A fast direct solver by structured matrix compression

Kenneth L. Ho and Leslie Greengard

Courant Institute of Mathematical Sciences
New York University, New York, NY, USA

Introduction

The direct solution of dense linear systems by classical methods requires $\mathcal{O}(N^3)$ operations, where N is the system size. This can be prohibitive when N is large. For system matrices with special structure, however, this cost can be dramatically decreased. Here, we present a fast algorithm for hierarchically structured matrices based on multilevel compression that exposes their underlying sparsity. Our method is an extension of the skeletonization scheme of Martinsson and Rokhlin, embedded in a linear algebraic framework that allows for significant generalization. It is particularly useful for the solution of the integral equations of potential theory, where, compared with iterative fast multipole-based approaches, its primary advantages are its reduced sensitivity to conditioning and its extremely fast solve time following precomputation.

Algorithm

Definition 1. A block matrix A is *block separable* if each off-diagonal block is the product of three low-rank matrices: $A_{ij} = L_i S_{ij} R_j$ for $i \neq j$.

Then

$$A = D + L\ S\ R,$$

so the linear system $Ax = b$ can be written as

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

where $z \equiv Rx$ and $y \equiv Sz$. This can be solved efficiently using standard sparse direct solver technology.

Definition 2. A block matrix A with a tree τ on its block indices is *hierarchically block separable (HBS)* if it is block separable on every level of τ.

This describes the matrix discretizations of many non-oscillatory Green's function integral operators (e.g., Laplace, Stokes, low-frequency Helmholtz), sorted by an orthtree.

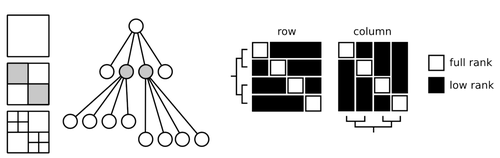

Figure 1: *Hierarchically block separable matrix structure.*

For such matrices, if we use the interpolative decomposition for low-rank compression, then the *skeleton matrix* S is also HBS and can be represented in the same form. This leads to a multilevel compression and inversion scheme:

- telescoping compressed representation:

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

- multilevel sparse embedding:

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

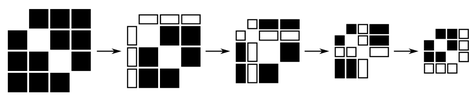

Figure 2: *One-level compression by skeletonization.*

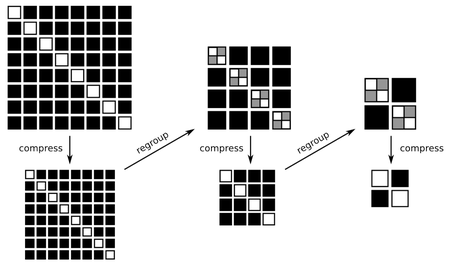

Figure 3: *Multilevel compression by recursive skeletonization.*

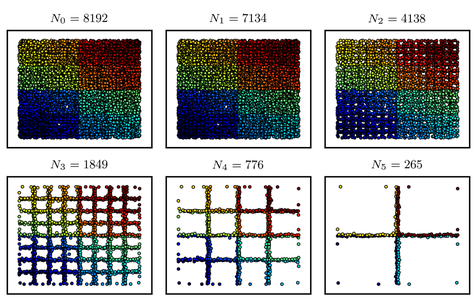

Figure 4: *Data sparsification (logarithmic interactions, $\epsilon = 10^{-3}$).*

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 efficient compression, we use *proxy surfaces* to represent global interactions locally.

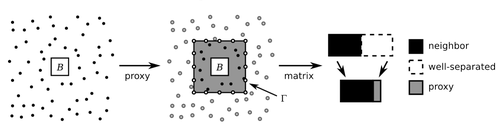

Figure 5: *Interactions are decomposed into contributions from the near field, which are handled directly, and the far field, which are represented using a proxy surface.*

Table 1: *Computational complexities for solving integral equations in d-dimensional domains.*

d	precomp	solve
1	$\mathcal{O}(N)$	$\mathcal{O}(N)$
2	$\mathcal{O}(N^{3/2})$	$\mathcal{O}(N \log N)$
3	$\mathcal{O}(N^2)$	$\mathcal{O}(N^{4/3})$

Numerical results

We implemented our algorithm in Fortran, using UMFPACK for sparse inversion. All results are for a single 2.66 GHz processor.

