Use of Semi-separable Approximate Factorization and Direction-preserving for Constructing Effective Preconditioners

X. Sherry Li

Lawrence Berkeley National Laboratory

Ming Gu (UC Berkeley)

Panayot Vassilevski (Lawrence Livermore National Laboratory)

SIAM Conference on Applied Linear Algebra

Oct. 26-29, 2009

Monterey Bay-Seaside

Terminologies (1/2)

- **(approximate) Cholesky factorization:** $A \cong R^T R$
  - **Direct solver, preconditioner**
- **Schur-monotonic – successive Schur complement matrices are positive definite**
- **Semi-Separable (SS) matrix – a tool from structured matrices to achieve low complexity**

Terminologies (2/2)

- **Direction-preserving – avoid any approximation along some known directions:** $R^T R\, Z = A\, Z,\ \ Z \in \Re^{N \times d}\ (d << N)$

- **MILU:** $LU\, e = Ae$ **maintain "row-sum" (d = 1)**
- $LU\, x = Ax + \Lambda D\, x$ **for a vector x, with diag. perturbations (d=1)**
  - **Dupont-Kendall, Axelsson-Gustafsson, Notay**
  - **Reduce condition number of elliptic discretization matrices (i.e., from $O(h^{-2})$ to $O(h^{-1})$)**
- **Frequency filtering (d = 1, 2) (Wittum et al., Axelsson-Polman)**

- **Algorithms unknown for general d, until now …**
  - **Elasticity problems with d rigid body modes**
  - **Application in AMG:**
    - **Vector preserving interpolation matrices**
    - **Kernel preserving Non-Galerkin coarse-grid matrices**

Outline

- **Construction algorithm for SS-approximate factorization with the desired properties**
- **Quality of the approximation as preconditioner**

Mathematical formulation

- **Block Cholesky factorization**

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ A_{1,2}^T & A_{2,2} & \cdots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1,n}^T & A_{2,n}^T & \cdots & A_{n,n}^T \end{pmatrix}$$

for block k = 1, 2, …, n

Cholesky factor $R_{k,k}^T R_{k,k} := A_{k,k}$

Triangular solve $R_{k,k+1:n} := R_{k,k}^{-T} A_{k,k+1:n}$

Schur complement $A_k := A_{k+1:n,k+1:n} - R_{k,k+1:n}^T R_{k,k+1:n}$

endfor

- **New approximate Cholesky factorization, satisfying**

1. $S^T S = A + \mathrm{O}(\sqrt{\| A \|_2}\, \tau)$ for tolerance τ, still SPD
2. $S^T S\, Z = A\, Z$

- **S is an upper triangular semi-separable matrix**

Semi-separable matrix (1/2)

- **Semi-separable matrix with 4 x 4 blocks**

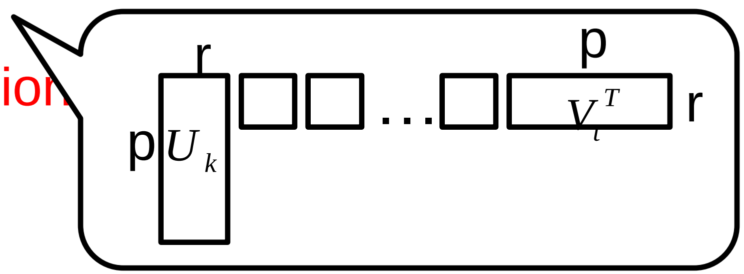
$$A \cong \begin{pmatrix} D_1 & U_1V_2^T & U_1W_2V_3^T & U_1W_2W_3V_4^T \\ V_2U_1^T & D_2 & U_2V_3^T & U_2W_3V_4^T \\ V_3W_2^TU_1^T & V_3U_2^T & D_3 & U_3V_4^T \\ V_4W_3^TW_2^TU_1 & V_4W_3U_2^T & V_4U_3^T & D_4 \end{pmatrix}$$

- First and second off-diagonal blocks of A are

$$U_1(V_2^T \quad W_2V_3^T \quad W_2W_3V_4^T) \quad \text{and} \quad \begin{pmatrix} U_1W_2 \\ U_2 \end{pmatrix}(V_3^T \quad W_3V_4^T)$$

- (k,t) block entry is $U_k W_{k+1} W_{k+2} \cdots W_{t-1} V_t^T$

U_i, W_i and V_i are of small dimension.



