Introduction to Simulation - Lecture 23

Fast Methods for Integral Equations

Jacob White

Thanks to Deepak Ramaswamy, Michal Rewienski, and Karen Veroy
Exterior Problem in Electrostatics

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

\( \Psi \) is given on Surface

“Dirichelet Problem”

First Kind Integral Equation For Charge:

\[ \Psi(x) = \int \frac{1}{\|x - x'\|} \sigma(x') dS' \]

Charge Density

Green's Function
Drag Force in a Microresonator

Resonator

Discretized Structure

Computed Forces
Bottom View

Computed Forces
Top View

Courtesy of Werner Hemmert, Ph.D. Used with permission.
Integral Equation: \[ \Psi(x) = \int_{\text{surface}} \frac{1}{|x - x'|} \sigma(x') dS' \]

Represent \( \sigma(x) \approx \sum_{i=1}^{n} \alpha_i \varphi_i(x) \)  

\( \varphi_j(x) = 1 \) if \( x \) is on panel \( j \)  
\( \varphi_j(x) = 0 \) otherwise

Piecewise Constant Basis
3-D Laplace’s Equation

Put collocation points at panel centroids

\[ \Psi(x_{ci}) = \sum_{j=1}^{n} \alpha_j \int_{\text{panel } j} G(x_{ci}, x') dS' \]

\[ \begin{bmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{n,1} & \cdots & A_{n,n} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} \Psi(x_{c_1}) \\ \vdots \\ \Psi(x_{c_n}) \end{bmatrix} \]
Basis Function Approach
Calculating Matrix Elements

3-D Laplace’s Equation

Panel $j$

Collocation point $x_{ci}$

One point quadrature approximation

Four point quadrature approximation

Panel Area

Approximation

$$A_{i,j} = \int_{\text{panel } j} \frac{1}{x_{ci} - x'} dS'$$

$$A_{i,j} \approx \frac{0.25 \times \text{Area}}{x_{ci} - x_{\text{centroid}_j}}$$

$$A_{i,j} \approx \sum_{j=1}^{4} \frac{0.25 \times \text{Area}}{x_{ci} - x_{\text{point}_j}}$$
Calculating “Self-Term”

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

One point quadrature Approximation

\[ A_{i,i} \approx \frac{1}{4} \left| x_{c_i} - x' \right| \]

is an integrable singularity
3-D Laplace’s Equation

Basis Function Approach
Calculating “Self-Term”
Tricks of the trade

\[ A_{i,i} = \int_{\text{disk}} \frac{1}{\|x_{c_i} - x'\|} dS' + \int_{\text{rest of panel}} \frac{1}{\|x_{c_i} - x'\|} dS' \]

Integrate in two pieces

Disk Integral has singularity but has analytic formula

\[ \int_{\text{disk}} \frac{1}{\|x_{c_i} - x'\|} dS' = \int_{0}^{R} \int_{0}^{2\pi} \frac{1}{r} r dr d\theta = 2\pi R \]
1) If panel is a flat polygon, analytical formulas exist
2) Curve panels can be handled with projection
3-D Laplace’s Equation

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

For piecewise constant Basis

\[ \int_{b_i} \Psi(x) \, dS' = \sum_{j=1}^{n} \alpha_j \int_{\text{panel } i} \int_{\text{panel } j} \frac{1}{\|x - x'\|} \, dS' \, dS \]

\[
\begin{bmatrix}
A_{1,1} & \cdots & \cdots & A_{1,n} \\
\vdots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
A_{n,1} & \cdots & \cdots & A_{n,n}
\end{bmatrix}
\begin{bmatrix}
\alpha_1 \\
\vdots \\
\vdots \\
\alpha_n
\end{bmatrix}
= 
\begin{bmatrix}
b_1 \\
\vdots \\
\vdots \\
b_n
\end{bmatrix}
\]
3-D Laplace’s Equation

Basis Function Approach

Problem with dense matrix

Integral Equation Method Generate Huge Dense Matrices

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

Gaussian Elimination Much Too Slow!
The Generalized Conjugate Residual Algorithm

The kth step of GCR

Solving Discretized Integral Equations

compute \( A p_k \)

