Search for robust algebraic preconditioners

Miroslav Tůma

Institute of Computer Science

Academy of Sciences of the Czech Republic

tuma@cs.cas.cz

based on joint work with **Michele Benzi, Rafael Bru, Jurjen Duintjer Tebbens, José Marín, José Mas, Miroslav Rozložník, Jennifer Scott** et al.

Emory University

November 2, 2009, Atlanta

Motivation: I.

Solving large, sparse systems of linear algebraic equations

$$Ax = b$$

Motivation: I.

Solving large, sparse systems of linear algebraic equations

$$Ax = b$$

Contemporary decompositional interpretation of the Gaussian elimination (GE): Householder at the end of the latest 50's.

Motivation: I.

Solving large, sparse systems of linear algebraic equations

$$Ax = b$$

Contemporary decompositional interpretation of the Gaussian elimination (GE): Householder at the end of the latest 50's.

Both different and similar role of GE in the two basic solving approaches:

- Direct methods and iterative methods

Case of our interest: Relaxed GE (incomplete decompositions of various kinds).

Motivation: II.

Incomplete decompositions and their implementation.

- GE: We need sparsity (in the input matrix, elimination graphs' estimates, intermediate data) and the speed of the whole computation.

Motivation: II.

Incomplete decompositions and their implementation.

- GE: We need sparsity (in the input matrix, elimination graphs' estimates, intermediate data) and the speed of the whole computation.
- The sparsity does not seem to be particularly critical when considering plain incomplete decompositions (ID). But, fast implementations of contemporary ID may cause problems.

Motivation: II.

Incomplete decompositions and their implementation.

- GE: We need sparsity (in the input matrix, elimination graphs' estimates, intermediate data) and the speed of the whole computation.
- The sparsity does not seem to be particularly critical when considering plain incomplete decompositions (ID). But, fast implementations of contemporary ID may cause problems.
- Fortunately, some data structures originally developed for direct methods (and not used there anymore) can be successfully used.

Fast implementations of sophisticated GE modifications are possible

Motivation: III.

Incomplete decompositions and robustness

- Robustness of ID jointly with the iterative method is what really critical is.

Motivation: III.

Incomplete decompositions and robustness

- Robustness of ID jointly with the iterative method is what really critical is.
- Partial robustness: in its evaluation (breakdown-free property).
  - May be based on relaxing accuracy of decomposition (decomposing a different matrix)
  - Or, may promote density of the decomposition (restricting the incompleteness (numerically or structurally))
- Stability of ID: important in combination with iterative methods.

Motivation: III.

Incomplete decompositions and robustness

- Robustness of ID jointly with the iterative method is what really critical is.
- Partial robustness: in its evaluation (breakdown-free property).
  - May be based on relaxing accuracy of decomposition (decomposing a different matrix)
  - Or, may promote density of the decomposition (restricting the incompleteness (numerically or structurally))
- Stability of ID: important in combination with iterative methods.

Is is to possible to guarantee more robustness for decompositions by relating them to GE?

Motivation: III.

Incomplete decompositions and robustness

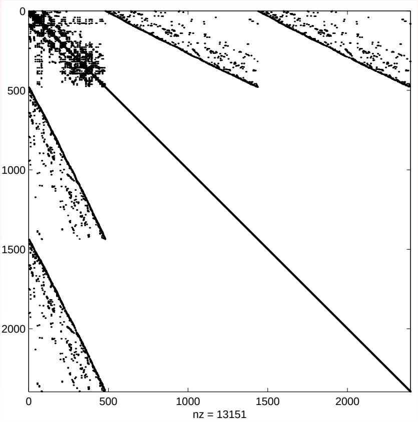
- Robustness of ID jointly with the iterative method is what really critical is.
- Partial robustness: in its evaluation (breakdown-free property).
  - May be based on relaxing accuracy of decomposition (decomposing a different matrix)
  - Or, may promote density of the decomposition (restricting the incompleteness (numerically or structurally))
- Stability of ID: important in combination with iterative methods.

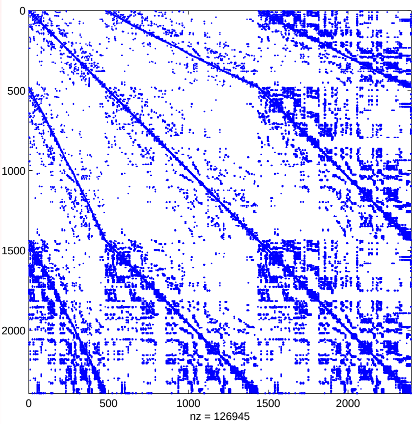
Is is to possible to guarantee more robustness for decompositions by relating them to GE?

In the other words, how far are we from GE-aware decompositions?

Motivation: IV.

ID affects the iterative method via its inverse.

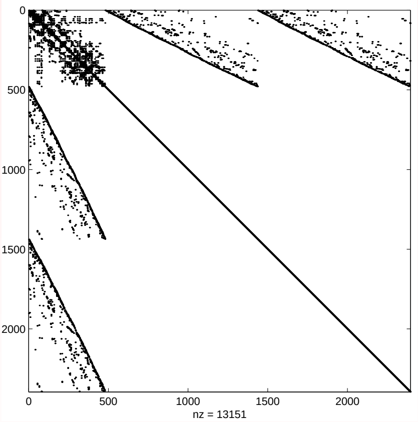




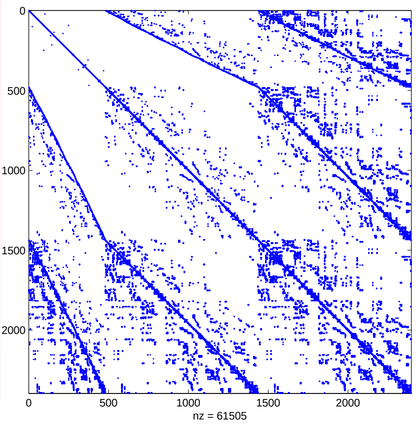
matrix ADD20

rather precise inverse
(2 its BiCGStab)

Motivation: IV.

