Solving sparse-dense least squares.

Jennifer Scott

University of Reading and STFC Rutherford Appleton Laboratory

Miroslav Tůma

Faculty of Mathematics and Physics, Charles University, Prague

Napoli, February 15th, 2022

Outline

1 Introduction

2 Towards the goal

3 1. Arbitrary sparse-dense (ASD) approach

4 2. Splitting large cliques by stretching

5 3. Null-space approach

6 Conclusions

Introduction

The Linear Least Squares Problem (LS)

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

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

Why the problem is so difficult?

Introduction

The Linear Least Squares Problem (LS)

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

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

Why the problem is so difficult?

- Enormous variability of LS problems even when considering them only algebraically
- The sparsity structure of $A^T A$ often harder than expected.
- Sparsity structure of $A^T A$ is always behind the scene in the Cholesky/QR approaches even when the normal equations are not formed.

Introduction

Why $A^T A$ is always behind the scene and what makes problems?

Introduction

Why $A^T A$ is always behind the scene and what makes problems?

- Undergraduate stuff:

$$A = QR \to A^T A = R^T Q^T Q R \to A^T A = R^T R$$

Introduction

Why $A^T A$ is always behind the scene and what makes problems?

- Undergraduate stuff:

$$A = QR \to A^T A = R^T Q^T QR \to A^T A = R^T R$$

- The fill is exactly as predicted by Cholesky of $A^T A$ if A has the strong Hall property

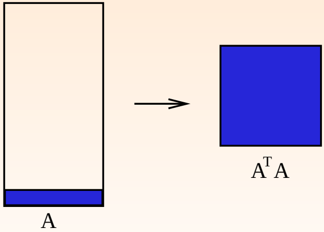
Introduction

Why $A^T A$ is always behind the scene and what makes problems?

- Undergraduate stuff:

$$A = QR \rightarrow A^T A = R^T Q^T Q R \rightarrow A^T A = R^T R$$

- The fill is exactly as predicted by Cholesky of $A^T A$ if A has the strong Hall property
- A trivial example of a problem with structure of $A^T A$.



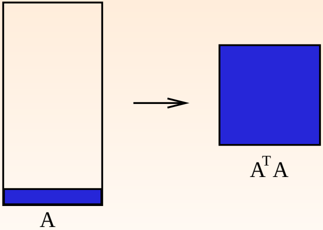
Introduction

Why $A^T A$ is always behind the scene and what makes problems?

- Undergraduate stuff:

$$A = QR \rightarrow A^T A = R^T Q^T QR \rightarrow A^T A = R^T R$$

- The fill is exactly as predicted by Cholesky of $A^T A$ if A has the strong Hall property
- A trivial example of a problem with structure of $A^T A$.



- This is only a simple case, but it may help to understand more complex situations.

Introduction

Trying to understand difficulties from structural point of view

- Denote rows of A by $a_i, i = 1, \ldots, n$. Then (adding rank-one terms)

$$A^T A = \sum_{i=1}^{n} a_i a_i^T$$

Introduction

Trying to understand difficulties from structural point of view

- Denote rows of A by $a_i, i = 1, \dots, n$. Then (adding rank-one terms)

$$A^T A = \sum_{i=1}^{n} a_i a_i^T$$

-

$$A = \begin{array}{c}\\ \vdots \\ i \\ \vdots \\ \vdots \\ j \\ \vdots\end{array}\begin{array}{c}\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array}\\ \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ * & * & & * & & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & * & & * & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}\end{array}$$

Introduction

Trying to understand difficulties from structural point of view

- Denote rows of A by $a_i, i = 1, \ldots, n$. Then (adding rank-one terms)

$$A^T A = \sum_{i=1}^{n} a_i a_i^T$$

-

$$A = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ i & * & * & & * & & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ j & & * & & * & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

-

$$A^T A = \ldots \oplus \begin{array}{c|cccc} & 1 & 2 & 4 & 6 \\ 1 & * & * & * & * \\ 2 & * & * & * & * \\ 4 & * & * & * & * \\ 6 & * & * & * & * \end{array} \oplus \ldots \oplus \begin{array}{c|cc} & 2 & 4 \\ 2 & * & * \\ 4 & * & * \end{array} \oplus \ldots$$

Introduction

Trying to understand difficulties from the structural point of view

- In other words, we are adding cliques (in graph terminology)

$$A^TA = \ldots \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \begin{array}{c|cccc} & 1 & 2 & 4 & 6 \\ \hline 1 & * & * & * & * \\ 2 & * & * & * & * \\ 4 & * & * & * & * \\ 6 & * & * & * & * \end{array} \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \ldots \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \begin{array}{c|cc} & 2 & 4 \\ \hline 2 & * & * \\ 4 & * & * \end{array} \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \ldots$$

Introduction

Trying to understand difficulties from the structural point of view

- In other words, we are adding cliques (in graph terminology)

$$A^T A = \ldots \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \begin{array}{c|cccc} & 1 & 2 & 4 & 6 \\ \hline 1 & * & * & * & * \\ 2 & * & * & * & * \\ 4 & * & * & * & * \\ 6 & * & * & * & * \end{array} \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \ldots \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \begin{array}{c|cc} & 2 & 4 \\ \hline 2 & * & * \\ 4 & * & * \end{array} \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \ldots$$

- But the set of operations in the subsequent Cholesky factorization of $A^T A$ is very similar (remember the multifrontal method). ☺

Introduction

Trying to understand difficulties from the structural point of view

- In other words, we are adding cliques (in graph terminology)

$$A^TA = \ldots \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \begin{array}{c|cccc} & 1 & 2 & 4 & 6 \\ \hline 1 & * & * & * & * \\ 2 & * & * & * & * \\ 4 & * & * & * & * \\ 6 & * & * & * & * \end{array} \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \ldots \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \begin{array}{c|cc} & 2 & 4 \\ \hline 2 & * & * \\ 4 & * & * \end{array} \mathbin{\updownarrow\!\!\!\!\leftrightarrow} \ldots$$

- But the set of operations in the subsequent Cholesky factorization of A^TA is very similar (remember the multifrontal method). ☺

-

$$A = \begin{pmatrix} a_{11} & v^T \\ v & C \end{pmatrix} = \begin{pmatrix} 1 & \\ v/a_{11} & I \end{pmatrix} \begin{pmatrix} a_{11} & \\ & C - vv^T/a_{11} \end{pmatrix} \begin{pmatrix} 1 & v^T/a_{11} \\ & I \end{pmatrix}$$

- Again, fill-in based on cliques (in predefined order)

Introduction

Normal equations, factorization and implications for us

- Solving LS via normal equations means structurally two layers of cliques

Introduction

Normal equations, factorization and implications for us

- Solving LS via normal equations means structurally two layers of cliques
- Why and when is this of interest? Why not considering QR (backward stable) directly?

Normal equations, factorization and implications for us

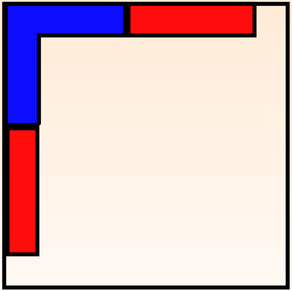
- Solving LS via normal equations means structurally two layers of cliques
- Why and when is this of interest? Why not considering QR (backward stable) directly?
  - In complete factorizations is this view probably of less interest (large fill-in)
  - In incomplete factorizations used as preconditioners this may be a relation to think about:

Normal equations, factorization and implications for us

- Solving LS via normal equations means structurally two layers of cliques
- Why and when is this of interest? Why not considering QR (backward stable) directly?
  - In complete factorizations is this view probably of less interest (large fill-in)
  - In incomplete factorizations used as preconditioners this may be a relation to think about:
    - ★ There is apparently no reliable incomplete QR for solving large least squares. So far, as I hope.
    - ★ Clique-based view: two levels of approximation possible: (1) for A^TA and (2) for subsequent Cholesky
    - ★ Motivating example for the approach: rank-one based preconditioner construction

Motivating example for one-level rank-one updates

- Rank-1 (rank-k) modifications of (approximate) factorizations from some rows of A^TA (Tismenetsky (1991); Kaporin (1998); Scott, T. (2014)) may generate dense contributions for the Schur complement.

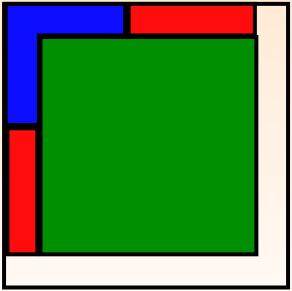


Before the update: blue: big entries, red: small entries

Introduction

Motivating example for one-level rank-one updates

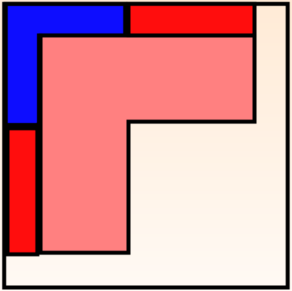
- Rank-1 (rank-k) modifications of (approximate) factorizations from some rows of $A^T A$ (Tismenetsky (1991); Kaporin (1998); Scott, T. (2014)) may generate dense contributions into the Schur complement.



Should be in the update

Motivating example for one-level rank-one updates

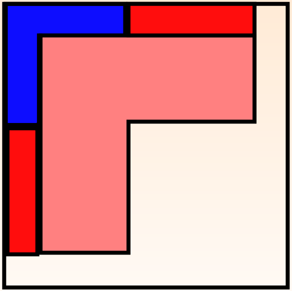
- Rank-1 (rank-k) modifications of (approximate) factorizations from some rows of A^TA (Tismenetsky (1991); Kaporin (1998); Scott, T. (2014)) may generate dense contributions into the Schur complement.



