Hierarchical interpolative factorization

Kenneth L. Ho (Stanford)

Joint work with Lexing Ying

SIAM AN 2014

Introduction

Integral equations associated with elliptic PDEs:

$$a(x)u(x) + \int_{\Omega} K(x,y)u(y) \, d\Omega(y) = f(x)$$

- Many applications in science and engineering
- Interested in 2D/3D, complex geometry
- Discretize \rightarrow structured linear system Au = f



Goal: fast and accurate algorithms for the discrete operator

- Fast matrix-vector multiplication, fast direct solver, good preconditioner
- Ideally, fast matrix factorization
- Linear or quasilinear complexity, high practical efficiency

Related work

Fast matrix-vector multiplication

- O(N) or $O(N \log N)$ using FMM, treecode, $\mathcal{H}/\mathcal{H}^2$ -matrices
- ► Fast iterative solver by combining with GMRES, CG, etc.



Fast direct solver

- HSS matrices/recursive skeletonization
 - O(N) in 1D, $O(N^{3/2})$ in 2D, $O(N^2)$ in 3D
- \mathcal{H} -matrices: $O(N \log^{\alpha} N)$ but with a large constant
- HSS/RS with structured matrix algebra: O(N) in 2D
 - Corona, Martinsson, Zorin (2013)
- More recent ideas: Siva's talk later this session

Overview

Hierarchical interpolative factorization

- ▶ RS + recursive dimensional reduction
- Same idea as using structured algebra but much simpler
- ► New matrix sparsification framework, generalized LU decomposition
- Linear or quasilinear complexity, small constants
- ▶ Works for 2D/3D, adaptive geometry

Tools: sparse elimination, interpolative decomposition, skeletonization

Sparse elimination

Let

$$A = \begin{bmatrix} A_{pp} & A_{pq} \\ A_{qp} & A_{qq} & A_{qr} \\ & A_{rq} & A_{rr} \end{bmatrix}.$$



(Think of A as a sparse matrix.) If A_{pp} is nonsingular, define

$$R_{\rho}^{*} = \begin{bmatrix} I & & \\ -A_{q\rho}A_{\rho\rho}^{-1} & I & \\ & & I \end{bmatrix}, \quad S_{\rho} = \begin{bmatrix} I & -A_{\rho\rho}^{-1}A_{\rhoq} & \\ & I & \\ & & I \end{bmatrix}$$

so that

$$R_{\rho}^*AS_{\rho} = \begin{bmatrix} A_{\rho\rho} & & \\ & * & A_{qr} \\ & A_{rq} & A_{rr} \end{bmatrix}.$$

- DOFs p have been eliminated
- Interactions involving r are unchanged

Interpolative decomposition

If $A_{:,q}$ is numerically low-rank, then there exist

- ▶ skeleton (\hat{q}) and redundant (\check{q}) columns partitioning $q = \hat{q} \cup \check{q}$
- ▶ an interpolation matrix T_q

such that

$$A_{:,\check{q}} \approx A_{:,\hat{q}} T_q.$$

Essentially a pivoted QR written slightly differently:

$$\begin{aligned} A_{:,(\hat{q},\check{q})} &= \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix} \approx Q_1 \begin{bmatrix} R_{11} & R_{12} \end{bmatrix} \\ &\implies A_{:,\check{q}} \approx Q_1 R_{12} = \underbrace{Q_1 R_{11}}_{A_{:,\check{q}}} \underbrace{(R_{11}^{-1} R_{12})}_{T_q} \end{aligned}$$

Interactions between separated regions are low-rank.

Skeletonization

▶ Efficient elimination of redundant DOFs from dense matrices

• Let
$$A = \begin{bmatrix} A_{pp} & A_{pq} \\ A_{qp} & A_{qq} \end{bmatrix}$$
 with A_{pq} and A_{qp} low-rank

• Apply ID to
$$\begin{bmatrix} A_{qp} \\ A_{pq}^* \end{bmatrix}$$
: $\begin{bmatrix} A_{q\check{p}} \\ A_{\check{p}\check{q}}^* \end{bmatrix} \approx \begin{bmatrix} A_{q\hat{p}} \\ A_{\check{p}\check{q}}^* \end{bmatrix} T_p \implies \begin{array}{c} A_{q\check{p}} \approx A_{q\hat{p}} T_p \\ A_{\check{p}\check{q}} \approx T_p^* A_{\check{p}\check{q}} \end{bmatrix}$

$$\blacktriangleright \text{ Reorder } A = \begin{bmatrix} A_{\check{p}\check{p}} & A_{\check{p}\check{p}} & A_{\check{p}\check{q}} \\ A_{\hat{p}\check{p}} & A_{\hat{p}\check{p}} & A_{\hat{p}\check{q}} \\ A_{q\check{p}} & A_{q\check{p}} & A_{q\check{q}} \end{bmatrix}, \text{ define } Q_p = \begin{bmatrix} I & & \\ -T_p & I & \\ & & I \end{bmatrix}$$