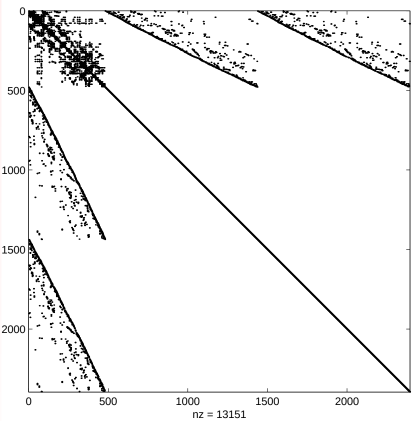


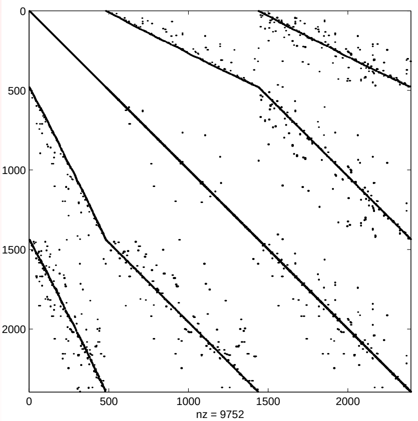
matrix ADD20



less precise inverse

Motivation: IV.

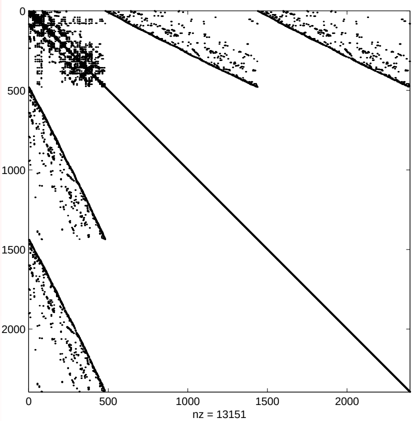


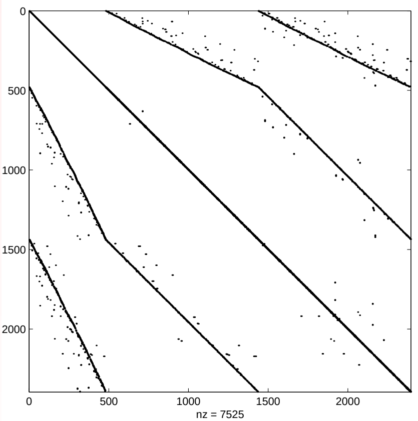


matrix ADD20

even less precise inverse

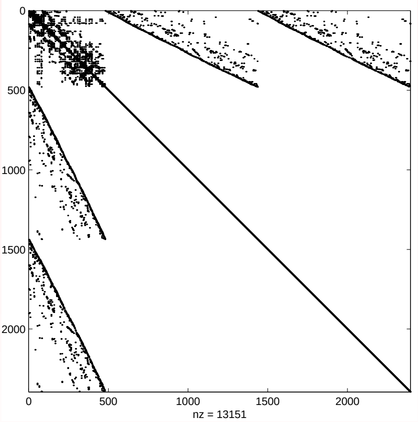
Motivation: IV.

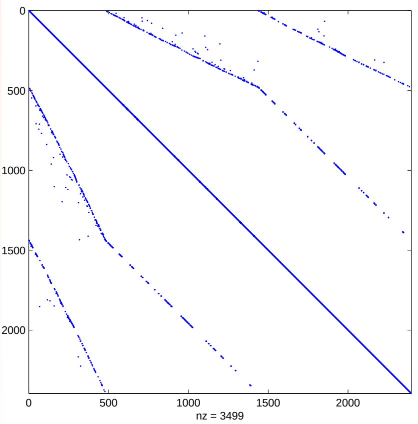




matrix ADD20

rough inverse

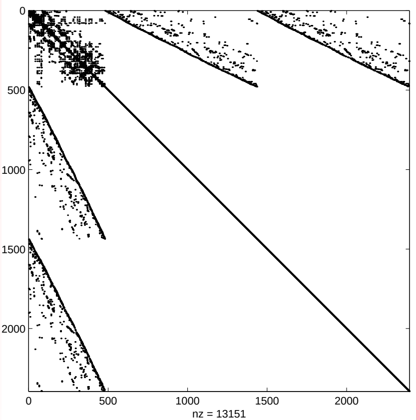


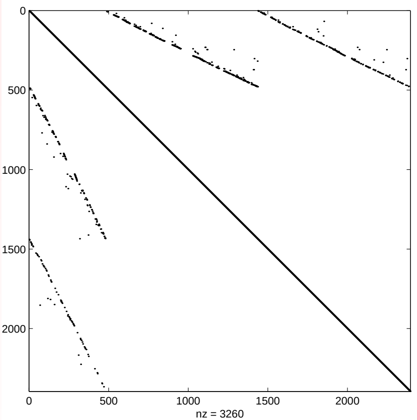


matrix ADD20

very rough inverse

Motivation: IV.

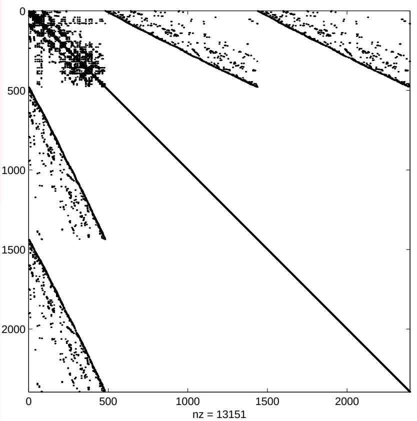


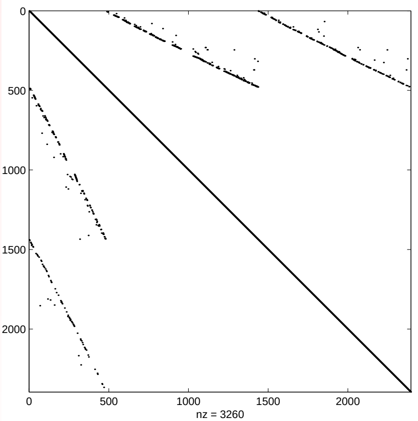


matrix ADD20

ILU decomposition
(similar size as the "very rough inverse")

Motivation: IV.





matrix ADD20

inverted ILU decomposition

Motivation: V.

Concluded motivation

- Consulting / employing matrix inverse may provide useful information

Motivation: V.

Concluded motivation

- Consulting / employing matrix inverse may provide useful information
- Two extreme cases of incomplete decompositions:
  - approximate inverse decompositions (direct ID)
  - direct incomplete decompositions (inverse ID)

Motivation: V.

Concluded motivation

- Consulting / employing matrix inverse may provide useful information
- Two extreme cases of incomplete decompositions:
  - approximate inverse decompositions (direct ID)
  - direct incomplete decompositions (inverse ID)
- Our tools: joint treatment of both direct and inverse decompositions.

Motivation: V.

Concluded motivation

- Consulting / employing matrix inverse may provide useful information
- Two extreme cases of incomplete decompositions:
  - approximate inverse decompositions (direct ID)
  - direct incomplete decompositions (inverse ID)
- Our tools: joint treatment of both direct and inverse decompositions.
- Is this a way to GE-aware decompositions?

