Fast direct solvers for integral equations in complex geometries

Kenneth L. Ho

Joint work with Leslie Greengard

Courant Institute, New York University

NSF RTG Symposium (Sep 22, 2012)

Model problem

- Laplace's equation with Dirichlet boundary conditions:

$$\Delta u = 0 \quad \text{in } \Omega \quad , \quad u = f \quad \text{on } \partial\Omega$$

- Fundamental to many areas of mathematical physics
- Solve using a Green's function representation (double-layer potential):

$$u(\mathbf{r}) = \int_{\partial\Omega} \frac{\partial G}{\partial \nu_{\mathbf{s}}}(\mathbf{r}, \mathbf{s})\, \sigma(\mathbf{s})\, dA_{\mathbf{s}} \quad \text{in } \Omega$$

- Integral equation for unknown surface density σ:

$$-\frac{1}{2}\sigma(\mathbf{r}) + \int_{\partial\Omega} \frac{\partial G}{\partial \nu_{\mathbf{s}}}(\mathbf{r}, \mathbf{s})\, \sigma(\mathbf{s})\, dA_{\mathbf{s}} = f(\mathbf{r}) \quad \text{on } \partial\Omega$$

- Discretize: $Ax = b$
- Good: well-conditioned, high-order, dimensional reduction
- Bad: dense matrices, computational cost

Numerical considerations

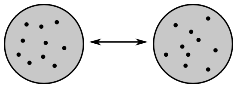
If $A \in \mathbb{C}^{N\times N}$, then:

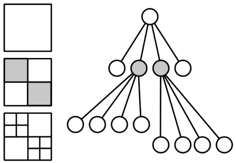
- Cost of applying A: $\mathcal{O}(N^2)$
- Cost of inverting A: $\mathcal{O}(N^3)$

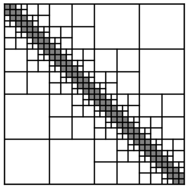
Fast algorithms are required!

Fortunately, such matrices are often structured.

- Analysis: non-oscillatory Green's functions have smooth far fields
- Algebra: off-diagonal matrix blocks are numerically low-rank







- Exploit smoothness with a hierarchical decomposition of space
- $\mathcal{O}(N \log N)$ matrix-vector multiplication (treecode, FMM, panel clustering)
- Combine with Krylov methods for fast iterative solvers

Beyond iterative solvers

Fast iterative solvers have been very successful, but they can still be inefficient:

- When A is ill-conditioned (multiphysics, singular geometries)
- When $Ax = b$ must be solved with many right-hand sides b or many perturbations of a base matrix A (optimization, design, time marching)

An alternative: fast direct solvers (construct A^{-1}).

- Robust: insensitive to conditioning, always works
- Fast solves and inverse updates following initial factorization

Various approaches in recent years:

- $\mathscr{H}$-matrices (Hackbusch, Börm, Grasedyck, Bebendorf et al.)
- HSS matrices (Chandrasekaran, Gu, Xia, Li et al.)
- Skeletonization (Martinsson, Rokhlin, Gillman, Greengard et al.)
    - BIEs in 2D
    - One-level BIEs in 3D

Fast direct solver for integral equations

Here, we describe a multilevel skeletonization-based solver in general dimension.

For BIEs:

	2D	3D
precomp	$\mathcal{O}(N)$	$\mathcal{O}(N^{3/2})$
solve	$\mathcal{O}(N)$	$\mathcal{O}(N \log N)$

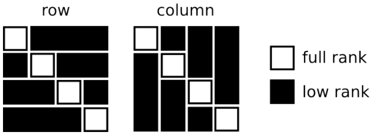
Main ideas/take-home messages :

- Kernel-independent: Laplace, Stokes, Yukawa, low-frequency Helmholtz
- Robust to geometry (e.g., boundary vs. volume, dimensionality)
- User-specified precision: trade accuracy for speed
- Naturally exposes the underlying sparsity of integral equation matrices
- Transparently takes advantage of sparse direct solver development
- Very fast solve times, beating the FMM by factors of 100–1000
- Simple framework: easy to analyze, implement, and optimize
- *Can be improved* (see Eduardo's talk coming up next)

