A Hierarchical Low-Rank Schur Complement Preconditioner for Nonsymmetric Linear Systems

Geoffrey Dillon

Preconditioning 2017

joint work with Yuanzhe Xi   Yousef Saad

Dept. of Computer Science
University of Minnesota

July 31, 2017

Outline

1. Introduction
2. Preconditioner Construction Process
3. Numerical Results
4. Conclusions/Future Work

Our goal is the efficient solution of

$$Ax = b \tag{1}$$

via Krylov subspace methods. We assume that $A \in \mathbb{R}^{n \times n}$ is large and sparse. In this work A can be nonsymmetric and indefinite.

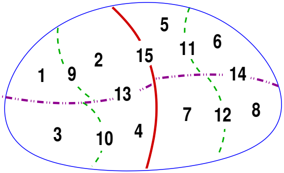
This is an extension of the MSLR preconditioner of Xi, Li, and Saad [3], which was for symmetric problems.

Reordering Algorithms $\rightarrow$ Multilevel Structure

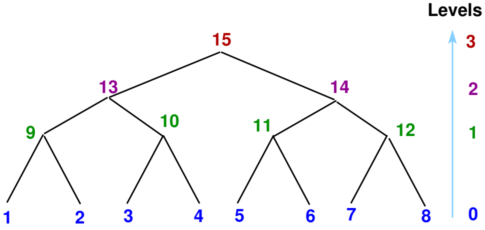
- Ordering algorithms such as nested dissection, multilevel coarsening, multicoloring, etc... can make (1) easier to solve.
- The *Hierarchical Interface Decomposition* (HID) is a multilevel combination of nested dissection and domain decomposition.
- An L level HID reordering takes a matrix A and recursively partitions it into the 2×2 block form

$$A_l = \begin{pmatrix} B_l & F_l \\ E_l & C_l \end{pmatrix}, \; C_l = A_{l+1}, \; l = 0 : L - 1. \tag{2}$$

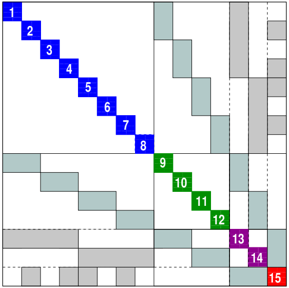
HID Example



3 level partition

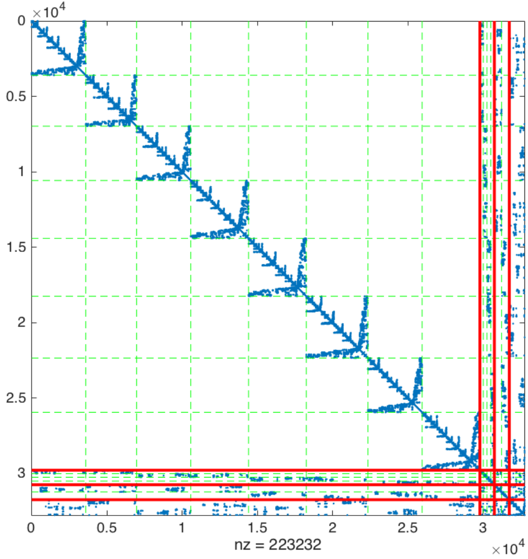


HID tree



Reordered matrix

Multilevel Structure Example $L = 3$



The Schur Complement

- At each level of (2) we have the Schur complement matrix:

$$S_l = C_l - E_l B_l^{-1} F_l.$$

- The Schur complement appears in the block-LU factorization of the matrix A_l:

$$A_l = \begin{pmatrix} B_l & F_l \\ E_l & C_l \end{pmatrix} = \begin{pmatrix} I & 0 \\ E_l B_l^{-1} & I \end{pmatrix} \begin{pmatrix} B_l & F_l \\ 0 & S_l \end{pmatrix} = LU \qquad (3)$$

where A_0 is the reordering of the original matrix A.