Motivation: V.

Concluded motivation

- Consulting / employing matrix inverse may provide useful information
- Two extreme cases of incomplete decompositions:
  - approximate inverse decompositions (direct ID)
  - direct incomplete decompositions (inverse ID)
- Our tools: joint treatment of both direct and inverse decompositions.
- Is this a way to GE-aware decompositions?

What we do not discuss here?

- Modifications of the basic algorithm (basic diagonal modifications, general diagonal compensations with respect to some matvecs etc.)
- a priori diagonal changes
- matrix pre/post processings
- embedding into a more general (e.g. multilevel) scheme.
- Analysis of the described schemes

Summarizing our starting points and goals

Starting points

- Approximate inverse decompositions (Kolotilina, Yeremin, 1993; Benzi, Meyer, T., 1996; Benzi, T., 1998 etc.)

Summarizing our starting points and goals

Starting points

- Approximate inverse decompositions (Kolotilina, Yeremin, 1993; Benzi, Meyer, T., 1996; Benzi, T., 1998 etc.)
- Successful use of parts of factorized matrix inverse used in inverse-based incomplete decompositions (Bollhöfer, Saad, 2002; Bollhöfer, 2003)

Summarizing our starting points and goals

Starting points

- Approximate inverse decompositions (Kolotilina, Yeremin, 1993; Benzi, Meyer, T., 1996; Benzi, T., 1998 etc.)
- Successful use of parts of factorized matrix inverse used in inverse-based incomplete decompositions (Bollhöfer, Saad, 2002; Bollhöfer, 2003)
- A particular goal: Combined use of direct and inverse incomplete decompositions

Summarizing our starting points and goals

Starting points

- Approximate inverse decompositions (Kolotilina, Yeremin, 1993; Benzi, Meyer, T., 1996; Benzi, T., 1998 etc.)
- Successful use of parts of factorized matrix inverse used in inverse-based incomplete decompositions (Bollhöfer, Saad, 2002; Bollhöfer, 2003)
- A particular goal: Combined use of direct and inverse incomplete decompositions
- One of the tools: generalized biconjugation formula

Summarizing our starting points and goals

Starting points

- Approximate inverse decompositions (Kolotilina, Yeremin, 1993; Benzi, Meyer, T., 1996; Benzi, T., 1998 etc.)
- Successful use of parts of factorized matrix inverse used in inverse-based incomplete decompositions (Bollhöfer, Saad, 2002; Bollhöfer, 2003)
- A particular goal: Combined use of direct and inverse incomplete decompositions
- One of the tools: generalized biconjugation formula

Here we try to get inside GE, not to study/defend a synthetic approach.

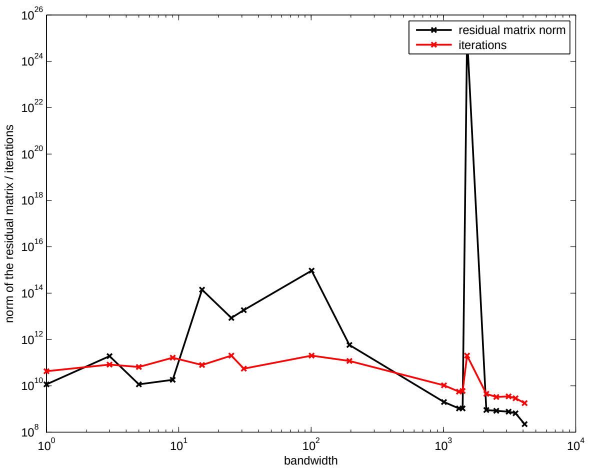
Outline

1. Limits of standard algebraic approaches
2. Standard biconjugation and matrix inverses
3. Direct-inverse decompositions
4. A flavor of applications different from preconditioning
5. Conclusions

Outline

Limits of ID: BCSSTK38, $n = 8032$, $nz = 181,746$

ID: Limitations in predictability and efficiency

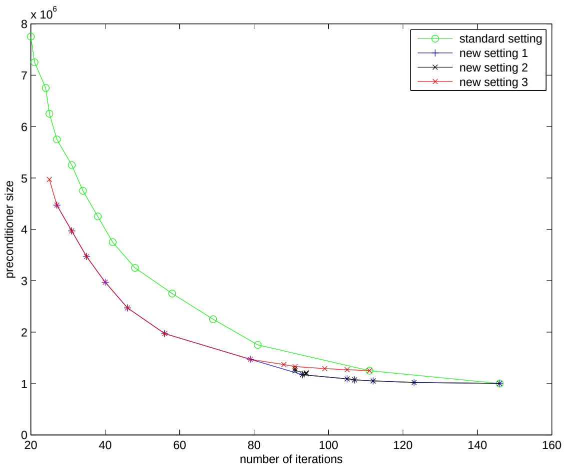


- Generally no clear dependence on the error size, pattern etc.
- This is a very common kind of behavior

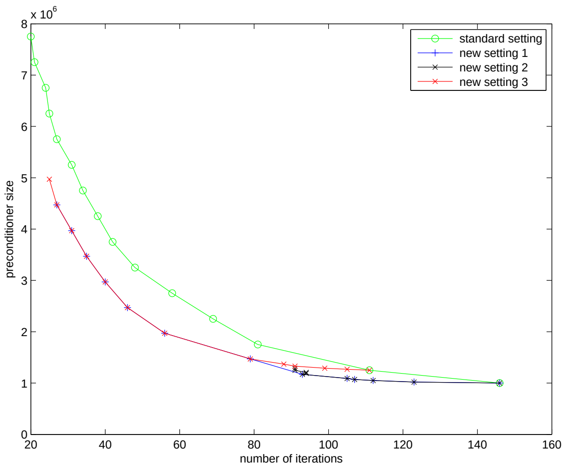
Overcoming some limits: individual level preassignments

Experiments: Kohn-Sham equation, n=250500

Effects individual level preassignments



Effects individual level preassignments



- see the talk of Jennifer Scott at SIAM ALA 2009, Monterey

Outline

Generalized Gram-Schmidt (GGS)

Generalized Gram-Schmidt: basics of SPD case

- Orthogonalize columns of I using the inner product $\langle \, , \, \rangle_A$
- We get (instead of $A = QDR$ with R unit upper triangular):

$$I = ZU$$

  - U is unit upper triangular, as usual ($U = L^T$ for $A = LL^T$).
  - Z is orthogonal in $\langle \, , \, \rangle_A$

$$Z^T AZ = D \quad \text{(Biconjugate decomposition)}$$

  - But: Z is unit upper triangular as well ($Z = L^{-T}$ for $A = LL^T$)
- Easy to reveal decomposed matrix inverse:

$$A^{-1} = ZD^{-1}Z^T,$$

Generalized Gram-Schmidt: II.

