The state-of-the-art of preconditioners for sparse linear least squares problems

Nick Gould and Jennifer Scott
STFC Rutherford Appleton Laboratory

SIAM Conference on Applied Linear Algebra,
Atlanta, 26–30 October 2015

Introduction

Least squares are used across a wide range of disciplines: everything from simple curve fitting, through the estimation of satellite image sensor characteristics, data assimilation for weather forecasting and for climate modelling, to powering internet mapping services, exploration seismology, NMR spectroscopy, ultrasound for medical imaging, aerospace systems, neural networks ...

Linear least squares (LS)

$$min_{x \in \mathbb{R}^n} \|Ax - b\|_2,$$

where $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ is large and sparse and $b \in \mathbb{R}^m$.

Introduction

Mathematically equivalent to solving $n \times n$ positive definite *normal equations*

$$Cx = A^T b, \qquad C = A^T A,$$

and this, in turn, is equivalent to solving $(m+n) \times (m+n)$ symmetric indefinite *augmented system*

$$K \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \qquad K = \begin{bmatrix} I_m & A \\ A^T & 0 \end{bmatrix},$$

where $r = b - Ax$ is the residual vector.

Test Problems

- Taken from CUTEst LP and UFL collections
- Selected all rectangular matrices, transposed if necessary and removed null rows/columns
- Removed "duplicates" (similar problems in same group). 921 problems.
- For test set, kept those for which solving with LSMR without preconditioning either requires > 10 secs or $> 10^5$ iterations
- Gives a set of 83 problems
- Range in size from $nz(A) \approx 5000$ to $nz(A) \approx 4 * 10^7$ (latter with $m \approx 10^6$ and $n \approx 2.7 * 10^5$)
- Right-hand side is either provided or taken to be vector of 1's

Stopping criteria

C1: Stop if $\|r_k\|_2 < \delta_1$ or

C2: Stop if $\frac{\|A^T r_k\|_2}{\|r_k\|_2} < \frac{\|A^T r_0\|_2}{\|r_0\|_2} * \delta_2$,

where tolerances δ_1 and δ_2 set to 10^{-8} and 10^{-6}, respectively.

- We take the initial solution guess to be $x_0 = 0$.
- Stopping criteria are **independent of the preconditioner**.
- We exclude the cost of computing residuals to test C1 and C2 from reported times.

Normal equation preconditioners

Consider preconditioners for the normal equations.

These will be used with LSMR (Fong and Saunders, 2011), which is mathematically equivalent to MINRES applied to normal equations.

LSMR monotonically decreases $||A^T r_k||$ **and** $||r_k||$.

Used in preference to LSQR (Paige and Saunders, 1982) as can terminate earlier.

- LSMR optionally uses local reorthogonalization
- Controlled by a parameter `localSize`
- Each new basis vector is reorthogonalized with respect to the previous `localSize` vectors

Normal equation preconditioners: IC

IC = Incomplete Cholesky ie

$$C \approx LL^T$$

with L lower triangular. Set $M = LL^T$.

- General purpose, many variants used in many applications
- Simplest is IC(0): no fill allowed. Often used in comparisons with other preconditioners but not generally powerful enough.
- Applying M requires two triangular solves.