► Sparsify via ID:
$$Q_p^* A Q_p \approx \begin{bmatrix} * & * \\ * & A_{\hat{p}\hat{p}} & A_{\hat{p}q} \\ A_{q\hat{p}} & A_{qq} \end{bmatrix}$$

► Sparse eliminate: $R_p^* Q_p^* A Q_p S_{\hat{p}} \approx \begin{bmatrix} * & & \\ * & A_{\hat{p}q} \\ A_{q\hat{p}} & A_{qq} \end{bmatrix}$

Algorithm: recursive skeletonization factorization

```
Build quadtree/octree.

for each level \ell = 0, 1, 2, \dots, L from finest to coarsest do

Let C_{\ell} be the set of all cells on level \ell.

for each cell c \in C_{\ell} do

Skeletonize remaining DOFs in c.

end for

end for
```

Old algorithm (RS) in new factorization form



domain

matrix



RSF in 2D: level 2



RSF in 2D: level 3



RSF in 3D: level 0



RSF in 3D: level 1



RSF in 3D: level 2



RSF analysis

Skeletonization operators:

$$U_{\ell} = \prod_{c \in C_{\ell}} Q_{c} R_{\check{c}}, \quad V_{\ell} = \prod_{c \in C_{\ell}} Q_{c} S_{\check{c}}$$
$$Q_{c} = \begin{bmatrix} I & & \\ * & I & \\ & & I \end{bmatrix}, \quad R_{\check{c}}, S_{\check{c}} = \begin{bmatrix} I & * & \\ & I & \\ & & I \end{bmatrix}$$

Block diagonalization:

$$D \approx U_{L-1}^* \cdots U_0^* A V_0 \cdots V_{L-1}$$

Generalized LU decomposition:

$$A \approx U_0^{-*} \cdots U_{L-1}^{-*} D V_{L-1}^{-1} \cdots V_0^{-1}$$
$$A^{-1} \approx V_0 \cdots V_{L-1} D^{-1} U_L^* \cdots U_0^*$$

► Fast direct solver or preconditioner

RSF analysis

The cost is determined by the skeleton size.

	1D	2D	3D
Skeleton size Factorization cost Solve cost	$O(\log N)$ $O(N)$ $O(N)$	$O(N^{1/2}) \\ O(N^{3/2}) \\ O(N \log N)$	$O(N^{2/3}) \ O(N^2) \ O(N^{4/3})$

Question: How to reduce the skeleton size in 2D and 3D?

- Skeletons cluster near cell interfaces (Green's theorem)
- Exploit skeleton geometry by further skeletonizing along interfaces
- Dimensional reduction

_

Algorithm: hierarchical interpolative factorization for IEs in 2D

Build quadtree.

```
for each level \ell = 0, 1, 2, \dots, L from finest to coarsest do

Let C_{\ell} be the set of all cells on level \ell.

for each cell c \in C_{\ell} do

Skeletonize remaining DOFs in c.

end for

Let C_{\ell+1/2} be the set of all edges on level \ell.

for each cell c \in C_{\ell+1/2} do

Skeletonize remaining DOFs in c.

end for

end for
```



domain

matrix









HIF-IE in 2D: level 3/2



HIF-IE in 2D: level 2





matrix

domain

HIF-IE in 2D: level 5/2



HIF-IE in 2D: level 3



RSF vs. HIF-IE in 2D



RSF

HIF-IE



Build octree for each level $\ell = 0, 1, 2, \dots, L$ from finest to coarsest **do** Let C_{ℓ} be the set of all cells on level ℓ . for each cell $c \in C_{\ell}$ do Skeletonize remaining DOFs in c. end for Let $C_{\ell+1/3}$ be the set of all faces on level ℓ . for each cell $c \in C_{\ell+1/3}$ do Skeletonize remaining DOFs in c. end for Let $C_{\ell+2/3}$ be the set of all edges on level ℓ . for each cell $c \in C_{\ell+2/3}$ do Skeletonize remaining DOFs in c. end for end for