Resulting direct and inverse ID may be practical in the incomplete case

$$I = ZDU$$

$$A \approx LL^T,\ U \approx L^T,\ Z \approx L^{-1}$$

- Origins: more papers in 40's and early 50's (Escalator method by Morris (1946), Vector method by Purcell (1952), Fox, Huskey, Wilkinson (1948)).
- The sparse incomplete method can be implemented: AINV (Benzi, Meyer, T., 1996; Benzi, T., 1998)
- Computational procedures to compute sparse incomplete U in this way: RIF (Benzi, T., 2003)
- As we will see, both Z and U can be computed breakdown-free, but this is not all that we may want.

Generalized Gram-Schmidt: III.

Generalized Gram-Schmidt: the (SPD) algorithm

$$I = ZU \equiv [z_1, \ldots, z_n]\ (u_{ij})_{i,j}$$

```
for i=1, n
     for j=1, i-1 with nonzero
```
$u_{ij} = e_j^T A {z_i}^{(j)}$

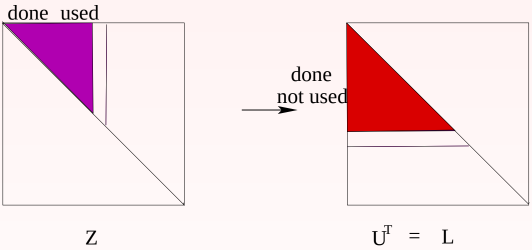
$${z_i}^{(j)} = {z_i}^{(j-1)} - {z_j}^{(j-1)} \frac{e_j^T A {z_i}^{(j-1)}}{e_j^T A {z_j}^{(j-1)}}$$

```
     end j
end i
```

- Forcing partial robustness: different formulas which are the same in exact arithmetic: the breakdown-free variant SAINV
- But: in order to get U we must get Z: direct factor is obtained via the inverse factor

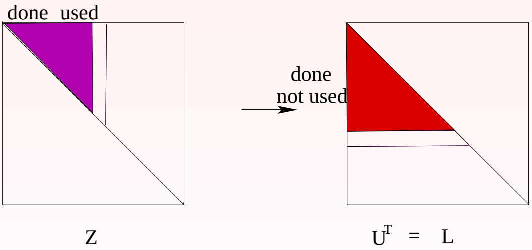
Generalized Gram-Schmidt: IV.

Generalized Gram-Schmidt $I = ZU$: the data dependence graphically



Generalized Gram-Schmidt: IV.

Generalized Gram-Schmidt $I = ZU$: the data dependence graphically



One way transfer of information

Summarization the two general problems

Two resulting general problems

Summarization the two general problems

Two resulting general problems

1. Is there a practical scheme of decomposition that would have an arbitrary transfer of information between direct and inverse factors?

Summarization the two general problems

Two resulting general problems

1. Is there a practical scheme of decomposition that would have an arbitrary transfer of information between direct and inverse factors?
2. Very good behavior of SAINV observed also in the incomplete case with respect to the non-stabilized algorithm cannot be explained just by breakdown-free property

Summarization the two general problems

Two resulting general problems

1. Is there a practical scheme of decomposition that would have an arbitrary transfer of information between direct and inverse factors?
2. Very good behavior of SAINV observed also in the incomplete case with respect to the non-stabilized algorithm cannot be explained just by breakdown-free property

   What is behind the clearly superior performance of the stabilized decomposition with respect to its standard form? Is it possible to get similar enhancement for direct decompositions?

Summarization the two general problems

Two resulting general problems

1. Is there a practical scheme of decomposition that would have an arbitrary transfer of information between direct and inverse factors?
2. Very good behavior of SAINV observed also in the incomplete case with respect to the non-stabilized algorithm cannot be explained just by breakdown-free property

   What is behind the clearly superior performance of the stabilized decomposition with respect to its standard form? Is it possible to get similar enhancement for direct decompositions?

- We have (some) answers for both of these problems
- 1. Arbitrary direct-inverse decompositions
- 2. Transforming the problem via projections.

Summarization the two general problems

Two resulting general problems

1. Is there a practical scheme of decomposition that would have an arbitrary transfer of information between direct and inverse factors?
2. Very good behavior of SAINV observed also in the incomplete case with respect to the non-stabilized algorithm cannot be explained just by breakdown-free property
   What is behind the clearly superior performance of the stabilized decomposition with respect to its standard form? Is it possible to get similar enhancement for direct decompositions?

- We have (some) answers for both of these problems
- 1. Arbitrary direct-inverse decompositions
- 2. Transforming the problem via projections.

Of course, it remains a lot to do to improve GE-based decompositions from inside.

Outline

New shifted biconjugation

- Note: general nonsymmetric formulation is used here

$$A^{-1} = ZZ^T \longleftarrow A^{-1} = ZD^{-1}W^T$$

New shifted biconjugation

- Note: general nonsymmetric formulation is used here

$$A^{-1} = ZZ^T \longleftarrow A^{-1} = ZD^{-1}W^T$$

Nonsymmetric recursions:

$$z_i^{(j)} = z_i^{(j-1)} - z_j^{(j-1)} \frac{a^j z_i^{(j-1)}}{a^j z_j^{(j-1)}}, \quad w_i^{(j)} = w_i^{(j-1)} - w_j^{(j-1)} \frac{a_j^T w_i^{(j-1)}}{a_j^T w_j^{(j-1)}}$$

New shifted biconjugation

- Note: general nonsymmetric formulation is used here

$$A^{-1} = ZZ^T \longleftarrow A^{-1} = ZD^{-1}W^T$$

Nonsymmetric recursions:

$$z_i^{(j)} = z_i^{(j-1)} - z_j^{(j-1)} \frac{a^j z_i^{(j-1)}}{a^j z_j^{(j-1)}}, \quad w_i^{(j)} = w_i^{(j-1)} - w_j^{(j-1)} \frac{a_j^T w_i^{(j-1)}}{a_j^T w_j^{(j-1)}}$$

$$\Downarrow$$

$$s^{-1} I - A^{-1} = ZD^{-1}V^T$$

New shifted biconjugation

- Note: general nonsymmetric formulation is used here

$$A^{-1} = ZZ^T \longleftarrow A^{-1} = ZD^{-1}W^T$$

Nonsymmetric recursions:

$$z_i^{(j)} = z_i^{(j-1)} - z_j^{(j-1)} \frac{a^j z_i^{(j-1)}}{a^j z_j^{(j-1)}}, \quad w_i^{(j)} = w_i^{(j-1)} - w_j^{(j-1)} \frac{a_j^T w_i^{(j-1)}}{a_j^T w_j^{(j-1)}}$$

$$\Downarrow$$

$$s^{-1} I - A^{-1} = ZD^{-1}V^T$$

Analogical recursions:

$$z_i = se_i - \sum_{j=1}^{i-1} \frac{v_j^T e_i}{d_j} z_j \quad , \; v_i = (a^i - se^i)^T - \sum_{j=1}^{i-1} \frac{z_j^T (a^i - se^i)}{d_j} v_j,$$