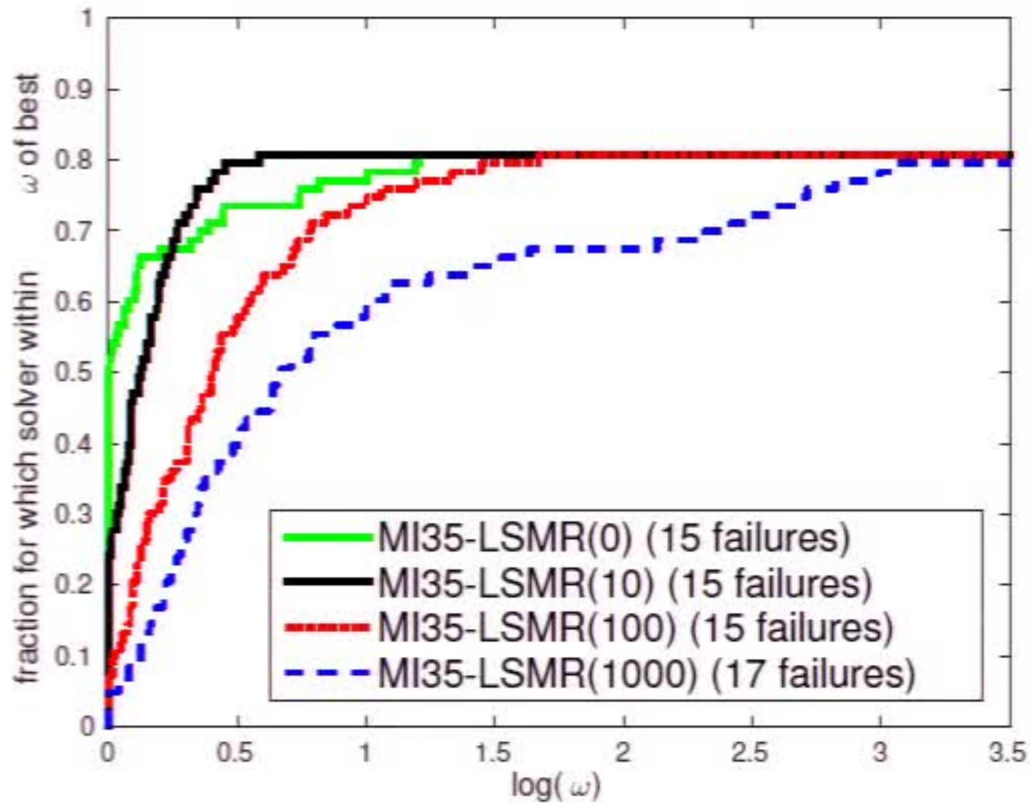
Normal equation preconditioners: HSL_MI35

Scott and Tůma have new IC code for LS called HSL_MI35. This is a variant of HSL_MI28 which was designed for symmetric positive definite systems.

HSL_MI35 implements a memory efficient approach.

- Only needs one column of C at a time.
- Incorporates ordering and scaling.
- Memory usage (number of entries in each column of L) is under the user's control.
- A global shift is used to prevent breakdown.

Time performance profile for LSMR with HSL_MI35 for range of values of reorthogonalization parameter localSize



Normal equation preconditioners: MIQR

Approximate orthogonal factorization of A

$$A \approx Q \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

Then $C \approx R^T R$ so set $M = RR^T$.

MIQR = Multilevel Incomplete QR (Li and Saad 2005)

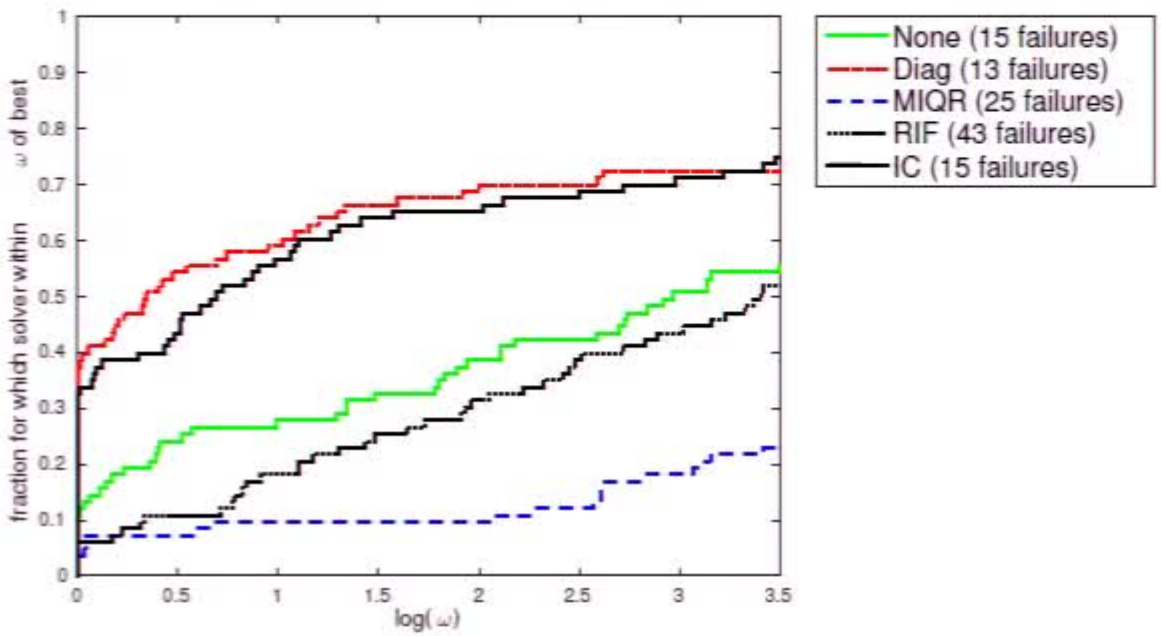
- Builds factorization by exploiting structural orthogonality in A
- At each stage, find set S of orthogonal cols; block orthogonalize other cols against S. Then repeat.

