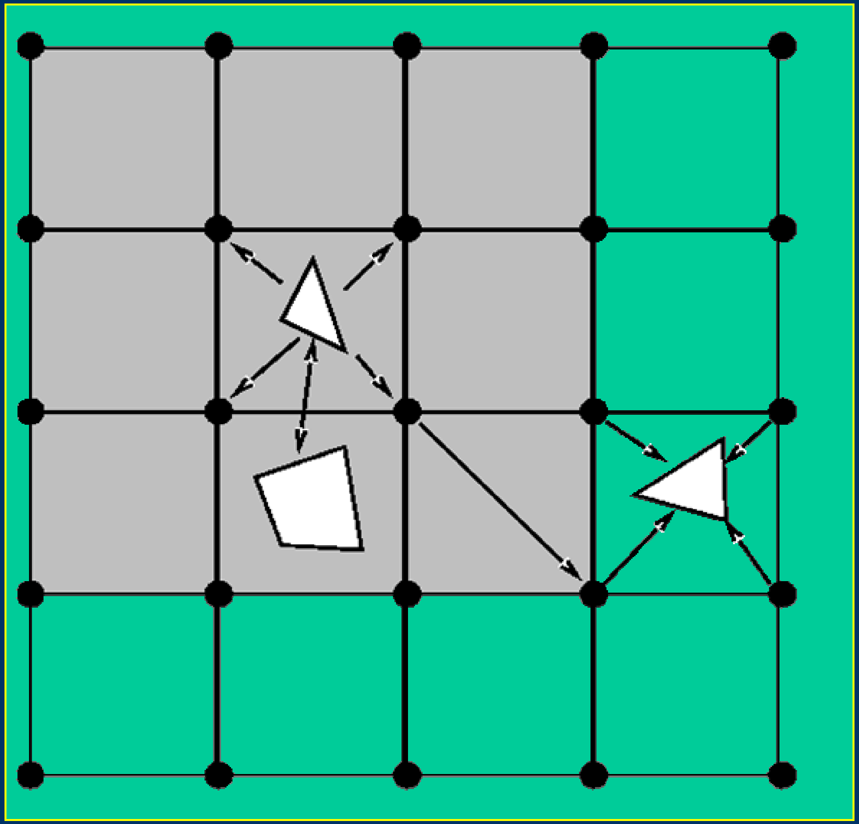
- Direct matrix-vector product cost increases like n^2.
- Multipole matrix-vector product cost increases like n.
- The slope for the multipole algorithm depends on accuracy.
- For order 2 expansions, breakpoint is about $n = 400$.

Complexity Summary

For an integral equation discretized with n panels:

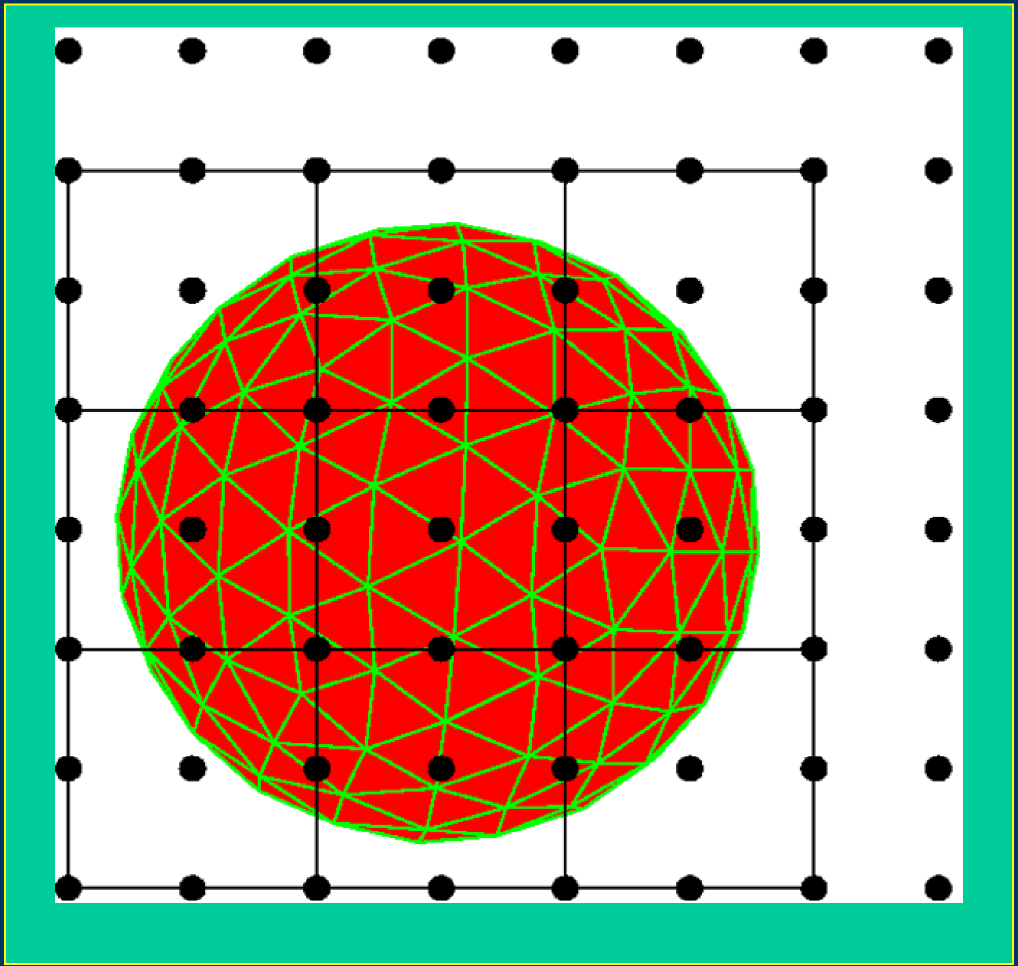
- Gaussian elimination: $O(n^3)$.
- GCR, direct M-V $O(n^2)$.
- Multipole accelerated GCR $O(mn)$.