- If we use the U factor as a right preconditioner to A_0, then in **exact** arithmetic we get

$$\begin{aligned} A_0 U^{-1} &= \begin{pmatrix} B_0 & F_0 \\ E_0 & C_0 \end{pmatrix} \begin{pmatrix} B_0 & F_0 \\ 0 & S_0 \end{pmatrix}^{-1} \\ &= \begin{pmatrix} B_0 & F_0 \\ E_0 & C_0 \end{pmatrix} \begin{pmatrix} B_0^{-1} & -B_0^{-1} F_0 S_0^{-1} \\ 0 & S_0^{-1} \end{pmatrix} \\ &= \begin{pmatrix} B_0 & F_0 \\ E_0 & C_0 \end{pmatrix} \begin{pmatrix} B_0^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & -F_0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & S_0^{-1} \end{pmatrix} = L \end{aligned}$$

  and GMRES will converge in two iterations [2].
- Cost of this preconditioner:
  - 2 linear solves (1 w/ B_0 and 1 w/ S_0)
  - 1 sparse matvec
- Problem: S_0 is dense! How can we compute S_0^{-1} in order to apply the preconditioner U^{-1}?

An approximation to S^{-1}

- To make this preconditioner practical, we have to approximate the Schur Complement. Based on the analysis in [1, 3], we claim that a good approximation to S_I^{-1} is

$$S_I^{-1} \approx C_I^{-1} + LRC \tag{4}$$

where LRC is a low rank correction matrix.

- We now will give some details of how LR is built.

Building the low-rank correction

- We assume that the B_l blocks have LU decompositions at all levels, i.e.

$$B_l = L_{B_l} U_{B_l}, \;\; l = 0 : L-1.$$

- Substitute these factorizations in for B_l in the expression of the Schur complement, we get

$$S_l = C_l - E_l U_{B_l}^{-1} L_{B_l}^{-1} F_l = (I - G_l)\, C_l \tag{5}$$

where

$$G_l = E_l U_{B_l}^{-1} L_{B_l}^{-1} F C_l^{-1}. \tag{6}$$

- From (5) it follows that

$$S_l^{-1} = C_l^{-1} (I - G_l)^{-1}. \tag{7}$$

The low-rank correction

- We then take the complex Schur decomposition of G_l:

$$G_l = W_l R_l W_l^H \tag{8}$$

- Substitute this into (7) and use the Sherman Morrison Woodbury identity, we get

$$S_l^{-1} = C_l^{-1}\left(I + W_l\left[(I - R_l)^{-1} - I\right] W_l^H\right). \tag{9}$$

- If $\widetilde{R}_l \approx R_l$ and then an approximate inverse of S_l is given by

$$\widetilde{S}_l^{-1} = C_l^{-1} + C_l^{-1} W_l\left[(I - \widetilde{R}_l)^{-1} - I\right] W_l^H = C_l^{-1}\left(I + W_l H_l W_l^H\right) \tag{10}$$

where $H_l = (I - \widetilde{R}_l)^{-1} - I$.

An Example with $L = 3$ for solving $Ax = b$

1. Perform a 3 level HID reordering and call the reordered matrix A_0.
2. We wish to use $\widetilde{U}^{-1}$ as a right preconditioner for A_0 where

$$\widetilde{U}^{-1} = \begin{pmatrix} \widetilde{B}_0^{-1} & \\ & I \end{pmatrix} \begin{pmatrix} I & -F_0 \\ & I \end{pmatrix} \begin{pmatrix} I & \\ & \widetilde{S}_0^{-1} \end{pmatrix}$$

   with $\widetilde{S}_0^{-1} = \widetilde{C}_0^{-1}\left(I + W_0 H_0 W_0^H\right)$ and $\widetilde{B}_0^{-1} \approx B_0^{-1}$. We assume that C_0 is too large to factor, so we use the fact that thanks to the HID reordering, $C_0 = A_1$ and move from level 0 to level 1.
3. At level 1, we have

$$\widetilde{C}_0^{-1} \approx A_1^{-1} = \begin{pmatrix} I & -\widetilde{B}_1^{-1}F_1 \\ & I \end{pmatrix} \begin{pmatrix} \widetilde{B}_1^{-1} & \\ & \widetilde{S}_1^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -E_1\widetilde{B}_1^{-1} & I \end{pmatrix}$$