Really kept in the incomplete update

Motivating example for one-level rank-one updates

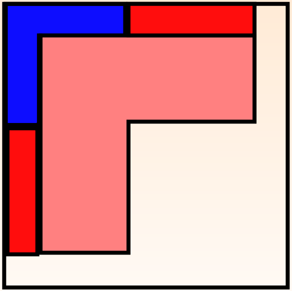
- Rank-1 (rank-k) modifications of (approximate) factorizations from some rows of $A^T A$ (Tismenetsky (1991); Kaporin (1998); Scott, T. (2014)) may generate dense contributions into the Schur complement.



How to exploit this in the two-level clique-based approximate construction?

Motivating example for one-level rank-one updates

- Rank-1 (rank-k) modifications of (approximate) factorizations from some rows of $A^T A$ (Tismenetsky (1991); Kaporin (1998); Scott, T. (2014)) may generate dense contributions into the Schur complement.



A note:: the same structure as in incomplete QR with complete Q

Towards the goal

Back to reality

- The talk mentions the approaches to solve the problem caused by one large clique of A only: implied by a set of dense rows in A.

Towards the goal

Back to reality

- The talk mentions the approaches to solve the problem caused by one large clique of A only: implied by a set of dense rows in A.
- We call this problem sparse-dense

Towards the goal

Back to reality

- The talk mentions the approaches to solve the problem caused by one large clique of A only: implied by a set of dense rows in A.
- We call this problem sparse-dense
- Of course, we could solve just the sparse problem and then update, but let us try more integrated approaches.

Towards the goal

Back to reality

- The talk mentions the approaches to solve the problem caused by one large clique of A only: implied by a set of dense rows in A.
- We call this problem sparse-dense
- Of course, we could solve just the sparse problem and then update, but let us try more integrated approaches.
- Notation for the mixed sparse-dense problem: sparse problem with a few dense rows (structurally a clique)

$$A = \begin{pmatrix} A_s \\ A_d \end{pmatrix}$$

$$C = \begin{pmatrix} A_s^T & A_d^T \end{pmatrix} \begin{pmatrix} A_s \\ A_d \end{pmatrix} = A_s^T A_s + A_d^T A_d \equiv C_s + C_d$$

  - $A_s \in \mathbb{R}^{m_s \times n}$ is sparse, $A_d \in \mathbb{R}^{m_d \times n}$ is dense, $(m_s \gg m_d)$.
  - Full column rank of A (not necessarily of A_s)

Towards the goal

The approaches

1. Combining sparse and dense parts of A
   - Arbitrary sparse-dense (ASD) approach (Scott., T., 2017)
   - Solver: iterative approach based on CG (CGLS1)
   - Specific modifications needed if $rank(A) > rank(A_s)$: .
   - In fact, an implicit combination of the dense (large clique) part and the rest (set of remaining cliques) coupled together inside CG to get

$$z = M^{-1}r.$$

Towards the goal

The approaches

1. Combining sparse and dense parts of A
   - Arbitrary sparse-dense (ASD) approach (Scott., T., 2017)
   - Solver: iterative approach based on CG (CGLS1)
   - Specific modifications needed if $rank(A) > rank(A_s)$: .
   - In fact, an implicit combination of the dense (large clique) part and the rest (set of remaining cliques) coupled together inside CG to get

$$z = M^{-1} r.$$

2. Transforming A_d to a sparse set of rows at the expense of getting the problem larger.
   - Sparsifying the dense part by matrix stretching (Scott, T., 2019)
   - Hoping to get overall "uniform problem sparsity"
   - Traps on the way: size increase / ill-conditioning
   - Attempts with QR factorization in extended space (Scott, T. 2021)

Towards the goal

The approaches (2)

3. Null-space approach (Scott, T., 2022)
    - Saddle-point structure
    - An approach to develop and test construction of null-space bases of wide matrices
4. Schur complement approach (Scott, T., 2018)

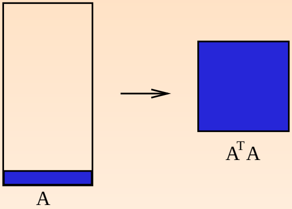
The approaches (2)

3. Null-space approach (Scott, T., 2022)
   - Saddle-point structure
   - An approach to develop and test construction of null-space bases of wide matrices
4. Schur complement approach (Scott, T., 2018)

- All mentioned approaches have specific strengths, weaknesses and a potential to be further developed.
- We intend to discuss here mainly ideas, not techniques.

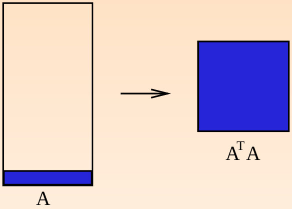
Combining sparse and dense parts of A

1. Combining sparse and dense parts of A



Combining sparse and dense parts of A

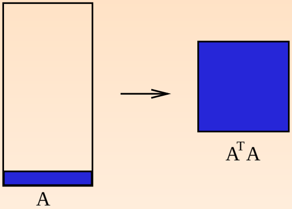
1. Combining sparse and dense parts of A



- Woodbury formulas (1949, 1950) rewritten for residual updates

Combining sparse and dense parts of A

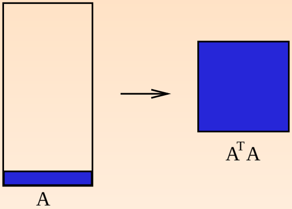
1. Combining sparse and dense parts of A



- Woodbury formulas (1949, 1950) rewritten for residual updates
  - Sometimes such approach interpreted as compute (sparse) and update (by dense)

Combining sparse and dense parts of A

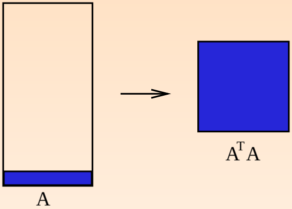
1. Combining sparse and dense parts of A



- Woodbury formulas (1949, 1950) rewritten for residual updates
  - Sometimes such approach interpreted as compute (sparse) and update (by dense)
  - But even in this one-clique case we have more possible ways: sparse → dense, dense → sparse.

Combining sparse and dense parts of A

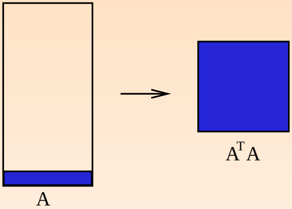
1. Combining sparse and dense parts of A



- Woodbury formulas (1949, 1950) rewritten for residual updates
  - Sometimes such approach interpreted as compute (sparse) and update (by dense)
  - But even in this one-clique case we have more possible ways: sparse → dense, dense → sparse.
  - Moreover, dense part can be structured. Moreover, our approach is: incomplete clique, incomplete update

Combining sparse and dense parts of A

1. Combining sparse and dense parts of A



- Woodbury formulas (1949, 1950) rewritten for residual updates
  - Sometimes such approach interpreted as compute (sparse) and update (by dense)
  - But even in this one-clique case we have more possible ways: sparse → dense, dense → sparse.
  - Moreover, dense part can be structured. Moreover, our approach is: incomplete clique, incomplete update
- There are ways to overcome rank deficiency of A_s.

Combining sparse and dense parts of A

1. Combining sparse and dense parts of A

- Example of (hidden) Woodbury-like formulas

Theorem

If $C_s = L_s L_s^T$ and ξ_1 minimizes $\|A_s L_s^{-T} z - b_s\|_2$ exactly, the exact least squares solution of our problem can be written as $x = L_s^{-T}(\xi_1 + \Gamma_1)$, $\rho_d = b_d - A_d L_s^{-T}\xi_1$ and

$$\Gamma_1 = L_s^{-1} A_d^T (I_{m_d} + A_d L_s^{-T} L_s^{-1} A_d^T)^{-1} \rho_d.$$

Combining sparse and dense parts of A

1. Combining sparse and dense parts of A

- Example of (hidden) Woodbury-like formulas

Theorem

If $C_s = L_s L_s^T$ *and* ξ_1 *minimizes* $\|A_s L_s^{-T} z - b_s\|_2$ *exactly, the exact least squares solution of our problem can be written as* $x = L_s^{-T}(\xi_1 + \Gamma_1)$, $\rho_d = b_d - A_d L_s^{-T} \xi_1$ *and*

$$\Gamma_1 = L_s^{-1} A_d^T (I_{m_d} + A_d L_s^{-T} L_s^{-1} A_d^T)^{-1} \rho_d.$$

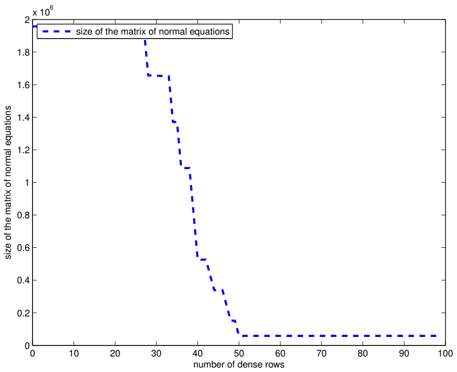
Theorem

If $C_s = L_s L_s^T$ *and* ξ_1 *is an approximate solution to the problem* $\min_z \left\|A_s L_s^{-T} z - b_s\right\|_2$*, the exact least squares solution of the equivalent problem above can be written as* $z = \xi_1 + \Gamma_1$*, where* $\rho_s = b_s - A_s L_s^{-T} \xi_1$, $\rho_d = b_d - A_d L_s^{-T} \xi_1$ *and*

$$\Gamma_1 = L_s^{-1} A_s^T \rho_s + L_s^{-1} A_d^T (I_{m_d} + A_d L_s^{-T} L_s^{-1} A_d^T)^{-1} (\rho_d - A_d L_s^{-T} L_s^{-1} A_s^T \rho_s).$$