Z and D are the same in both recursions

More on the new biconjugation

- The $(s^{-1}I - A^{-1})^{-1})$ biconjugation introduced by Bru, Cerdán, Marín, Mas, 2003. The incomplete algorithm was proposed as an approximate inverse preconditioner. (factor Z)

More on the new biconjugation

- The $(s^{-1}I - A^{-1})^{-1})$ biconjugation introduced by Bru, Cerdán, Marín, Mas, 2003. The incomplete algorithm was proposed as an approximate inverse preconditioner. (factor Z)
- It was shown that this new biconjugation can be used to get a direct decomposition (factor U) as well, Bru, Marín, Mas, T., 2008.

$$s^{-1}I - A^{-1} = ZD^{-1}V^T \text{ and } A = LDU \text{ and } Z = U^{-1}$$

$$s^{-1}I - U^{-1}D^{-1}L^{-1} = U^{-1}D^{-1}V^T$$

$$s^{-1}I = U^{-1}D^{-1}(L^{-1} + V^T)$$

upper triangular ↗ ↖ lower triangular

More on the new biconjugation: II.

Pictorially

$$V = \begin{bmatrix} \ddots & & -sL^{-T} \\ & \ddots & \\ U^T D & & \ddots \end{bmatrix}, \qquad \operatorname{diag}(V) = D - sI. \qquad (1)$$

More on the new biconjugation: II.

Pictorially

$$V = \begin{bmatrix} \ddots & & -sL^{-T} \\ & \ddots & \\ U^T D & & \ddots \end{bmatrix}, \qquad \mathrm{diag}(V) = D - sI. \qquad (1)$$

- V obtained by a simple recursion for its columns

More on the new biconjugation: II.

Pictorially

$$V = \begin{bmatrix} \ddots & & -sL^{-T} \\ & & \\ & \ddots & \\ & & \\ U^T D & & \ddots \end{bmatrix}, \qquad \mathrm{diag}(V) = D - sI. \quad (1)$$

- V obtained by a simple recursion for its columns
- The new recursions provide scaled U and L^{-1} at the same time!
- Dropping can interconnect their computation.

New biconjugation in the SPD case

- Note that $s^{-1}I - A^{-1} = ZD^{-1}V^T, V = LD - sL^{-T}, Z = L^{-T}$

$$v_i = (a^i - se^i)^T - \sum_{j=1}^{i-1} \frac{z_j^T(a^i - se^i)}{d_j} v_j,$$

New biconjugation in the SPD case

- Note that $s^{-1}I - A^{-1} = ZD^{-1}V^T, V = LD - sL^{-T}, Z = L^{-T}$

$$v_i = (a^i - se^i)^T - \sum_{j=1}^{i-1} \frac{z_j^T(a^i - se^i)}{d_j} v_j,$$

- We do not need to compute Z at all!

New biconjugation in the SPD case

- Note that $s^{-1}I - A^{-1} = ZD^{-1}V^T, V = LD - sL^{-T}, Z = L^{-T}$

$$v_i = (a^i - se^i)^T - \sum_{j=1}^{i-1} \frac{z_j^T(a^i - se^i)}{d_j} v_j,$$

- We do not need to compute Z at all!
- This is correct strictly mathematically, but computationally?

New biconjugation in the SPD case

- Note that $s^{-1}I - A^{-1} = ZD^{-1}V^T, V = LD - sL^{-T}, Z = L^{-T}$

$$v_i = (a^i - se^i)^T - \sum_{j=1}^{i-1} \frac{z_j^T(a^i - se^i)}{d_j} v_j,$$

- We do not need to compute Z at all!
- This is correct strictly mathematically, but computationally?
- Still the inverse factor influences the direct factor.

$$L^{-1} \longrightarrow L$$

New biconjugation in the SPD case

- Note that $s^{-1}I - A^{-1} = ZD^{-1}V^T, V = LD - sL^{-T}, Z = L^{-T}$

$$v_i = (a^i - se^i)^T - \sum_{j=1}^{i-1} \frac{z_j^T(a^i - se^i)}{d_j} v_j,$$

- We do not need to compute Z at all!
- This is correct strictly mathematically, but computationally?
- Still the inverse factor influences the direct factor.

$$L^{-1} \longrightarrow L$$

- But, dropping can interconnect computation of both L and L^{-1}.

New biconjugation in the SPD case

- Note that $s^{-1}I - A^{-1} = ZD^{-1}V^T, V = LD - sL^{-T}, Z = L^{-T}$

$$v_i = (a^i - se^i)^T - \sum_{j=1}^{i-1} \frac{z_j^T(a^i - se^i)}{d_j} v_j,$$

- We do not need to compute Z at all!
- This is correct strictly mathematically, but computationally?
- Still the inverse factor influences the direct factor.

$$L^{-1} \longrightarrow L$$

- But, dropping can interconnect computation of both L and L^{-1}.
- We drop L using sizes of entries in L^{-1} and vice versa: balanced incomplete factorization, Bru, Mas, Marín, T. 2008.
- Is is the best strategy we can do?

Balanced incomplete factorization (BIF) experiments

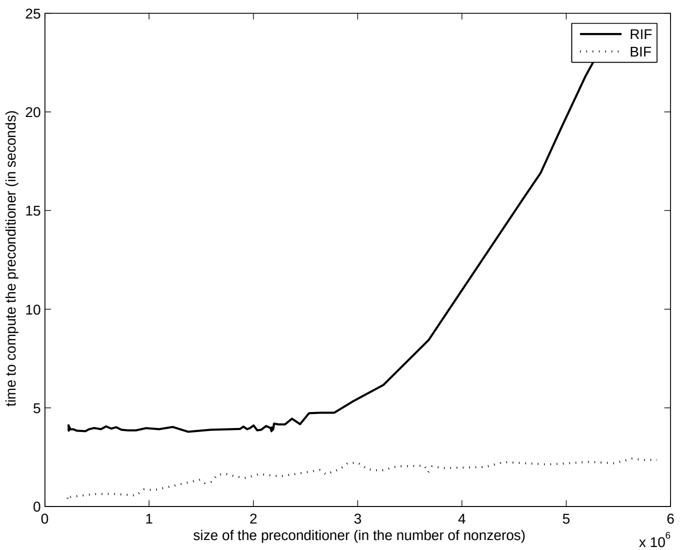
SPD experiments: I.

