A fast direct solver for structured matrices

Kenneth L. Ho and *Leslie Greengard*

Courant Institute of Mathematical Sciences
New York University, 251 Mercer Street, New York, NY 10012
greengard@courant.nyu.edu

Introduction

We have developed a fast direct solver for structured linear systems based on multilevel matrix compression. Starting with a hierarchically block-separable matrix [2], we embed an approximation of the original matrix into a larger, but highly structured sparse one. The resulting representation allows for efficient storage, fast matrix-vector multiplication, fast matrix factorization, and fast application of the inverse.

The algorithm proceeds in two phases: a precomputation phase, consisting of matrix compression and factorization, followed by a solution phase to apply the matrix inverse. For boundary integral equations which are not too oscillatory, e.g., based on the Green's functions for the Laplace or low-frequency Helmholtz equations, both phases typically have complexity $\mathcal{O}(N)$ in two dimensions, where N is the number of discretization points. In our current three-dimensional implementation, the corresponding costs are $\mathcal{O}(N^{3/2})$ and $\mathcal{O}(N \log N)$ for precomputation and solution, respectively.

Extensive numerical experiments show a speedup of ~ 100 for the solution phase over modern fast multipole methods; however, the cost of precomputation remains high. Thus, the solver is particularly suited to problems where large numbers of iterations would be required.

(1) Reduced sensitivity to the conditioning of A.
(2) Fast application of A^{-1} to multiple right-hand sides.
(3) Efficient handling of low-rank perturbations of A.

Several closely related efforts:

- H-matrices (Hackbusch et al.)
- HSS matrices (Gu, Chandrasekaran, et al.)
- skeletonization-based schemes (Martinsson and Rokhlin, G-, Gueyffier, Martinsson, Rokhlin).

Structured matrices

Let $A\mathbf{x} = \mathbf{f}$ be written in the form

$$\begin{aligned} A_{11}\mathbf{x}_1 + A_{12}\mathbf{x}_2 + \cdots + A_{1p}\mathbf{x}_p &= \mathbf{f}_1 \\ A_{21}\mathbf{x}_1 + A_{22}\mathbf{x}_2 + \cdots + A_{2p}\mathbf{x}_p &= \mathbf{f}_2 \\ &\cdots \\ A_{p1}\mathbf{x}_1 + A_{p2}\mathbf{x}_2 + \cdots + A_{pp}\mathbf{x}_p &= \mathbf{f}_p\,, \end{aligned} \tag{1}$$

where $\mathbf{x}_i$ and $\mathbf{f}_i$ are complex vectors of dimension N_i and $A_{ij} \in \mathbf{C}^{N_i \times N_j}$. Assuming $N = \sum_{i=1}^{p} N_i$, solution of the full linear system by classical Gaussian elimination is well-known to require $O(N^3)$ work.

Definition The matrix A is said to be *block-separable* [2] if each off-diagonal submatrix A_{ij} can be decomposed as the product of three low-rank matrices:

$$A_{ij} = L_i S_{ij} R_j \tag{2}$$

where $L_i \in \mathbf{C}^{N_i \times k_i^r}$, $S_{ij} \in \mathbf{C}^{k_i^r \times k_j^c}$, and $R_j \in \mathbf{C}^{k_j^c \times N_j}$, with $k_i^r \ll N_i$ and $k_j^c \ll N_j$. Note that in (2), the left matrix L_i depends only the index i and the right matrix R_j depends only on the index j.

$$A = D + LSR, \tag{3}$$

where

$$D = \begin{bmatrix} A_{11} & & \\ & \ddots & \\ & & A_{pp} \end{bmatrix} \tag{4}$$

is a block-diagonal $N \times N$ matrix consisting of the diagonal blocks of A, and

$$S = \begin{bmatrix} 0 & S_{12} & \cdots & S_{1p} \\ S_{21} & 0 & \cdots & S_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ S_{p1} & S_{p2} & \cdots & 0 \end{bmatrix} \tag{5}$$

is a dense $K_\mathrm{r} \times K_\mathrm{c}$ matrix, where $K_\mathrm{r} = \sum_{i=1}^{p} k_i^r$ and $K_\mathrm{c} = \sum_{i=1}^{p} k_i^c$, with zero diagonal blocks. L and R are given by

$$L = \begin{bmatrix} L_1 & & \\ & \ddots & \\ & & L_p \end{bmatrix}, \qquad R = \begin{bmatrix} R_1 & & \\ & \ddots & \\ & & R_p \end{bmatrix}. \tag{6}$$

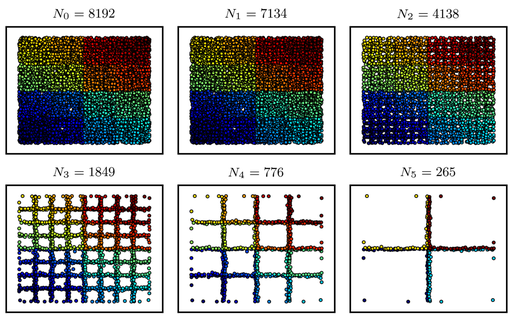
They are block-diagonal $N \times K_\mathrm{r}$ and $K_\mathrm{c} \times N$ matrices, respectively. This is, of course, an effective compression of A only when $K_\mathrm{r}, K_\mathrm{c} \ll N$. A useful feature of the representation (3) is that it permits rapid inversion. To see this, let $z \equiv Rx$ and let $y \equiv Sz$. We can then write the original system in the form

$$\begin{bmatrix} D & L & \\ R & & -I \\ & -I & S \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}. \tag{7}$$