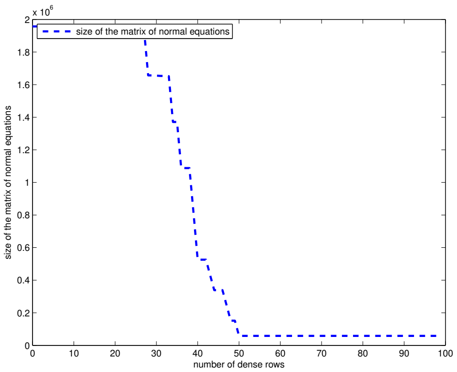
ASD: Arbitrary sparse-dense preconditioning

SCSD8-2r_a (m=60,550; n=8,650): size of C_s



ASD: Arbitrary sparse-dense preconditioning

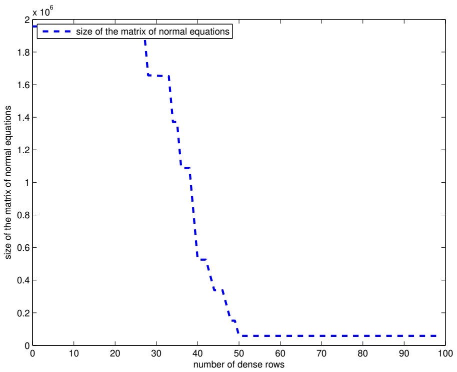
SCSD8-2r_a (m=60,550; n=8,650): size of C_s



- This is not only one clique but 50 cliques merged together

ASD: Arbitrary sparse-dense preconditioning

SCSD8-2r_a (m=60,550; n=8,650): size of C_s



- This is not only one clique but 50 cliques merged together
- They are far from being dense. Can be split into more dense blocks!

ASD: Moving rows one by one from A_s to A_d

SCSD8-2r_a: iteration counts + $size\_p/size(A^TA)$

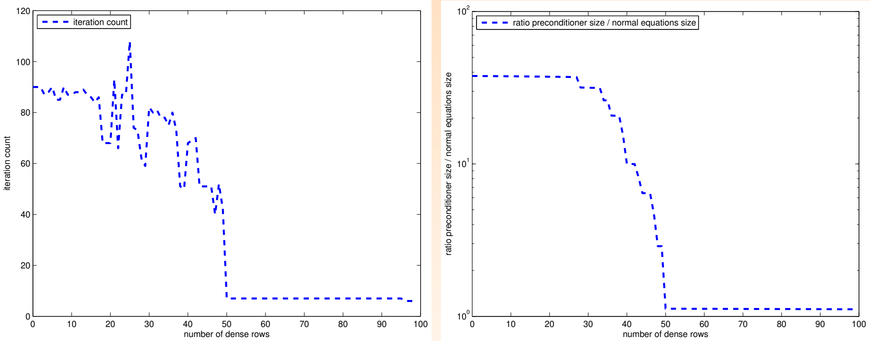


Figure: Problem $Meszaros/scsd8-2r$. Iteration counts (left), and ratio of the preconditioner size to the size of A^TA (right) as the number of dense rows that are removed from A is increased.

ASD: Moving rows one by one from A_s to A_d

SCSD8-2r_a: timings

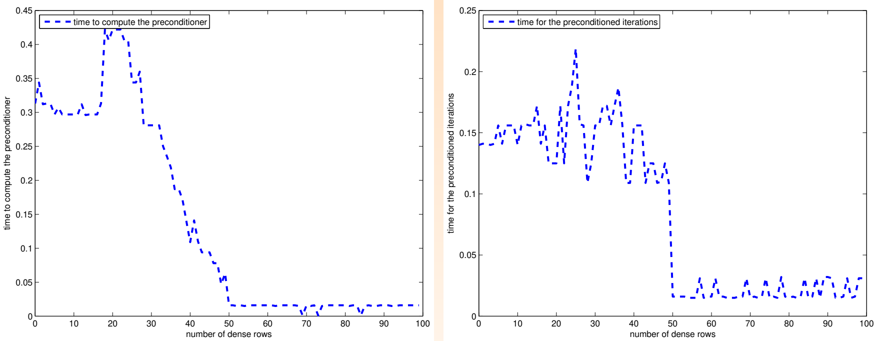


Figure: Problem $Meszaros/scsd8-2r$. Time to compute the preconditioner (left) and time for CGLS (right) as the number of dense rows that are removed from A is increased.

Experimental evaluation of ASD

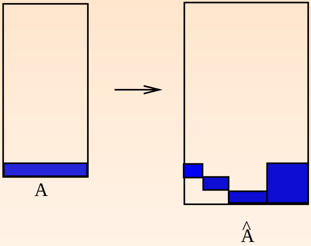
	Dense rows not exploited				Dense rows exploited				
Identifier	*size_p*	*T_p*	*Its*	*T_i*	m_d	*size_ps*	*T_p*	*Its*	*T_i*
lp_fit2p	17,985	0.26	‡	‡	25	4,940	0.09	1	0.01
scsd8-2r	51,885	0.25	90	0.11	50	51,855	0.05	7	0.02
scagr7-2r	197,067	3,34	244	0.53	7	152,977	0.06	1	0.01
scfxm1-2r	227,835	0.59	187	0.51	58	227,823	0.14	33	0.23
neos1	789,471	†	†	†	74	789,471	5.27	132	3.71
neos2	†	†	†	†	90	795,323	5.46	157	4.84
stormg2-125	395,595	0.27	‡	‡	121	7,978,135	0.22	16	0.29
PDE1	†	†	†	†	1	1,623,531	12.7	696	1.28
neos	†	†	†	†	20	2,874,699	4.93	232	15.0
stormg2_1000	3,157,095	19.1	‡	‡	121	3,125,987	19.1	18	2.92
cont1_l	†	†	†	†	1	11,510,370	4.82	1	0.33

2. Matrix stretching

- Stretching: a specific sparsification by splitting dense rows into sparse pieces.

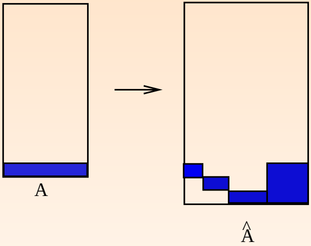
2. Matrix stretching

- Stretching: a specific sparsification by splitting dense rows into sparse pieces.
- The problem is augmented both by rows and columns.



2. Matrix stretching

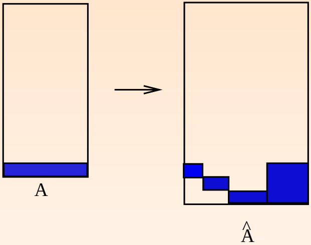
- Stretching: a specific sparsification by splitting dense rows into sparse pieces.
- The problem is augmented both by rows and columns.



- Such strategy called stretching discussed (among others) by Grcar (1990), Vanderbei (1991), Gondzio (1991), Alvarado (1997), Adler (2000), Adler, Björck (2000), Duff, Scott (2005).

2. Matrix stretching

- Stretching: a specific sparsification by splitting dense rows into sparse pieces.
- The problem is augmented both by rows and columns.



- Such strategy called stretching discussed (among others) by Grcar (1990), Vanderbei (1991), Gondzio (1991), Alvarado (1997), Adler (2000), Adler, Björck (2000), Duff, Scott (2005).
- Up to now it has not been an approach of choice

2. Matrix stretching

- An example of one-row stretching

$$\begin{pmatrix} A_{se} & A_{sf} \\ e & f \end{pmatrix} \longrightarrow \begin{pmatrix} A_{se} & A_{sf} & 0 \\ \sqrt{2}\,e & 0 & 1 \\ 0 & \sqrt{2}\,f & -1 \end{pmatrix}$$

2. Matrix stretching

- An example of one-row stretching

$$\begin{pmatrix} A_{se} & A_{sf} \\ e & f \end{pmatrix} \longrightarrow \begin{pmatrix} A_{se} & A_{sf} & 0 \\ \sqrt{2}\,e & 0 & 1 \\ 0 & \sqrt{2}\,f & -1 \end{pmatrix}$$

- Behind: splitting and an orthogonal transformation.