Semi-separable matrix (2/2)

- A is N x N, uses O(N p) memory, good for p << N
- Examples: banded matrices and their inverses
- Representation can be numerically constructed

- Related work on structured matrices
  - $\mathcal{H}$-matrix, $\mathcal{H}^2$-matrix (Hackbusch, Starr and Roklin, et al.) (hierarchical matrices)
  - FMM matrix (Greengard and Roklin, et al.)
  - HSS matrix (Hierarchical SS) (Chandrasekaran et al.)

The Construction Algorithm

- **Embed SS construction scheme in Cholesky factorization to ensure each approximate Schur complement positive definite, and A*Z unchanged at each step**

Semi-separable Cholesky factor

$$A \approx S^T S \quad \text{and} \quad AZ = SZ \ \text{ for } \ Z = (Z_1, \cdots, Z_n)^T$$

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ A_{1,2}^T & A_{2,2} & \cdots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1,n}^T & A_{2,n}^T & \cdots & A_{n,n}^T \end{pmatrix} \qquad S = \begin{pmatrix} D_1 & S_{1,2} & \cdots & S_{1,2} \\ & D_2 & \cdots & S_{2,n} \\ & & \ddots & \vdots \\ & & & D_n \end{pmatrix}$$

$$\text{where, } S_{k,t} = U_k W_{k+1} \cdots W_{t-1} V_t^T$$

- D_i's are upper triangular, $p \times p$
- $U_i, W_i,$ and V_i are of small dimensions (related to numerical rank)

Main tools

- **Perform low-rank approximations via rank revealing QR, or** τ-accurate SVD
  1. **Decompose intermediate matrices in SVD form**

     $$H = (U \ \ \hat{U})(V \ \ \hat{V})^T \approx UV^T$$

     where, $(U \ \ \hat{U})$ is column-orthogonal, and $\|\hat{V}\|_2 = O(\tau)$

     make U have fewer columns

  2. **Furthermore, want direction-preserving**

     Given $H \in R^{m\times n}$, and two direction matrices $F \in R^{n\times d}$, $G \in R^{m\times d}$

     Seek $H \approx UV^T$, which preserves $HF = UV^TF$ and $G^TH = G^TUV^T$

✓ **Denote this procedure as: ("constrained" SVD)**

$$[U, V] \leftarrow DPsvd(H, F, G)$$

Construction: STEP 1

- **Factor:** $D_1^T D_1 = A_{11}$ and $H_1 = D_1^{-T} A_{1,2:n}$ $\leftarrow$ first block row of R
- **Approximate H_1 and preserve A*Z:** $$AZ = \begin{pmatrix} D_1^T D_1 Z_1 + \boxed{D_1^T H_1 Z_{2:n}} \\ \boxed{H_1^T D_1 Z_1} + A_{2:n,2:n} Z_{2:n} \end{pmatrix}$$

Use procedure "DPsvd":

$$[U_1, Q_1] \leftarrow DPsvd\ (H_1, Z_{2:n}, D_1 Z_1)$$

so that $\tilde{H}_1 = U_1 Q_1^T \approx H_1$ and $H_1^T H_1 = Q_1 Q_1^T + \hat{Q}_1 \hat{Q}_1^T$

- **Approximate Schur complement**

$$\text{Exact}: \ A_1 = A_{2:n,2:n} - H_1^T H_1 = A_{2:n,2:n} - Q_1 Q_1^T - \hat{Q}_1 \hat{Q}_1^T$$

$$\text{Approximate}: \ \tilde{A}_1 = A_{2:n,2:n} - Q_1 Q_1^T = A_1 + \hat{Q}_1 \hat{Q}_1^T = A_1 + O(\tau^2)$$, **still SPD**