Example: matrix PWTK, n=217,918, nnz=5,926,171

Balanced incomplete factorization (BIF) experiments

SPD experiments: I.

Example: matrix PWTK, n=217,918, nnz=5,926,171



Balanced incomplete factorization (BIF) experiments: II.

Of course: not only pros; cons as well

- Taking approximate inverses into account, dropping must be always strong. Prefiltration of entries of A is a must.

Balanced incomplete factorization (BIF) experiments: II.

Of course: not only pros; cons as well

- Taking approximate inverses into account, dropping must be always strong. Prefiltration of entries of A is a must.
- We used the inverse-based dropping rules based on Saad, Bollhöfer, 2002, but dropping should be further investigated. It seems that sometimes any rules influence entries of the factors nonuniformly. Also, our dropping often forces skipping a lot of updates in the decomposition. Is this really the right way to go?

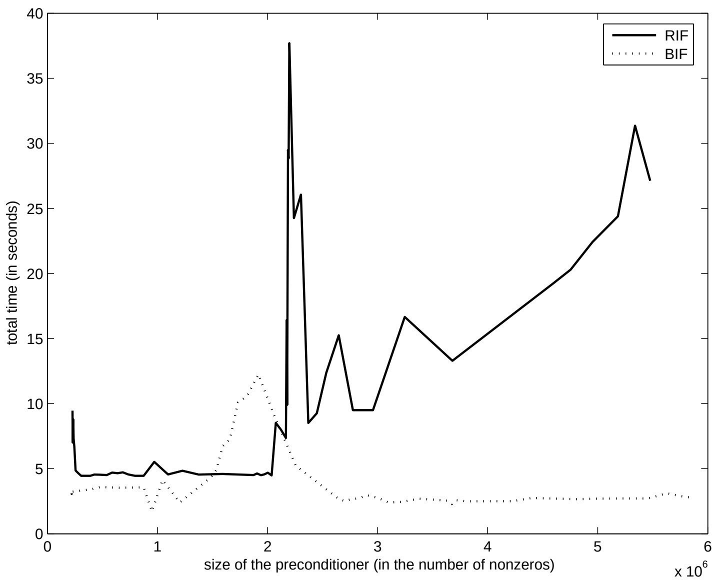
Balanced incomplete factorization (BIF) experiments: II.

Of course: not only pros; cons as well

- Taking approximate inverses into account, dropping must be always strong. Prefiltration of entries of A is a must.
- We used the inverse-based dropping rules based on Saad, Bollhöfer, 2002, but dropping should be further investigated. It seems that sometimes any rules influence entries of the factors nonuniformly. Also, our dropping often forces skipping a lot of updates in the decomposition. Is this really the right way to go?
- The convergence curve is often rather flat if we run many iterations. Is the accuracy sufficient for solving sequences from nonlinear solvers?

Balanced incomplete factorization (BIF) experiments: III.

SPD experiments: II.



Direct-inverse decomposition

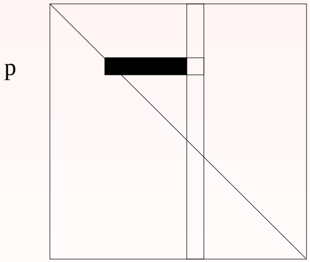
- Vector formulation of the shifted biconjugation can hide important details Bru, Mas, Marín, T. 2009

$$v_i = (a^i - se^i)^T - \sum_{j=1}^{i-1} \frac{z_j^T (a^i - se^i)}{d_j} v_j,$$

Direct-inverse decomposition

- Vector formulation of the shifted biconjugation can hide important details Bru, Mas, Marín, T. 2009

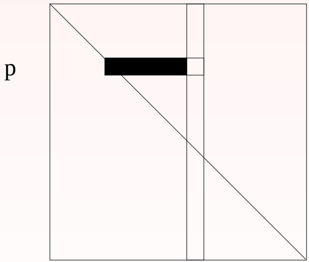
$$v_i = (a^i - se^i)^T - \sum_{j=1}^{i-1} \frac{z_j^T (a^i - se^i)}{d_j} v_j,$$



Direct-inverse decomposition

- Vector formulation of the shifted biconjugation can hide important details Bru, Mas, Marín, T. 2009

$$v_i = (a^i - se^i)^T - \sum_{j=1}^{i-1} \frac{z_j^T(a^i - se^i)}{d_j} v_j,$$

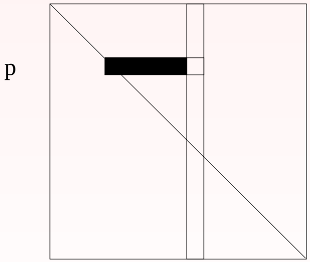


- v_{pi}: just the entries of V with indices $p+1, \ldots, i-1$ are involved

Direct-inverse decomposition

- Vector formulation of the shifted biconjugation can hide important details Bru, Mas, Marín, T. 2009

$$v_i = (a^i - se^i)^T - \sum_{j=1}^{i-1} \frac{z_j^T(a^i - se^i)}{d_j} v_j,$$



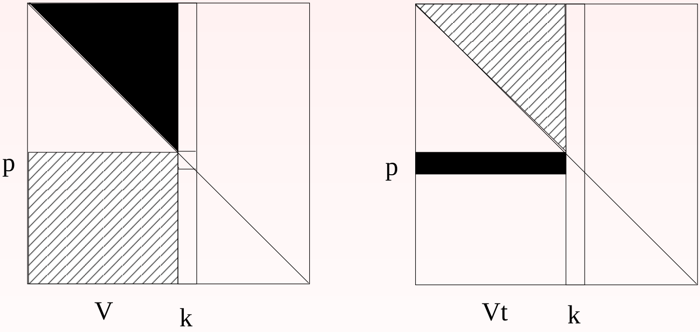
- v_{pi}: just the entries of V with indices $p+1, \ldots, i-1$ are involved
- good, but not enough: the inverse factor still updated only by entries of the inverse factor

Direct-inverse decomposition: II.