Normal equation preconditioners: RIF

Robust Incomplete Factorization (Benzi and Tůma 2003)

- Computes an incomplete LDLT factorization of C without explicitly forming it.
- Utilizes a conjugate Gram-Schmidt process to compute factorization
$$Z^T CZ = D$$
Z unit upper triangular and D diagonal.
- Follows that $L^{-1} = Z^T$.
- Entries dropped as factorization proceeds to give $M = \hat{L}\hat{D}\hat{L}^T$.

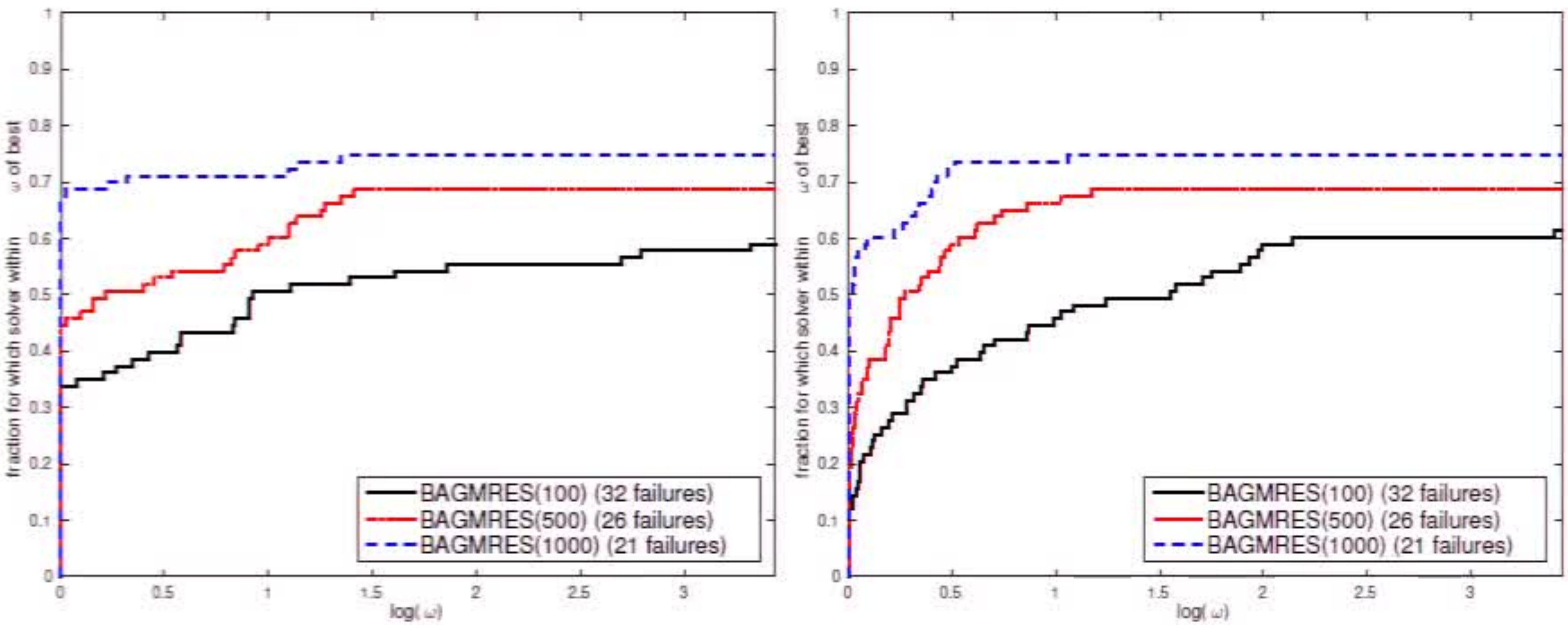
Preconditioned LSMR (time): 83 test problems



In terms of time, simple diagonal preconditioning and IC (HSL_MI35) are the winners.