   with $\widetilde{S}_1^{-1} = \widetilde{C}_1^{-1}(I + W_1 H_1 W_1^T)$ and $\widetilde{B}_1^{-1} \approx B_1^{-1}$. We again assume that C_1 is too large to factor, so we move up a level.
4. At level 2, we have something similar:

$$\widetilde{C}_1^{-1} \approx A_2^{-1} = \begin{pmatrix} I & -\widetilde{B}_2^{-1}F_2 \\ & I \end{pmatrix} \begin{pmatrix} \widetilde{B}_2^{-1} & \\ & \widetilde{S}_2^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -E_2\widetilde{B}_2^{-1} & I \end{pmatrix}$$

   with $\widetilde{S}_2^{-1} = \widetilde{C}_2^{-1}(I + W_2 H_2 W_2^T)$, $\widetilde{B}_2^{-1} \approx B_2^{-1}$. Here we assume that C_2 is small enough to factor, so we compute its ILU factorization $C_2 \approx L_{C_2} U_{C_2}$.
5. The actual implementation starts with level 2 and moves down the tree:

$$L_{C_2} U_{C_2} \to \widetilde{C}_2^{-1} \to \widetilde{S}_2^{-1} \to A_2^{-1} \to \widetilde{S}_1^{-1} \to A_1^{-1} \to \widetilde{S}_0^{-1} \to \widetilde{U}^{-1}.$$

An Example with $L = 3$ for solving $Ax = b$

1. Perform a 3 level HID reordering and call the reordered matrix A_0.
2. We wish to use $\widetilde{U}^{-1}$ as a right preconditioner for A_0 where

$$\widetilde{U}^{-1} = \begin{pmatrix} \widetilde{B}_0^{-1} & \\ & I \end{pmatrix} \begin{pmatrix} I & -F_0 \\ & I \end{pmatrix} \begin{pmatrix} I & \\ & \widetilde{S}_0^{-1} \end{pmatrix}$$

with $\widetilde{S}_0^{-1} = \widetilde{C}_0^{-1}\left(I + W_0 H_0 W_0^H\right)$ and $\widetilde{B}_0^{-1} \approx B_0^{-1}$. We assume that C_0 is too large to factor, so we use the fact that thanks to the HID reordering, $C_0 = A_1$ and move from level 0 to level 1.

3. At level 1, we have

$$\widetilde{C}_0^{-1} \approx A_1^{-1} = \begin{pmatrix} I & -\widetilde{B}_1^{-1} F_1 \\ & I \end{pmatrix} \begin{pmatrix} \widetilde{B}_1^{-1} & \\ & \widetilde{S}_1^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -E_1 \widetilde{B}_1^{-1} & I \end{pmatrix}$$

with $\widetilde{S}_1^{-1} = \widetilde{C}_1^{-1}(I + W_1 H_1 W_1^T)$ and $\widetilde{B}_1^{-1} \approx B_1^{-1}$. We again assume that C_1 is too large to factor, so we move up a level.

4. At level 2, we have something similar:

$$\widetilde{C}_1^{-1} \approx A_2^{-1} = \begin{pmatrix} I & -\widetilde{B}_2^{-1} F_2 \\ & I \end{pmatrix} \begin{pmatrix} \widetilde{B}_2^{-1} & \\ & \widetilde{S}_2^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -E_2 \widetilde{B}_2^{-1} & I \end{pmatrix}$$

with $\widetilde{S}_2^{-1} = \widetilde{C}_2^{-1}(I + W_2 H_2 W_2^T)$, $\widetilde{B}_2^{-1} \approx B_2^{-1}$. Here we assume that C_2 is small enough to factor, so we compute its ILU factorization $\widetilde{C}_2 \approx L_{C_2} U_{C_2}$.