For discretized Integral equations, \( A \) is dense

\[
\alpha_k = \frac{(r^k)^T (A p_k)}{(A p_k)^T (A p_k)}
\]

Determine optimal stepsize in kth search direction

\[
x^{k+1} = x^k + \alpha_k p_k
\]

Update the solution and the residual

\[
r^{k+1} = r^k - \alpha_k A p_k
\]

Compute the new orthogonalized search direction

\[
p_{k+1} = r^{k+1} - \sum_{j=0}^{k} \frac{(A r^{k+1})^T (A p_j)}{(A p_j)^T (A p_j)} p_j
\]
The Generalized Conjugate Residual Algorithm

Complexity of GCR

Solving Discretized Integral Equations

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

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

Vector inner products, \(O(n)\)

Vector Adds, \(O(n)\)

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

\[
x^{k+1} = x^k + \alpha_k p_k
\]

\[
r^{k+1} = r^k - \alpha_k Ap_k
\]

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

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

compute \(Ap_k\)

\(N(n)\) inner products, \(O(n)\)
exactly compute $A p_k$
Dense Matrix-vector product costs $O(n^2)$

approximately compute $A p_k$
Reduces Matrix-vector product costs to $O(n)$ or $O(n \log n)$
Fast Solvers

Computational Costs

- 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

<table>
<thead>
<tr>
<th>N</th>
<th>Gaussian Elim</th>
<th>“Fast” $O(N)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>30 MFLOPS</td>
<td>30 MFLOPS</td>
</tr>
<tr>
<td>5e4</td>
<td>3 days, 20GB</td>
<td>80 sec, 130M</td>
</tr>
<tr>
<td>1e5</td>
<td>25 days, 80GB</td>
<td>2.5 min, 300M</td>
</tr>
<tr>
<td>5e5</td>
<td>8.8 yrs, 2TB</td>
<td>15 min, 1.5GB</td>
</tr>
</tbody>
</table>
• 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.
Approximate potential at point $i$:

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

Multipole Representation

Multipole coefficients function of panel charges:

\[ M^k_j \triangleq \sum_{i=1}^d \frac{q_i}{A_i} \int_{\text{panel}} \rho^j \gamma_{j-k} \alpha, \beta \, dA. \]

- 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

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

Error \leq K \left( \frac{3R}{3r} \right)^{order+1}
Hierarchy guarantees:

• Bounded error:

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

\[ \leq K \left( \frac{1}{2} \right)^{\text{order}+1} \]

\( \text{order} = 2 \) 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.
Local expansion summarizes the influence of distant charge for clusters of evaluation points.
Multipole Optimizations

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

Local Expansions
Summary of Operations

- L = local expansion
- M = multipole expansion

N4
Multipole Optimizations

- 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).

Local Expansions

...Summary of Operations
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.
- 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.
Direct Evaluation

\[ v_4(x, y, z) = q_1 P_{41} + q_2 P_{42} + q_3 P_{43} \]
Multipole Optimizations

Adaptive Algorithm

...Multipole Inefficiency

Multipole Evaluation

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

Using Multipole MORE expensive than Direct.
Multipole Optimizations

Adaptive Algorithm

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

- 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

$\frac{1}{(\# \text{ panels})}$

Integrated Error

- $\text{tolerance} = 0.001$, order = 2
- $\text{tolerance} = 0.0001$, order = 3
- $\text{tolerance} = 0.0001$, order = 4
- $\text{tolerance} = 0.0001$, order = up to 6
Computational Examples

- Error should decay like $\frac{1}{n}$.
- Multipole approximations eventually interfere.
- Higher-order multipole expansions needed for higher accuracy.
Potential on each sphere: \( \psi(x) = -\frac{x_3}{2\|x\|^3} \).

Does not correspond to a simple physical problem.
Two Sphere Example

Matrix-Vector Product Cost

![Graph showing the relationship between Total Operation Count and Number of Panels.]
Computational Examples

- 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 $n = 400$. 
For an integral equation discretized with $n$ panels:

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