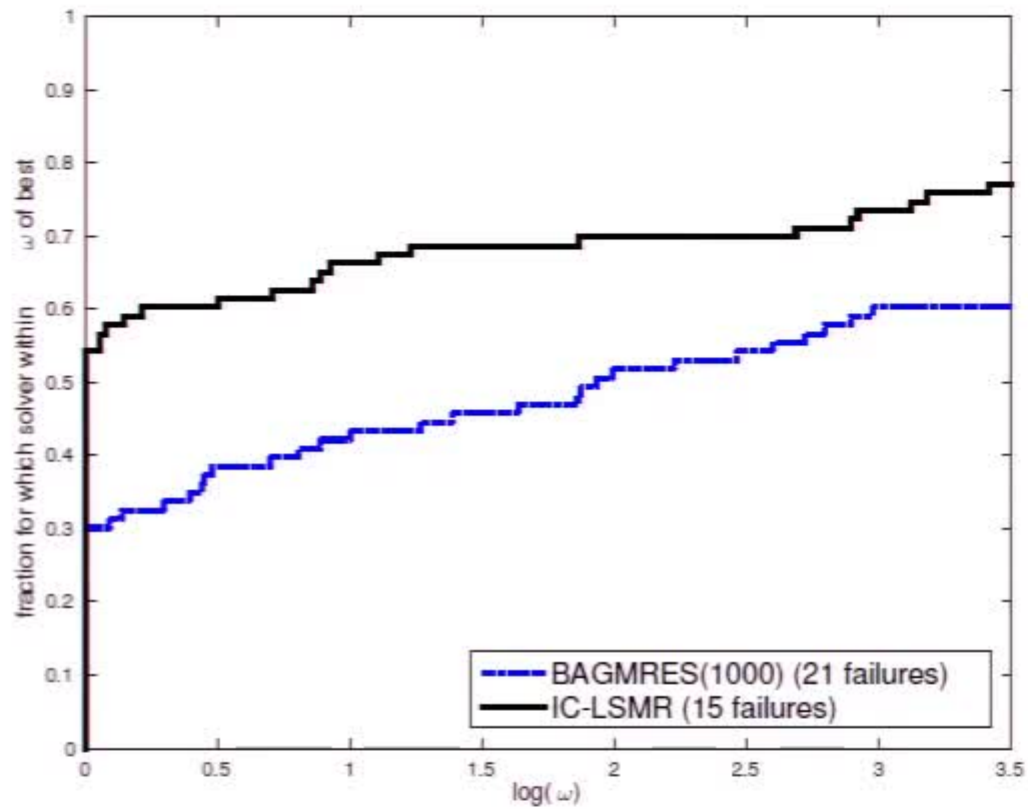
- Thus at each GMRES iteration, another system of normal equations is solved approximately using a stationary iterative method.
- This can be done without forming any entries of C explicitly (need repeated products with A and A^T).
- Memory used determined by number of steps of GMRES that are performed before restarting.

Iteration (left) and time (right) performance profiles for BA-GMRES(k)