5. The actual implementation starts with level 2 and moves down the tree:

$$L_{C_2} U_{C_2} \rightarrow C_2^{-1} \rightarrow \widetilde{S}_2^{-1} \rightarrow A_2^{-1} \rightarrow \widetilde{S}_1^{-1} \rightarrow A_1^{-1} \rightarrow \widetilde{S}_0^{-1} \rightarrow \widetilde{U}^{-1}.$$

An Example with $L = 3$ for solving $Ax = b$

1. Perform a 3 level HID reordering and call the reordered matrix A_0.
2. We wish to use $\widetilde{U}^{-1}$ as a right preconditioner for A_0 where

$$\widetilde{U}^{-1} = \begin{pmatrix} \widetilde{B}_0^{-1} & \\ & I \end{pmatrix} \begin{pmatrix} I & -F_0 \\ & I \end{pmatrix} \begin{pmatrix} I & \\ & \widetilde{S}_0^{-1} \end{pmatrix}$$

   with $\widetilde{S}_0^{-1} = \widetilde{C}_0^{-1}\left(I + W_0 H_0 W_0^H\right)$ and $\widetilde{B}_0^{-1} \approx B_0^{-1}$. We assume that C_0 is too large to factor, so we use the fact that thanks to the HID reordering, $C_0 = A_1$ and move from level 0 to level 1.
3. At level 1, we have

$$\widetilde{C}_0^{-1} \approx A_1^{-1} = \begin{pmatrix} I & -\widetilde{B}_1^{-1} F_1 \\ & I \end{pmatrix} \begin{pmatrix} \widetilde{B}_1^{-1} & \\ & \widetilde{S}_1^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -E_1 \widetilde{B}_1^{-1} & I \end{pmatrix}$$

   with $\widetilde{S}_1^{-1} = \widetilde{C}_1^{-1}(I + W_1 H_1 W_1^T)$ and $\widetilde{B}_1^{-1} \approx B_1^{-1}$. We again assume that C_1 is too large to factor, so we move up a level.
4. At level 2, we have something similar:

$$\widetilde{C}_1^{-1} \approx A_2^{-1} = \begin{pmatrix} I & -\widetilde{B}_2^{-1} F_2 \\ & I \end{pmatrix} \begin{pmatrix} \widetilde{B}_2^{-1} & \\ & \widetilde{S}_2^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -E_2 \widetilde{B}_2^{-1} & I \end{pmatrix}$$

   with $\widetilde{S}_2^{-1} = \widetilde{C}_2^{-1}(I + W_2 H_2 W_2^T)$, $\widetilde{B}_2^{-1} \approx B_2^{-1}$. Here we assume that C_2 is small enough to factor, so we compute its ILU factorization $\widetilde{C}_2 \approx L_{C_2} U_{C_2}$.
5. The actual implementation starts with level 2 and moves down the tree:

$$L_{C_2} U_{C_2} \to C_2^{-1} \to \widetilde{S}_2^{-1} \to A_2^{-1} \to \widetilde{S}_1^{-1} \to A_1^{-1} \to \widetilde{S}_0^{-1} \to \widetilde{U}^{-1}.$$

An Example with $L = 3$ for solving $Ax = b$

1. Perform a 3 level HID reordering and call the reordered matrix A_0.
2. We wish to use $\widetilde{U}^{-1}$ as a right preconditioner for A_0 where

$$\widetilde{U}^{-1} = \begin{pmatrix} \widetilde{B}_0^{-1} & \\ & I \end{pmatrix} \begin{pmatrix} I & -F_0 \\ & I \end{pmatrix} \begin{pmatrix} I & \\ & \widetilde{S}_0^{-1} \end{pmatrix}$$

   with $\widetilde{S}_0^{-1} = \widetilde{C}_0^{-1}\left(I + W_0 H_0 W_0^H\right)$ and $\widetilde{B}_0^{-1} \approx B_0^{-1}$. We assume that C_0 is too large to factor, so we use the fact that thanks to the HID reordering, $C_0 = A_1$ and move from level 0 to level 1.
3. At level 1, we have

$$\widetilde{C}_0^{-1} \approx A_1^{-1} = \begin{pmatrix} I & -\widetilde{B}_1^{-1} F_1 \\ & I \end{pmatrix} \begin{pmatrix} \widetilde{B}_1^{-1} & \\ & \widetilde{S}_1^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -E_1 \widetilde{B}_1^{-1} & I \end{pmatrix}$$