2. Matrix stretching

- An example of one-row stretching

$$\begin{pmatrix} A_{se} & A_{sf} \\ e & f \end{pmatrix} \longrightarrow \begin{pmatrix} A_{se} & A_{sf} & 0 \\ \sqrt{2}\,e & 0 & 1 \\ 0 & \sqrt{2}\,f & -1 \end{pmatrix}$$

- Behind: splitting and an orthogonal transformation.
- The transformation can be used for more rows and more parts

2. Matrix stretching

- An example of one-row stretching

$$\begin{pmatrix} A_{se} & A_{sf} \\ e & f \end{pmatrix} \longrightarrow \begin{pmatrix} A_{se} & A_{sf} & 0 \\ \sqrt{2}\,e & 0 & 1 \\ 0 & \sqrt{2}\,f & -1 \end{pmatrix}$$

- Behind: splitting and an orthogonal transformation.
- The transformation can be used for more rows and more parts
- But, there are problems with stretching. The first of them: how many parts? Grcar (1990):" the main challenge ... lies in determinining the appropriate choice of the number of rows ... to split into ... "

2. Matrix stretching

- An example of one-row stretching

$$\begin{pmatrix} A_{se} & A_{sf} \\ e & f \end{pmatrix} \longrightarrow \begin{pmatrix} A_{se} & A_{sf} & 0 \\ \sqrt{2}\,e & 0 & 1 \\ 0 & \sqrt{2}\,f & -1 \end{pmatrix}$$

- Behind: splitting and an orthogonal transformation.
- The transformation can be used for more rows and more parts
- But, there are problems with stretching. The first of them: how many parts? Grcar (1990):“ the main challenge ... lies in determinining the appropriate choice of the number of rows ... to split into ... ”
- Our answer: Dense cliques should be compatible with the remaining (sparse) part A_s of A.

2. Matrix stretching

How to do this: back to cliques

- $$A^T A = \sum_{i=1}^{n} \mathbf{a_i}^T \mathbf{a_i},\ a_i, i = 1, \ldots, n \text{ are rows of } A.$$

2. Matrix stretching

How to do this: back to cliques

- $$A^T A = \sum_{i=1}^{n} \mathbf{a_i}^T \mathbf{a_i},\ a_i, i = 1, \ldots, n \text{ are rows of } A.$$
- What if a pattern of a row $\mathbf{a_j}$ is contained in the pattern of a row $\mathbf{a_i}$ (dominated by $\mathbf{a_i}$)?

2. Matrix stretching

How to do this: back to cliques

- $$A^T A = \sum_{i=1}^{n} \mathbf{a_i}^T \mathbf{a_i},\ a_i, i = 1, \ldots, n \text{ are rows of } A.$$
- What if a pattern of a row $\mathbf{a_j}$ is contained in the pattern of a row $\mathbf{a_i}$ (dominated by $\mathbf{a_i}$)?
- ⇒ Pattern of $\mathbf{a_j}$ is not needed to get the pattern of $A^T A$.

$$A = \begin{array}{cc} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} \vdots \\ \mathbf{a_i} \\ \vdots \\ \mathbf{a_j} \\ \vdots \end{array} & \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ * & & & * & & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ * & & & * & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \end{array}, \qquad \hat{A} : A \text{ without the row } a_j$$

How to do this: back to cliques

- $$A^T A = \sum_{i=1}^{n} \mathbf{a_i}^T \mathbf{a_i},\ a_i, i = 1, \ldots, n \text{ are rows of } A.$$
- What if a pattern of a row $\mathbf{a_j}$ is contained in the pattern of a row $\mathbf{a_i}$ (dominated by $\mathbf{a_i}$)?
- ⇒ Pattern of $\mathbf{a_j}$ is not needed to get the pattern of $A^T A$.

$$A = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{a_i} & * & & & * & & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{a_j} & * & & & * & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}, \qquad \hat{A} : A \text{ without the row } a_j$$

- $A^T A$ and $\hat{A}^T \hat{A}$ have the same sparsity patterns.

Matrix stretching

- The idea: split a_j into (noncontiguous) subvectors dominated by rows in A_s!

$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} \\ \\ \\ \\ \\ \mathbf{a_j} \end{array} & \left(\begin{array}{ccccc} \ast & \ast & & & \\ & & \ast & & \\ \ast & & \ast & & \ast \\ & \ast & & \ast & \\ \ast & & & & \\ \ast & \ast & \ast & \ast & \ast \end{array}\right) \end{array}$$

Matrix stretching

- The idea: split a_j into (noncontiguous) subvectors dominated by rows in A_s!

$$\begin{array}{c c}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} \\ \\ \\ \\ \\ \mathbf{a_j} \end{array} & \left(\begin{array}{ccccc}
* & * & & & \\
 & & * & & \\
\textcolor{red}{*} & & \textcolor{red}{*} & & \textcolor{red}{*} \\
 & * & & * & \\
* & & & & \\
\colorbox{red}{$*$} & \colorbox{white}{$*$} & \colorbox{red}{$*$} & \colorbox{white}{$*$} & \colorbox{red}{$*$}
\end{array} \right)
\end{array}$$

Matrix stretching

- The idea: split a_j into (noncontiguous) subvectors dominated by rows in A_s!

$$
\begin{array}{c c}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} \\ \\ \\ \\ \\ \mathbf{a_j} \end{array} &
\left(\begin{array}{ccccc}
* & * & & & \\
 & & * & & \\
{\color{red}*} & & {\color{red}*} & & {\color{red}*} \\
 & {\color{blue}*} & & {\color{blue}*} & \\
* & & & & \\
\colorbox{red}{$*$} & \colorbox{blue}{$*$} & \colorbox{red}{$*$} & \colorbox{blue}{$*$} & \colorbox{red}{$*$}
\end{array}\right)
\end{array}
$$

Matrix stretching

- The idea: split a_j into (noncontiguous) subvectors dominated by rows in A_s!

$$
\begin{array}{cc}
 & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} \\ \\ \\ \\ \\ \mathbf{a_j} \end{array} &
\left(\begin{array}{ccccc}
* & * & & & \\
 & & * & & \\
* & & * & & * \\
 & * & & * & \\
* & & & & \\
\boxed{*} & \boxed{*} & \boxed{*} & \boxed{*} & \boxed{*}
\end{array}\right)
\end{array}
$$

And then stretch

Matrix stretching

- The idea: split a_j into (noncontiguous) subvectors dominated by rows in A_s!

$$\begin{array}{cc}
 & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{array}{c} \\ \\ \\ \\ \\ \mathbf{a_{j1}} \\ \mathbf{a_{j2}} \end{array} &
\begin{pmatrix}
* & * & & & & \\
 & & * & & & \\
* & & * & & * & \\
 & * & & * & & \\
* & & & & & \\
\boxed{*} & & \boxed{*} & & \boxed{*} & * \\
 & \boxed{*} & & \boxed{*} & & *
\end{pmatrix}
\end{array}$$

Matrix stretching

- The idea: split a_j into (noncontiguous) subvectors dominated by rows in A_s!

$$\begin{array}{cc}
 & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{array}{c} \\ \\ \\ \\ \\ \mathbf{a_{j1}} \\ \mathbf{a_{j2}} \end{array} &
\left(\begin{array}{cccccc}
* & * & & & & \\
 & & * & & & \\
{\color{red}*} & & {\color{red}*} & & {\color{red}*} & \\
 & {\color{blue}*} & & {\color{blue}*} & & \\
* & & & & & \\
\boxed{*} & & \boxed{*} & & \boxed{*} & * \\
 & \boxed{*} & & \boxed{*} & & *
\end{array}\right)
\end{array}$$

Covering a row by other rows can be casted as a minimum cover problem

Matrix stretching

$$\begin{pmatrix}
* & & & & & & & & & & & * \\
& * & & & & & & * & & * & & \\
& & * & & & & & & & & & \\
& & & * & & & & & & & & \\
* & * & & & * & & * & & & & & \\
& & & & & * & & & & & & \\
& & & & & & * & & & & * & \\
& & * & & & & & & & & & * \\
& & * & * & & * & & & & & & \\
& & * & * & & & & & & & & \\
& & & & & & * & & & & & \\
& & & & & & & & * & & * & \\
& & & & & & & & & & & * \\
& & * & & & & & & & * & & \\
* & * & * & * & * & * & * & * & & * & & *
\end{pmatrix}$$

Matrix stretching

$$\begin{pmatrix}
* & & & & & & & & & & & * \\
 & \color{red}{*} & & & & & & \color{red}{*} & & \color{red}{*} & & \\
 & & * & & & & & & & & & \\
 & & & * & & & & & & & & \\
* & * & & & * & & * & & & & & \\
 & & & & & * & & & & & & \\
 & & & & & & * & & & & * & \\
 & & * & & & & & & & & & * \\
 & & * & * & & * & & & & & & \\
 & & * & * & & & & & & & & \\
 & & & & & & * & & & & & \\
 & & & & & & & & * & & * & \\
 & & & & & & & & & & & * \\
 & & * & & & & & & & * & & \\
* & \colorbox{red}{$*$} & * & * & * & * & * & \colorbox{red}{$*$} & & \colorbox{red}{$*$} & & *
\end{pmatrix}$$

Matrix stretching

$$\left(\begin{array}{cccccccccccc}
* & & & & & & & & & & & * \\
 & \color{red}{*} & & & & & & \color{red}{*} & & \color{red}{*} & & \\
 & & * & & & & & & & & & \\
 & & & * & & & & & & & & \\
\color{blue}{*} & \color{blue}{*} & & & \color{blue}{*} & & \color{blue}{*} & & & & & \\
 & & & & & * & & & & & & \\
 & & & & & & * & & & & * & \\
 & & * & & & & & & & & & * \\
 & & * & * & & * & & & & & & \\
 & & * & * & & & & & & & & \\
 & & & & & & * & & & & & \\
 & & & & & & & & * & & * & \\
 & & & & & & & & & & & * \\
 & & * & & & & & & & * & & \\
\colorbox{blue}{$*$} & \colorbox{red}{$*$} & * & * & \colorbox{blue}{$*$} & * & \colorbox{blue}{$*$} & \colorbox{red}{$*$} & & \colorbox{red}{$*$} & & *
\end{array}\right)$$