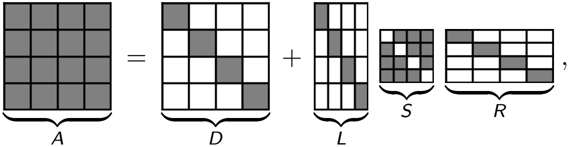
Block separable matrices

A block matrix A is block separable if

$$\underbrace{\begin{bmatrix} \times & \times \\ \times & \times \end{bmatrix}}_{A_{ij}} = \underbrace{\begin{bmatrix} \times \\ \times \end{bmatrix}}_{L_i} \underbrace{\begin{bmatrix} \times \end{bmatrix}}_{S_{ij}} \underbrace{\begin{bmatrix} \times & \times \end{bmatrix}}_{R_j}, \quad i \neq j.$$



Then



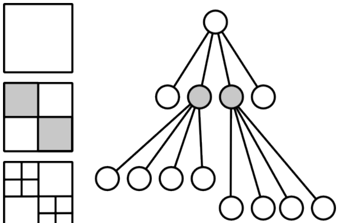
so $Ax = b$ is equivalent to the structured sparse system

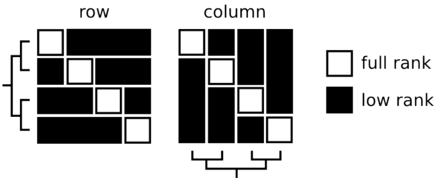
$$\begin{bmatrix} D & L & \\ R & & -I \\ & -I & S \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$

with $z \equiv Rx$ and $y \equiv Sz$. Factor using UMFPACK, SuperLU, WSMP, etc.

Hierarchically block separable matrices

Integral equation matrices are, in fact, hierarchically block separable, i.e., they are block separable at every level of an octree-type ordering.





In this setting, much more powerful algorithms can be developed.

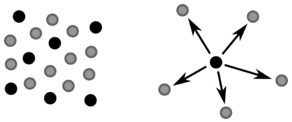
Interpolative decomposition

An interpolative decomposition of a rank-k matrix is a factorization

$$\underbrace{A}_{m\times n} = \underbrace{B}_{m\times k}\underbrace{P}_{k\times n},$$

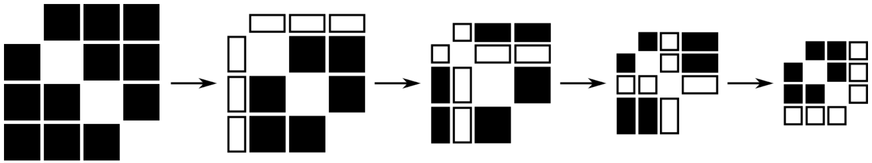
where B is a column-submatrix of A (with $\|P\|$ small).

- The ID compresses the column space; to compress the row space, apply the ID to A^{T}. We call the retained rows and columns skeletons.
- Adaptive algorithms can compute the ID to any specified precision $\epsilon > 0$.
- Related factorizations: SVD, RRQR, pseudoskeleton (CUR), ACA