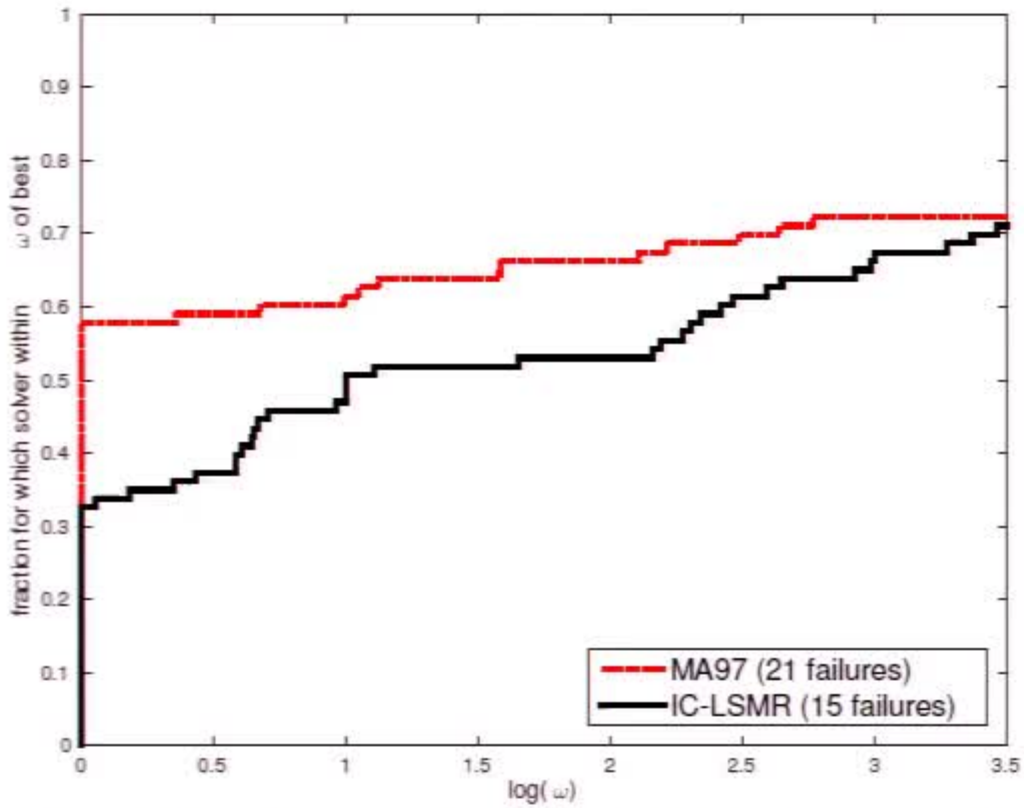


As expected, increasing k improves reliability and reduces the iteration count and time.

Time performance profile for BA-GMRES(1000) and IC preconditioned LSMR: 83 test problems



Time performance profile for direct solver HSL_MA97 and IC preconditioned LSMR: 83 test problems



It is hard to beat the (parallel) direct solver!

Parallel implementations of preconditioning required (eg Chow and Patel, Chow and Scott 2015)

Preconditioning the augmented system

Recall

$$K = \begin{pmatrix} I_m & A \\ A^T & 0 \end{pmatrix}.$$

Two approaches to incomplete factorization

- Signed incomplete Cholesky factorization (exploit structure).
- General incomplete LDL^T factorization (ignore structure).

In each case, we use GMRES as the solver.

Problem: stopping criteria based on K (not A)
(GMRES not aware of structure of K).

Signed IC preconditioner

Compute

$$K = \begin{bmatrix} I_m & A \\ A^T & 0 \end{bmatrix} \approx L \begin{bmatrix} I_m & \\ & -I_n \end{bmatrix} L^T$$

- Exploits structure of augmented system
- Avoids need for numerical pivoting
- Shifts are used to avoid breakdown

$$\tilde{K} = \begin{pmatrix} I_m & A \\ A^T & 0 \end{pmatrix} + \begin{pmatrix} \alpha_1 I_m & 0 \\ & -\alpha_2 I_n \end{pmatrix}$$

- Software that implements this is HSL_MI30

General indefinite incomplete factorization LDL^T

Many challenges:

- Must prevent growth in the entries of the factors.
- Pivoting using 1×1 and 2×2 pivots needed.
- Pivoting potentially expensive: localize the pivot search?
- Preprocess to improve efficiency and reliability using matching-based ordering?
- What about shifting?
- Is the use of intermediate memory beneficial?
- Can we monitor instability as factorization proceeds?

	Signed IC + GMRES			General indef. + GMRES			IC + LMSR		
	α_2	iters	time	α	iters	time	α	iters	time
Maraguel_4	0.26	78	0.11	0.01	12	0.04	0.13	115	0.07
TF16	2.0	807	15.6	2.0	3972	99	0.26	86,715	141
mri1	0.02	51	2.9	2.0	–	> 600	2.0	3978	27
208bit	0.01	3654	104	2.0	–	> 600	0.02	2611	8.2

- In these examples, Signed IC uses $\alpha_1 = 0$ and α_2 is small.
- No consistent winner!

Concluding remarks I

- Preconditioning least squares problems is hard.
- A number of methods have been proposed and can work well for some problems.
- BUT
  - Often slower than (parallel) sparse direct solver.
  - Parallel implementations of preconditioning not generally available.
  - Current preconditioners can fail to give good convergence for many problems.
  - Simple diagonal preconditioning can often be sufficient.
  - There are other approaches (eg based on LU factorization) but not included here as robust, efficient implementations not currently available.

Concluding remarks II

- The NA Group at RAL has been awarded a £1M grant to work on least squares problems over the next 4 years.
- Understanding the state-of-the-art, including preconditioners for linear least squares, is a first step in this project.
- We will be working on practical applications (including problems from ISIS neutron synchrotron, RAL Space and Diamond Light Source).
- If you have an interest/expertise in sparse least squares, we would be keen to explore ideas with you and possible collaborations.