Matrix stretching

$$\begin{pmatrix}
* & & & & & & & & & & & * \\
& * & & & & & & * & & * & & \\
& & * & & & & & & & & & \\
& & & * & & & & & & & & \\
* & * & & & * & & * & & & & & \\
& & & & & * & & & & & & \\
& & & & & & * & & & & * & \\
& & * & & & & & & & & & * \\
& & * & * & & * & & & & & & \\
& & * & * & & & & & & & & \\
& & & & & & * & & & & & \\
& & & & & & & & * & & * & \\
& & & & & & & & & & & * \\
& & * & & & & & & & * & & \\
* & * & * & * & * & * & * & * & & * & & *
\end{pmatrix}$$

Matrix stretching

$$\begin{pmatrix}
* & & & & & & & & & & & * \\
 & {\color{red}*} & & & & & & {\color{red}*} & & {\color{red}*} & & \\
 & & * & & & & & & & & & \\
 & & & * & & & & & & & & \\
{\color{blue}*} & {\color{blue}*} & & & {\color{blue}*} & & {\color{blue}*} & & & & & \\
 & & & & & * & & & & & & \\
 & & & & & & * & & & & * & \\
 & & {\color{magenta}*} & & & & & & & & & {\color{magenta}*} \\
 & & {\color{green}*} & {\color{green}*} & & {\color{green}*} & & & & & & \\
 & & * & * & & & & & & & & \\
 & & & & & & * & & & & & \\
 & & & & & & & & * & & * & \\
 & & & & & & & & & & & * \\
 & & * & & & & & & & * & & \\
\colorbox{blue}{$*$} & \colorbox{red}{$*$} & \colorbox{magenta}{$*$} & \colorbox{green}{$*$} & \colorbox{blue}{$*$} & \colorbox{green}{$*$} & \colorbox{blue}{$*$} & \colorbox{red}{$*$} & & \colorbox{red}{$*$} & & \colorbox{magenta}{$*$}
\end{pmatrix}$$

Matrix stretching

- Segments made to be disjoint.

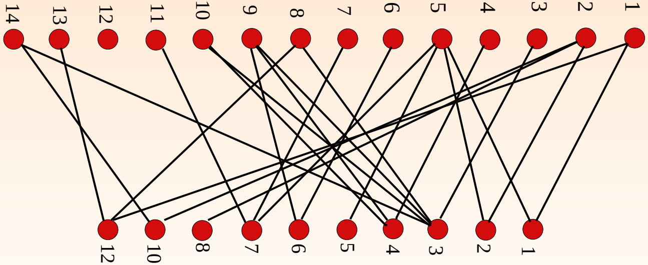
Matrix stretching

- Segments made to be disjoint.
- Finding segments to be stretched: can be formulated as vertex cover of a related bipartite graph

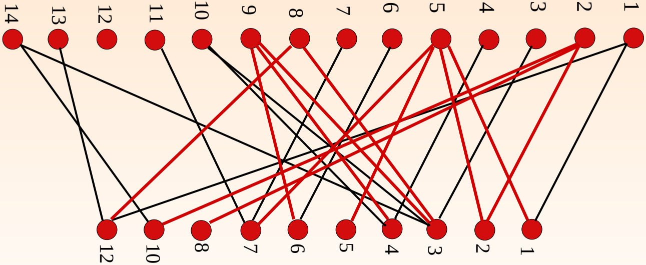
Matrix stretching

- Segments made to be disjoint.
- Finding segments to be stretched: can be formulated as vertex cover of a related bipartite graph
- There are efficient heuristics to do this.

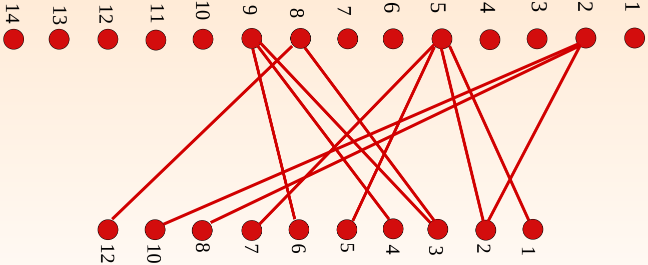
Matrix stretching

- Segments made to be disjoint.
- Finding segments to be stretched: can be formulated as vertex cover of a related bipartite graph
- There are efficient heuristics to do this.
- 

Matrix stretching

- Segments made to be disjoint.
- Finding segments to be stretched: can be formulated as vertex cover of a related bipartite graph
- There are efficient heuristics to do this.
- 

Matrix stretching

- Segments made to be disjoint.
- Finding segments to be stretched: can be formulated as vertex cover of a related bipartite graph
- There are efficient heuristics to do this.
- 

Matrix stretching

- $\mathbf{a_j} \to F,\ A \to \begin{pmatrix} \hat{A} \\ F \end{pmatrix}$

Matrix stretching

- $\mathbf{a_j} \to F$, $A \to \begin{pmatrix} \hat{A} \\ F \end{pmatrix}$
- Splitting F to more rows and stretching

$$\begin{pmatrix} \hat{A} \\ F \end{pmatrix} \longrightarrow \begin{pmatrix} \hat{A} & 0 \\ \hat{F} & S \end{pmatrix}$$

Matrix stretching

- $\mathbf{a_j} \to F,\ A \to \begin{pmatrix} \hat{A} \\ F \end{pmatrix}$
- Splitting F to more rows and stretching

$$\begin{pmatrix} \hat{A} \\ F \end{pmatrix} \longrightarrow \begin{pmatrix} \hat{A} & 0 \\ \hat{F} & S \end{pmatrix}$$

- $$\begin{pmatrix} \hat{A}^T & \hat{F}^T \\ 0 & S^T \end{pmatrix} \begin{pmatrix} \hat{A} & 0 \\ \hat{F} & S \end{pmatrix} = \begin{pmatrix} \hat{A}^T \hat{A} + \hat{F}^T \hat{F} & S^T \hat{F} \\ \hat{F}^T S & S^T S \end{pmatrix}$$

Matrix stretching

- $\mathbf{a_j} \to F$, $A \to \begin{pmatrix} \hat{A} \\ F \end{pmatrix}$
- Splitting F to more rows and stretching

$$\begin{pmatrix} \hat{A} \\ F \end{pmatrix} \longrightarrow \begin{pmatrix} \hat{A} & 0 \\ \hat{F} & S \end{pmatrix}$$

- $$\begin{pmatrix} \hat{A}^T & \hat{F}^T \\ 0 & S^T \end{pmatrix} \begin{pmatrix} \hat{A} & 0 \\ \hat{F} & S \end{pmatrix} = \begin{pmatrix} \hat{A}^T\hat{A} + \hat{F}^T\hat{F} & S^T\hat{F} \\ \hat{F}^T S & S^T S \end{pmatrix}$$

- $S^T S$ is tridiag $\begin{pmatrix} -1 & 2 & -1 \end{pmatrix}$
  - Saddle-point problem?
  - Scaling?

Matrix stretching

Ad subsequent experimental results

- Matrix transformed by the stretching parameter-free based on the minimum set cover heuristic

Matrix stretching

Ad subsequent experimental results

- Matrix transformed by the stretching parameter-free based on the minimum set cover heuristic
- Achieved simultaneously sparsity of normal equations, Cholesky factor size, reasonable iteration counts if used for preconditioning

Matrix stretching

Ad subsequent experimental results

- Matrix transformed by the stretching parameter-free based on the minimum set cover heuristic
- Achieved simultaneously sparsity of normal equations, Cholesky factor size, reasonable iteration counts if used for preconditioning
- This (set cover-based) stretching compared against ad hoc splits.

Matrix stretching

- Number of entries: $A^T A$ and Cholesky factor versus number of parts
- Red dot means the result for the set cover-based stretching

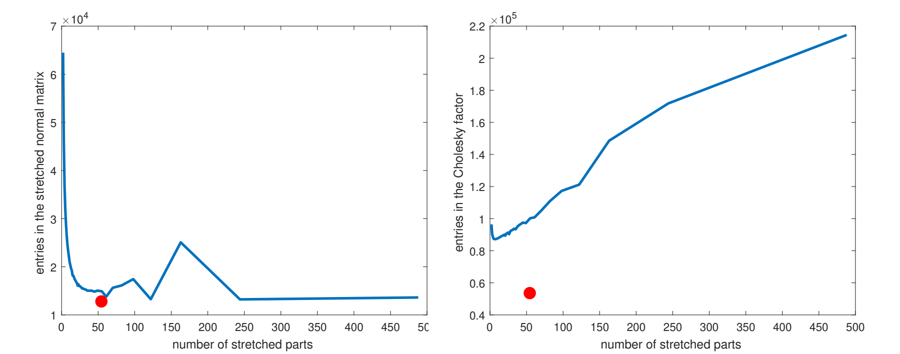


Figure: Comparison of the entries in the stretched normal matrix (left) and its Cholesky factor (right) for problem `LP_AGG` with one dense row appended.