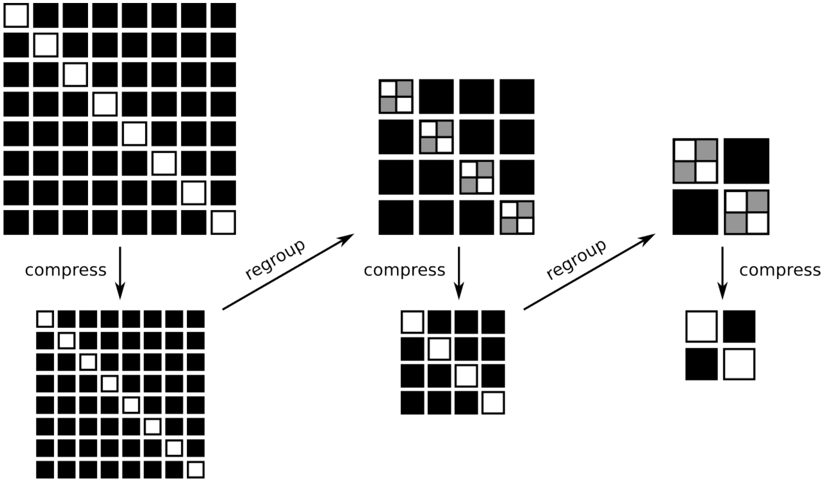
One-level matrix compression

- Compress the row space of each off-diagonal block row.
  Let the L_i be the corresponding row interpolation matrices.
- Compress the column space of each off-diagonal block column.
  Let the R_j be the corresponding column interpolation matrices.
- Approximate the off-diagonal blocks by $A_{ij} \approx L_i S_{ij} R_j$ for $i \neq j$.
- S is a skeleton submatrix of A



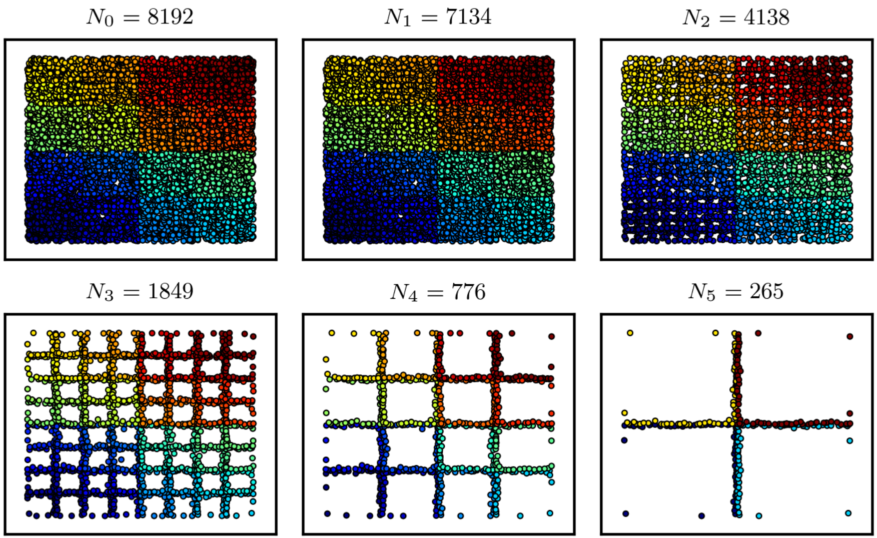
Skeletonization

Multilevel matrix compression



Recursive skeletonization

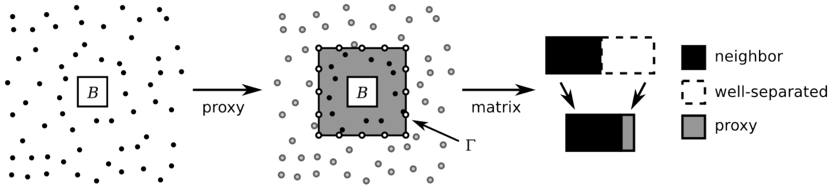
Data sparsification



$$G(\mathbf{r}, \mathbf{s}) = -\frac{1}{2\pi} \log |\mathbf{r} - \mathbf{s}| \ , \quad \epsilon = 10^{-3}$$

Accelerated compression for PDEs

- General compression algorithm is global and so at least $\mathcal{O}(N^2)$
- For potential fields, use Green's theorem to accelerate
- Represent well-separated interactions via a local proxy surface
- Can be generalized to non-PDE kernels using sparse grids