HIF-IE in 3D: level $1/3\,$



HIF-IE in 3D: level $2/3\,$



HIF-IE in 3D: level 1



HIF-IE in 3D: level 4/3



HIF-IE in 3D: level 5/3



HIF-IE in 3D: level 2





HIF-IE analysis

► 2D: $A \approx U_0^{-*} U_{1/2}^{-*} \cdots U_{L-1/2}^{-*} D V_{L-1/2}^{-1} \cdots V_{1/2}^{-1} V_0^{-1}$ $A^{-1} \approx V_0 V_{1/2} \cdots V_{L-1/2} D^{-1} U_{L-1/2}^* \cdots U_{1/2}^* U_0^*$

► 3D:
$$A \approx U_0^{-*} U_{1/3}^{-*} U_{2/3}^{-*} \cdots U_{L-1/3}^{-*} DV_{L-1/3}^{-1} \cdots V_{2/3}^{-1} V_{1/3}^{-1} V_0^{-1}$$
$$A^{-1} \approx V_0 V_{1/3} V_{2/3} \cdots V_{L-1/3} D^{-1} U_{L-1/3}^* \cdots U_{2/3}^* U_{1/3}^* U_0^*$$

Conjecture:	Skeleton size: Factorization cost: Solve cost:	$\frac{O(\log N)}{O(N)}$
		0(11)

Numerical results in 2D

First-kind volume IE on the unit square with

$$a(x) \equiv 0, \quad K(x, y) = -\frac{1}{2\pi} \log ||x - y||.$$



- rskelf2 (white), hifie2 (black)
- ▶ Factorization time (\circ), solve time (\Box), memory (\diamond)
- Precision $\epsilon = 10^{-6}$

Numerical results in 3D

Second-kind boundary IE on the unit sphere with



- rskelf3 (white), hifie3 (gray), hifie3x (black)
- ► Factorization time (○), solve time (□), memory (◊)
- Precision $\epsilon = 10^{-3}$

Numerical results in 3D

First-kind volume IE on the unit cube with



- rskelf3 (white), hifie3 (black)
- ► Factorization time (◦), solve time (□), memory (◊)
- Precision $\epsilon = 10^{-3}$

Conclusions

- ▶ Efficient factorization of structured operators in 2D and 3D
 - Fast matrix-vector multiplication
 - Fast direct solver at high accuracy, preconditioner otherwise
 - Empirical linear complexity but no proof yet
- Sparsification and elimination (skeletonization) via the ID
- Dimensional reduction by alternating between cells, faces, and edges
- Can be viewed as adaptive numerical upscaling
- **Extensions**: PDEs, $A^{1/2}$, log det A, diag A^{-1}
- Naturally parallelizable, block-sweep structure
- > Perspective: structured dense matrices can be sparsified very efficiently
- Can borrow directly from sparse algorithms, e.g., RSF = MF
- What other features of sparse matrices can be exploited?

MATLAB codes available at https://github.com/klho/FLAM/.

Proxy compression

- Main cost of algorithm is computing IDs
- ► Global operation can be reduced to local operation using Green's theorem
- Suffices to compress against neighbors plus "proxy" surface
- Crucial for beating $O(N^2)$ complexity



Second-kind IEs

- IEs of the form $u(x) + \int_{\Omega} K(x, y)u(y) d\Omega(y) = f(x)$
- High contrast in diagonal vs. off-diagonal entries
- Mixing of cell, face, edge in HIF-IE leads to error
- Need to use effective precision $O(\epsilon/N)$
- Quasilinear complexity estimates:

	2D	3D
Factorization cost Solve cost	$O(N \log N)$ $O(N \log \log N)$	$\frac{O(N\log^6 N)}{O(N\log^2 N)}$

Second-kind results in 2D

Second-kind volume IE on the unit square with

$$a(x) \equiv 1$$
, $K(x, y) = -\frac{1}{2\pi} \log ||x - y||$.



- rskelf2 (white), hifie2 (gray), hifie2x (black)
- ▶ Factorization time (\circ), solve time (\Box), memory (\diamond)
- Precision $\epsilon = 10^{-6}$

Second-kind results in 3D

Second-kind volume IE on the unit cube with

$$a(x) \equiv 1, \quad K(x,y) = \frac{1}{4\pi ||x - y||}.$$



- rskelf3 (white), hifie3 (gray), hifie3x (black)
- ► Factorization time (○), solve time (□), memory (◊)
- Precision $\epsilon = 10^{-3}$