SAVING: NOT to compute $\tilde{A}_1$ explicitly, but only store Q_1

Partition $Q_1^T = (V_2^T \quad \hat{H}_1)$, then $\tilde{H}_1 = (U_1 V_2^T \quad U_1 \hat{H}_1)$

Schur complement becomes $$\tilde{A}_1 = \begin{pmatrix} A_{2,2} - V_2 V_2^T & A_{2,3:n} - V_2 \hat{H}_1 \\ \left(A_{2,3:n} - V_2 \hat{H}_1\right)^T & A_{3:n,3:n} - \hat{H}_1 \hat{H}_1^T \end{pmatrix}$$

Construction: STEP 2

- **Updates** $A_{2,2} := A_{2,2} - V_2 V_2^T, \quad A_{2,3:n} := A_{2,3:n} - V_2 \hat{H}_1$
- **Factor** $D_2^T D_2 = A_{22}$ and $H_2 = D_2^{-T} A_{2,3:n} \leftarrow$ second block row of R

Define $\Delta_2 = \begin{pmatrix} D_1 & U_1 V_2^T \\ & D_2 \end{pmatrix}$ and $\Gamma_2 = \begin{pmatrix} U_1 & \\ & I \end{pmatrix}$

Then

$$A \approx \begin{pmatrix} \Delta_2^T \Delta_2 & \Delta_2^T \Gamma_2 \begin{pmatrix} \hat{H}_1 \\ H_2 \end{pmatrix} \\ \begin{pmatrix} \hat{H}_1 \\ H_2 \end{pmatrix}^T \Gamma_2^T \Delta_2 & A_{3:n,3:n} \end{pmatrix}$$

- **Approximate** $\begin{pmatrix} \hat{H}_1 \\ H_2 \end{pmatrix}$ **and preserve A*Z:**

Use procedure: $[Y_2 \quad Q_2] \leftarrow DPsvd\left(\begin{pmatrix} \hat{H}_1 \\ H_2 \end{pmatrix}, Z_{3:n}, \Gamma_2^T D_2 Z_{1:2} \right)$

- **Approximate Schur complement**

$$\mathrm{A}_2 = A_{3:n,3:n} - \begin{pmatrix} \hat{H}_1 \\ H_2 \end{pmatrix}^T \begin{pmatrix} \hat{H}_1 \\ H_2 \end{pmatrix} \approx A_{3:n,3:n} - Q_2 Q_2^T$$

$$\tilde{\mathrm{A}}_2 = \mathrm{A}_2 + \hat{Q}_2 \hat{Q}_2^T = \mathrm{A}_2 + O(\tau^2),$$ still SPD