- Even more sophisticated computation possible
- Here we demonstrate the computation in the fully nonsymmetric case

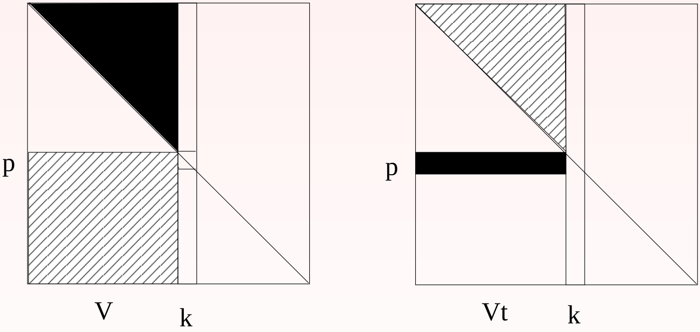
Direct-inverse decomposition: II.

- Even more sophisticated computation possible
- Here we demonstrate the computation in the fully nonsymmetric case



Direct-inverse decomposition: II.

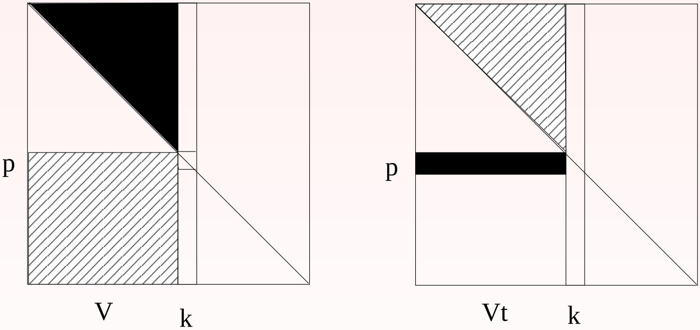
- Even more sophisticated computation possible
- Here we demonstrate the computation in the fully nonsymmetric case



- $v_{1:p-1}$ computed using fully filled areas

Direct-inverse decomposition: II.

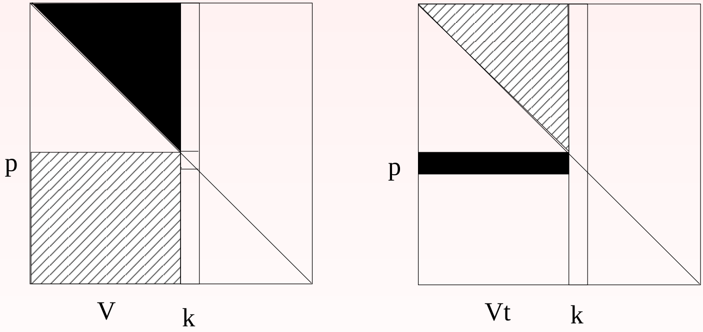
- Even more sophisticated computation possible
- Here we demonstrate the computation in the fully nonsymmetric case



- $v_{1:p-1}$ computed using fully filled areas
- $v_{p+1:n}$ computed using dashed areas

Direct-inverse decomposition: II.

- Even more sophisticated computation possible
- Here we demonstrate the computation in the fully nonsymmetric case



- $v_{1:p-1}$ computed using fully filled areas
- $v_{p+1:n}$ computed using dashed areas
- direct and inverse factors influence each other

Direct-inverse (NBIF) decomposition: experiments: II.

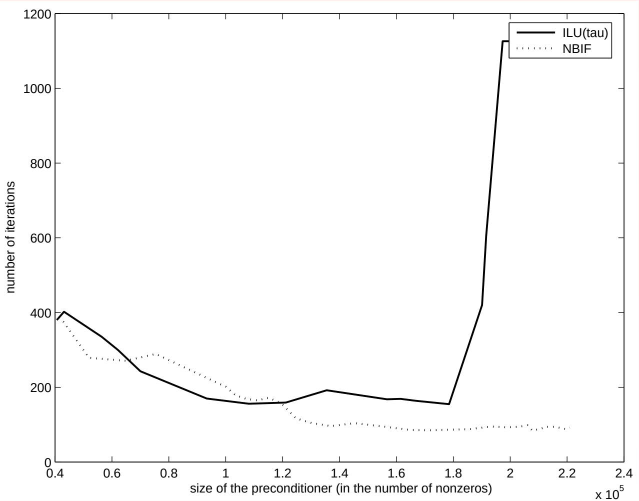
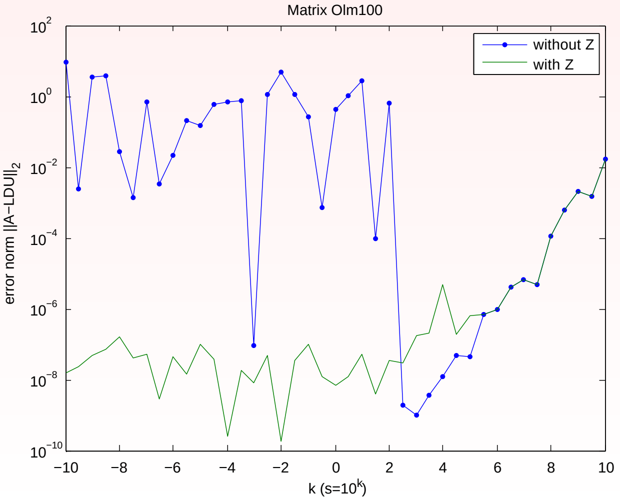


Figure: Sizes of NBIF and ILU(τ) preconditioners versus iteration counts of the preconditioned BiCGStab method for the matrix CHEM_MASTER1.

Scaling parameter

- Choice of scaling parameter s / computational procedures should be coordinated
- Here we demonstrate the computation in the fully nonsymmetric case



Outline

Condition number estimation

Condition number estimation in the 2-norm
(Duintjer Tebbens et al., 2009)

- Two basic approaches: Incremental condition estimation using left singular vectors (ICE, Bischof, 1990) and Incremental norm estimation using right singular vectors (INE, Duff, Vömel, 2002)

Condition number estimation

Condition number estimation in the 2-norm
(Duintjer Tebbens et al., 2009)

- Two basic approaches: Incremental condition estimation using left singular vectors (ICE, Bischof, 1990) and Incremental norm estimation using right singular vectors (INE, Duff, Vömel, 2002)
- Availability of simultaneously computed inverse factor $\Longrightarrow$: four possible ways to estimate the condition number

Condition number estimation

Condition number estimation in the 2-norm
(Duintjer Tebbens et al., 2009)