   with $\widetilde{S}_1^{-1} = \widetilde{C}_1^{-1}(I + W_1 H_1 W_1^T)$ and $\widetilde{B}_1^{-1} \approx B_1^{-1}$. We again assume that C_1 is too large to factor, so we move up a level.
4. At level 2, we have something similar:

$$\widetilde{C}_1^{-1} \approx A_2^{-1} = \begin{pmatrix} I & -\widetilde{B}_2^{-1} F_2 \\ & I \end{pmatrix} \begin{pmatrix} \widetilde{B}_2^{-1} & \\ & \widetilde{S}_2^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -E_2 \widetilde{B}_2^{-1} & I \end{pmatrix}$$

   with $\widetilde{S}_2^{-1} = \widetilde{C}_2^{-1}(I + W_2 H_2 W_2^T)$, $\widetilde{B}_2^{-1} \approx B_2^{-1}$. Here we assume that C_2 is small enough to factor, so we compute its ILU factorization $\widetilde{C}_2 \approx L_{C_2} U_{C_2}$.
5. The actual implementation starts with level 2 and moves down the tree:

$$L_{C_2} U_{C_2} \to C_2^{-1} \to \widetilde{S}_2^{-1} \to A_2^{-1} \to \widetilde{S}_1^{-1} \to A_1^{-1} \to \widetilde{S}_0^{-1} \to \widetilde{U}^{-1}.$$

An Example with $L = 3$ for solving $Ax = b$

1. Perform a 3 level HID reordering and call the reordered matrix A_0.
2. We wish to use $\widetilde{U}^{-1}$ as a right preconditioner for A_0 where

$$\widetilde{U}^{-1} = \begin{pmatrix} \widetilde{B}_0^{-1} & \\ & I \end{pmatrix} \begin{pmatrix} I & -F_0 \\ & I \end{pmatrix} \begin{pmatrix} I & \\ & \widetilde{S}_0^{-1} \end{pmatrix}$$

   with $\widetilde{S}_0^{-1} = \widetilde{C}_0^{-1}\left(I + W_0 H_0 W_0^H\right)$ and $\widetilde{B}_0^{-1} \approx B_0^{-1}$. We assume that C_0 is too large to factor, so we use the fact that thanks to the HID reordering, $C_0 = A_1$ and move from level 0 to level 1.
3. At level 1, we have

$$\widetilde{C}_0^{-1} \approx A_1^{-1} = \begin{pmatrix} I & -\widetilde{B}_1^{-1} F_1 \\ & I \end{pmatrix} \begin{pmatrix} \widetilde{B}_1^{-1} & \\ & \widetilde{S}_1^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -E_1 \widetilde{B}_1^{-1} & I \end{pmatrix}$$

   with $\widetilde{S}_1^{-1} = \widetilde{C}_1^{-1}(I + W_1 H_1 W_1^T)$ and $\widetilde{B}_1^{-1} \approx B_1^{-1}$. We again assume that C_1 is too large to factor, so we move up a level.
4. At level 2, we have something similar:

$$\widetilde{C}_1^{-1} \approx A_2^{-1} = \begin{pmatrix} I & -\widetilde{B}_2^{-1} F_2 \\ & I \end{pmatrix} \begin{pmatrix} \widetilde{B}_2^{-1} & \\ & \widetilde{S}_2^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -E_2 \widetilde{B}_2^{-1} & I \end{pmatrix}$$

   with $\widetilde{S}_2^{-1} = \widetilde{C}_2^{-1}(I + W_2 H_2 W_2^T)$, $\widetilde{B}_2^{-1} \approx B_2^{-1}$. Here we assume that C_2 is small enough to factor, so we compute its ILU factorization $\widetilde{C}_2 \approx L_{C_2} U_{C_2}$.
5. The actual implementation starts with level 2 and moves down the tree:

$$L_{C_2} U_{C_2} \to C_2^{-1} \to \widetilde{S}_2^{-1} \to A_2^{-1} \to \widetilde{S}_1^{-1} \to A_1^{-1} \to \widetilde{S}_0^{-1} \to \widetilde{U}^{-1}.$$

Building the preconditioner

Here is the L level version of the algorithm:

Algorithm 1 Generalized Multilevel Schur Low-Rank

procedure GMSLR
- Apply an L-level reordering to A (A_0 = reordered matrix).
- **for** level l from $L-1$ to 0 **do**
  - **if** $l = L-1$ **then**
    - Compute ILU factorization of C_{L-1}, $C_{L-1} \approx L_{C_{L-1}} U_{C_{L-1}}$
  - **end if**
  - Compute ILU factorization of B_l, $B_l \approx L_{B_l} U_{B_l}$. The B_l blocks are block diagonal, so this can be done in parallel.
  - Compute k_l largest eigenpairs by Arnoldi's method ▷ *Call Algorithm 2 (next slide) to apply* $\widetilde{C}_l^{-1}$

$$[V_l, K_l] = \text{Arnoldi}(E_l U_{B_l}^{-1} L_{B_l}^{-1} F_l \widetilde{C}_l^{-1}, k_l)$$

  - Compute the complex Schur decomposition $K_l = W_0 T_0 W_0^T$.
  - Compute $W_{l,k_l} = V_l W_0$ and set $R_{k_l} = T_{0(1:k_l,1:k_l)}$.
  - Compute $H_l = (I - R_{k_l})^{-1} - I$.
- **end for**

end procedure

Application of $\widetilde{C}_l^{-1}$

We still need a way to apply $\widetilde{C}_l^{-1}, l = 0, \ldots, L-1$ to a vector.

Algorithm 2 A recursive formula for the approximation of $y = \widetilde{C}_l^{-1} b$.

procedure `RecursiveSolve`(l, b)
  if $l = L-1$ **then**
    return $y = U_{L-1}^{-1} L_{L-1}^{-1} b$
  else
    Split $b = (b_1^T, b_2^T)^T$ conformingly with the blocking of $\widetilde{C}_l$
    Compute $z_1 = U_{B_l}^{-1} L_{B_l}^{-1} b_1$
    Compute $z_2 = b_2 - E_l z_1$
    if $1 \leq l < L-1$ **then**
      Compute $w_2 = W_{l,k_l} H_l W_{l,k_l}^T z_2$
      Compute $y_2 =$ `RecursiveSolve`$(l+1, z_2 + w_2)$
      Compute $y_1 = z_1 - U_{B_l}^{-1} L_{B_l}^{-1} F_l y_2$
    else
      Solve the system $S_0 y_2 = z_2$ right preconditioned by $\widetilde{S}_0^{-1}$.
      Compute $y_1 = U_{B_0}^{-1} L_{B_0}^{-1} (b_1 - F_0 y_2)$
    end if
    return $y = (y_1^T, y_2^T)^T$
  end if
end procedure

Eigenvalue Analysis

We can analyze the spectrum of the preconditioned system $A_0 U^{-1}$ by looking at the generalized eigenvalue problem

$$\begin{pmatrix} B_0 & F_0 \\ E_0 & C_0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} \widetilde{B}_0 & F_0 \\ 0 & \widetilde{S}_0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

If, for simplicity, we assume that $B_0 = \widetilde{B}_0$, then the eigenvalues of $A_0 U^{-1}$ are

$$\lambda(A_0 U^{-1}) = \{1, \lambda(S_0 \widetilde{S}_0^{-1})\}.$$

Example: `SiH4`

This matrix is 5041×5041 and is symmetric indefinite.

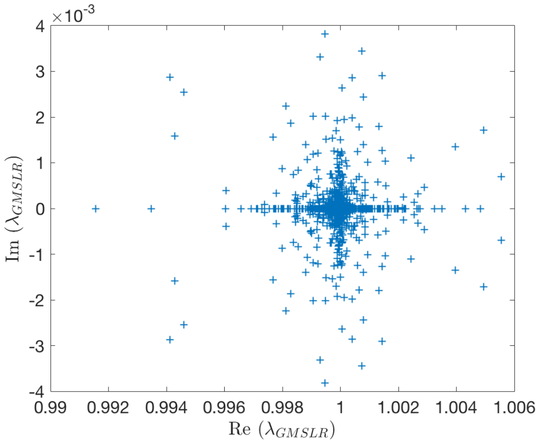


Figure: Eigenvalues of `SiH4` preconditioned by GMSLR. fGMRES converges in 4 iterations.