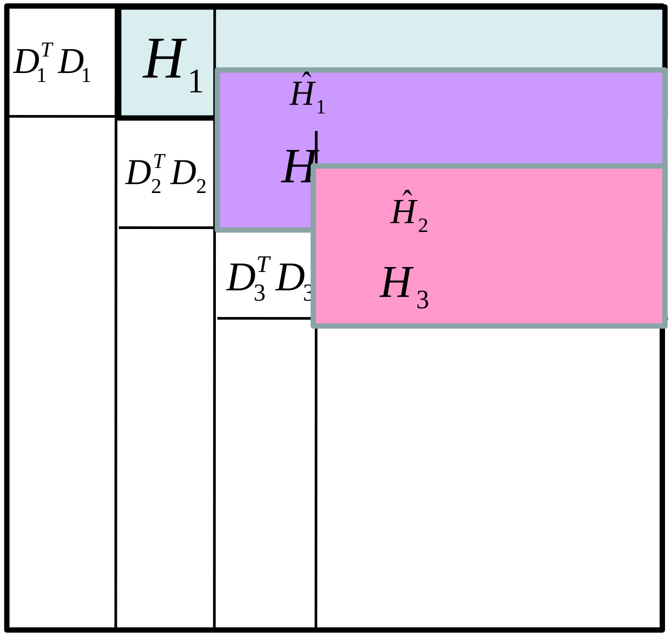
Construction: end of STEP 2

- **Use partition** $Y_2 = \begin{pmatrix} W_2 \\ U_2 \end{pmatrix}$

  the first two blocks of the (approx.) Cholesky factor is

$$\begin{pmatrix} D_1 & U_1 V_2^T & U_1 W_2 Q_2^T \\ & D_2 & U_2 Q_2^T \end{pmatrix}$$

- **Pictorial view**



Complexity

- **Operations**
  - **Let p be the maximum dimension in all the diagonal blocks**
  - **Cost of each step (update & compression):** $O(N\,p^2)$
  - **Total :** $O(N\,p^2 \times n) = O(N^2\,p)$

- **Storage**
  - **Only need to store** $D_{i,}\,U_i, W_i, \text{and}\,V_i$ **each of dimension <=** $p \times p$
  - **Total :** $O(n\,p^2) = O(N\,p)$

  (Implementation: 4 arrays of size (N, p))

Quality of the approximation

Example 1: 2D anisotropic diffusion equation on [0,1]x[0,1]

$$-div\,(k(x,y)\nabla u) = f(x,y)$$

where $k(x,y) = \varepsilon I + \mathbf{b}\mathbf{b}^T = \begin{bmatrix} \varepsilon + b_1^{\,2} & b_1 b_2 \\ b_1 b_2 & \varepsilon + b_2^{\,2} \end{bmatrix}$, $\quad \mathbf{b} = \begin{bmatrix} \cos\alpha(1 - x\cos\alpha) \\ \sin\alpha(1 - y\sin\alpha) \end{bmatrix}$

That is,

$$(\varepsilon + b_1^{\,2})\frac{\partial^2 u}{\partial x^2} + 2b_1 b_2 \frac{\partial^2 u}{\partial x \partial y} + (\varepsilon + b_2^{\,2})\frac{\partial^2 u}{\partial y^2} = f(x,y)$$

- **Assume a mixture of Dirichlet and Neumann boundary conditions**
- **Use lexicographic ordering of the unknowns**
- **Direction vectors:**
  - **d=1: constant vector**
  - **d=2, 3: linear functions x and y evaluated at the mesh nodes**

Diffusion problem

- $\varepsilon = 0.01, \quad \alpha = \frac{\pi}{3}$ **p – block size, r – max rank**
- **Number of PCG iterations decreases with more directions**
- **Larger rank results in better approximation**
- **Doubling block size and rank, construction time less than doubled**

h^{-1}	$p=8, \quad r=2d+2$				$p=20, \quad r=2d+10$			
	d=0 time	d=1	d=2	d=3 time	d=0 time	d=1	d=2	d=3 time
12	28 0.00	24	21	20 0.01	7 0.00	1	1	1 0.00
24	61 0.05	55	51	51 0.07	28 0.05	24	23	20 0.13
48	115 0.57	113	121	110 0.91	77 1.00	65	65	53 1.14
96	233 8.52	221	216	210 13.74	158 15.48	139	185	118 18.49

Example 2: 2D linear elasticity equation

$$-(\mu \Delta u + (\lambda + \mu)\nabla\nabla \bullet u) = f \ \text{ in } \ \Omega = (0,1)\times(0,1)$$

$$u = 0 \ \text{ on } \ \partial\Omega$$

where, $u \in \Re^2$ is displacement vector field

λ and μ are the Lame constants

- **PDE is very ill-conditioned when λ / μ is very large. Iterative methods, including MG, diverge or converge very slowly**
- **Direction vectors: d=2 corresponds to two rigid-body modes with entries alternating (1,0) and (0,1):**

  Let u=(u_1, u_2), one of the modes is such that all discretized u_1 nodes are 1 and u_2 nodes are 0; the other mode is vice versa