Such a factorization can be computed recursively if the matrix supports the block-separable property at each level of the hierarchy. The result is a telescoping representation

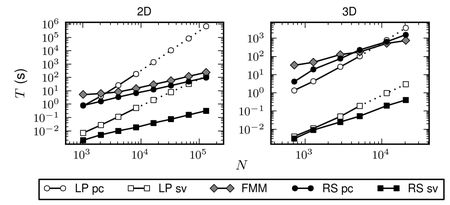
$$A \approx D^{(1)} + L^{(1)} \left[D^{(2)} + L^{(2)} \left(\cdots D^{(\lambda)} + L^{(\lambda)} S R^{(\lambda)} \cdots \right) R^{(2)} \right] R^{(1)}$$

Example: Suppose $\phi(x) = \int K(x,y)\,\rho(y)\,dy$ with $K(x,y) = log\|x-y\|$, discretized at $N = 8192$ points in the unit square, and compress to relative precision $\epsilon = 10^{-3}$ using a five-level quadtree-based scheme.



Data sparsification by recursive skeletonization. At each level, the surviving skeletons are shown, colored by block index, with the total number of skeletons remaining given by N_l.

Numerical results



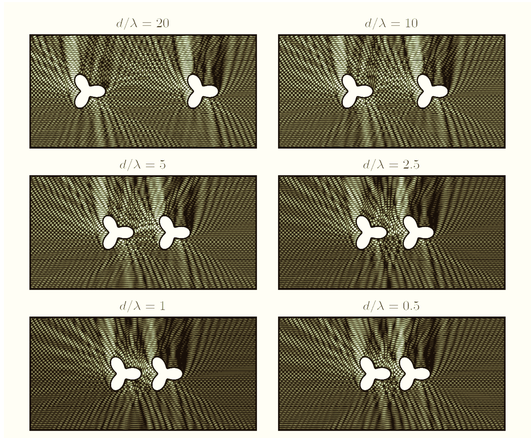
CPU times for solving the Helmholtz equation in various cases at low frequency ($\omega = 10$ in 2D and $\omega \approx 3.18$ in 3D) using LAPACK/ATLAS, FMM/GMRES, and recursive skeletonization; The precision was set to $\epsilon = 10^{-9}$ in 2D and $\epsilon = 10^{-6}$ in 3D.

Multiple scattering

We show how our solver can be combined with fast iterative methods to great effect by considering a multiple scattering problem.

$$\left(\Delta + k^2\right) u = 0 \quad \text{in } \mathbf{R}^2 \setminus \bigcup_{i=1}^{p} \Omega_i$$

with, e.g., Neumann (sound-hard) boundary conditions



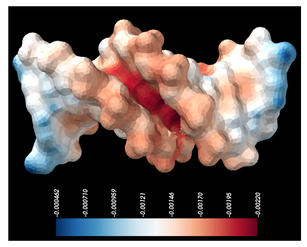
Real part $\Re(u)$ of the pressure field in response to a vertical plane wave for various multiple scattering geometries characterized by the separation distance d/λ in wavelengths.

Molecular electrostatics

Letting Σ be a molecular surface, a classical model for the electrostatic potential φ is Poisson's equation:

$$-\nabla \cdot [\varepsilon(x)\,\nabla\varphi(x)] = \sum_{i=1}^{n} q_i \delta(x - x_i)\,, \tag{8}$$

We generated a molecular surface for a short segment of DNA consisting of $N = 19752$ triangles and solved an integral equation for the induced polarization charge to precision $\epsilon = 10^{-3}$.



The total solution time was ≈ 592 s, with an inverse application time of ≈ 0.08 s, to be compared with ≈ 27 s using FMM/GMRES.

Conclusions

We have developed a multilevel matrix compresssion and solution algorithm. The matrix structure required is fairly general and relies only on the assumption that the matrix have low-rank off-diagonal blocks. It is competitive with fast iterative methods based on FMM/GMRES in both 2D and 3D, provided that the kernel of the integral equation is not too oscillatory.

A principal limitation of all current hierarchical direct solvers is the growth in skeleton size in 3D, which prohibits the scheme from achieving optimal $\mathcal{O}(N)$ complexity. New methods to curtail this growth are under development, at least for non-oscillatory cases.

Finally, although all numerical results are reported here for a single CPU core, the algorithm is naturally suited for HPC systems.

References

[0] K. L. Ho and L. Greengard, *A fast direct solver for structured linear systems by recursive skeletonization*, submitted.

[1] L. Greengard, D. Gueyffier, P.-G. Martinsson, and V. Rokhlin, *Fast direct solvers for integral equations in complex three-dimensional domains*, Acta Numer, 18 (2009), pp. 243–275.

[2] A. Gillman, P. Young, and P.G. Martinsson, *A direct solver with $O(N)$ complexity for integral equations on one-dimensional domains*, in review.

[3] P.-G. Martinsson and V. Rokhlin, *A fast direct solver for boundary integral equations in two dimensions*, J Comput Phys, 205 (2005), pp. 1–23.

[4] S. Chandrasekaran, M. Gu, and T. Pals, *A fast ULV decomposition solver for hierarchically semiseparable representations*, SIAM J Matrix Anal Appl, 28 (2006), pp. 603–622.

[5] W. Hackbusch and S. Börm, *Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices*, Computing, 69 (2002), pp. 1–35.

Acknowledgments

Supported by DOE under contract DEFGO288ER25053. Thanks to Zydrunas Gimbutas and NSF (K.L.H.).