Precorrected-FFT Acceleration



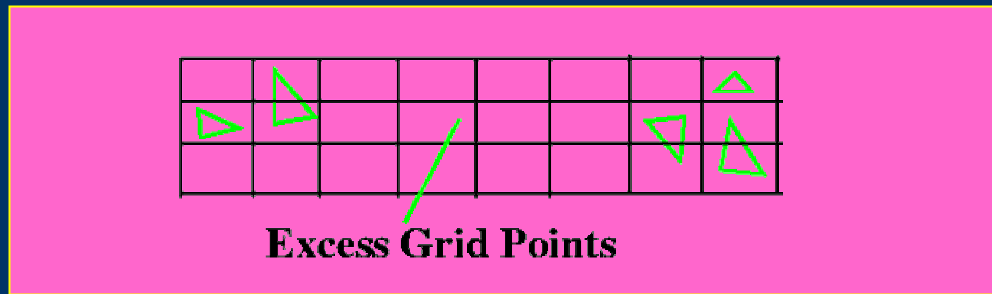
- Project panel charges on grid $q_g = Wq$.
- Compute using FFT's grid potentials due to grid charges $\psi_g = Hq_g$.
- Interpolate grid potentials onto panels $\psi = V\psi_g$.
- Compute near interactions directly $\psi_{a,b} = P_{a,b}q_b$.

The FFT Grid Selected To Balance Costs



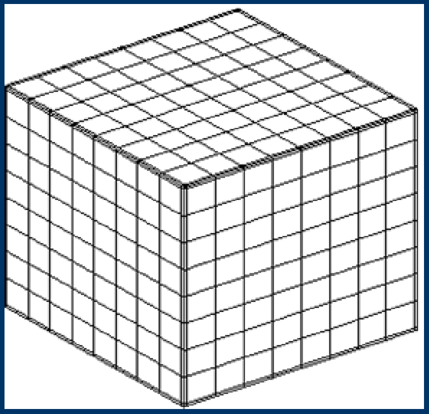
- Grid Selected So Direct Cost equals FFT Cost.
- Finer Problem Discretizations Usually Yield Finer Grids.

Inhomogeneity Problem

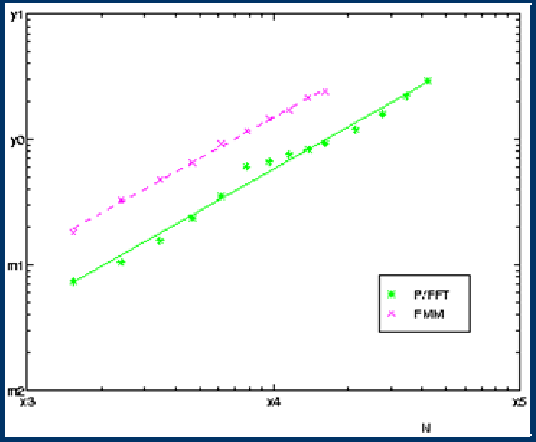


- Inhomogeneity - Empty Grid due to FFT - Inefficiency

Refining Cube Discretization - Worsening Inhomogenity



MV Product Time



Summary

Solving Discretized Integral Equations

- GCR plus Fast Matrix-Vector Products

Multipole Algorithms

- Multipole Representation.
- Basic Hierarchy

Algorithmic Improvements

- Local Expansions
- Adaptive Algorithms

Computational Results

Precorrected-FFT Algorithms