Model Problems

- Problem 1: Shifted Laplacians (Symmetric and Indefinite)

$$\begin{aligned} -\Delta u - cu &= f \text{ in } (0,1)^3, \\ u &= 0 \text{ on the boundary} \end{aligned} \tag{11}$$

- Problem 2: Steady-State Convection-Diffusion (Nonsymmetric)

$$\begin{aligned} -\Delta u - \alpha \cdot \nabla u - cu &= f \text{ in } (0,1)^3, \\ u &= 0 \text{ on the boundary} \end{aligned} \tag{12}$$

- Problem 3: Helmholtz w/ PML boundary condition (complex non-Hermitian)

$$\left(-\Delta - \frac{\omega^2}{v(x)^2}\right) u(x,\omega) = s(x,\omega) \text{ in } (0,1)^3$$

More Problems

Other nonsymmetric problems from Tim Davis' collection:

Matrix	Order	nnz	SPD	Origin
`CoupCons`	416,800	22,322,336	no	structural problem
`AtmosModd`	1,270,432	8,814,880	no	atmospheric model
`AtmosModL`	1,489,752	10,319,760	no	atmospheric model
`Cage14`	1,505,785	27,130,349	no	directed weighted graph
`Transport`	1,602,111	23,500,731	no	CFD problem

Notation

The GMSLR preconditioner has many parameters to tune. We use the following abbreviations when reporting performance:

- fill = $\frac{nnz(prec)}{nnz(A)}$
- p-t: wall clock time to build preconditioner (in seconds)
- its: number of iterations of GMRES(40) required for $\|r_k\|_2 < 10^{-6}$. We use "F" to indicate that GMRES(40) did not converge after 500 iterations.
- i-t: wall clock time for the iteration phase of the solver. This time is not reported when GMRES does not converge, as indicated by "–".
- rk: maximum rank used in building low-rank corrections (i.e. the number of Arnoldi steps)
- nlev: number of levels

Problem 1 - 3D: $N = 64^3$, $c = 0$

nlev	fill	p-t	i-t	its
2	14.54	5.96	4.29	13
3	13.57	2.93	2.31	12
4	12.7	1.47	1.24	11
5	11.3	.863	.97	10
6	9.79	.876	.93	10
7	8.35	.526	.527	9
8	6.67	.285	.466	8
9	5.21	.246	.354	6
10	4.1	.391	.728	6
11	3.26	.452	.532	6

Table: Results for Problem 1 in 3D. Max rank is fixed at 20. For this experiment, we only vary the number of levels. The inner solve runs for a maximum of 10 iterations.

Problem 1 - 3D: $N = 32^3$, $c = 0.5$

This shifted Laplacian matrix has 163 negative eigenvalues.

rank	LU fill	LRC fill	p-t	i-t	its
10	5.56	.792	.074	2.66	45
20	5.56	1.58	.101	3.04	49
30	5.56	2.37	.18	2.08	21
40	5.56	3.17	.158	1.11	19
50	5.56	3.96	.288	1.19	18
60	5.56	4.75	.217	1.55	25
70	5.56	5.24	.247	1.35	22
80	5.56	5.99	.322	1.14	16
90	5.56	6.73	.367	1.5	16
100	5.56	7.48	.41	1.13	45

Table: Results for Problem 1 in 3D on a 32^3 regular grid. Max rank varies from 10 to 100. The number of levels stays fixed at 6. Since we use the same number of levels, the LU fill stays constant. However, since this problem is fairly indefinite, we increase the maximum rank to 50 and allow the inner solve to run for 50 iterations.

Problem 2 - 3D

- For this first problem, $\alpha = [.2, .2, .2]$. The inner solve runs a maximum of 10 iterations.

size	nlev	fill	p-t	i-t	its
32^3	10	2.56	.059	.098	6
64^3	10	4.04	.363	.649	7
128^3	10	7.93	2.54	4.87	9

Table: Results for Problem 2 in 3D. Max rank is fixed at 4. For this experiment, we only vary the problem size.