Matrix stretching

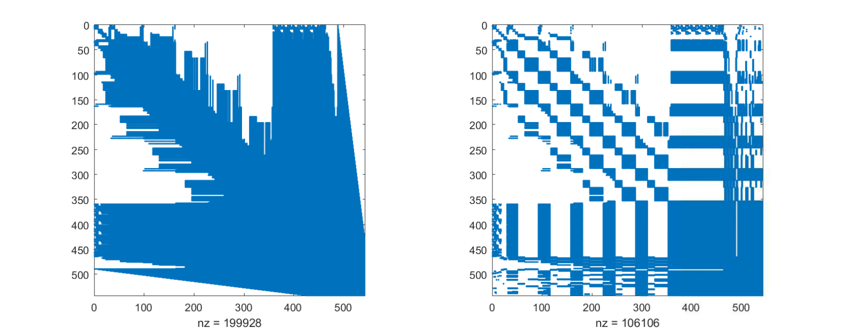


Figure: For problem LP_AGG with one dense row appended, the sparsity pattern of $\hat{L} + \hat{L}^T$ of the Cholesky factor of the stretched normal matrix for ad hoc stretching (left) and set cover-based stretching (right).

Matrix stretching

- The sizes really transfer into the iteration counts

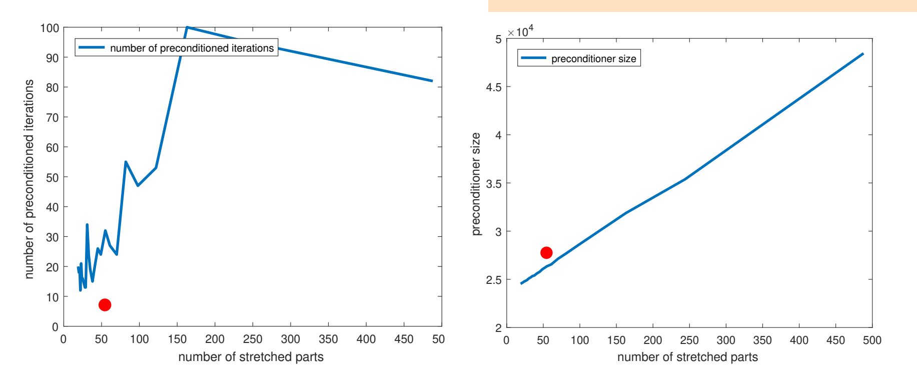


Figure: Comparison of the iteration counts (left) and preconditioner size (right) for the matrix LPAGG. The curve corresponds to the number of entries varying with the number of parts into which is the dense row stretched. CGLS preconditioned by HSL_MI35.

Matrix stretching

Problems with ill-conditioning

- Ill-conditioning in practice is in agreement with (non-optimistic) theoretical bounds.

Matrix stretching

Problems with ill-conditioning

- Ill-conditioning in practice is in agreement with (non-optimistic) theoretical bounds.
- Adlers-Björck theory (see Adlers, Björck, 2000; Scott, T., 2019)

Theorem

An upper bound for the condition number of the stretched matrix (p stretched rows, k parts) $\hat{A}$ with $\gamma = 1/2\sqrt{p\,k}||A_d||_2$ is

$$\kappa^2(\hat{A}) \le \kappa^2(A)k\left(1 + \frac{2\,p\,k||A_d||_2^2}{||A||_2^2}\right)\left(k + 1 + \frac{\sigma_n(A)^2}{||A_d||_2^2}\right).$$

Matrix stretching

Problems with ill-conditioning

- Ill-conditioning in practice is in agreement with (non-optimistic) theoretical bounds.
- Adlers-Björck theory (see Adlers, Björck, 2000; Scott, T., 2019)

Theorem

An upper bound for the condition number of the stretched matrix (p stretched rows, k parts) $\hat{A}$ with $\gamma = 1/2\sqrt{p\,k}||A_d||_2$ is

$$\kappa^2(\hat{A}) \le \kappa^2(A)k\left(1 + \frac{2\,p\,k||A_d||_2^2}{||A||_2^2}\right)\left(k + 1 + \frac{\sigma_n(A)^2}{||A_d||_2^2}\right).$$

- Do we really need to stretch everything?

Matrix stretching

- Condition number increase when stretching more rows

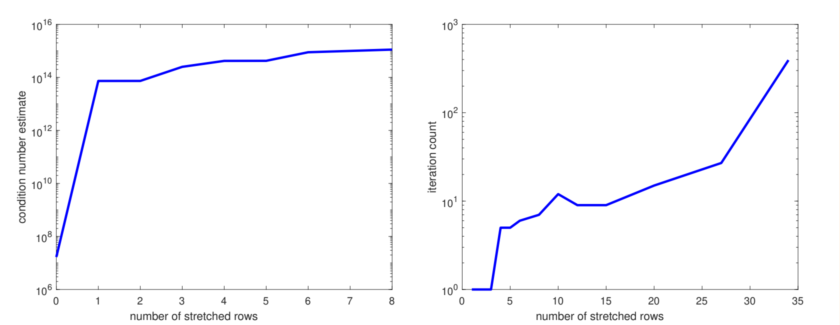


Figure: Condition number estimate (right) and iteration count (left) for problem sctap1-2b as the number of dense rows increases.

Matrix stretching

- Condition number increase when stretching more rows

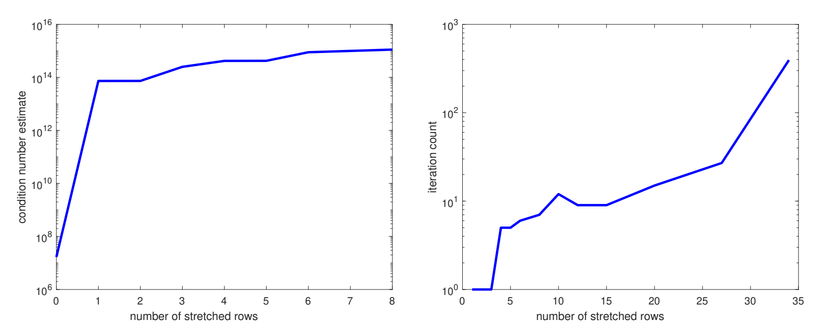


Figure: Condition number estimate (right) and iteration count (left) for problem sctap1-2b as the number of dense rows increases.

- Do we really need to stretch everything (hide all nasty cliques)?

Hiding cliques: partial summary

Combined approach? What can we do?

- Some cliques can be moved to the dense part (that can be itself structured, banded, block triangular etc.) This processing is cheap. Can be even approximate. And the interaction can be combined within a preconditioner.

Hiding cliques: partial summary

Combined approach? What can we do?

- Some cliques can be moved to the dense part (that can be itself structured, banded, block triangular etc.) This processing is cheap. Can be even approximate. And the interaction can be combined within a preconditioner.
- Some cliques can be stretched or embedded. Only some, in order to keep condition number increase only moderate.

Hiding cliques: partial summary

Combined approach? What can we do?

- Some cliques can be moved to the dense part (that can be itself structured, banded, block triangular etc.) This processing is cheap. Can be even approximate. And the interaction can be combined within a preconditioner.
- Some cliques can be stretched or embedded. Only some, in order to keep condition number increase only moderate.
- There are other motivations for stretching, partial stretching.

Hiding cliques: partial summary

Combined approach? What can we do?

- Some cliques can be moved to the dense part (that can be itself structured, banded, block triangular etc.) This processing is cheap. Can be even approximate. And the interaction can be combined within a preconditioner.
- Some cliques can be stretched or embedded. Only some, in order to keep condition number increase only moderate.
- There are other motivations for stretching, partial stretching.
- But, first consider an example that motivated our interest in stretching + direct methods.

Stretching and QR

QR factorization for an unstretched system $(A \to R)$

$$\begin{pmatrix}
* & * & * & * & & & & \\
* & * & * & * & & & & \\
* & * & * & * & & & & \\
* & * & * & * & & & & \\
 & & & & * & * & * & * \\
 & & & & * & * & * & * \\
 & & & & * & * & * & * \\
 & & & & * & * & * & * \\
* & * & * & * & * & * & * & *
\end{pmatrix}
\to
\begin{pmatrix}
* & * & * & * & * & * & * & * \\
 & * & * & * & * & * & * & * \\
 & & * & * & * & * & * & * \\
 & & & * & * & * & * & * \\
 & & & & * & * & * & * \\
 & & & & & * & * & * \\
 & & & & & & * & * \\
 & & & & & & & * \\
 & & & & & & &
\end{pmatrix}$$

Stretching and QR

QR factorization for stretched system ($A \to R$)

$$
\begin{pmatrix}
* & * & * & * & & & & & \\
* & * & * & * & & & & & \\
* & * & * & * & & & & & \\
* & * & * & * & & & & & \\
 & & & & * & * & * & * & \\
 & & & & * & * & * & * & \\
 & & & & * & * & * & * & \\
 & & & & * & * & * & * & \\
* & * & * & * & & & & & * \\
 & & & & * & * & * & * & *
\end{pmatrix}
\to
\begin{pmatrix}
* & * & * & * & & & & & * \\
 & * & * & * & & & & & * \\
 & & * & * & & & & & * \\
 & & & * & & & & & * \\
 & & & & * & * & * & * & * \\
 & & & & & * & * & * & * \\
 & & & & & & * & * & * \\
 & & & & & & & * & * \\
 & & & & & & & & * \\
 & & & & & & & &
\end{pmatrix}
$$

Matrix stretching and QR

- Despite the sharp contrast between stretched/unstretched, but in our experiments more theoretical than really cutting down efficiency in practice (our experience)
- A flavor of other motivations.
- Like solving rank deficient problems.