Compressed matrix representation

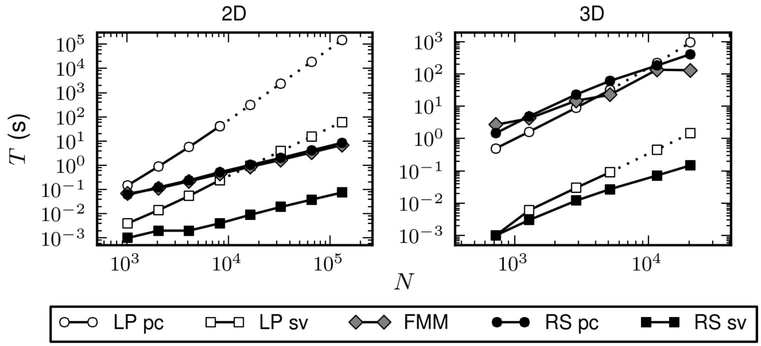
- Telescoping formula:

$$A \approx D^{(1)} + L^{(1)} \left[D^{(2)} + L^{(2)} \left(\cdots D^{(\lambda)} + L^{(\lambda)} S R^{(\lambda)} \cdots \right) R^{(2)} \right] R^{(1)}$$

- Efficient storage, fast matrix-vector multiplication (generalized FMM)
- Structured sparse inversion:

$$\begin{bmatrix} D^{(1)} & L^{(1)} & & & & & \\ R^{(1)} & & -I & & & & \\ & -I & D^{(2)} & L^{(2)} & & & \\ & & R^{(2)} & \ddots & \ddots & & \\ & & & \ddots & D^{(\lambda)} & L^{(\lambda)} & \\ & & & & R^{(\lambda)} & & -I \\ & & & & & -I & S \end{bmatrix} \begin{bmatrix} x \\ y^{(1)} \\ z^{(1)} \\ \vdots \\ \vdots \\ y^{(\lambda)} \\ z^{(\lambda)} \end{bmatrix} = \begin{bmatrix} b \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

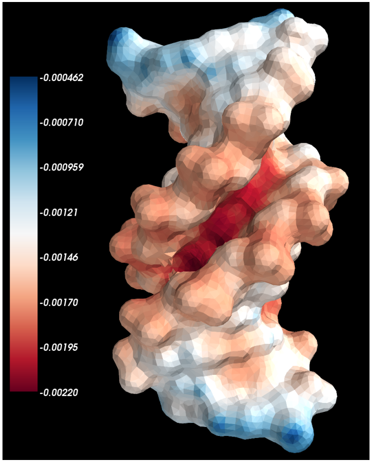
Laplace BIE solver



- Less memory-efficient than FMM/GMRES
- Each solve is extremely fast (in elements/sec)

ϵ	10^{-3}	10^{-6}	10^{-9}
2D	3.3×10^6	2.0×10^6	1.7×10^6
3D	6.0×10^5	1.4×10^5	6.2×10^4

Poisson electrostatics

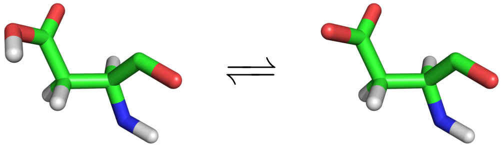


$$
\begin{aligned}
-\Delta\varphi &= 0 && \text{in } \Omega_0 \\
-\Delta\varphi &= \frac{1}{\varepsilon_1}\sum_i q_i \delta\left(\mathbf{r}-\mathbf{r}_i\right) && \text{in } \Omega_1 \\
[\varphi] &= \left[\varepsilon \frac{\partial\varphi}{\partial\nu}\right] = 0 && \text{on } \Sigma
\end{aligned}
$$

N	7612	19752
FMM/GMRES	12.6 s	26.9 s
RS precomp	151 s	592 s
RS solve	0.03 s	0.08 s