Laplace equation

- second-kind boundary integral formulation:

$$-\frac{1}{2}\sigma(\boldsymbol{r}) + \int_\Gamma \frac{\partial G}{\partial \nu_{\boldsymbol{s}}}(\boldsymbol{r}, \boldsymbol{s})\, dS_{\boldsymbol{s}} = f(\boldsymbol{r})$$

- 2D: $2:1$ ellipse (points), $\epsilon = 10^{-9}$
- 3D: unit sphere (triangles), $\epsilon = 10^{-6}$
- compare with LAPACK/ATLAS and FMM/GMRES

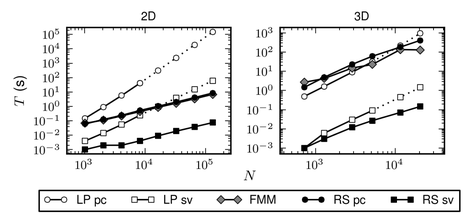

Figure 6: *CPU times for solving the boundary integral formulation of the Laplace equation.*

Molecular electrostatics

- piecewise constant dielectric Poisson:

$$-\nabla \cdot (\varepsilon \nabla \varphi) = \sum_i q_i \delta(\boldsymbol{r} - \boldsymbol{r}_i)$$

- DNA: $N = 19752$ triangles, $\epsilon = 10^{-3}$
- precomp: 592 s; solve: 0.08 s; FMM/GMRES: 27 s

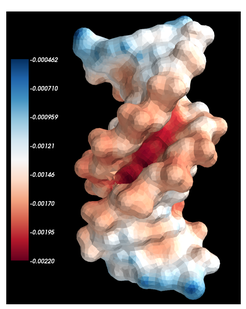

Figure 7: *Surface potential of DNA.*

Multiple scattering

- sound-hard acoustics on two scatterers:

$$\left(\Delta + k^2\right) u = 0 \quad \text{in } \mathbb{R}^2 \setminus (\Omega_1 \cup \Omega_2)$$

- block system:

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

- FMM/GMRES with RS preconditioner ($N = 1024$, $\epsilon = 10^{-6}$)

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

- precon: 8 s (10 iterations each)
- unprecon: 474 s (700 iterations each)

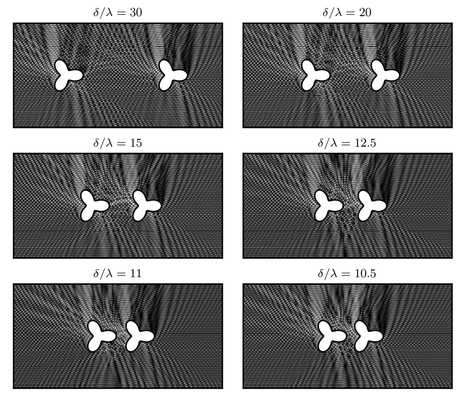

Figure 8: *Acoustic pressure field energy.*

Generalizations and conclusions

We have developed a matrix compression algorithm that exposes the sparsity of structured dense matrices and provides a platform for fast matrix algebra. As a direct solver, its principal novelties are its generality (kernel independence) and its low cost per solve (after precomputation), often beating the FMM by several orders of magnitude. Thus, it is especially suited to problems involving multiple right-hand sides. Our overall approach is similar to that of work on $\mathscr{H}$-matrices and HSS matrices.

Several extensions are immediate, including low-rank inverse updates (e.g., for local geometric perturbations) via preconditioning or the Sherman-Morrison-Woodbury formula, and the solution of least squares problems using a sparse QR factorization. Research is ongoing.

Acknowledgments

We thank Zydrunas Gimbutas and Mark Tygert. This work was supported by NYU, NSF (DGE-0333389), DOE (DEFG0288ER25053), and AFOSR (NSSEFF FA9550-10-1-0180).