Schur complement approach

The Schur complement approach

- Fully embedded in the Schur complement approach that combines a direct solver, modifications, regularization to get a preconditioner:
- System matrix varying α

$$K(\alpha) = \begin{pmatrix} C_s(\alpha) & A_d^T \\ A_d & -I_{m_d} \end{pmatrix}.$$

Schur complement approach

The Schur complement approach

- Fully embedded in the Schur complement approach that combines a direct solver, modifications, regularization to get a preconditioner:
- System matrix varying α

$$K(\alpha) = \begin{pmatrix} C_s(\alpha) & A_d^T \\ A_d & -I_{m_d} \end{pmatrix}.$$

- Once the dense rows are clearly detected, the preconditioned iterative method can be extremely successful in solving some hard problems.

Partial stretching versus Schur complement approach

- A variation of stretching for rank-deficient problems: stretch only the rows needed to have A_s full column rank.

Matrix	Meth	$\widetilde{m}$	$\widetilde{n}$	$nnz(\widetilde{R}_s)$	$flops$	its	$ratio$
aircraft	PStr	10517	6754	4.719×10^4	9.33×10^5	9	4.947×10^{-12}
	Regu	11271	3754	3.754×10^3	3.37×10^4	7	5.476×10^{-7}
sc205-2r	PStr	64023	36813	3.175×10^5	8.25×10^6	6	9.983×10^{-9}
	Regu	97636	35213	2.704×10^5	1.21×10^7	7	1.002×10^{-8}
scagr7-2b	PStr	15127	11023	1.186×10^5	5.03×10^6	7	2.129×10^{-12}
	Regu	23590	9743	6.027×10^4	3.67×10^6	8	3.979×10^{-9}
scagr7-2br	PStr	50999	37167	4.673×10^5	1.88×10^7	7	1.046×10^{-10}
	Regu	79526	32847	2.273×10^5	1.28×10^7	8	3.470×10^{-8}
scrs8-2r	PStr	32820	19493	4.242×10^5	4.66×10^7	7	3.311×10^{-12}
	Regu	42055	14364	8.200×10^4	2.84×10^6	16	3.532×10^{-7}

Significantly better (quality) than just regularization (and the Schur complement approach)

Null-space approach

Another trick to annihilate large cliques: use them for the null-space projection

Null-space approach

Another trick to annihilate large cliques: use them for the null-space projection

- Note that we emphasize here only one motivation for the null-space approach!!! There are other ones.

Null-space approach

Another trick to annihilate large cliques: use them for the null-space projection

- Note that we emphasize here only one motivation for the null-space approach!!! There are other ones.
- Is it possible to make this transformation such that the structure of the transformed system is not fully destroyed (filled)?

Null-space approach

Optimization motivation of the null-space approach

- $$\begin{aligned} \text{minimize} \qquad & f(u) \\ \text{subject to} \qquad & B\,u = g, \end{aligned} \tag{1}$$

Null-space approach

Optimization motivation of the null-space approach

- $$\begin{aligned} &\text{minimize} && f(u) \\ &\text{subject to} && B\,u = g, \end{aligned} \qquad (1)$$

- Getting the saddle-point problem for the direction vector u.

$$\begin{pmatrix} H & B^T \\ B & 0_{k,k} \end{pmatrix} \begin{pmatrix} \bar{u} \\ v \end{pmatrix} = \begin{pmatrix} f - H\hat{u} \\ g \end{pmatrix}, \bar{u} = u - \hat{u}, H \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{k\times n} \text{ (full rank)}. \qquad (2)$$

Null-space approach

Optimization motivation of the null-space approach

-

$$\begin{aligned} \text{minimize} \quad & f(u) \\ \text{subject to} \quad & B\,u = g, \end{aligned} \tag{1}$$

- Getting the saddle-point problem for the direction vector u.

$$\begin{pmatrix} H & B^T \\ B & 0_{k,k} \end{pmatrix} \begin{pmatrix} \bar{u} \\ v \end{pmatrix} = \begin{pmatrix} f - H\hat{u} \\ g \end{pmatrix}, \bar{u} = u - \hat{u}, H \in \mathbf{R}^{n\times n}, B \in \mathbf{R}^{k\times n} \text{ (full rank).} \tag{2}$$

- The second equation is equivalent to finding $z \in \mathbb{R}^{n-k}$ such that $\bar{u} = Zz$, columns of $Z \in \mathbf{R}^{n\times(n-k)}$ form a basis for $\mathcal{N}(B)$.

Null-space approach

Algorithm

Dual variable (null-space) method for solving the saddle-point problem

1. *Find Z with columns forming a basis for $\mathcal{N}(B)$*
2. *Find $\hat{u}$ such that $B\hat{u} = g$.*
3. *Solve $Z^T H Z z = Z^T (f - H\hat{u})$.*
4. *Set $x = \hat{u} + Zz$.*
5. *Solve $BB^T v = B(f - Hu)$ for $v \in \mathbf{R}^k$.*

-

Null-space approach

- Standard null-space method uses the fact that the bottom right block of the saddle-point matrix is zero

Null-space approach

- Standard null-space method uses the fact that the bottom right block of the saddle-point matrix is zero
- Saddle-point from the LS problem
  The LS problem can be written as solving the following system

$$\begin{pmatrix} C_s & A_d^T \\ A_d & -I \end{pmatrix} \begin{pmatrix} x \\ A_d x \end{pmatrix} = \begin{pmatrix} c \\ 0 \end{pmatrix}. \tag{3}$$

Null-space approach

- Standard null-space method uses the fact that the bottom right block of the saddle-point matrix is zero
- Saddle-point from the LS problem

  The LS problem can be written as solving the following system

$$\begin{pmatrix} C_s & A_d^T \\ A_d & -I \end{pmatrix} \begin{pmatrix} x \\ A_d x \end{pmatrix} = \begin{pmatrix} c \\ 0 \end{pmatrix}. \tag{3}$$

- We will show how the problem with nonzero C can be overcome.

Null-space approach

- Standard null-space method uses the fact that the bottom right block of the saddle-point matrix is zero
- Saddle-point from the LS problem
  The LS problem can be written as solving the following system

$$\begin{pmatrix} C_s & A_d^T \\ A_d & -I \end{pmatrix} \begin{pmatrix} x \\ A_d x \end{pmatrix} = \begin{pmatrix} c \\ 0 \end{pmatrix}. \tag{3}$$

- We will show how the problem with nonzero C can be overcome.
- Use more general notation as

$$\mathcal{A} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} H & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{4}$$

The null space approach to solve sparse/dense LS problems

Theorem

Consider the saddle-point problem above, $rank(B) = r \leq k$, H, C SPSD, $\mathcal{N}(H) \cap \mathcal{N}(B) = \{0\}$, $\mathcal{N}(C) \cap \mathcal{N}(B^T) = \{0\}$. Then the solution of the system above with generally nonzero C can be obtained by solving a transformed saddle point problem of the order $n + k$ with a symmetric principal leading matrix of order $n - r$.

The null space approach to solve sparse/dense LS problems

Theorem

Consider the saddle-point problem above, $rank(B) = r \leq k$, H, C SPSD, $\mathcal{N}(H) \cap \mathcal{N}(B) = \{0\}$, $\mathcal{N}(C) \cap \mathcal{N}(B^T) = \{0\}$. Then the solution of the system above with generally nonzero C can be obtained by solving a transformed saddle point problem of the order $n + k$ with a symmetric principal leading matrix of order $n - r$.

- Transformation uses the nonsingular matrix $E = \begin{pmatrix} Z & Y \end{pmatrix} \in \mathbb{R}^{n \times n}$, $Z \in \mathbb{R}^{n \times n-r}$ is such that $BE = \begin{pmatrix} 0_{k,n-r} & B_r \end{pmatrix}$, $B_r \in \mathbb{R}^{k \times r}, B_r \neq 0$.

The null space approach to solve sparse/dense LS problems

Theorem

Consider the saddle-point problem above, $rank(B) = r \le k$, H, C SPSD, $\mathcal{N}(H) \cap \mathcal{N}(B) = \{0\}$, $\mathcal{N}(C) \cap \mathcal{N}(B^T) = \{0\}$. Then the solution of the system above with generally nonzero C can be obtained by solving a transformed saddle point problem of the order $n + k$ with a symmetric principal leading matrix of order $n - r$.

- Transformation uses the nonsingular matrix $E = \begin{pmatrix} Z & Y \end{pmatrix} \in \mathbb{R}^{n \times n}$, $Z \in \mathbb{R}^{n \times n-r}$ is such that $BE = \begin{pmatrix} 0_{k,n-r} & B_r \end{pmatrix}$, $B_r \in \mathbb{R}^{k \times r}, B_r \neq 0$.

Lemma

Consider $A = \begin{pmatrix} A_s \\ A_d \end{pmatrix}$, $A_s \in \mathbb{R}^{m_s \times n}$, $A_d \in \mathbb{R}^{m_d \times n}$. If A is of full rank, then $C_s = A_s^T A_s$ is positive definite on $\mathcal{N}(A_d)$.

The null space approach to solve sparse/dense LS problems

Theorem

Consider the saddle-point problem above, $rank(B) = r \le k$, H, C SPSD, $\mathcal{N}(H) \cap \mathcal{N}(B) = \{0\}$, $\mathcal{N}(C) \cap \mathcal{N}(B^T) = \{0\}$. Then the solution of the system above with generally nonzero C can be obtained by solving a transformed saddle point problem of the order $n + k$ with a symmetric principal leading matrix of order $n - r$.