Break-even point: 10–25 solves

Protein pK_a calculations



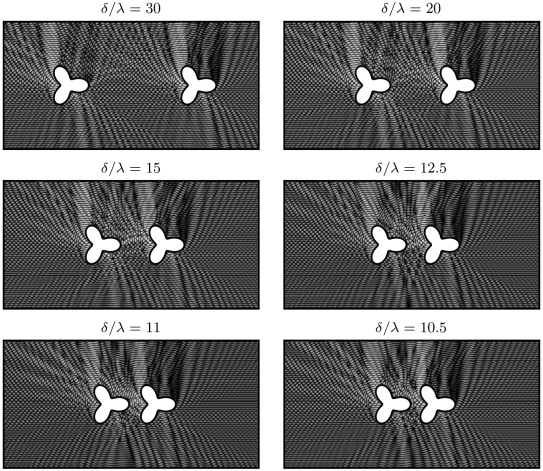
- Characterizes free energy of ionization reaction
- Important for binding affinities, enzymatic activities, structural properties
- Main bottleneck: solving the same BIE with multiple right-hand sides
- Use recursive skeletonization with linearized Poisson-Boltzmann model

Results:

- DoFs: 10,000–30,000
- Precomp time: 1–2 hr
- Solve time: 10 s
- Speedup: 2–5×

name	PDB ID	residues	atoms	sites
BPTI	4PTI	58	891	18
OMTKY3	2OVO	56	813	15
HEWL	2LZT	129	1965	30
RNase A	3RN3	124	1865	34
RNase H	2RN2	155	2474	53

Multiple scattering



- Each object: 10λ

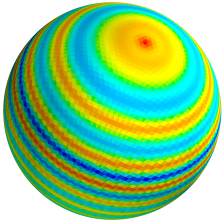
$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

- FMM/GMRES with block preconditioner via RS

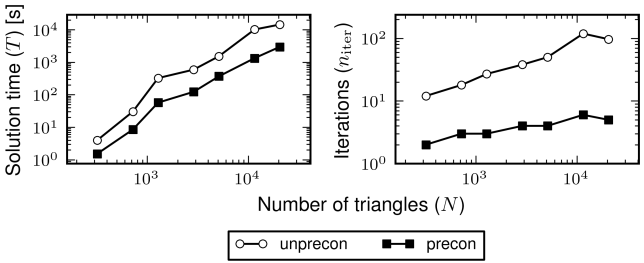
$$\begin{bmatrix} A_{11}^{-1} & \\ & A_{22}^{-1} \end{bmatrix}$$

- Unprecon: 700 iterations
- Precon: 10 iterations
- $50\times$ speedup

Helmholtz preconditioning



- $N = 20480$, 10λ
- Precondition with low-precision inverse ($\epsilon = 10^{-3}$)
- Iterate for full accuracy ($\epsilon = 10^{-12}$)
- Unprecon: 190 iterations
- Precon: 6 iterations
- $10\times$ speedup



Summary

Complexities in d dimensions (BIEs in $d+1$ dimensions):

$$\text{precomp} \sim \begin{cases} N & \text{if } d = 1, \\ N^{3(1-1/d)} & \text{if } d > 1, \end{cases} \qquad \text{solve} \sim \begin{cases} N & \text{if } d = 1, \\ N \log N & \text{if } d = 2, \\ N^{2(1-1/d)} & \text{if } d > 2 \end{cases}$$

- Mild assumptions: low-rank off-diagonal blocks, Green's theorem
- Can generalize to asymptotically smooth kernels
- Very fast solves following precomputation ($\sim$ 0.1 s)
- Highly effective for preconditioning
- Reveals connection with sparse matrices (Chandrasekaran, Gu et al.)
- Naturally parallelizable via block-sweep structure
- Extensions: low-rank updates, geometric perturbations, least squares
- Similar ideas also apply for PDE formulations (Xia, Gillman et al.)

Relevant publications

Published/in press:

- KL Ho, *Fast direct methods for molecular electrostatics*, PhD thesis, New York University, 2012.
- KL Ho and L Greengard, *A fast direct solver for structured linear systems by recursive skeletonization*, SIAM J Sci Comput, in press.

Submitted/in preparation:

- KL Ho and L Greengard, *A fast direct least squares algorithm for hierarchically block separable matrices*, in preparation.
- KL Ho, S Jung, and L Greengard, *Protein pK_a calculations using a fast direct boundary element solver*, in preparation.
- KL Ho, J Sifuentes, Z Gimbutas, and L Greengard, *Approximate inverse preconditioning for integral equations on two-dimensional domains*, in preparation.