Elasticity problem

- **Define** $\hat{A} = R^{-T} A R^{-1}$
- **When** $\lambda / \mu = 1$ **, directions and larger block/rank are helpful**
- **When** $\lambda / \mu = 10^4$ **, directions are helpful, but larger block/rank do not help**

(λ, μ)	h^{-1}	$p=8, \quad r=2d+2$				$p=20, \quad r=2d+10$			
		d=0	$\kappa(\hat{A})$	d=2	$\kappa(\hat{A})$	d=0	$\kappa(\hat{A})$	d=2	$\kappa(\hat{A})$
(1.0, 1.0)	8	32	1.5e+1	25	9.7e+1	16	2.9e+1	11	1.9e+1
	16	62	6.4e+2	48	4.7e+2	64	8.6e+2	31	2.0e+2
	32	123	2.5e+3	92	1.7e+3	83	3.0e+3	62	1.2e+3
(1.0, 10^{-4})	8	243	3.1e+5	236	3.5e+5	12	1.3e+1	9	1.3e+1
	16	549	1.1e+6	440	9.7e+5	1230	1.7e+6	1203	2.0e+6
	32	1216	4.5e+6	1258	4.3e+6	1867	7.0e+6	1996	8.6e+6

Elasticity problem : last Schur complement

- **Construct SS-approximate Cholesky for the last Schur complement**
- **Schur preconditioner much more effective than the whole**

λ / μ	1.0	10^4	10^8
$\kappa(S)$	2.4e+02	1.9e+04	3.9e+09
$\kappa(\hat{S} = R^{-T} S R^{-1})$	2.9	7.1e+01	7.2e+01
CG iters	58	354	648
PCG iters	7	20	27

Application of D.P. block-factorization to AMG

- **Two-grid "c" - "f" partition:** $A = \begin{bmatrix} A_{ff} & A_{fc} \\ A_{cf} & A_{cc} \end{bmatrix}$
- **Choices of vectors:**
  - **Schur complements can be viewed as coarse discretization matrices if they preserve the near null-space of the fine-grid matrix**
  - **In adaptive AMG, it is important that the coarse space contains several "algebraically smooth" vectors, i.e., the smoother M cannot damp successfully:** $(I - M^{-1}A)\,\mathbf{v} \approx \mathbf{v}$
  - **Constant vector for scalar diffusion equations**
  - **Rigid body modes for elasticity equations**

Application to AMG (cont.)

- **Let v = (v_f, v_c) be a vector in null-space of A**

$$A_{ff} v_f + A_{fc} v_c = 0, \text{ then } Av = \begin{bmatrix} 0 \\ S v_c \end{bmatrix}$$

- **<u>Coarse-grid matrix</u>: the Schur complement matrix** $S = A_{cc} - A_{cf} A_{ff}^{-1} A_{fc}$ **can be approximated by a sparse A_c, satisfying** $A_c v_c = S v_c = 0$

- **<u>Interpolation matrix</u>: let M_{ff} be a factored s.p.d. matrix approximating A, such that** $M_{ff} v_f = A_{ff} v_f$

  Then define an interpolation matrix $P = \begin{bmatrix} -M_{ff}^{-1} A_{fc} \\ I \end{bmatrix}$ **satisfying** $P v_c = v$ **, which interpolate back onto the fine grid**

Perspectives

- We now have a fast algebraic approx. factorization procedure that achieves D.P. and Schur monotonicity ….

- May not be good enough as a general solver or preconditioner
  - Need analytical study for different PDEs
  - Whether cond. number of preconditioned matrix depends only on approximation precision, not discretization dofs N ?
  - Compare with traditional IC, ILU, etc.
- Incorporate into the superfast multifrontal sparse Cholesky procedure (Xia's talk)
- Analyze, test robustness of new AMG preconditioner
  - Interpolation matrix and coarse-grid matrix in AMG
- Parallelization, performance tuning (of small matrices)
  - More scalable than traditional factorization with smaller amount data to communicate