- Transformation uses the nonsingular matrix $E = \begin{pmatrix} Z & Y \end{pmatrix} \in \mathbb{R}^{n \times n}$, $Z \in \mathbb{R}^{n \times n-r}$ is such that $BE = \begin{pmatrix} 0_{k,n-r} & B_r \end{pmatrix}$, $B_r \in \mathbb{R}^{k \times r}, B_r \ne 0$.

Lemma

Consider $A = \begin{pmatrix} A_s \\ A_d \end{pmatrix}$, $A_s \in \mathbb{R}^{m_s \times n}$, $A_d \in \mathbb{R}^{m_d \times n}$. If A is of full rank, then $C_s = A_s^T A_s$ is positive definite on $\mathcal{N}(A_d)$.

- Are we able to construct a suitable Z for a wide matrix?

The null space approach to solve sparse/dense LS problems

Sparse Z for a wide (possibly dense) matrix: example

- An example of $BP = QR$

$$B = \begin{pmatrix} 1 & 2 & 3 & 10 & 4 \end{pmatrix}, \qquad BP = \begin{pmatrix} 10 & 2 & 3 & 1 & 4 \end{pmatrix},$$

$$\widetilde{Z}_1 = \begin{pmatrix} 0.2 & & & \\ -1 & 1.5 & & \\ & -1 & 1/3 & \\ & & -1 & 4 \\ & & & -1 \end{pmatrix}, \; \widetilde{Z}_2 = \begin{pmatrix} 1/5 & 3/10 & 1/10 & 4/10 \\ -1 & & & \\ & -1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix}.$$

The null space approach to solve sparse/dense LS problems

Sparse Z for a wide (possibly dense) matrix: example

- An example of $BP = QR$

$$B = \begin{pmatrix} 1 & 2 & 3 & 10 & 4 \end{pmatrix}, \qquad BP = \begin{pmatrix} 10 & 2 & 3 & 1 & 4 \end{pmatrix},$$

$$\widetilde{Z}_1 = \begin{pmatrix} 0.2 & & & \\ -1 & 1.5 & & \\ & -1 & 1/3 & \\ & & -1 & 4 \\ & & & -1 \end{pmatrix}, \; \widetilde{Z}_2 = \begin{pmatrix} 1/5 & 3/10 & 1/10 & 4/10 \\ -1 & & & \\ & -1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix}.$$

- $Z = P\widetilde{Z}$ (this does not change suitability of Z)

The null space approach to solve sparse/dense LS problems

Sparse Z for a wide (possibly dense) matrix: example

- An example of $BP = QR$

$$B = \begin{pmatrix} 1 & 2 & 3 & 10 & 4 \end{pmatrix}, \qquad BP = \begin{pmatrix} 10 & 2 & 3 & 1 & 4 \end{pmatrix},$$

$$\widetilde{Z}_1 = \begin{pmatrix} 0.2 & & & \\ -1 & 1.5 & & \\ & -1 & 1/3 & \\ & & -1 & 4 \\ & & & -1 \end{pmatrix}, \; \widetilde{Z}_2 = \begin{pmatrix} 1/5 & 3/10 & 1/10 & 4/10 \\ -1 & & & \\ & -1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix}.$$

- $Z = P\widetilde{Z}$ (this does not change suitability of Z)
- Z_1 is OK, Z_2 is not OK

The null space approach to solve sparse/dense LS problems

Sparse Z for a wide (possibly dense) matrix: example

- An example of $BP = QR$

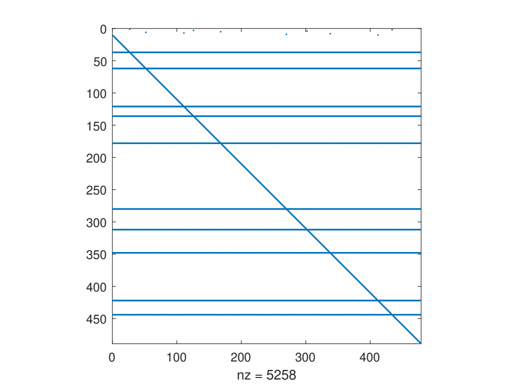
$$B = \begin{pmatrix} 1 & 2 & 3 & 10 & 4 \end{pmatrix}, \qquad BP = \begin{pmatrix} 10 & 2 & 3 & 1 & 4 \end{pmatrix},$$

$$\widetilde{Z}_1 = \begin{pmatrix} 0.2 & & & \\ -1 & 1.5 & & \\ & -1 & 1/3 & \\ & & -1 & 4 \\ & & & -1 \end{pmatrix}, \; \widetilde{Z}_2 = \begin{pmatrix} 1/5 & 3/10 & 1/10 & 4/10 \\ -1 & & & \\ & -1 & & \\ & & -1 & \\ & & & -1 \end{pmatrix}.$$

- $Z = P\widetilde{Z}$ (this does not change suitability of Z)
- Z_1 is OK, Z_2 is not OK
- QR factorization with threshold pivoting to keep locality: this pivoting offers a suitable compromise.

The null space approach to solve sparse/dense LS problems

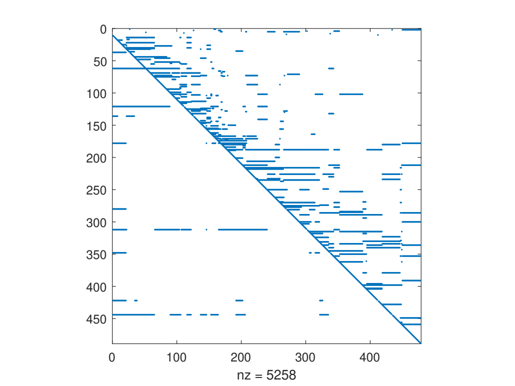
Threshold $thresh = 1$



LPAGG (615×488, UFL Sparse Matrix Collection) + 10 dense rows.

The null space approach to solve sparse/dense LS problems

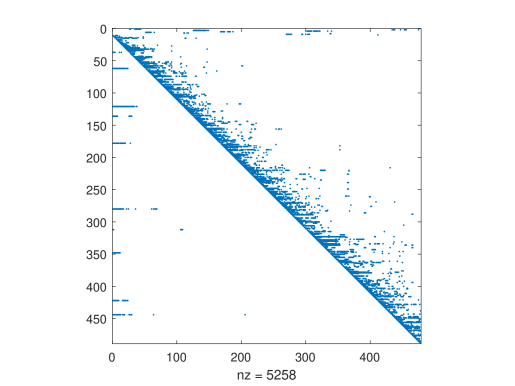
Threshold $thresh = 0.9$



LPAGG (615×488, UFL Sparse Matrix Collection) + 10 dense rows.

The null space approach to solve sparse/dense LS problems

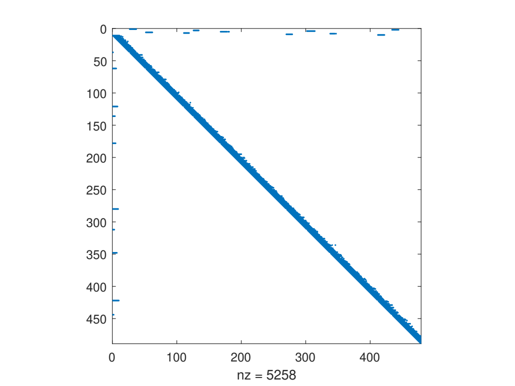
Threshold $thresh = 0.73$



LPAGG (615×488, UFL Sparse Matrix Collection) + 10 dense rows.

The null space approach to solve sparse/dense LS problems

Threshold $thresh = 0.2$



LPAGG (615×488, UFL Sparse Matrix Collection) + 10 dense rows.

Conclusions

We will comment mainly on the space for subsequent research

Conclusions

We will comment mainly on the space for subsequent research

- ASD: still a lot of theoretical challenges (different substitutions and combinations).

Conclusions

We will comment mainly on the space for subsequent research

- ASD: still a lot of theoretical challenges (different substitutions and combinations).
- Stretching - a cute idea but needs to be developed (ill-conditioning, motivations in saddle-point approach?)

Conclusions

We will comment mainly on the space for subsequent research

- ASD: still a lot of theoretical challenges (different substitutions and combinations).
- Stretching - a cute idea but needs to be developed (ill-conditioning, motivations in saddle-point approach?)
- Towards the double layer of cliques?

Conclusions

We will comment mainly on the space for subsequent research

- ASD: still a lot of theoretical challenges (different substitutions and combinations).
- Stretching - a cute idea but needs to be developed (ill-conditioning, motivations in saddle-point approach?)
- Towards the double layer of cliques?
- Null-space approach: a viable way to get over the singularity of A_s.

Conclusions

We will comment mainly on the space for subsequent research

- ASD: still a lot of theoretical challenges (different substitutions and combinations).
- Stretching - a cute idea but needs to be developed (ill-conditioning, motivations in saddle-point approach?)
- Towards the double layer of cliques?
- Null-space approach: a viable way to get over the singularity of A_s.
- Many more questions than expected at the beginning.

Last but not least

Thank you for your attention!

- Great thanks to the organizers!
- Thanks also to our institution and the Doctoral school (in CZ) supported by ESF in Doctoral school for education in mathematical methods and tools in HPC project, CZ.02.2.69/0.0/0.0/16_018/0002713.