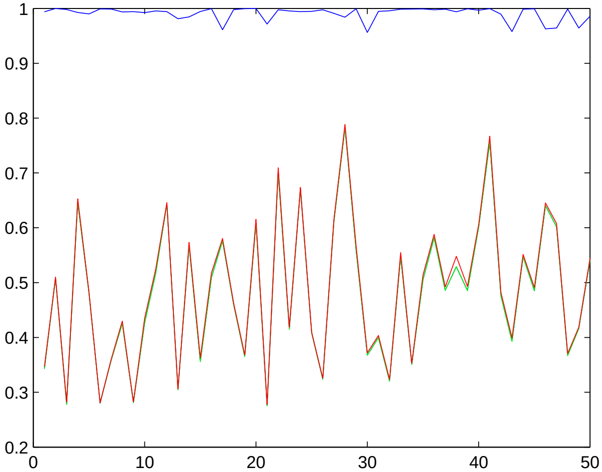
- Two basic approaches: Incremental condition estimation using left singular vectors (ICE, Bischof, 1990) and Incremental norm estimation using right singular vectors (INE, Duff, Vömel, 2002)
- Availability of simultaneously computed inverse factor $\Longrightarrow$: four possible ways to estimate the condition number

$$\kappa(R) \approx \frac{\sigma_{maxL}(R)}{\sigma_{minL}(R)} \ (ICE) \longrightarrow \ldots \longrightarrow \kappa(R) \approx \sigma_{maxR}(R)\sigma_{minRI}(R)$$

yellow (green) curve **blue curve**

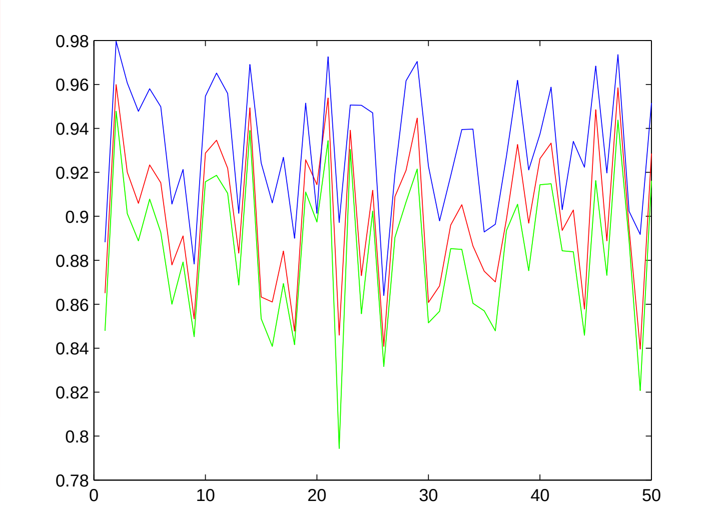
Condition number estimation: II.

50 Random matrices A forming AA^T



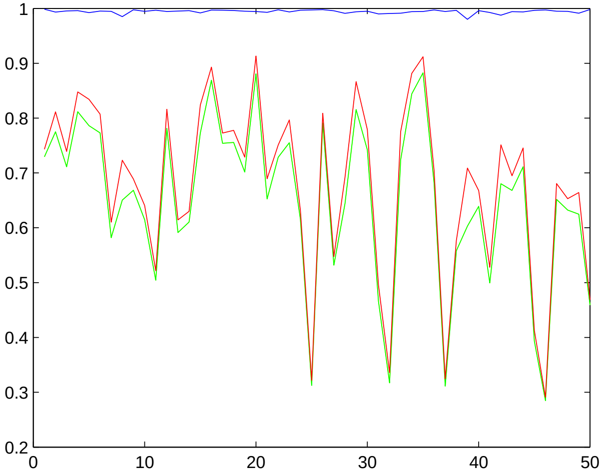
Condition number estimation: III.

50 Random matrices A forming $A + A^T$ with an additional shift



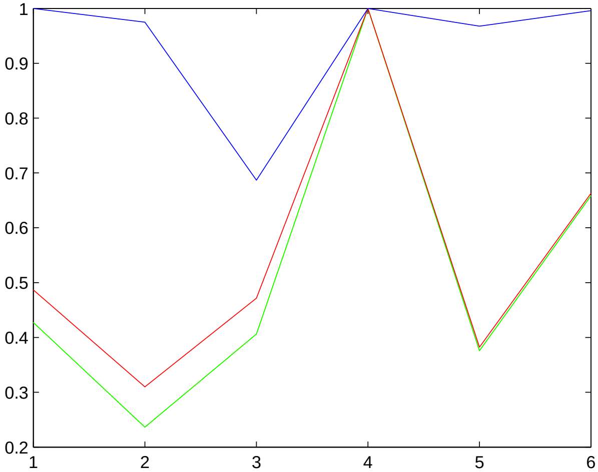
Condition number estimation: IV.

50 Random matrices A forming $A + A^T$, different shift



Condition number estimation: V.

6 Harwell-Boeing matrices, not via BIF



Outline

Conclusions

- Development of algebraic decompositions far from being finished.

Conclusions

- Development of algebraic decompositions far from being finished.
- In particular, a new direct-inverse decomposition BIF/NBIF for preconditioning was introduced.

Conclusions

- Development of algebraic decompositions far from being finished.
- In particular, a new direct-inverse decomposition BIF/NBIF for preconditioning was introduced.
- BIF/NBIF may be useful in other applications, e.g. in construction of condition estimators.

Conclusions

- Development of algebraic decompositions far from being finished.
- In particular, a new direct-inverse decomposition BIF/NBIF for preconditioning was introduced.
- BIF/NBIF may be useful in other applications, e.g. in construction of condition estimators.
- It is possible to construct a direct (GE-like) decomposition using orthogonal projections.

Conclusions

- Development of algebraic decompositions far from being finished.
- In particular, a new direct-inverse decomposition BIF/NBIF for preconditioning was introduced.
- BIF/NBIF may be useful in other applications, e.g. in construction of condition estimators.
- It is possible to construct a direct (GE-like) decomposition using orthogonal projections.
- Do we really understand Gaussian elimination in the sense to expect all future improvements of GE-like decompositions from the inside?

Conclusions

- Development of algebraic decompositions far from being finished.
- In particular, a new direct-inverse decomposition BIF/NBIF for preconditioning was introduced.
- BIF/NBIF may be useful in other applications, e.g. in construction of condition estimators.
- It is possible to construct a direct (GE-like) decomposition using orthogonal projections.
- Do we really understand Gaussian elimination in the sense to expect all future improvements of GE-like decompositions from the inside? Of course, not.

Conclusions

- Development of algebraic decompositions far from being finished.
- In particular, a new direct-inverse decomposition BIF/NBIF for preconditioning was introduced.
- BIF/NBIF may be useful in other applications, e.g. in construction of condition estimators.
- It is possible to construct a direct (GE-like) decomposition using orthogonal projections.
- Do we really understand Gaussian elimination in the sense to expect all future improvements of GE-like decompositions from the inside? Of course, not.

The way from efficient rules of decomposition to fully GE-aware algorithms may be very long

Last but not least

Thank you for your attention!

Last but not least

Thank you for your attention!

Last but not least

Thank you for your attention!

Last but not least

Thank you for your attention!