- Now we add a shift of $c = 0.1$:

size	nlev	rank	maxits	fill	p-t	i-t	its
32^3	10	4	10	2.57	.056	.412	14
64^3	10	50	30	11.6	1.73	17.49	61

Table: Results for Problem 2 in 3D. Max rank is fixed at 4. For this experiment, we only vary the problem size. This problem is more difficult, so we have to vary the rank and number of inner iterations.

Problem 2 - 3D

- For this first problem, $\alpha = [.2, .2, .2]$. The inner solve runs a maximum of 10 iterations.

size	nlev	fill	p-t	i-t	its
32^3	10	2.56	.059	.098	6
64^3	10	4.04	.363	.649	7
128^3	10	7.93	2.54	4.87	9

Table: Results for Problem 2 in 3D. Max rank is fixed at 4. For this experiment, we only vary the problem size.

- Now we add a shift of $c = 0.1$:

size	nlev	rank	maxits	fill	p-t	i-t	its
32^3	10	4	10	2.57	.056	.412	14
64^3	10	50	30	11.6	1.73	17.49	61

Table: Results for Problem 2 in 3D. Max rank is fixed at 4. For this experiment, we only vary the problem size. This problem is more difficult, so we have to vary the rank and number of inner iterations.

Problem 3 - 3D Helmholtz w/ PML BCs

Here we list the results of solving the 3D Helmholtz equation on a sequence of meshes. These matrices are complex non-Hermitian. The number of levels and max rank shown led to the best results.

$\omega/(2\pi)$	q	$n = N^3$	nlev	rk	fill	p-t	i-t	its
2.5	8	20^3	4	16	5.43	.05	.035	6
3	8	30^3	5	16	6.65	.128	.147	8
5	8	40^3	6	16	7.56	.318	.775	8
6	8	50^3	6	16	10.45	1.01	1.72	9
8	8	60^3	6	20	15.09	3.35	3.33	9
10	8	80^3	6	40	22.44	19.34	14.54	13

Table: *Results from solving Problem 3 on a sequence of 3D meshes with GMSLR. Here the wave number is $\omega/(2\pi)$ and q denotes the number of points per wavelength.*

Nonsymmetric problems

Results for some large, nonsymmetric problems taken from Tim Davis' collection. For reference, we compare to ILUT.

Matrix	GMSLR						ILUT			
	fill	nlev	rank	p-t	i-t	its	fill	p-t	i-t	its
`CoupCons`	1.82	10	16	1.68	.64	5	1.64	17.49	2.03	23
`AtmosModd`	5.86	10	4	1.23	3.05	11	5.68	8.1	8.6	47
`AtmosModL`	5.81	11	4	1.67	2.12	7	6.03	11.35	6.37	30
`Cage14`	1.54	6	4	3.1	.89	4	1.57	5.09	0.7	4
`Transport`	2.52	11	4	1.85	7.45	23	2.96	17.92	76.93	`F`

Table: *Comparison between GMSLR and ILUT preconditioners for solving nonsymmetric test problems. ILUT parameters were chosen so that the fill factor was close to that of GMSLR. Both sets of tests use the same reordered matrix.*

Conclusions/Future Work

Conclusions

- Combined ideas from block preconditioning and the MSLR preconditioner.
- Presented a multilevel method for computing the low rank corrections.
- Developed a preconditioner capable of solving symmetric and nonsymmetric indefinite linear systems.

Future Work

- Try this Schur complement approximation on multiphysics problems.
- Develop a non-recursive way of performing the C_l solves.
- Look at different (cheaper) ways of computing the low rank correction (randomized SVD, Lanczos bidiagonalization, etc).
- Come up with heuristics for automatically choosing parameters such as nlev, rank, etc...

References I

[1] R. Li, Y. Xi, and Y. Saad, *Schur complement-based domain decomposition preconditioners with low-rank corrections*, Numerical Linear Algebra with Applications, 23 (2016), pp. 706–729.

[2] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM Journal on Scientific Computing, 21 (2000), pp. 1969–1972.

[3] Y. Xi, R. Li, and Y. Saad, *An Algebraic Multilevel Preconditioner with Low-Rank Corrections for Sparse Symmetric Matrices*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 235–259.